

UNISYS

**A Series
Help Utility
Operations
Guide**

UNISYS

June 1991
Distribution Code SE
Printed in U S America
8600 0510-000

Priced Item

UNISYS

**A Series
Help Utility
Operations
Guide**

Copyright © 1991 Unisys Corporation

All rights reserved.

Unisys is a registered trademark of Unisys Corporation

Release Mark 3.9.0

June 1991

Distribution Code SE

Printed in U S America

8600 0510-000

Priced Item

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

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

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

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

Correspondence regarding this publication should be forwarded to Unisys Corporation either by using the Business Reply Mail form at the back of this manual or by addressing remarks directly to Unisys Corporation, Technical Publications, 25725 Jeronimo Road, Mission Viejo, CA 92691.

Page Status

Page	Issue
iii	-000
iv	Blank
v through vii	-000
viii	Blank
ix through xi	-000
xii	Blank
xiii	-000
xiv	Blank
xv	-000
xvi	Blank
1-1 through 1-16	-000
2-1 through 2-22	-000
3-1 through 3-8	-000
4-1 through 4-4	-000
5-1 through 5-4	-000
A-1 through A-5	-000
A-6	Blank
B-1 through B-9	-000
B-10	Blank
Glossary-1 through Glossary-4	-000
Bibliography-1	-000
Bibliography-2	Blank
Index-1 through Index-4	-000

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document level. The first digit (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

The purpose of this guide is to explain how to use help commands to create or modify a Help Utility source file, and how to process such a file with the Help Utility to generate a help book. A help book provides online help for users of A Series systems.

Scope

This guide presents all aspects of using help commands in a file and processing the file to create a help book. This guide does not describe how to create programmatic interfaces to books of help text.

Audience

The audience for this guide is anyone responsible for writing or maintaining text that is intended to be processed with the Help Utility for eventual online display. The audience might include programmers, technical writers, translators, and support personnel.

Prerequisites

Anyone intending to write help text or use the Help Utility should have a basic understanding of how to use an A Series system, and of how to create and edit a text file.

How To Use This Guide

If you are creating new help text, you should read all five sections of this guide. You might find it helpful to compare the examples of command usage in Section 4, "Sample Source File," with the descriptions of the commands in Section 2, "Commands." If you are unfamiliar with railroad diagrams, you should read Appendix B, "Understanding Railroad Diagrams."

If your goal is simply to translate the existing help text from one language into another, only a cursory understanding of the utility is necessary. You need only to translate help text and keywords. Knowledge of the command syntax and the structure of help books is not required. Reading Section 1, "Help Utility, Help

About This Guide

Text, and Source Files,” and Section 5, “Running the Help Utility,” should be sufficient for your needs.

Organization

This guide consists of the following sections and appendixes.

In addition, a glossary, a bibliography, and an index appear at the end of the guide.

Section 1. Help Utility, Help Text, and Source Files

This section provides an overview of the Help Utility and describes the basic concepts such as help books and keywords.

Section 2. Commands

This section provides a detailed, formal description of the function and syntax of each command. Basic constructs are also defined.

Section 3. Structured Help Text

This sections describes the concepts of structured help text and the methods for implementing it.

Section 4. Sample Source File

This section provides an extended example of a source file.

Section 5. Running the Help Utility

This section provides instructions for running the Help Utility.

Appendix A. Error Messages

This appendix describes the error messages that are generated when the Help Utility encounters errors in a source file.

Appendix B. Understanding Railroad Diagrams

This appendix explains the railroad diagram notation that is used to formally describe the command syntax of the Help Utility.

Results

After reading this guide, you will be able to correctly place help commands in a text file and process that file to generate a help book.

Related Product Information

The following documents are published by Unisys Corporation and can be ordered through Corporate Software and Publication Operations (CSPO).

A Series CANDE Operations Reference Manual (form 8600 1500)

This manual describes how CANDE operates to allow 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 Editor Operations Guide (form 8600 0551)

This guide describes the operation of the Editor, an interactive tool for creating and modifying text and program files. This guide is written for experienced and inexperienced users who are responsible for creating and maintaining text and program files.

A Series Interactive Menugraph Generator (IMG) Operations Guide (form 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, and IMG screens and forms. This guide is written for users who want to customize these screens for their system.

A Series Menu-Assisted Resource Control (MARC) Operations Guide (form 8600 0403)

This guide provides an overview of MARC, a description of the menu structure, and information on how to use help text, commands, security features, and Communications Management System (COMS) windows from MARC. The guide also explains how to run programs from MARC, how to customize MARC to meet user needs, and how to use MARC in a multinational environment. This guide is written for a wide audience, ranging from experienced system administrators to end users with no previous knowledge of MARC or A Series systems.

A Series Screen Design Facility (SDF) Operations and Programming Guide (form 1185295)

This guide explains how to install SDF. It gives detailed instructions on interactively defining fields, forms, and formlibraries, painting form images, and generating formlibraries. It also provides suggestions and examples for writing applications that use SDF features effectively. This guide is written for application programmers.

A Series Work Flow Language (WFL) Programming Reference Manual (form 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. Help Utility, Help Text, and Source Files

Overview	1-1
Help Utility Concepts	1-2
Help Text	1-2
Source Files	1-2
Help Books	1-2
One Help Book for Short and Long Help Text ..	1-3
Two Help Books	1-3
Elements of a Source File	1-3
Keywords	1-3
Synonyms	1-4
Secondary Keywords	1-4
Embedded Keywords	1-4
Blanks in Embedded Keywords	1-4
Reserved Keywords	1-4
Keyword Formats	1-5
Simple Keywords	1-5
Composite Keywords	1-5
Case in Composite Keywords	1-6
Blanks in Composite Keywords	1-6
Embedded Composite Keywords	1-6
Avoiding Asterisks within Composite Keywords	1-6
Metatoken Keywords	1-7
Embedded Metatoken Keywords	1-7
Blanks in Metatoken Keywords	1-7
Keyword Length in Metatoken Keywords	1-7
Avoiding Angle Brackets within Metatoken Keywords	1-8
Multiple Keywords in the KEYWORD Command ...	1-8
Sections and Section Titles	1-8
Length of a Source File Record	1-9
Help Text Display	1-9
Help Screens	1-10
Boundaries of Displayed Text	1-11
Long Help Text Boundaries	1-11
Short Help Text Boundaries	1-12

Contents

Short Help Text with SECTION Command	
Boundaries	1-12
Short Help Text with KEYWORD Command	
Boundaries	1-12
Making Help Text Available	1-14
Using the SL System Command on	
SYSTEM/HELP	1-14
Making New Versions of Books Available	1-14
Using the NEW BOOK Command	1-15
Using the SL System Command	1-15
Changing the SECURITYTYPE of Help Books	1-15

Section 2. Commands

General Information	2-1
Basic Constructs	2-2
AUTONUMBER	2-4
BOOK	2-5
FILETITLE Option	2-5
COMMENT	2-7
KEYWORD	2-8
Using Metatoken and Composite Keywords	2-9
Using Embedded Keywords	2-9
LANGUAGE	2-10
MANUALNUMBER	2-11
ORGANIZATION	2-12
FLAT Option	2-12
STRUCTURED Option	2-12
REFERENCE	2-13
Continuation Lines	2-13
File Title	2-14
Reference	2-14
Using the Reference to Access Another Help Book ..	2-15
Embedded References	2-15
Creating a Catalog of Help Books	2-16
SECTION	2-17
Section Numbers in a Structured Source File	2-18
With AUTONUMBER	2-18
With MANUALNUMBER	2-19
Section Numbers in a Flat Source File	2-20
Effects of the SECTION Command on Scrolling	2-20
TITLE	2-21
VERSION	2-22

Section 3. Structured Help Text

Section Numbers and Level Numbers	3-1
Parent Topics, Related Topics, and Subtopics	3-2
Parent Topics	3-2
Related Topics	3-3

Subtopics	3-3
Help Screen Actions for Structured Help Books	3-5
Section Numbers and the SECTION Command	3-6
Section Numbering with MANUALNUMBER in Effect	3-6
Section Numbering with AUTONUMBER in Effect ..	3-7

Section 4. Sample Source File

Section 5. Running the Help Utility

Processing the Source File through CANDE	5-1
Processing the Source File in Batch Mode	5-2
Processing the Source File through MARC	5-2
In Command Mode	5-2
Using MARC Screens	5-3
After Processing Is Completed	5-3

Appendix A. Error Messages

Appendix B. Understanding Railroad Diagrams

What Are Railroad Diagrams?	B-1
Constants and Variables	B-2
Constraints	B-2
Vertical Bar	B-3
Percent Sign	B-3
Right Arrow	B-3
Required Items	B-3
User-Selected Items	B-3
Loop	B-4
Bridge	B-4
Following the Paths of a Railroad Diagram	B-5
Railroad Diagram Examples with Sample Input	B-6

Glossary

Bibliography

Index

Figures

1-1.	KEYWORD Command with Multiple Keywords	1-8
1-2.	Sample Help Screen	1-10
3-1.	Help Screen for a Structured Help Book	3-5

Examples

1-1.	Short Help Text with KEYWORD Command Boundaries	1-13
4-1.	Sample Source File	4-2

Section 1

Help Utility, Help Text, and Source Files

This section provides an overview of the Help Utility and explains the elements you need to use the Help Utility.

Overview

The Help Utility is part of the Unisys InterPro® family of software products for the A Series systems. The Help Utility is one element of a system designed to supply online information for the A Series user and your users.

In order for the system to display helpful information on the user's terminal, the system must have access to a file that contains this information. The file contains keywords and help text. In this guide, such a file is called a *help book*.

The function of the Help Utility is to create a help book. A help book contains help text and keywords. *Help text* is the online information that is intended to assist the user in working with the computer. *Keywords* are the link between the user and the help text.

The Help Utility creates a help book by processing a *source file*, which is a file containing embedded commands and help text. This guide teaches you how to create and modify a source file and how to process it with the Help Utility. Once a source file has been processed into a help book, the user can view the help text online through programs that have access to the help book, such as the Menu-Assisted Resource Control (MARC) interface and the Interactive Datacomm Configurator (IDC).

The Screen Design Facility (SDF) also uses help text provided by a Help Utility source file. Instructions for making and modifying long help text and short help text for screens created with the Screen Design Facility (SDF) are provided in the *A Series Screen Design Facility (SDF) Operations and Programming Guide*.

InterPro is a trademark of Unisys Corporation.

Help Utility Concepts

The following paragraphs explain basic terms and concepts you need to understand to use the Help Utility.

Help Text

Help text is documentation that the user can access through a terminal for display *online*. The user can request information about a particular field or task to be performed, for example, and the appropriate help text will appear on the terminal screen. Because it is retrieved electronically, online documentation can be accessed more quickly than printed information in a manual.

There are two kinds of help information available. *Short help text* consists of one or two lines of helpful information that appear on the screen on which the help was requested. Long help text can have as many lines of information as are necessary, and appears on a special help screen.

Source Files

The source file is a text file made up of help text and embedded commands. You create the content of the help text. When you are writing help text, keep in mind the user who will be reading it. This information helps you tailor your help text to fit the specific needs and skill level of the user.

The help text format is limited by the display capabilities of the terminal screen on which the text appears. The maximum number of characters on a single line of a source file is 72. A section of short help text cannot be longer than two lines. A section of long help text can be as long as is necessary to provide complete information.

Embedded commands instruct the Help Utility on how the source file is to be processed. With these commands, you control the structure of the final help text, indicating, for example, where section breaks occur and how the sections of help are structurally related to each other. You choose the keywords that can be used to reference the help text and link these keywords to specific lines of text. These embedded commands do not appear on the screen when help text is displayed to the user.

Help Books

The help book is the file that is created when the Help Utility processes a source file. A help book contains the help text that is divided into sections. Each section has a section title assigned to it and keywords associated with it. The help book also contains a *keyword directory*, which summarizes the associations between keywords and lines of help text.

One Help Book for Short and Long Help Text

A single help book can contain both short help text and long help text. However, if you put short help text and long help text in the same help book, then either the short help text and long help text for a given screen item must have different keywords, or the short help text and long help text must be carefully integrated together.

If each screen item has two different keywords, one for short help and one for long help, then the program accessing the help book, such as MARC, can differentiate between short help text and long help text. You can write different text for each kind of help.

If there is only one keyword for both the short help text and long help text in the same help book, the program accessing the help book, such as MARC, gets the same block of text for short help and help. The first two lines of this block of text appear as short help text. The entire block is displayed as long help text. Therefore, the first two lines must be written so that they can function both as short help text and as the leading lines of long help text.

Two Help Books

If you have short help text in one help book and long help text in another help book, then you can use the same keywords to link a screen item to short help text and to link the screen item to help text.

In order to have short help text in one help book and long help text in another help book, you make one source file that contains only short help text and another source file that contains only long help text. When these two source files are processed by the Help Utility, two help books are created. One help book contains only short help text, and the other contains only long help text.

Elements of a Source File

In addition to help text, a source file contains the embedded Help Utility commands **KEYWORD** and **SECTION**. These elements are explained in the following paragraphs.

Keywords

A *keyword* is a word or group of words that provides the connection between a screen item and the help text that supplies information about that item. Keywords are designated in the **KEYWORD** command. The **KEYWORD** command links the keyword to the first line of online help text following the command.

When you are choosing keywords, select terms that accurately reflect the information contained in the text they are linked to. Also, try to keep keywords short, because the user has the option of typing in keywords to get help text.

Help Utility, Help Text, and Source Files

Synonyms

When more than one keyword is linked with the same line of text, these keywords are referred to as *synonyms*. A synonym is a keyword that can be used to access the same block of help text as some other keyword. If a block of help text has several synonyms associated with it, those synonyms act as extra tools with which the user can work.

Secondary Keywords

A *secondary keyword* is one that is linked with a line of text that falls within a larger section of text. Secondary keywords enable the user to easily access information on relatively minor subjects—for example, subtopics of sections of help text.

Embedded Keywords

When you are writing long help text, you have the option of embedding a keyword within the text. Then you can write another section of help text to provide information on that embedded keyword. This capability enables you to provide the user with helpful information about technical words and phrases that appear in long help text. If an embedded keyword is more than a single word, it cannot be split over two or more lines—it must appear on a single line.

Blanks in Embedded Keywords

When the Help Utility processes a source file, blanks are not removed from keywords embedded in help text. However, when a user selects an embedded keyword—for example, in MARC, by placing the cursor somewhere on the keyword and pressing the SPCFY key—the program that retrieves the help text reads the embedded keyword, removing extra blanks according to the rules for composite and metatoken keywords. The result is that the appropriate help text is retrieved for the user.

Reserved Keywords

There are three words that you cannot use when you are choosing keywords. KEYS, REFERENCES, and TOPICS are *reserved keywords* that the Help Utility automatically places in every help book it creates. The user can use these keywords in the following manner:

- KEYS can be used to display a list of all the keywords in the help book.
- REFERENCES can be used to display a list of all the references defined in a help book. A *reference* is a name used in help text to refer to another help book.
- TOPICS can be used to display a list of all the section titles in the help book.

Keyword Formats

There are three formats your keywords can take: simple, composite, and metatoken. These formats cover the different kinds of keywords you might want to use. The rules for these formats apply both to keywords embedded in help text and keywords entered in the KEYWORD command.

Simple Keywords

A keyword that consists of a single word is called a *simple keyword*. The characters of a simple keyword can consist of letters, digits, and hyphens (-) arranged in any order. The utility does not distinguish between uppercase and lowercase letters in simple keywords.

The following are examples of KEYWORD commands with simple keywords:

```
.KEYWORD acquire
.KEYWORD ACQUIRE
.KEYWORD 1st-run
.KEYWORD -insert-
.KEYWORD 12345
```

Because the case of the letters in simple keywords is unimportant, the first two commands in this example designate the same keyword.

When you enter a simple keyword in the KEYWORD command, or when you enter it as an embedded keyword in help text, you can type it in exactly as it is. If two simple keywords appear next to each other in a KEYWORD command, then at least one blank space must separate them. (For more information on the KEYWORD command, refer to Section 2, "Commands.")

Composite Keywords

A keyword that consists of more than one word or that has characters in it in addition to letters, digits, and hyphens is called a *composite keyword*. The special characters you can use in the composite keywords are the following:

@ \$ { } ~ % ! . = &

To indicate to the Help Utility that a keyword is composite, you must set off the keyword with delimiters. A *delimiter* is a character that comes at the beginning and at the end of an item in order to indicate the boundaries of that item. The delimiter of composite keywords is the asterisk (*).

The following are examples of KEYWORD commands with composite keywords:

Help Utility, Help Text, and Source Files

```
.KEYWORD *ODT commands*
```

```
.KEYWORD *The case when b1 := b2*
```

```
.KEYWORD *Error message #11: (undefined variable)*
```

Case in Composite Keywords

With a composite keyword, the utility treats the uppercase and lowercase forms of a letter as the same letter, just as it does with simple keywords. For example, in the following commands the utility does not distinguish between uppercase and lowercase letters:

```
.KEYWORD *function designators*
```

```
.KEYWORD *Function Designators*
```

Therefore, these two commands define exactly the same keyword.

Blanks in Composite Keywords

In a composite keyword, all blanks are removed between the leading asterisk delimiter and the keyword, and between the keyword and the trailing asterisk delimiter. If two or more blanks appear together within the keyword, the blanks are reduced to a single blank. For example, after processing, the keywords in the following two commands are treated as the same keyword:

```
.KEYWORD * Arithmetic Expressions *
```

```
.KEYWORD *Arithmetic Expressions*
```

Embedded Composite Keywords

A composite keyword entered as an embedded keyword in help text cannot be split over two or more lines.

Avoiding Asterisks within Composite Keywords

Because the asterisk is used as the delimiter of the composite keyword, the asterisk is one special character you cannot use in your composite keyword. For example, the following KEYWORD command is not allowed:

```
.KEYWORD *The * Function*
```

Metatoken Keywords

A *metatoken* is a variable that appears within angle brackets (< >) in certain kinds of language. What if you want to use a metatoken as a keyword? In this case, the utility enables you to enter the metatoken exactly the way it appears. Between the left angle bracket (<) and right angle bracket (>) you can have more than one word. As with composite keywords, the words of *metatoken keywords* can consist of letters, digits, hyphens, and special characters. With metatoken keywords, the utility treats the uppercase and lowercase forms of a letter as the same letter, just as it does with simple keywords and composite keywords.

The following are examples of KEYWORD commands with metatoken keywords:

```
.KEYWORD <digit>
.KEYWORD <integer type>
.KEYWORD <the % function>
```

Embedded Metatoken Keywords

A metatoken keyword that is entered as an embedded keyword in help text cannot be split over two or more lines.

Blanks in Metatoken Keywords

In a metatoken keyword, all blanks are removed between the left angle bracket and the first character, and between the last character and the right angle bracket. If two or more blanks appear together within the keyword, the blanks are reduced to a single blank. For example, after processing, the keywords in the following two commands are treated as the same keyword:

```
.KEYWORD < real array identifier >
.KEYWORD <real array identifier>
```

Keyword Length in Metatoken Keywords

If a metatoken keyword is more than 30 characters long, including the angle brackets, the keyword is shortened to 30 characters by the Help Utility. Before the keyword is shortened, any extra blanks are removed. If this new version of the keyword is still more than 30 characters long, the utility shortens it by removing all characters after the 30th character, except for the right angle bracket.

In the following example, the original statement is truncated to 30 characters:

Help Utility, Help Text, and Source Files

.KEYWORD <this keyword is more than 30 characters long>

.KEYWORD <this keyword is more than 30>

Avoiding Angle Brackets within Metatoken Keywords

The left and right angle brackets are part of the metatoken keyword, and they also serve to delimit the metatoken keyword. Because the left and right angle brackets serve to delimit metatoken keywords, they cannot be used as special characters in metatokens. For example, the following KEYWORD commands are not permitted:

.KEYWORD <the > function>

.KEYWORD <arithmetic < symbol>

Multiple Keywords in the KEYWORD Command

In most of the previous keyword examples, only one keyword follows each KEYWORD command to keep the examples simple. In fact, more than one keyword can appear in the KEYWORD command line, and these keywords can be of different types.

Figure 1-1 shows more than one keyword following the KEYWORD command.

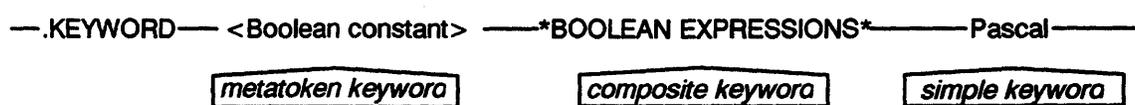


Figure 1-1. KEYWORD Command with Multiple Keywords

For more information on the KEYWORD command, refer to Section 2, "Commands."

Sections and Section Titles

A *section* of help text is a piece of text occurring between two consecutive SECTION commands. Each section of help text normally has keywords linked to it and a section title assigned to it.

The *section title* is a name you give to a section of text. It can be a single word or a group of words. The section title appears on the screen along with the help text when the user asks for information.

Length of a Source File Record

A *record* is one line of text on your terminal. When the Help Utility processes your source file, it looks only at the first 72 characters of each record. If your source file has records that are more than 72 characters long, the extra characters are lost when the utility processes the file.

The easiest way to make sure that the lengths of records in your source file are compatible with the Help Utility is to make your source file of file type TEXT before you begin writing the source file. If you use the Command and Edit (CANDE) message control system (MCS) to edit your source file, you can do so by entering *MAKE <file name> TEXT*, and substituting the name of your source file for <file name>. See Section 4, "Sample Source File," for an example.

Help Text Display

Short help text is displayed at the bottom of the screen from which it was requested. Long help text is displayed on its own help screen.

You use the SECTION and KEYWORD commands to delineate the beginning and end of each piece of help text that appears for the user.

The following paragraphs describe the layout of a help screen in the MARC interface and explain how to set boundaries for help text.

Note: *Products that have online help text can differ in their methods of providing help text access to the user.*

This guide gives examples of how a user can access help text that are always true for MARC; however, these examples might not be true for certain other products. If you are interested in the help text access methods for a particular product, refer to the operations guide for that product.

Help Screens

When the user requests short help text in MARC by pressing the SPCFY key once, one or two lines of information appear on the bottom one or two lines of the screen the user is currently viewing. When the user requests long help text, by pressing the SPCFY key twice, a special *help screen* appears. Figure 1-2 is an example of a MARC help screen.

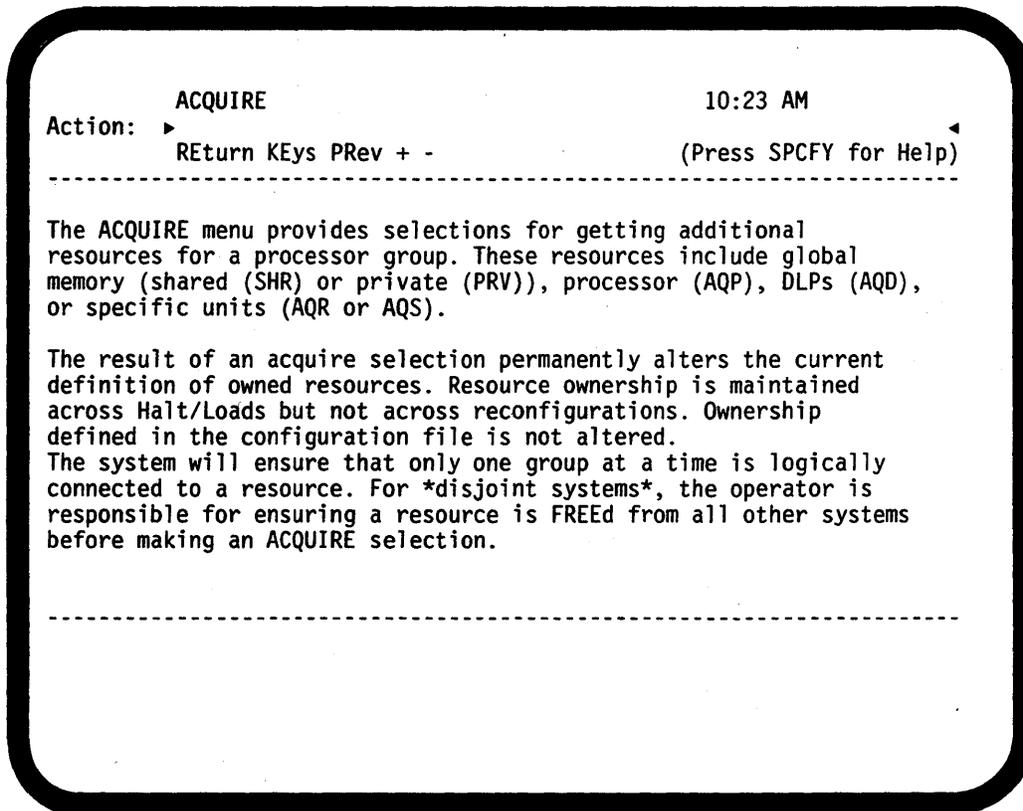


Figure 1-2. Sample Help Screen

In Figure 1-2, the three lines at the top of the screen supply the user with the following information about the help text displayed and give the user the ability to manipulate the screen:

- On the first line, two items appear. *ACQUIRE* is the title of this section of text. *10:23 AM* is the time of day when the screen is displayed.
- The second line is the Action field. Between the left and right arrows, the user can enter any action that appears on the third line.
- The third line lists the actions available on this screen. The actions available on a help screen vary depending on the kind of help book the displayed help text came from. The two different kinds of help books and the screen actions

associated with them are described in a later section. On this screen you can enter the following in the Action field:

- *REturn* displays the screen on which help information was originally requested.
- *KEys* displays a list of all the keywords in the book.
- *PReV* (an abbreviation of *previous*) displays the last screen of help text that was shown to the user.
- The plus (+) and minus (-) actions scroll text one screen page forward and one screen page backward, respectively, in those cases where a section of text is too long to fit on a single help screen.

The message *Press SPCFY for Help* is a reminder to the user that by moving the cursor to certain help screen items, such as those appearing between asterisks or angle brackets, and then pressing the SPCFY key, he or she can view help text for that screen item.

The help screen item that the user can retrieve help text on in Figure 1-2 is the *disjoint systems* embedded keyword.

In Figure 1-2, the information that comes from the help book is the section title ACQUIRE and the help text itself. All the other items are supplied by the menu-driven system that has access to the help text. In this example, the menu-driven system is MARC.

The format of the help text the user sees is exactly the format of the source file. Blank lines in a source file are displayed as blank lines in help text. See Section 4 for an example of a source file.

Boundaries of Displayed Text

The following paragraphs describe the boundaries of long help text and short help text for the Menu-Assisted Resource Control (MARC) interface, the Interactive Menugraph Generator (IMG), and the Interactive Datacomm Configurator (IDC), and for the SECTION and KEYWORD commands.

Long Help Text Boundaries

The keywords in the KEYWORD command mark the only points where a user can access long help text. The KEYWORD command links keywords to the first line in a section of long help text, or, in the case of secondary keywords, the command links keywords to lines of text that fall within a section of help text.

In MARC and programs that access help text in a similar way, when a user requests help information, the entire help book section containing that information is made available to the user. The first line of help text displayed for the user is the line of text linked to the keyword that was used to access the information.

As a result, if a secondary keyword is used to access help text, the first line displayed for the user is a line from within a section of help text. Whether the keyword used to access information is a secondary keyword or a keyword linked to the first line of a section, the user can use the plus (+) and minus (-) actions to scroll through the entire section of help text.

Short Help Text Boundaries

Short help text is different from long help text in the following ways:

- Short help text is displayed on the last two lines of the screen from which help was requested.
- Two lines of short help text are always displayed.

You can set the boundaries of short help text by using either the SECTION command or the KEYWORD command.

Short Help Text with SECTION Command Boundaries

As with long help text, each segment of short help text can begin with a SECTION command and one or more KEYWORD commands. If used in short help, the SECTION command for one segment of short help text marks the end of the preceding segment of short help text. When the user requests short help, the Help Utility retrieves the entire segment of short help text—all the text between two successive SECTION commands.

The text is displayed according to the following rules:

- If there is only one line of short help text between the SECTION commands, that line of text and a second line that is blank (empty of any text) are displayed to the user.
- If there are two or more lines of short help text between the SECTION commands, only the first two lines of text are displayed to the user. Since the user cannot scroll within short help, any extra lines are never displayed.

Note that even though SECTION commands appear in a source file of short help text, short help section titles are never displayed to the user.

Short Help Text with KEYWORD Command Boundaries

The KEYWORD command provides another method for marking the leading boundaries of segments of short help text. This method requires that each segment begin with a KEYWORD command. The command indicates only where the short help text begins, not where it ends. When the user requests short help, the Help Utility retrieves the short help text beginning with the first line of text following the KEYWORD command, and if no SECTION commands appear first, ending with the last line of short help text in the help book.

The text is displayed according to the following rules:

- If only one line of short help text appears between KEYWORD commands and the next KEYWORD command appears on the line immediately following the first, that one line of text and the first line of text following the next KEYWORD command are displayed. The Help Utility does not recognize the next KEYWORD command as marking the end of the segment of short help text.
- If there are two or more lines of short help text between the KEYWORD commands, only the first two lines of text are displayed to the user. Any extra lines will never be seen as short help text.

Because the Help Utility always displays two lines of short help text, in order for this method of marking boundaries to succeed, any time there is only one line of short help text between KEYWORD commands, you must add a second blank.

Example 1-1 shows short help text with KEYWORD command boundaries.

```
.ORGANIZATION FLAT
.COMMENT THE ORGANIZATION COMMAND MUST COME BEFORE THE FIRST
.COMMENT SECTION COMMAND.
.SECTION " "
.COMMENT A SECTION COMMAND IS REQUIRED AT THE BEGINNING OF ANY
.COMMENT SOURCE FILE OF HELP TEXT OR SHORT HELP TEXT.
.KEYWORD menucopy
This selection is the first step in copying a menu or form from a
different menugraph to the current menugraph.
.KEYWORD hmenucrea
This selection is the first step in creating a new menu or form.

.KEYWORD copyfname
Enter the name of the menugraph that contains the form to be copied

.KEYWORD copyform
Enter the screen name of the form that is to be copied. This is the
screen name of the form as it appears in the original menugraph

.KEYWORD copymname
Enter the name of the menugraph that contains the menu to be copied

.KEYWORD copymenu
Enter the screen name of the menu that is to be copied. This is the
screen name of the menu as it appears in the original menugraph.
```

Example 1-1. Short Help Text with KEYWORD Command Boundaries

Making Help Text Available

Access to help books by the products that use those help books is made possible by a program called SYSTEM/HELP, which runs as a library on your system.

Using the SL System Command on SYSTEM/HELP

In order for long help text and short help text to be available to the A Series users, SYSTEM/HELP must be a system library associated with the HELPSUPPORT on that system. You can make SYSTEM/HELP a system library by means of the SL (System Library) system command. This command links the function HELPSUPPORT to the library SYSTEM/HELP.

To learn whether or not SYSTEM/HELP is linked to the HELPSUPPORT library, enter the SL System command as follows on an appropriate terminal:

```
SL
```

A list is then displayed on the terminal screen. If the following entry appears in the list, SYSTEM/HELP is already linked to HELPSUPPORT:

```
SL HELPSUPPORT = *SYSTEM/HELP ON <family name>
```

If there is no entry like this, then you can make SYSTEM/HELP a system library for HELPSUPPORT by entering the SL System command as follows:

```
SL HELPSUPPORT = *SYSTEM/HELP ON <family name>
```

Substitute the family name of the disk pack on which SYSTEM/HELP is to reside for <family name>.

Unisys recommends that this procedure be carried out only by system operators or others who are familiar with system commands.

Making New Versions of Books Available

The HELPSUPPORT library has a *book cache* that contains information about recently accessed help books. This book cache is maintained on a least recently used basis: when the cache is full and a new help book is to be added, the least recently used help book is removed from the cache.

When the HELPSUPPORT library receives a request for help information, it first checks its book cache to see whether the help book containing the specified information is present. If the help book is present, the HELPSUPPORT library draws the help information from the book cache, without having to reopen the help book.

If the help book is not present in the book cache, the HELPSUPPORT library searches for a help book with the specified title. If the HELPSUPPORT library

finds the help book, it updates its book cache with the current help book information. If necessary, the library also removes from the book cache the help book with the oldest previous access date. Thereafter, any requests for information from that help book are drawn from the version that is stored in the book cache. As a consequence of this cache system, when the Help Utility is used to make changes to an existing book, the new version of the book is not available to the user if the HELPSUPPORT library continues to access the old version of the book stored in its book cache.

There are two ways to make a new version of a book part of the book cache so that it is available to the user:

- The MARC *NEW BOOK* command
- The SL system command

Using the NEW BOOK Command

To use the MARC *NEW BOOK* command, MARC must be available on your system. From the Action field of any MARC screen, enter

```
CO NEW BOOK <file title>
```

Instead of <file title>, you enter the name of the new help book.

A description of the *NEW BOOK* command appears in the *Menu-Assisted Resource Control (MARC) Operations Guide*.

Using the SL System Command

If MARC is not available on your system, then you must repeat the process of making SYSTEM/HELP a system library associated with the HELPSUPPORT library. To do so, from an appropriate terminal, enter the SL System command:

```
SL HELPSUPPORT = *SYSTEM/HELP ON <family name>
```

Instead of <family name>, you enter the family name of the disk pack on which SYSTEM/HELP is to reside.

Unisys recommends that this procedure be carried out only by system operators or others who are familiar with system commands.

Changing the SECURITYTYPE of Help Books

When the Help Utility creates each help book, it assigns PRIVATE as the value of the SECURITYTYPE attribute. For all users to be able to access a help book, you might have to make the help book a public file. You can do so by changing the value of the SECURITYTYPE attribute to PUBLIC. Enter the *CANDE SECURITY* command as follows:

Help Utility, Help Text, and Source Files

SECURITY <file title> PUBLIC

In place of <file title>, you enter the file title of the help book. You can also change the value of the SECURITYTYPE attribute on the MARC Changing Disk File Security screen.

MARC has the ability to access both public and nonpublic books, but it allows only an authorized user access to a nonpublic book. One way to ensure that all users will have access to a book through MARC is to make the book public. For example, to allow all users access to the MARC help book and the MARC short help book, make those files public.

Section 2

Commands

Commands are the means by which you give instructions to the Help Utility. The following commands are recognized by the Help Utility:

- AUTONUMBER
- BOOK
- COMMENT
- KEYWORD
- LANGUAGE
- MANUALNUMBER
- ORGANIZATION
- REFERENCE
- SECTION
- TITLE
- VERSION

This section provides syntax and explanation for each of these Help Utility commands.

General rules are defined under “General Information,” and repetitive syntax elements are explained under “Basic Constructs.”

General Information

The term *command line* refers to a line in a source file consisting of a command and any information that is required by that command.

The following is the syntax for a command:

```
- .<command> _____|
```

There are some general rules that apply to all commands. In order for the Help Utility to successfully process your source file, you must follow these rules:

- The period (.) in the command must appear in the leftmost column (column 1).
- At least one blank must appear between the command itself and any other characters that appear in the command line.

Commands

- At least one blank must appear between each item in a command line.
- Only one command can appear in each command line.
- A command line cannot be longer than a single line (72 characters).
- Although the examples in this guide show commands in uppercase (capital) letters, the Help Utility recognizes any combination of uppercase and lowercase letters, as long as the command is correctly spelled.

Basic Constructs

<space>

One or more blank characters.

<digit>

Any integer from 0 to 9, inclusive.

<special character>

Any one of the following characters:

! # \$ % & ' () = _ ^
' @ { } [] * : + < >
. ? / | ; , \ ~ "

<file identifier>

- <name> _____|

<file title>

- <file name> _____|
 └── ON - <family name> ─┘

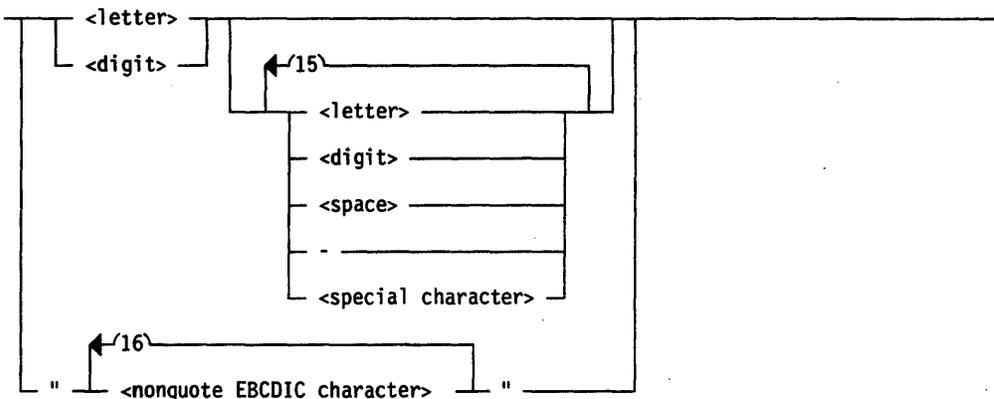
<file name>

┌──────────────────┐
└── (- <usercode> -) ─┘
* _____|
 ┌── / ─┘
 |
 └── <name> ─┘

<usercode>

- <name> _____|

<name>



<family name>

The name of the pack on which <file name> is stored. This name can be any combination of up to 17 letters and digits.

AUTONUMBER

The AUTONUMBER command directs the Help Utility to automatically number the long help text each time the SECTION command appears.

- .AUTONUMBER

When AUTONUMBER is enabled, then each time the SECTION command appears, the following section of text is assigned a number that is one greater than the immediately preceding section of text. This procedure holds true whether the preceding section was automatically numbered or manually numbered. The SECTION command is used to specify the level at which the numbering is done. Refer to "SECTION" later in this section for more information about the SECTION command.

The AUTONUMBER command is optional. If the AUTONUMBER command is used, it can appear anywhere in the source file. If the AUTONUMBER command appears after a MANUALNUMBER command, the MANUALNUMBER command is turned off and automatic numbering is put into effect. If the AUTONUMBER command does not appear in the source file, then MANUALNUMBER is the default mode, and you must assign section numbers yourself. Refer to "MANUALNUMBER" later in this section for more information about the MANUALNUMBER command.

BOOK

The BOOK command enables you to influence the file title of the help book that is produced when the Help Utility is run on the source file.

```

- .BOOK FILETITLE - <help book file subtitle>
      FILEID - <file identifier>
  
```

<help book file subtitle>

Any valid file title that is not more than 60 characters long.

The BOOK command is optional. If the BOOK command is used, it can appear only once in a source file, and it must appear before any SECTION commands and any help text. If the BOOK command is not used, and if DEST is not file-equated, then the help book file created when the Help Utility processes the source file is

BOOK/<source file title>/<language name>

Instead of <source file title>, the title of your source file appears.

FILETITLE Option

If you use the FILETITLE option, the BOOK command causes the file created by running the Help Utility to be named

BOOK/<help book file subtitle>/<language name>

The language name is either English or the language specified by the LANGUAGE command. (Refer to "LANGUAGE" later in this section for more information on <language name>.)

For example, assume you enter the following BOOK command:

```
.BOOK FILETITLE FIRST/HELP/TEXT
```

The help book created when the Help Utility processes the source file containing this BOOK command is

BOOK/FIRST/HELP/TEXT/<language name>

If a help book file subtitle is so long that the BOOK command plus the help book file subtitle runs more than a single line (72 characters), you must use the FILEID option. In this case, when the Help Utility is run, the file identifier is equated to the help book file subtitle. (Refer to "Basic Constructs" earlier in this section for a definition of a valid file identifier.)

For example, assume the BOOK command is

Commands

.BOOK FILEID LONGNAME

Then at the time you run the Help Utility, you equate LONGNAME with the file subtitle you want as the middle part of the file title of the help book. (Refer to "Basic Constructs" earlier in this section for a definition of a valid file title.)

The action of the BOOK command is overridden if the internal file name DEST is file-equated with a file title at the time the source file is processed. For a description of DEST file equation, refer to "Processing the Source File in Batch Mode" in Section 5, "Running the Help Utility."

COMMENT

The **COMMENT** command enables you to place comments about your text anywhere in the source file. These comments do **not** appear in the help book.

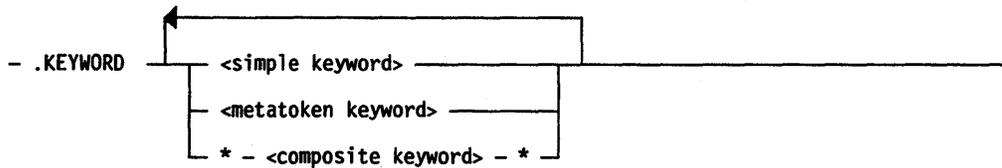
```
.COMMENT <comment text>
% _____|
```

The comment text is text intended to supply documentation for the help text in the source file. The comment text can be composed of any combination of characters and blanks.

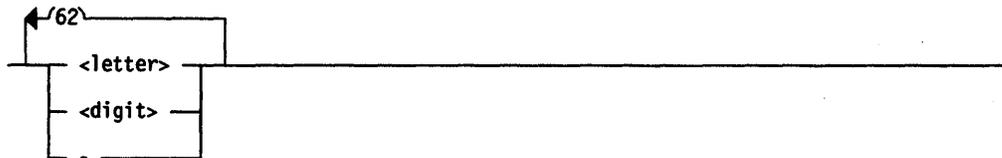
The two forms of the **COMMENT** command function the same. Comments can be used to document the text, to remind you why help text was written the way it was. Comments can provide useful information when it comes time to update existing help text or add new text.

KEYWORD

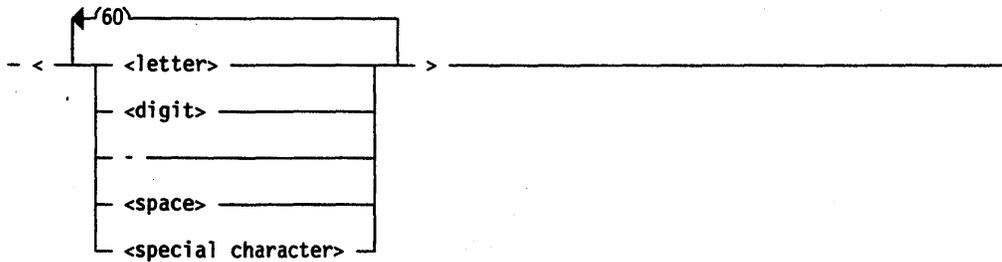
The **KEYWORD** command links one or more keywords with the first line of help text that follows the **KEYWORD** command line and, consequently, with the entire block of text following the keyword. Keywords provide the user with a means of referencing help information.



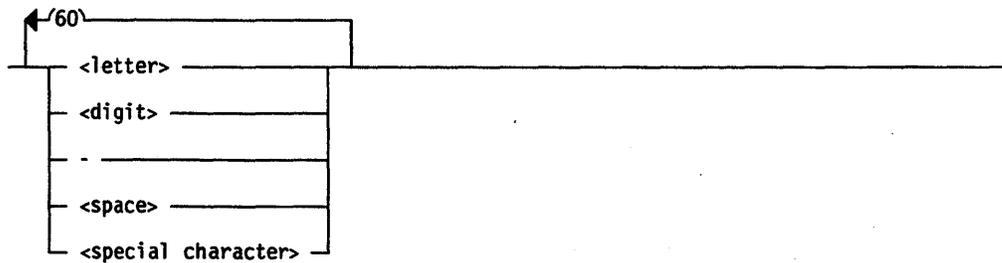
<simple keyword>



<metatoken keyword>



<composite keyword>



The following rules apply to all keywords:

- A source file cannot define more than 20,000 keywords.
- A specific keyword can appear in only one KEYWORD command in a source file.
- A SECTION command must appear at least once somewhere in the source file before the first KEYWORD command can appear. After that, KEYWORD commands can appear anywhere in the source file. Keywords that do not appear in conjunction with SECTION commands are secondary keywords.
- TOPICS, REFERENCES, and KEYS are keywords reserved for use by the Help Utility. They cannot be assigned as keywords to sections of help or shorthelp text.
- Although up to 63 characters are allowed in a keyword, only the first 30 characters, including the delimiters in metatoken keywords, are significant in distinguishing between keywords.
- You can link as many keywords with a line of text as you wish. If the KEYWORD command and its keywords run longer than a single line (72 characters), then the list of keywords can be continued on the next line. Each additional line must begin with the KEYWORD command.
- A single keyword cannot be broken over two or more lines.

Using Metatoken and Composite Keywords

For a metatoken or composite keyword, all blanks are removed between the left delimiter and the first character, and between the last character and the right delimiter. If two or more blanks appear together within a metatoken or composite keyword, the blanks are reduced to a single blank.

If a metatoken keyword is more than 30 characters long—including the angle brackets (<>)—the keyword is shortened to 30 characters by the Help Utility. Before the keyword is shortened, any extra blanks are removed. If this new version of the keyword is still more than 30 characters long, it is shortened by the removal of trailing characters other than the right angle bracket.

The left angle bracket (<) and the right angle bracket (>) are not allowed as special characters in a metatoken keyword.

The asterisk (*) is not allowed as a special character in a composite keyword.

Using Embedded Keywords

When entering keywords as embedded keywords in help text, follow the appropriate syntax for simple keywords, metatoken keywords, and composite keywords as described in the preceding railroad diagrams.

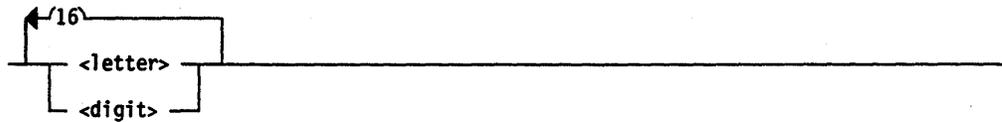
Embedded keywords that are more than a single word cannot be split over two or more lines.

LANGUAGE

The **LANGUAGE** command designates the language in which the help text is written.

`..LANGUAGE - <language name>` _____

<language name>



The language name specified in the **LANGUAGE** command is appended to the end of the help book file title (refer to the **BOOK** command earlier in this section). For example, assume the following **LANGUAGE** command is included in the source file:

```
.LANGUAGE DEUTSCH
```

As a result, the help book file title would be

```
BOOK/<help book file subtitle>/DEUTSCH
```

The **LANGUAGE** command is optional. If the **LANGUAGE** command is used, it must appear before the first **SECTION** command. If the **LANGUAGE** command is not used, the language name defaults to **ENGLISH**.

MANUALNUMBER

The **MANUALNUMBER** command enables you to number sections of text yourself. The **MANUALNUMBER** command allows greater flexibility in numbering sections than you have with the **AUTONUMBER** command. (Refer to “**AUTONUMBER**” earlier in this section for more information about the **AUTONUMBER** command.)

— .MANUALNUMBER _____

To number a section when the **MANUALNUMBER** command is in effect, you must enter the section number as part of the **SECTION** command. For example, in the following **SECTION** command, the number 3.2 is assigned to a section:

```
.SECTION 3.2 "<section title>"
```

Refer to “**SECTION**” later in this section for more information about the **SECTION** command.

The rules for numbering sections when the **MANUALNUMBER** command is in effect are as follows:

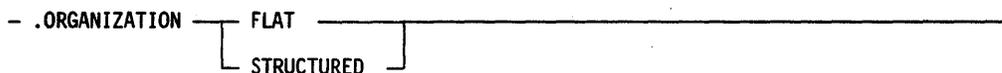
- The section number for a given section must be greater than the section number of the immediately preceding section. However, the section number for a given section does not have to be just one greater than the section number of the immediately preceding section.
- The level at which the current section is numbered may be less than, equal to, or one greater than the level at which the immediately preceding section was numbered. The level may not be more than one greater than the level at which the immediately preceding section was numbered.

The **MANUALNUMBER** command can appear anywhere in a source file. If the **MANUALNUMBER** command appears after an **AUTONUMBER** command, the **AUTONUMBER** command is turned off and manual numbering is put into effect.

If neither the **AUTONUMBER** command nor the **MANUALNUMBER** command appears in the source file, **MANUALNUMBER** is the default mode.

ORGANIZATION

The ORGANIZATION command enables you to specify the structural implications of the section numbers of help text.



The ORGANIZATION command is optional. If the ORGANIZATION command is used, it must appear somewhere before the first SECTION command in a source file. If the ORGANIZATION command is not used, then the STRUCTURED option is the default.

FLAT Option

If you use the FLAT option, all sections of help text are treated as being structurally equal to each other, and section numbers are optional. If the organization is FLAT and section numbers do appear, the Help Utility ignores them.

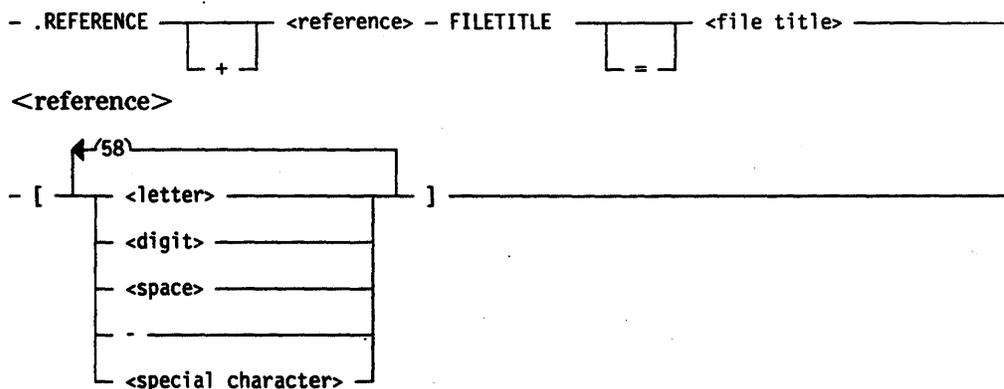
STRUCTURED Option

If you use the STRUCTURED option, the help book will have a structure that is determined by the order in which the sections of text are numbered. When help text is accessed, the section numbers that determine the structure are displayed to the user along with the section title. These section numbers also serve as keywords for the help book and can be used to display sections of help text. The structure of the help book is significant to the help text user because with structured help books, the actions PARnt, RLtd, and SUB are available as methods of moving through sections of text.

Refer to "Section Numbers and Level Numbers" in Section 3, "Structured Help Text," for more information about structured help text.

REFERENCE

The REFERENCE command links a help book reference in long help text to the file title of an online help book. This link enables the reader of help text to easily call up the online version of the referenced help book.



The REFERENCE command is optional. If the REFERENCE command is used, it can appear anywhere in the source file.

Continuation Lines

The plus sign (+) indicates that the REFERENCE command is continued on the next line. The continuation line must begin with *.REFERENCE*. A reference command can be broken over as many lines as are necessary. However, the reference, the word FILETITLE, and the names within the file title cannot be split across lines. The following example shows a REFERENCE command broken over two lines:

```
.REFERENCE + [Employees' Handbook]
.REFERENCE FILETITLE = *BOOK/EMPLOYEE/1/ENGLISH ON PACK
```

The following example shows the same command broken over four lines:

```
.REFERENCE + [Employees' Handbook]
.REFERENCE + FILETITLE =
.REFERENCE + *BOOK/EMPLOYEE
.REFERENCE /1/ENGLISH ON PACK
```

The following example shows the same command broken in a way that is not allowed and that results in an error when the Help Utility processes the file:

Commands

```
.REFERENCE + [Employees'  
.REFERENCE + Handbook]  
.REFERENCE + FILETITLE =  
.REFERENCE + *BOOK/EMPLOYEE  
.REFERENCE /1/ENGLISH ON PACK
```

File Title

The file title identifies the online help book the reference is associated with. For example:

```
.REFERENCE + [MARC Help Text] FILETITLE = BOOK/MARC/HELP  
.REFERENCE /ENGLISH ON PACK
```

In this example, the reference *[MARC Help Text]* is linked with the help book BOOK/MARC/HELP/ENGLISH ON PACK. In MARC, this enables a user to place the cursor on *[MARC Help Text]* when it appears within the help text and press the SPCFY key to enter the Topics section of BOOK/MARC/HELP/ENGLISH.

Note that if the file title of a referenced help book changes—for example, if the help book is moved to a different disk pack or its file name changes—then you must update the file title in every REFERENCE command of every help book that refers to the changed file title. To do this, you must make the appropriate changes to the affected source files and then process the updated source files with the Help Utility.

Reference

The <reference> value is the name used in the help text as a substitute for the file title of a help book. When the Help Utility processes a reference in a REFERENCE command, all blanks are removed between the left square bracket and the first character, and between the last character and the right square bracket. If two or more blanks appear between characters in a reference, the blanks are reduced to a single blank. For example, the following are treated as the same reference:

```
.REFERENCE [ MARC Help Text ] FILETITLE *BOOK/MARC/ENGLISH  
.REFERENCE [MARC Help Text] FILETITLE *BOOK/MARC/ENGLISH
```

A reference can be any combination of up to 59 letters, digits, special characters, spaces, and hyphens, which are bordered by left and right square brackets ([]). Because square brackets mark the borders of a reference, they cannot appear elsewhere in the reference. Although a reference can be up to 61 characters long (including square brackets), only the first 30 characters (including square brackets) are significant in distinguishing between references.

If a reference is more than 30 characters long (including the square brackets), the reference is shortened to 30 characters by the Help Utility. Before the reference is shortened, any extra blanks are removed, as described in the preceding

paragraph. If this new version of the reference is still more than 30 characters long, it is shortened by the removal of trailing characters other than the right square bracket. For example, the reference in the first of the following examples is truncated to the reference shown in the second:

```
.REFERENCE [this reference is more than 30 characters long]
```

```
.REFERENCE [this reference is more than]
```

Although a specific reference can be used as often as desired in the help text, a specific reference can be defined only once; that is, it can appear in only one REFERENCE command in a source file. Wherever a reference appears, it must appear entirely on a single line; it cannot be split over two or more lines.

Using the Reference to Access Another Help Book

In MARC, a user reading help text can place the cursor anywhere on a reference and press the SPCFY key to display the Topics section of the referenced help book. (The Topics section lists all the section titles of the referenced help book.) The reader can then either use the *HELP <keyword>* command or place the cursor anywhere on a section number and press the SPCFY key to get the help text for a specific topic.

For example, suppose the help text of an online version of the *Pseudocorp Catalog of Corporate Documents* makes a reference to the *Pseudocorp Employees' Handbook*, for which there is also an online version. Assume the file title of the employees' handbook help book is *BOOK/EMPLOYEE/1/ENGLISH ON PACK.

To provide a user of the catalog help text with the ability to easily access the employees' handbook text, the following command must appear in the catalog help book source file:

```
.REFERENCE + [Employees' Handbook]
.REFERENCE FILETITLE = *BOOK/EMPLOYEE/1/ENGLISH ON PACK
```

If this command appears in the source file for the catalog, then a user of that help book can place the cursor on *[Employees' Handbook]* wherever it appears in help text and press the SPCFY key. As a result, the Topics section of the employees' handbook is displayed, and the user can readily access the particular topic he or she needs.

Embedded References

When a reference is defined in a REFERENCE command, it can be used as an embedded reference anywhere within the help text. When the Help Utility processes a source file, embedded references are not truncated, nor are blanks removed from them. However, when a user selects an embedded reference, the program that retrieves the help text reads the embedded reference, removing extra blanks and truncating according to the rules outlined in the preceding paragraphs. The result is that the appropriate help book is retrieved for the user.

Creating a Catalog of Help Books

The primary use of the REFERENCE command is to make help books that are catalogs of other online help books. For example, suppose the help text of an online version of the *Pseudocorp Catalog of Corporate Documents* makes a reference to the *Pseudocorp Employees' Handbook*, for which there is also an online version. The complete source file for the *Pseudocorp Catalog of Corporate Documents* could be as follows:

```
.BOOK FILETITLE CORPORATE/DOCUMENTS
.TITLE "Pseudocorp Catalog of Corporate Documents"
.REFERENCE + [Corporate Procedures Guide]
.REFERENCE FILETITLE = *BOOK/CORP/PROCEDURES/ENGLISH ON PACK
.REFERENCE + [Managers' Guidelines]
.REFERENCE FILETITLE = *BOOK/MANAGER/1/ENGLISH ON PACK
.REFERENCE + [Administrative Handbook]
.REFERENCE FILETITLE = *BOOK/ADMINISTRATIVE/1/ENGLISH ON PACK
.REFERENCE + [Employees' Handbook]
.REFERENCE FILETITLE = *BOOK/EMPLOYEE/1/ENGLISH ON PACK
.REFERENCE + [Employees' Benefits Handbook]
.REFERENCE FILETITLE = *BOOK/BENEFITS/1/ENGLISH ON PACK
.SECTION 1 "REQUIRED SECTION COMMAND"
```

This source file is processed with the Help Utility to create a help book titled BOOK/CORPORATE/DOCUMENTS/ENGLISH.

In MARC, a user can enter the following in the Action field:

```
HELP REFERENCES IN BOOK/CORPORATE/DOCUMENTS/ENGLISH
```

The reserved keyword REFERENCES causes the References section of the designated help book to be displayed. (The References section contains a list of all references declared in the source file.) The References section is automatically created when the Help Utility processes a source file containing REFERENCE commands. The References section for the example help book appears as follows:

```
[Administrative Handbook]      [Corporate Procedures Guide]
[Employees' Benefits Handbook] [Employees' Handbook]
[Managers' Guidelines]
```

The user can place the cursor on any one of these entries and press the SPCFY key to display the Topics section for the corresponding help book.

You can create more elaborate catalog source files than the one described in the preceding paragraphs. In a catalog, help books can be grouped by subject matter and organized into sections of help text. Descriptions of the contents of a help book can be given in each section, and references can be embedded in the titles of these sections. If you create such a source file, the Topics section of the catalog can provide the user with a mechanism either for obtaining information about a referenced help book or on accessing a referenced help book.

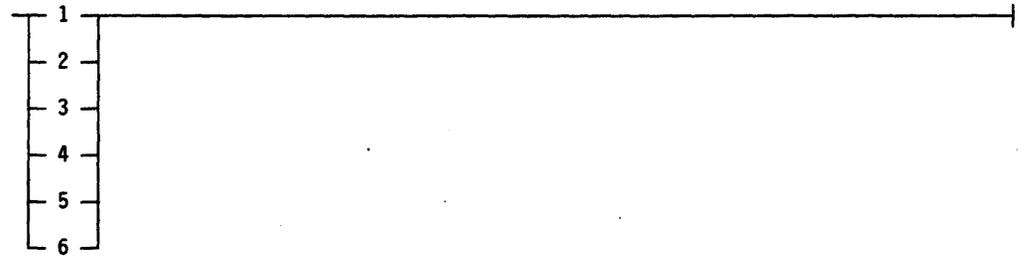
SECTION

The SECTION command enables you to divide blocks of text into sections and to assign titles and numbers to these sections.

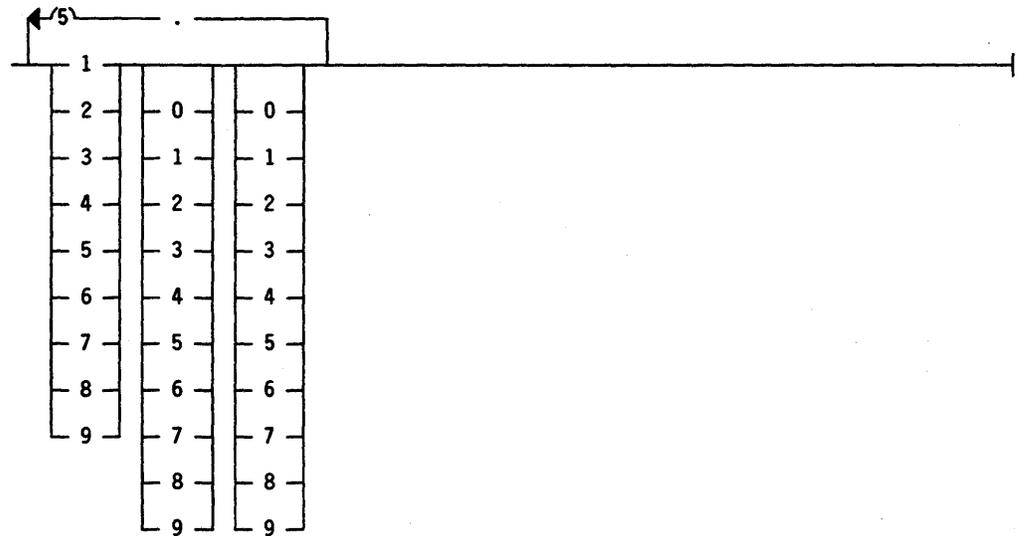
```

- .SECTION <level number> " - <section title> - "
          <section number>
  
```

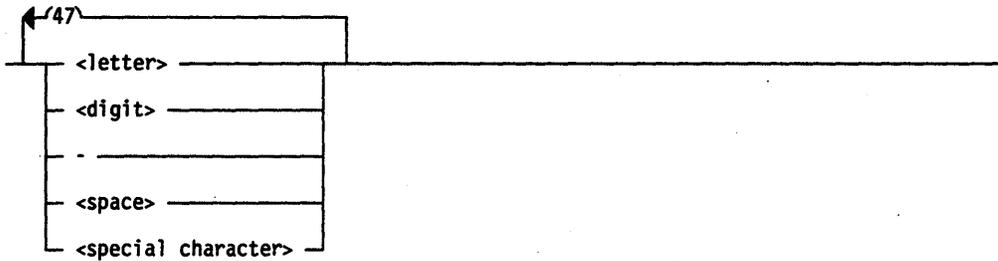
<level number>



<section number>



<section title>



A block of text between two consecutive SECTION commands is called a *section*.

The double quotation mark (") is not allowed as a special character in a section title.

The SECTION command must appear once in the source file before any text. A source file cannot contain more than 10,000 SECTION commands.

Section Numbers in a Structured Source File

When the ORGANIZATION STRUCTURED command appears in a source file, sections must be assigned section numbers. Section numbers determine the structure of the help book. The section numbers are displayed along with the section titles when help text is accessed. The section numbers also serve as keywords, so the user can go to a section of help text by using the section number associated with that section.

There are two forms of the SECTION command that allow you to assign section numbers. The form you use depends on whether AUTONUMBER or MANUALNUMBER is in effect.

With AUTONUMBER

When AUTONUMBER is in effect, the form to use is

```
.SECTION <level number> "<section title>"
```

Under AUTONUMBER, section numbers are generated automatically according to the following rules:

- The section number assigned to the current section is the section number of the immediately preceding section increased by 1 at the designated level.
- The level at which the immediately preceding section number is increased by 1 is specified by the level number that appears on the command line right after the SECTION command.

- The level number in the current SECTION command must not be more than one level deeper than the level of the immediately preceding section number.
- A section is numbered only to the depth of the level number.

For example, suppose the previous section is a level 3 section numbered 8.5.41. The following section numbers then result from the indicated SECTION commands:

Current SECTION Command	Resulting Section Number
.SECTION 1 "<section title>"	9
.SECTION 2 "<section title>"	8.6
.SECTION 3 "<section title>"	8.5.42
.SECTION 4 "<section title>"	8.5.41.1
.SECTION 5 "<section title>"	No section number is generated. Moving more than a single level deeper is not allowed. An error message is generated at run time.

Refer to "AUTONUMBER" earlier in this section for more information about the AUTONUMBER command.

With MANUALNUMBER

When MANUALNUMBER is in effect, the form to use is

```
.SECTION <section number> "<section title>"
```

Under MANUALNUMBER, the rules for numbering sections are as follows:

- The section number for a given section must be larger than the section number of the immediately preceding section. However, the section number for a given section does not have to be just one greater than the section number of the immediately preceding section.
- The level at which the current section is numbered may be less than, equal to, or one greater than the level at which the immediately preceding section was numbered. The level may not be more than one greater than the level at which the immediately preceding section was numbered.

Refer to "MANUALNUMBER" earlier in this section for more information about the MANUALNUMBER command.

Section Numbers in a Flat Source File

When the ORGANIZATION FLAT command appears in a source file, section numbers are optional. In this case, section numbers can be omitted, and the SECTION command can take the following form:

```
.SECTION "<section title>"
```

If section numbers appear in a source file when the FLAT option is in effect, they are ignored by the Help Utility.

Effects of the SECTION Command on Scrolling

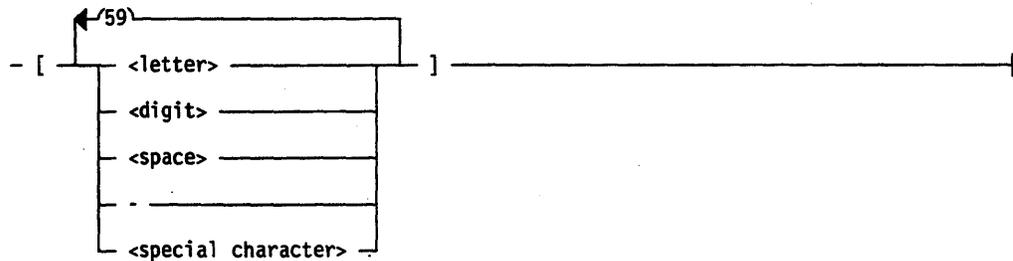
The possible effects of the SECTION command on how a user can scroll through help text are described under "Boundaries of Displayed Text" in Section 1, "Help Utility, Help Text, and Source Files."

TITLE

The TITLE command enables you to designate a book title for a help book.

```
.TITLE - " - <help book title> - "
```

<help book title>



The help book title is displayed on the first line of the Topics section of help text. (The Topics section lists all the section titles in a help book.) The help book title can be any combination of up to 60 letters, digits, special characters, spaces, and hyphens. For example, an appropriate help book title for the MARC help text might be "Menu-Assisted Resource Control (MARC) Help Text." The command that makes this title the help book title is as follows:

```
.TITLE "Menu-Assisted Resource Control (MARC) Help Text"
```

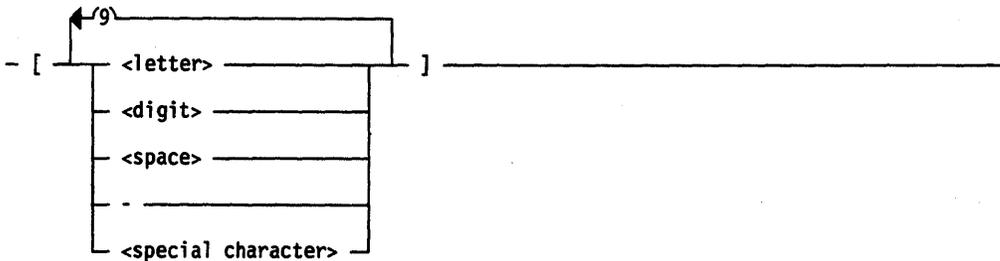
The TITLE command is optional. If the TITLE command is used, it must appear only once in the source file, and it must come before any SECTION commands and any help text. If it is not used, no help book title appears on the first line of the Topics section.

VERSION

The **VERSION** command enables you to designate for a help book a version identifier that can be used to distinguish the help book from earlier and later versions of the same help book.

```
.VERSION - " - <version ID> - "
```

<version ID>



The <version ID> is the version identifier for the help book. It appears on the right side of the first line of the Topics section of help text.

The version ID can be any combination of up to 10 letters, digits, special characters, spaces, and hyphens. For example, suppose there are three versions of a help book. It might be desirable to identify each version so that a help text user knows which version he or she is reading. In the source file for the earliest version of the help book, the following **VERSION** command could appear:

```
.VERSION "1.0"
```

In the second version of the help book, the following **VERSION** command could appear:

```
.VERSION "2.0"
```

In the latest version of the help book, the following **VERSION** command could appear:

```
.VERSION "3.0"
```

The **VERSION** command is optional. If the **VERSION** command is used, it must appear only once in the source file, and it must come before any **SECTION** commands and any help text. If the **VERSION** command is not used, no version identifier appears on the first line of the Topics section.

Section 3

Structured Help Text

When help text is structured, clearly defined relationships exist among sections of help text. These relationships reflect the structure of the text.

Long help text is structured from the top down. At the highest level, the book is split into major divisions (level 1). These major divisions can be split into subdivisions (level 2). These subdivisions can be further divided (level 3). This process can continue through level 6.

This guide can serve as an example of structured text. The table of contents reveals the organization of the guide. The sections, such as "Commands" and "Running the Help Utility," are major divisions of the entire guide and comprise level 1 in the structure of the guide.

The section you are now reading, "Structured Help Text," is a major division (level 1). Within this section, there is further structuring: "Section Numbers and Level Numbers," "Parent Topics, Related Topics, and Subtopics," and "Section Numbers and the SECTION Command" are divisions of this section (level 2). The level 2 division "Parent Topics, Related Topics, and Subtopics" is split into three further divisions: "Parent Topics," "Related Topics," and "Subtopics." These three divisions occur at level 3 in the structure of this guide because they are divisions of a level 2 division.

You, as the writer of help text, decide the exact structure of the help book. You can use all six structural levels if the logical organization of your help material seems to require it. In many cases, two or three structural levels are sufficient. Making an outline of the subjects you are covering in your help book should help you decide how many levels are called for.

Section Numbers and Level Numbers

When the help text is structured, every block of text between two consecutive SECTION commands has a unique section number assigned to it. If MANUALNUMBER is in effect, this section number must be manually assigned by you. If AUTONUMBER is in effect, the number is automatically assigned by the Help Utility according to the level number you specify in the SECTION command. These section numbers determine the structure of the help text. Sections of text are numbered as follows (n is any integer from 1 to 300):

Structured Help Text

- Level 1: n (major divisions)
- Level 2: $n.n$ (divisions of level 1)
- Level 3: $n.n.n$ (divisions of level 2)
- Level 4: $n.n.n.n$ (divisions of level 3)
- Level 5: $n.n.n.n.n$ (divisions of level 4)
- Level 6: $n.n.n.n.n.n$ (divisions of level 5)

The structural level of a section of help text is indicated by the depth of the section number assigned to that section, as shown in the following table:

Section Number	Structural Depth of Section
7	Level 1, a major division of the help book
7.9	Level 2, a division of section 7
7.9.1	Level 3, a division of section 7.9
7.9.1.8	Level 4, a division of section 7.9.1
7.9.1.8.2	Level 5, a division of section 7.9.1.8
7.9.1.8.2.1	Level 6, a division of section 7.9.1.8.2

Keep in mind that for any level, the number can be any integer from 1 to 300. The smallest section number is 1 at level 1, the first major division. The largest section number is 300.300.300.300.300.300 at level 6, with each level split into 300 divisions.

Parent Topics, Related Topics, and Subtopics

The terms *parent topic*, *related topic*, and *subtopic* are defined by the structural relationships among sections of help text. Remember that this structure is determined by the way the sections of text are numbered.

Consider, for example, a help book consisting of just four sections of text. Suppose one section of text treats the subject of mathematics in general. A second section deals with geometry, a third section deals with algebra, and a fourth section deals with trigonometry.

Parent Topics

The three sections of text about geometry, algebra, and trigonometry treat topics that are divisions of the general mathematics section. The general mathematics section is the *parent topic* of the other three sections.

More generally, the parent topic of a given section of help text is that section of text numbered one level less deep than the given section but otherwise numbered the same as the given section, as shown in the following table:

Section Number of Current Help Text	Section Number of Parent Topic Help Text
2.4	2
12.6.3	12.6
5.11.5.6.4.2	5.11.5.6.4
3	No parent topic exists for level 1 sections.

Note that there are never parent topics for level 1 sections. However, for any section at a level deeper than level 1, there is always one and only one parent topic.

Related Topics

In this example of a help book on mathematics, the sections on geometry, algebra, and trigonometry deal with *related topics*, topics that are on an equal level with each other and that have the same parent. Except at level 1, related sections have the same parent.

In a more general sense, the related topics of a given section of help text are those sections of text that are numbered exactly the same except at their deepest level, where they are numbered differently. For example:

Section Number of Current Help Text	Section Numbers of Related Help Text
5	1, 2, 3, 4, 6, 7, 8, ...
14.5.2	14.5.1, 14.5.3, 14.5.4, ...

Note that for any given section of text, related topics might not exist; their existence depends entirely on the structure of your help book.

Subtopics

In our example of a help book, the subject matter of the section on general mathematics is treated in greater detail in sections about geometry, algebra, and trigonometry. Each of these three sections is a *subtopic* of the section on general mathematics.

More generally, the subtopics of a given section of help text are those sections of text that are numbered one level deeper than the given section and that have the same numbers as the given section at all levels where they are both numbered.

Structured Help Text

For example:

Section Number of Current Help Text	Section Numbers of Help Text Subtopics
7	7.1, 7.2, 7.3, ...
1.9	1.9.1, 1.9.2, 1.9.3, ...
3.1.6.3.8.7	No subtopics exist for level 6 sections.

Note that there are never subtopics for Level 6 sections. In fact, for any given section, subtopics might not exist; their existence depends entirely on the structure of your help book.

Help Screen Actions for Structured Help Books

When a book is structured, the relationships between parent topics, related topics, and subtopics, along with the order in which the sections occur, supply the basis for a method of moving between sections of help text.

A help screen for a structured help book might look like the screen in Figure 3-1.

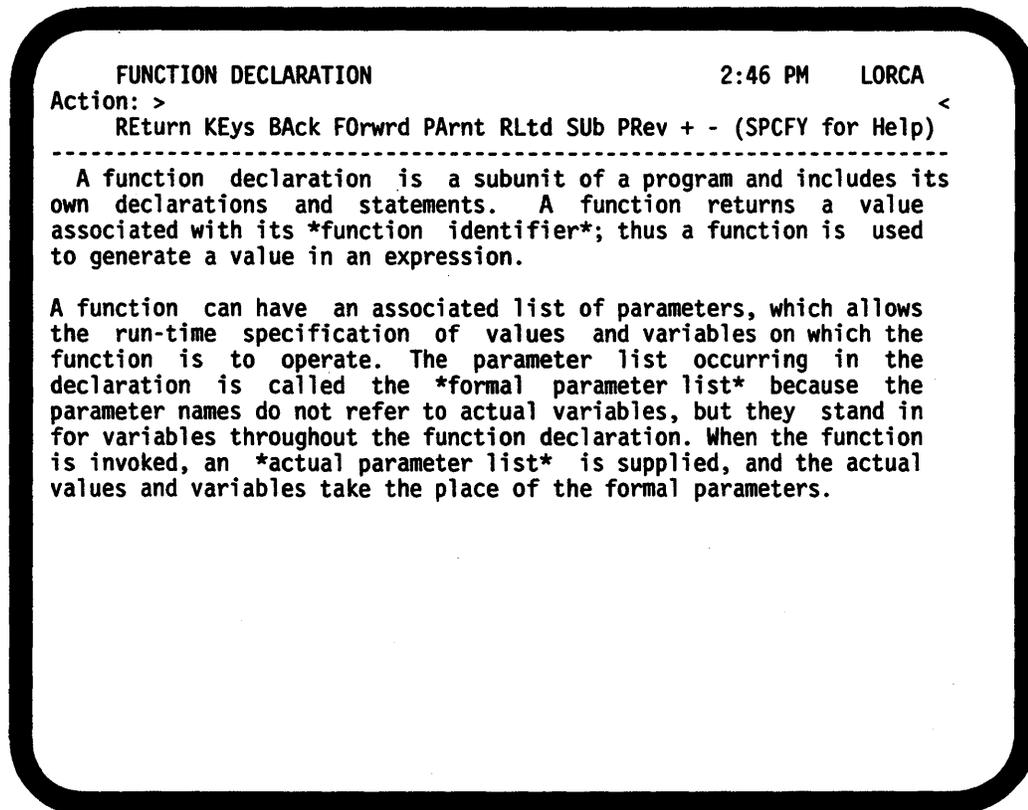


Figure 3-1. Help Screen for a Structured Help Book

The action list on this help screen includes the same actions that appear in Figure 1-2 in Section 1, “Help Utility, Help Text, and Source Files.” But because this is a help screen for a structured help book, there are five additional actions available to the user:

- If the user enters *BA* (an abbreviation of *back*) in the Action field, the section of text is presented that immediately precedes the FUNCTION DECLARATION screen in the help book structure.
- If the user enters *FO* (an abbreviation of *forward*) in the Action field, the section of text is presented that immediately follows the FUNCTION DECLARATION screen in the help book structure.

Structured Help Text

- If the user enters *PA* (an abbreviation of *parent*) in the Action field, the section of text is presented that is the parent topic of FUNCTION DECLARATION.
- If the user enters *RL* (for *related*) in the Action field, a list is presented of the names of help text sections that are related topics of FUNCTION DECLARATION.
- If the user enters *SU* (an abbreviation of *subtopic*) in the Action field, a list is presented of the names of help text sections that are subtopics of FUNCTION DECLARATION.

In those cases where the relationship indicated by the action entered does not apply, an appropriate message appears on the user's terminal.

In order for this method of moving around in help text to be useful, you must organize your source file in a logical fashion and assign section numbers that reflect the desired structure of the help book. By organizing the source file this way, you ensure that the BACK, FORWARD, PARENT, RELATED, and SUB actions provide a handy way for the user to move between sections of text.

Section Numbers and the SECTION Command

The concepts of section numbering, structured text, and levels of structure are implemented through the SECTION command. You use the SECTION command to number sections of text.

Section Numbering with MANUALNUMBER in Effect

When MANUALNUMBER is in effect, you must number each section of text by means of the SECTION command. In this case, the form of the SECTION command is

```
.SECTION <section number> "<section title>"
```

Under MANUALNUMBER, the rules for numbering sections are as follows:

- You must assign a section number as part of the SECTION command.
- The section number you choose for a given section must be larger than the section number of the immediately preceding section. However, the section number you choose for a given section does not have to be just one larger than the section number of the immediately preceding section.
- The level at which you number the current section can be less than, equal to, or one greater than the level at which the immediately preceding section was numbered. It cannot be more than one greater than the level at which the immediately preceding section was numbered.

In other words, you cannot skip levels when moving from one division to a smaller division. For example, you cannot go directly from a level 1 section to a level 3 section.

- The section numbers allowed are those discussed earlier in this section under "Section Numbers and Level Numbers."

Suppose the section number of the previous section is 3.4.1. The following are examples of valid SECTION commands for the current section:

```
.SECTION 3.4.2 "<section title>"  
.SECTION 3.4.1.1 "<section title>"  
.SECTION 3.5 "<section title>"  
.SECTION 7 "<section title>"
```

The following are examples of invalid SECTION commands for the current section:

```
.SECTION 3.4.1 "<section title>"      (Breaks the second rule)  
.SECTION 2 "<section title>"          (Breaks the second rule)  
.SECTION 3.4.1.1.1 "<section title>"  (Breaks the third rule)  
.SECTION 4.0.1 "<section title>"      (Breaks the fourth rule)
```

Section Numbering with AUTONUMBER in Effect

When AUTONUMBER is in effect, each section of text is automatically numbered according to a level number you assign the sections. In this case, the form of the SECTION command is

```
.SECTION <level number> "<section title>"
```

Under AUTONUMBER, section numbers are generated automatically by the following rules:

- You must assign a level number as part of the SECTION command.
- The section number assigned by the Help Utility to the current section is the section number of the immediately preceding section increased by 1 at the level you designate.
- The level at which the immediately preceding section number is increased by 1 is specified by the level number, which appears on the command line, right after the SECTION command.
- The level number in the current SECTION command must not be more than one level deeper than the level of the immediately preceding section number.
- A section is numbered only to the depth specified by the level number in the SECTION command.

Structured Help Text

These rules apply whether the immediately preceding section was numbered manually or automatically.

Remember, when AUTONUMBER is in effect, the number that appears in the SECTION command is not the section number for that section of text. It is the level at which that section of text is to be automatically numbered by the utility.

Suppose the previous section is a level 4 section numbered 7.5.8.1. The following section numbers then result from the indicated SECTION commands:

Current SECTION Command	Resulting Section Number
.SECTION 1 "<section title>"	8
.SECTION 2 "<section title>"	7.6
.SECTION 3 "<section title>"	7.5.9
.SECTION 4 "<section title>"	7.5.8.2
.SECTION 5 "<section title>"	7.5.8.1.1
.SECTION 6 "<section title>"	No section number is generated. Moving more than a single level deeper is not allowed. An error message is generated when the utility attempts to process the file.

Section 4

Sample Source File

This section leads you through an example of a source file—consisting of long help text only—that can be processed by the Help Utility to generate a help book. The Help Utility gives you a great deal of flexibility in the structure and composition of a source file. Although this example does not cover every allowable construct, it can help familiarize you with the commands and the correct way to use them.

The first step is to make a file. If you make your file of type **TEXT**, you can be certain the record length of your file is compatible with the Help Utility. In the Command and Edit (**CANDE**) message control system (**MCS**) you can make a new **TEXT** file by entering the following:

```
MAKE <source file title> TEXT
```

For example, you could enter *MAKE SAMPLE/SOURCE/FILE TEXT*. From there, you would proceed to writing your help text and correctly entering the embedded commands in the text. You can create the file in **CANDE** or in the A Series Editor, if the Editor utility is available to you.

If you were to use the Help Utility to process the sample file, the section numbers, section titles, and structural relationships that result would be those indicated in the following outline:

- 1 Introduction
- 2 An Example of a Related Topic
 - 2.1 Subtopics
 - 2.2 Another Subtopic
 - 2.2.1 A Level 3 Section
- 3 A Few Words about Formatting
- 4 The Last Section

Keep in mind that the section numbers appear along with the section title only in structured help books. The section numbers are shown here as examples of the way section numbers are tied to help book structure when the **ORGANIZATION STRUCTURED** option is in effect. The section titles are indented to reflect the structural relationships of the corresponding sections of text.

Sample Source File

Everything that appears between the two horizontal lines in Example 4-1 is included in the source file. If you would like to, you can enter the lines of the file exactly as they appear in Example 4-1. When you run the Help Utility on the resulting CANDE or Editor file, you create a help book.

```
.BOOK FILETITLE A/SAMPLE/BOOK

.LANGUAGE ENGLISH

.ORGANIZATION STRUCTURED

.AUTONUMBER

.SECTION 1 "Introduction"

.KEYWORD Introduction *Getting Started*
```

The first thing to note is that each command begins with a period (.) in the leftmost position (column 1) of the line. Remember the command rules: all commands must begin with a period (.) in the leftmost position (column 1) of the line, only one command can appear on each line, and at least one blank space must come between the command itself and any command items that follow it.

In this sample source file, blank lines appear before and after each set of commands. Putting blank lines in like this is entirely optional. It is done here for the purpose of making the commands stand out more. If blank lines appear in your source file, they will also appear when your help text is displayed.

This sample file begins with the commands `.BOOK FILETITLE A/SAMPLE/BOOK` and `.LANGUAGE ENGLISH`, so when the Help Utility is run on this source file, the help book created will be titled

```
BOOK/A/SAMPLE/BOOK/ENGLISH
```

The `LANGUAGE` command is unnecessary in this example, because anytime the `LANGUAGE` command does not appear in a source file, the Help Utility automatically assigns `ENGLISH` as the language of the help book.

The third command, `.ORGANIZATION STRUCTURED`, means that in the help book, the parent topic, related topic, and subtopic relationships will exist among sections of text.

The fourth command, `.AUTONUMBER`, puts automatic numbering of sections into effect. That way, the Help Utility will number sections for you at the level you indicate in the `SECTION` command.

Example 4-1. Sample Source File (cont.)

The first SECTION command in this example is .SECTION 1 Introduction. Because the AUTONUMBER command is in effect, the 1 in the SECTION command is a level number, not a section number.

When the Help Utility is run on this source file, this section of text will have the section number 1 assigned to it. Remember that section numbers serve only as a means of defining the structural relationships among sections of text when the ORGANIZATION option is STRUCTURED.

This section will have the keywords Introduction and Getting Started associated with it. What this means is that a user can employ the keyword Introduction or the keyword Getting Started to call up this section of text. Either keyword can be entered in any mixture of uppercase and lowercase letters, as long as the keyword is correctly spelled.

```
.SECTION 1 "An Example of a Related Topic"
```

```
.KEYWORD related *parent of 2.1 & 2.2*
```

This section of text will have the section number 2 assigned to it. The first section, Introduction, and this section, An Example of a Related Topic, are related topics. They are both numbered at level 1.

```
.SECTION 2 "Subtopics"
```

```
.KEYWORD *level 2 section* subtopic1
```

The level number in the SECTION command for this section makes it a subtopic of the previous section, An Example of a Related Topic. This section will have the number 2.1 assigned to it.

```
.SECTION 2 "Another Subtopic"
```

```
.KEYWORD *another level 2 section* subtopic2
```

The section number 2.2 will be assigned to this section of text. This section is also a subtopic of An Example of a Related Topic. It is a related topic of Subtopics.

```
.SECTION 3 "A Level 3 Section"
```

```
.KEYWORD *level 3 section* *section 2.2.1*
```

The section number 2.2.1 will be assigned to this section. It is a subtopic of Another Subtopic. This section has no related topics, because there are no other sections with the same parent. From here the sample file goes on to another subject, and section numbering moves up to level 1 to indicate this change.

Example 4-1. Sample Source File (cont.)

Sample Source File

.SECTION 1 "A Few Words about Formatting"

.KEYWORD formatting

The section number assigned by the Help Utility to this section will be 3. The subject of this section is the formatting of your help book.

Remember that the format of your help book is precisely the format of the source file you used to create the help book. This means that if you enter text in your source file like this, it will come out looking like this when it is displayed for the user.

So you want to be careful about the way you type the help text in your source file.

If you have the Editor Utility available to you, then you may want to use the Editor PARAGRAPH command when you are writing your source file. The PARAGRAPH command lets you set the left and right margins and, if you like, right-justify your text. (This means the right end of each line terminates in the same column, as do the lines in this manual.)

For complete descriptions of text-formatting commands available in the Editor, refer to the information on the PARAGRAPH command, the SHIFT command, and the ENTER command in the *A Series Editor Operations Guide*.

.SECTION 1 "The Last Section"

.KEYWORD end

The section number 4 will be assigned to this section. It marks the end of the sample source file. The next step after a source file is created is to process that source file with the Help Utility. Directions for processing a source file with the utility are given in Section 5, "Running the Help Utility."

Example 4-1. Sample Source File

Section 5

Running the Help Utility

Processing the Source File through CANDE

To run the Help Utility to process a source file, the Help Utility must be available on your system.

If the utility is not available, the following message appears when you try to process your source file through the Command and Edit (CANDE) message control system (MCS):

```
#UTILITY NOT PRESENT: SYSTEM/HELP/UTILITY
```

If this message appears on your terminal screen, talk with the people who supervise the computer system so they can arrange to make the Help Utility available to you.

To process your source file through CANDE, follow these steps:

1. Do one of the following:
 - In CANDE, run the utility by entering the following:
U SYSTEM/HELP/UTILITY <source file title>
Replace <source file title> with the name of the source file.
 - In CANDE:
 - Get the source file as the work file by entering the following:
GET <source file title>
Replace <source file title> with the name of the source file.
 - With the source file as the work file, run the utility by entering the following:
U SYSTEM/HELP/UTILITY
2. If the BOOK command with the FILEID option appears in the source file, equate the file identifier to the file title you want to give the help book by doing one of the following:
 - If you are running the Help Utility directly on your source file, enter
U SYSTEM/HELP/UTILITY <source file title>;FILE <file identifier>
(TITLE = <help book file title>)
 - If the source file is present as your work file, enter the following:

Running the Help Utility

```
U SYSTEM/HELP/UTILITY; FILE <file identifier>(TITLE = <help book file title>)
```

In either case, replace <source file title> with the name of your source file, replace <file identifier> with the actual file identifier that appears in the BOOK command of the source file, and replace <help book file title> with the name you want to give the help book.

Processing the Source File in Batch Mode

If you need to process your source file in batch mode, insert the following into your Work Flow Language (WFL) job:

```
COMMENT-RUN HELP UTILITY  
RUN SYSTEM/HELP/UTILITY ("<source file title>");  
FILE DEST=<help book file title>;
```

Replace <source file title> with the name of your source file, and replace <help book file title> with the name you want to give the help book.

The <help book file title> overrides any <help book file subtitle> or <file identifier> you entered as part of a BOOK command in your source file. The <help book file title> also overrides any <language name> you entered as part of a LANGUAGE command in your source file. The <help book file title> becomes the entire name of your help book file.

All systems that access help text expect the name of your help book to be of the form

```
BOOK/<help book file subtitle>/<language>
```

As a result, you must enter your <help book file title> in this form. Otherwise, the program designed to access the help book is unable to locate it.

In order for your WFL job to process your source file, SYSTEM/HELP/UTILITY and <source file title> must be available to the WFL job. If errors occur when the Help Utility processes your file, these errors are saved in a file named ERRORS, under the usercode on which the WFL job was run. To look at this file from CANDE, enter *LIST ERRORS*.

Processing the Source File through MARC

You can process a source file with the Help Utility through MARC either in command mode or by using MARC screens.

In Command Mode

The Help Utility is available through MARC. If you use MARC in command mode, you must enter the name of your source file at the time you run the utility.

You can do so in either of the following two ways:

- If the BOOK command appears in your source file with the FILETITLE option, or if the BOOK command does not appear, then to process a source file with the Help Utility, enter the following in the Action field:

```
CO RUN SYSTEM/HELP/UTILITY ("<source file title>")
```

Replace **<source file title>** with the name of the source file.

- If the BOOK command appears in your source file with the FILEID option, then file-equate the file identifier to the file title you want to give the help book by doing the following:
 1. Enter the RUN menu selection on the home menu. A new form is displayed.
 2. After **Task name**, enter *SYSTEM/HELP/UTILITY*.
 3. Place an x in the box next to **File equations are needed**. After you transmit this information, a new screen is displayed.
 4. Next to **File internal name**, enter the file identifier that appears in the BOOK command in your source file.
 5. Next to **Equation**, enter the file name you want to give the help book.

Using MARC Screens

The Help Utility is also available as a menu selection on a general MARC utilities menu. If you make this selection, MARC displays a form that asks for the source file name. This method cannot be used if the BOOK command with FILEID option appears in the source file.

Short and long help text are available on each of these screen items to answer any specific questions you might have when using MARC to run the Help Utility. For further information, refer to the *A Series Menu-Assisted Resource Control (MARC) Operations Guide*.

After Processing Is Completed

If you process your source file with the Help Utility through CANDE or MARC, the following message is displayed when the processing is complete:

```
RECORDS PROCESSED:  COMMAND = <integer>,  
TEXT = <integer>, TOTAL = <integer>
```

This message reports the number of command records, the number of text records, and the total number of records the Help Utility processed.

Whether you process your source file through CANDE, in batch mode, or through MARC, when the utility has successfully finished running, a help book file is created and automatically saved. The help book is named as follows:

Running the Help Utility

- If you processed your file through CANDE or MARC, the help book is named according to the rules described under “BOOK” and “LANGUAGE” in Section 2, “Commands.”
- If you batch-processed your source file, the help book file is given the name you designated as the FILE DEST in your WFL job.

Appendix A

Error Messages

In the context of this guide, an error is the incorrect use of one of the Help Utility commands. If the Help Utility finds errors when it processes a source file, it does not create a help book. Generally, there are two kinds of errors:

- *Syntactic errors* are errors in the way the elements of a command are arranged. For example, in the BOOK command it is a syntactic error not to follow the word BOOK with either FILETITLE or FILEID.
- *Semantic errors* are errors in the values or words of a command. For example, it is a semantic error to define \$DOLLARS as a keyword without enclosing it in quotation marks (" "), asterisks (*), or angle brackets (<>).

The Help Utility checks for specific kinds of errors in the source file when it runs. When errors occur, the lines containing the errors, and error messages corresponding to the types of errors are given. Some error messages refer to specific characters within lines in the source file. These messages can be especially helpful in correcting a source file.

If the Help Utility was run through CANDE or MARC, these error messages are sent to the terminal and appear on the screen. If the utility was run in batch mode, these error messages are saved in a file named ERRORS. To look at this file from CANDE, enter *LIST ERRORS*.

If you are in MARC, you can go to the MARC home menu and open a CANDE window by entering *CANDE* in the Choice field. Then, to look at the error messages, enter *LIST ERRORS*.

The following list presents each Help Utility error message along with an explanation of what caused the error and a recommended response. The response either tells you how to correct the error or refers you to information elsewhere in this guide.

COMMAND NOT CONTINUED CORRECTLY: COMMAND VERB NOT REPEATED

- A plus sign (+) appears immediately after the command verb in a continuable command, indicating that the command is to be continued on the next line; however, the command verb does not appear at the beginning of the next line.
- When a continuable command is split over two or more lines, each line must begin with the command verb. For example, if a REFERENCE command is continued over more than one line, each line must begin with *.REFERENCE*. Add the command verb.

COMMAND NOT CONTINUED CORRECTLY: END OF SOURCE DETECTED

- A plus sign (+) appears immediately after the command verb in a continuable command, indicating that the command is to be continued on the next line; however, this command is the last record in the source file.
- Remove the plus sign from the last record or add the missing command.

COMMAND NOT CONTINUED CORRECTLY: MISSING PERIOD IN COLUMN 1

- A plus sign (+) appears immediately after the command verb in a continuable command, indicating that the command is to be continued on the next line; however, the command verb in the next line is not preceded by a period (.).
- Add the missing period before the command.

'FILETITLE' OF REFERENCE COMMAND NOT FOUND

- The required word FILETITLE does not appear in a REFERENCE command.
- Refer to Section 2, "Commands," for the correct syntax for the REFERENCE command.

INCORRECT BOOK FILETITLE OR FILEID SPECIFICATION

- The book file subtitle or file identifier specified does not conform to the formal syntax for book file subtitles or file identifiers.
- Refer to Section 2, "Commands," under "BOOK," for the correct syntax for a book file subtitle. Refer to Section 2, "Commands," under "Basic Constructs," for the correct syntax for a file identifier.

INCORRECT <BOOK TITLE>

- The book title does not conform to the syntax rule for book titles.
- Refer to Section 2, "Commands," under "TITLE," for the correct syntax for a book title.

INCORRECT <KEYWORD>

- The keyword does not conform to the syntax rules for keywords.

- Refer to Section 2, "Commands," under "KEYWORD," for the correct syntax for a keyword.

INCORRECT LANGUAGE SPECIFICATION

- The language name does not conform to the syntax rules for language names.
- Refer to Section 2, "Commands," under "LANGUAGE," for the correct syntax for a language name.

INCORRECT <LEVEL NUMBER>

- The level number following the SECTION command is not an integer from 1 to 6, inclusive, or the level number for a given section is more than one larger than the level number for the immediately preceding section.
- Refer to "Section Numbers and Level Numbers" in Section 3, "Structured Help Text," and to "SECTION" in Section 2, "Commands," for a description of the use of level numbers.

INCORRECT ORGANIZATION

- The ORGANIZATION command appears, but neither FLAT nor STRUCTURED is specified.
- Refer to Section 2, "Commands," under "ORGANIZATION," for the correct syntax for the ORGANIZATION command.

INCORRECT <REFERENCE>

- The reference does not conform to the syntax rules for references.
- Refer to Section 2, "Commands," under "REFERENCE," for the correct syntax for a book reference.

INCORRECT REFERENCE <FILE TITLE>

- The file title in a REFERENCE command does not conform to the syntax rules for file titles.
- Refer to "Basic Constructs" in Section 2, "Commands," for the correct syntax for a file title.

INCORRECT <SECTION NUMBER>

- The section number does not conform to the syntax for section numbers.
- Refer to "Section Numbering with MANUALNUMBER in Effect" in Section 3, "Structured Help Text," and to "SECTION" in Section 2, "Commands," for the correct syntax for section numbers.

INCORRECT <SECTION TITLE>

- The section title does not follow the syntax rules for section titles.

Error Messages

- Refer to Section 2, "Commands," for the correct syntax for a section title.

INCORRECT SOURCE FILE FORMAT

- The format of the source file is incompatible with the Help Utility.
- The source file should have a text field with a length of 72 characters. If you create the source file as file type TEXT, you avoid this problem.

INCORRECT <VERSION ID>

- The version identifier does not conform to the syntax rules for version identifiers.
- Refer to Section 2, "Commands," under "VERSION," for the correct syntax for a version ID.

MAXIMUM NUMBER OF KEYWORDS EXCEEDED

- More than 20,000 keywords are defined in the source file.
- A help text source file cannot define more than 20,000 keywords. Reduce the number of keywords in your source file.

MAXIMUM NUMBER OF SECTIONS EXCEEDED

- More than 10,000 sections are defined in the source file.
- A help-text source file cannot contain more than 10,000 SECTION commands. Reduce the number of SECTION commands in your source file.

PROGRAM ENDED, TOO MANY ERRORS

- During the processing of the source file, the Help Utility found errors and ended the program.

SAME KEYWORD APPEARS IN MORE THAN ONE KEYWORD COMMAND

- A specific keyword appears in two or more KEYWORD commands.
- A keyword can be used in a KEYWORD command only once in a source file. Replace each duplicate keyword with a new keyword.

SAME REFERENCE APPEARS IN MORE THAN ONE REFERENCE COMMAND

- A specific reference appears in two or more REFERENCE commands.
- A reference can be used in a REFERENCE command only once in a source file. Replace the reference or delete the REFERENCE command.

TEXT OCCURS BEFORE FIRST SECTION COMMAND

- Text appears in the source file before the very first SECTION command.
- The only items that can appear in a source file before the first SECTION

command are the AUTONUMBER command, the BOOK command, the COMMENT command, the LANGUAGE command, and the ORGANIZATION command. Move the text to the appropriate location in the source file.

UNRECOGNIZED COMMAND ENCOUNTERED

- A period (.) appears in column 1 of a line in the source file and is followed by something other than one of the Help Utility commands.
- Add the appropriate command or delete the period.

UNRECOGNIZED INPUT ENCOUNTERED

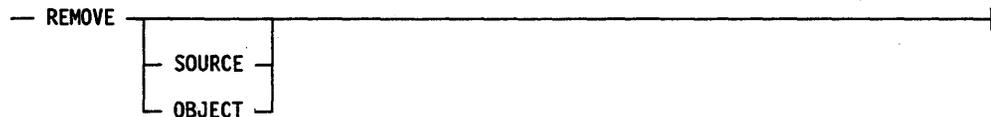
- One or more characters following a command do not coincide with accepted command syntax or semantics.
- Refer to the syntax for the command in Section 2, "Commands," and correct the characters.

Appendix B

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.

Understanding Railroad Diagrams

They all consist of paths that represent the allowable structure, constants, and variables for commands and statements.

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, and <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 that 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 that 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:

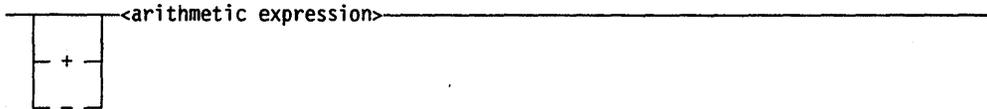
— EVENT —<identifier>—————|

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, the plus

Understanding Railroad Diagrams

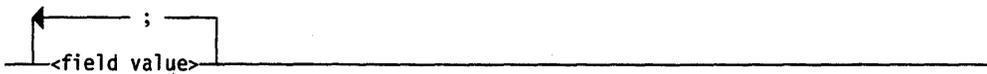
sign (+) or minus sign (-) can be entered before the required variable <arithmetic expression>, or the symbols can be disregarded because the diagram also contains an empty path.



Loop

A loop represents an item or a 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.

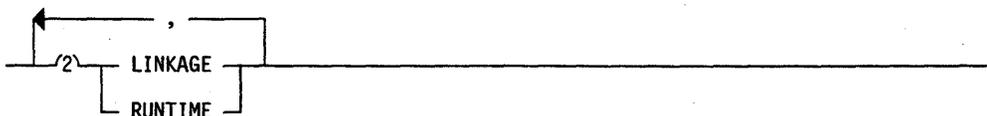


Bridge

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.

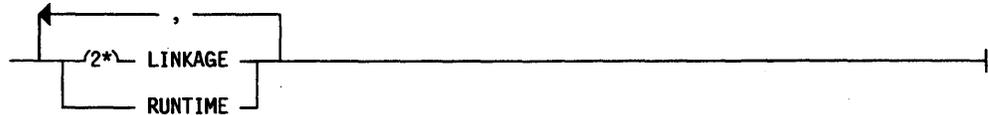
The bridge determines the number of times you can cross that point in the diagram. The bridge is an integer enclosed in curved lines (/ \). Not all loops have bridges. Those that do not can be repeated any number of times until all valid entries have been used.

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 table illustrates the constraints used in railroad diagrams.

Symbol	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 item. Indicates the constants, variables, and punctuation that must be entered in a command or statement.
	User-selected items. You select the item or items to include.
	Loop. Indicates that an item or group of items can be repeated.
	Bridge. Indicates the maximum number of times a loop can be repeated.

Following the Paths of a Railroad Diagram

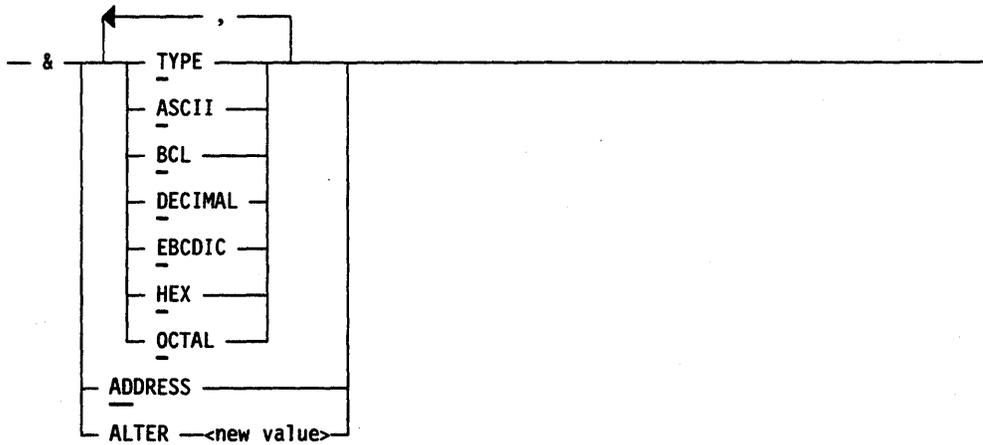
The paths of a railroad diagram lead you through the command or statement from beginning to end. Some railroad diagrams have only one path, while others have several alternate paths. The following railroad diagram indicates there is only one path that requires the constant LINKAGE and the variable <linkage mnemonic>:

Understanding Railroad Diagrams

— 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 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 followed by 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 — (—<file identifier>—) —

Sample Input

LOCK (F1)

LOCK (FILE4)

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 F1 and FILE4 are sample file identifiers.

Example 2

<open statement>



Sample Input

OPEN DATABASE1

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.

OPEN INQUIRY DATABASE1

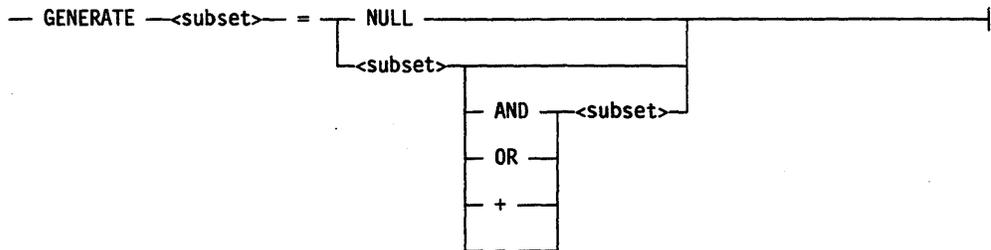
The constant OPEN is followed by the user-selected constant INQUIRY and the variable DATABASE1.

OPEN UPDATE DATABASE1

The constant OPEN is followed by the user-selected constant UPDATE and the variable DATABASE1.

Example 3

<generate statement>



Sample Input

GENERATE Z = NULL

The GENERATE constant is followed by the variable Z, an equal sign (=), and the user-selected constant NULL.

Understanding Railroad Diagrams

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

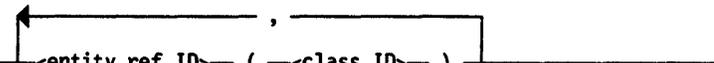
The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the AND command (from the list of user-selected items in the nested path), and a third variable, B.

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.

Example 4

<entity reference declaration>

— ENTITY REFERENCE —  (—<entity ref ID> — (—<class ID> —))

Sample Input

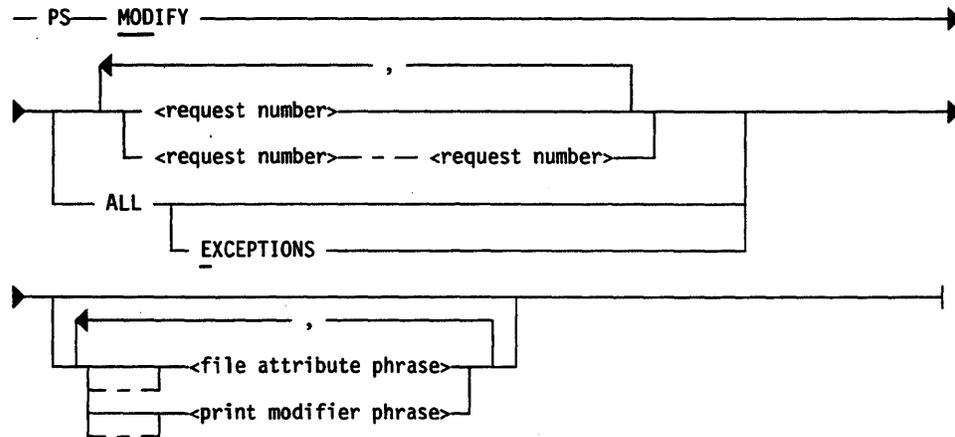
ENTITY REFERENCE ADVISOR1 (INSTRUCTOR)

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)

This sample illustrates the use of a loop by showing the input that appears in the first sample followed by a comma, the variable ADVISOR2, and the variable ASST_INSTRUCTOR. The parentheses are required.

Example 5



Sample Input

PS MODIFY 11159

The constants PS and MODIFY are followed by the variable 11159, which is a request number.

PS MODIFY 11159,11160,11163

This sample illustrates the use of a loop by showing the input that appears in the first sample followed by a comma, the variable 11160, another comma, and the final variable 11163.

PS MODIFY 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.

PS MOD ALL EXCEPTIONS

The constants PS and MODIFY are followed by the user-selected constant ALL and the user-selected constant EXCEPTIONS. Note that in this sample input, the constant MODIFY has been abbreviated.

Glossary

B

book

See help book.

book cache

An internal structure, maintained by the HELPSUPPORT library, that stores recently accessed help books.

C

CANDE

See Command and Edit.

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 line

In the Editor, the top line of the edit-mode display used for error messages, pending command displays, information about the edit line, and other information. Editor commands can be entered on the command line without the leading command character, which is the right square bracket (]) by default.

D

delimiter

A character that indicates the boundary of a field or token.

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 Unisys utility program designed to create and modify program source and data files.

Glossary

embedded command

In the Help Utility, a command that appears within a source file. Embedded commands are not displayed when help information is accessed by a user.

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.

H

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 library

A library that contains help text.

help text

Information about forms, screens, windows, fields, embedded keywords, and system messages available to a user online.

I

IDC

See Interactive Datacomm Configurator.

IMG

See Interactive Menugraph Generator.

Interactive Datacomm Configurator (IDC)

A Unisys interactive, menu-driven utility that enables the user to create, interrogate, and modify data communications network configurations.

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.

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

level number

In the Help Utility, a number designated in the SECTION command when the AUTONUMBER command is in effect; that is, when the Help Utility is automatically generating section numbers. The level number indicates at what structural level a section of help text is numbered.

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.

M

MARC

See Menu-Assisted Resource Control.

MCS

See message control system.

Menu-Assisted Resource Control (MARC)

A menu-driven interface to A Series systems that also enables direct entry of commands.

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.

metatoken

An item that appears in syntax notation as a variable item.

O

online

Pertaining to the state of being capable of immediate communication with a central computer.

Glossary

R

record

A group of logically related items of data in a file that are treated as a unit.

S

section

In the Help Utility, any block of help text between two consecutive SECTION commands. Each section has keywords and a section title linked to it.

short help text

A brief, online explanation of a form, screen, window, or field.

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.

W

WFL

See Work Flow Language.

Work Flow Language (WFL)

A Unisys language used for constructing jobs that compile or run programs on A Series systems. WFL includes variables, expressions, and flow-of-control statements that offer the programmer a wide range of capabilities with regard to task control.

Bibliography

A Series CANDE Operations Reference Manual (form 8600 1500). Unisys Corporation.

A Series Editor Operations Guide (form 8600 0551). Unisys Corporation.

A Series Interactive Menugraph Generator (IMG) Operations Guide (form 8600 0411). Unisys Corporation.

A Series Menu-Assisted Resource Control (MARC) Operations Guide (form 8600 0403). Unisys Corporation.

A Series Screen Design Facility (SDF) Operations and Programming Guide (form 1185295). Unisys Corporation.

A Series Work Flow Language (WFL) Programming Reference Manual (form 8600 1047). Unisys Corporation.

Index

A

- accessing help text, 1-9
- action field, 1-10
- actions, 3-5
 - BAck, 3-5
 - choices, 1-10, 3-5
 - FORward, 3-5
 - KEys, 1-11
 - PArent, 3-6
 - PREvious, 1-11
 - ReLated, 3-6
 - REturn, 1-11
 - scrolling, 1-11
 - SUBtopic, 3-6
- automatic section numbering command
 - (See AUTONUMBER command)
- AUTONUMBER command
 - syntax, 2-4
 - with SECTION command, 3-7

B

- BAck action, 3-5
- batch mode
 - to run the Help Utility, 5-2
- blanks
 - in composite keywords, 1-6
 - in embedded keywords, 1-4
 - in embedded references, 2-15
 - in metatoken keywords, 1-7
- book (See help book)
- BOOK command
 - FILEID option, 2-5
 - FILETITLE option, 2-5
 - syntax, 2-5
- boundaries of
 - displayed long help text, 1-11
 - short help text, 1-12

C

- characters, special
 - in composite keywords, 1-5
- CANDE (See Command and Edit)
- Command and Edit (CANDE)
 - to equate files, 5-1
 - to run the Help Utility, 5-1
- command line, 2-1
- commands, 2-1
 - AUTONUMBER, 2-4
 - BOOK, 2-5
 - COMMENT, 2-7
 - KEYWORD, 2-8
 - LANGUAGE, 2-10
 - MANUALNUMBER, 2-11
 - ORGANIZATION, 2-12
 - REFERENCE, 2-13
 - SECTION, 2-18
 - syntax rules, 2-1
 - TITLE, 2-21
 - VERSION, 2-22
- COMMENT command
 - syntax, 2-7
 - <comment text>, 2-7
 - <composite keyword>, 2-8
- composite keyword format, 1-5
- composite keywords
 - blanks in, 1-6
 - format of, 1-5
 - spaces in, 1-6
- creating
 - a help book, 1-2
 - a source file, 4-1

D

- delimiter, 1-5
 - of composite keyword, 1-5
 - of metatoken keyword, 1-8
- <digit>, 2-2

E

embedded keywords, 1-4
 example, 1-11
 removing blanks from, 1-4
 removing spaces from, 1-4
 syntax, 2-9
embedded references
 removing blanks from, 2-15
 removing spaces from, 2-15
entering keywords, 1-5
Error, A-1
 messages, A-1
 semantic, A-1
 syntactic, A-1
ERRORS file, 5-2
 displaying, 5-2, A-1
examples
 help screen, 1-10, 3-5
 source file, 4-2
 structured help text, 3-1

F

<family name>, 2-3
file equation
 with BOOK command, 2-5
 with CANDE, 5-1
 with MARC, 5-3
 with WFL, 5-2
<file identifier>, 2-2
 in BOOK command, 2-5
<file name>, 2-2
<file title>, 2-2
file title
 in the REFERENCE COMMAND, 2-16
FILEID (BOOK command option)
 syntax, 2-5
FILETITLE (BOOK command option)
 syntax, 2-5
FLAT (ORGANIZATION command option)
 syntax, 2-12
FORward action, 3-5

H

HELP action, 1-4
help and short help text for the Screen
 Design Facility (SDF), 1-1

help book
 availability of new
 the MARC NEW BOOK
 command, 1-15
 creating, 1-2
 value of SECURITYTYPE
 attribute, 1-15
<help book file subtitle>, 2-5
help screen, 1-10
 actions, 1-10, 3-5
 example, 1-10, 3-5
help text, 1-2
 boundaries, 1-11
 structured, 3-1
Help Utility
 running (*See* running the Help Utility)

K

KEys action, 1-11
KEYS (reserved keyword), 1-4
KEYWORD command
 syntax, 2-8
keyword directory, 1-2
keyword formats
 composite, 1-5
 metatoken, 1-7
 simple, 1-5
keywords, 1-3
 as a boundary of displayed text, 1-11
 blanks
 in composite keywords, 1-6
 in metatoken keywords, 1-7
 embedded, 1-4
 entering, 1-5
 formats (*See* keyword formats)
 length, 2-9
 multiple, 1-8
 reserved, 1-4
 secondary, 1-4, 1-11
 spaces
 in composite keywords, 1-6
 in metatoken keywords, 1-7
 synonyms, 1-4

L

LANGUAGE command
 syntax, 2-10
<language name>, 2-10

<level number>, 2-20
 level numbers
 of structured help text, 3-1

M

MANUALNUMBER command
 syntax, 2-11
 MARC (*See* Menu-Assisted Resource Control)
 MARC NEW BOOK command, 1-15
 Menu-Assisted Resource Control (MARC)
 to equate files, 5-3
 to run the Help Utility, 5-3
 <metatoken keyword>, 2-8
 metatoken keywords
 blanks in, 1-7
 format of, 1-7
 spaces in, 1-7
 multiple keywords, 1-8

N

<name>, 2-2
 NEW BOOK
 MARC command, 1-15

O

ORGANIZATION command
 FLAT option, 2-12
 STRUCTURED option, 2-12
 syntax, 2-12

P

PArent action, 3-6
 parent topic, 3-3
 PRevious action, 1-11
 processing a source file (*See* running the Help Utility)

R

record, 1-9
 <reference>, 2-13

REFERENCE command
 file title in the REFERENCE command, 2-16
 syntax, 2-13
 REFERENCES (reserved keyword), 1-4
 Related (RL) action, 3-6
 related topic, 3-3
 reserved keywords, 1-4
 REturn action, 1-11
 running the Help Utility
 in batch mode, 5-2
 through CANDE, 5-1
 through MARC, 5-3

S

sample source file, 4-1
 Screen Design Facility (SDF)
 help and short help text, 1-1
 scrolling, 1-11
 secondary keywords, 1-4
 as a boundary of displayed text, 1-11
 section, 1-8
 SECTION command
 syntax, 2-18
 with AUTONUMBER, 2-18, 3-7
 with MANUALNUMBER, 2-19, 3-6
 <section number>, 2-20
 section number
 of structured help text, 3-1
 section numbering
 with AUTONUMBER, 3-7
 with MANUALNUMBER, 3-6
 section numbering command, 3-6
 section title, 1-8
 <section title>, 2-20
 SECURITYTYPE attribute
 value for help books, 1-15
 short help text, 1-2
 with SECTION command
 boundaries, 1-12
 <simple keyword>, 2-8
 simple keyword format, 1-5
 source file, 1-2
 creation, 4-1
 example, 4-2
 file type, 1-9
 processing in batch mode, 5-2
 processing through CANDE, 5-1
 processing through MARC, 5-3
 <space>, 2-2

Index

spaces

- in composite keywords, 1-6
- in embedded keywords, 1-4
- in embedded references, 2-15
- in metatoken keywords, 1-7

<special character>, 2-2

special characters

- in composite keywords, 1-5

structured help text, 3-1

STRUCTURED (ORGANIZATION
command option)

- syntax, 2-12

SUBtopic action, 3-6

subtopics, 3-3

synonyms

- keywords as, 1-4

SYSTEM/HELP, 1-14

T

TEACH action, 1-4

TITLE command

- syntax, 2-21

TOPICS (reserved keyword), 1-4

U

<usercode>, 2-3

V

VERSION command

- syntax, 2-22

W

WFL (*See* Work Flow Language)

Work Flow Language (WFL)

- to equate files, 5-2

- to run the Help Utility, 5-2

Help Us To Help You

Publication Title _____

Form Number _____ Date _____

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

Addition Deletion Revision Error

Comments _____

Name _____

Title _____ Company _____

Address (Street, City, State, Zip) _____

Telephone Number _____

Help Us To Help You

Publication Title _____

Form Number _____ Date _____

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

Addition Deletion Revision Error

Comments _____

Name _____

Title _____ Company _____

Address (Street, City, State, Zip) _____

Telephone Number _____

Help Us To Help You

Publication Title _____

Form Number _____ Date _____

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

Addition Deletion Revision Error

Comments _____

Name _____

Title _____ Company _____

Address (Street, City, State, Zip) _____

Telephone Number _____



No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Publications, M/S 170
25725 Jeronimo Road
Mission Viejo, CA 92691-9826 USA



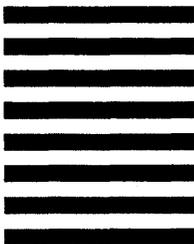
No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Publications, M/S 170
25725 Jeronimo Road
Mission Viejo, CA 92691-9826 USA



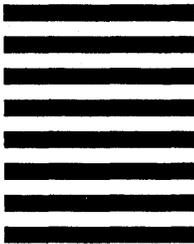
No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Publications, M/S 170
25725 Jeronimo Road
Mission Viejo, CA 92691-9826 USA





8600510-000