

# VAX DOCUMENT Using Global Tags

Order Number: AA-JT84C-TE

**February 1991**

This manual describes, for the new or experienced user, the VAX DOCUMENT doctype-independent, or global, tags used in all types of VAX DOCUMENT documents.

**Revision/Update Information:** This revised manual supersedes the *VAX DOCUMENT User Manual, Volume 1 Version 1.1* (Order Number AA-JT84B-TE).

**Operating System and Version:** VMS Version 5.3 or higher

**Software Version:** VAX DOCUMENT Version 2.0

---

**First printing, July 1987**  
**Revised, July 1988**  
**Revised, February 1991**

---

Copyright ©1987, 1988, 1991 Digital Equipment Corporation

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.


The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem-10	PDP	VT
DECSYSTEM-20	PDT	
DECUS	RSTS	
DECwriter	RSX	

The following is a third-party trademark:

PostScript is a registered trademark of Adobe Systems, Inc.

This document is available on CDROM.

ZK5352

---

# Contents

---

PREFACE vii

---

NEW FEATURES xi

---

## CHAPTER 1 GLOBAL TAG DESCRIPTIONS 1-1

<ABSTRACT>	1-2
<ACCENT>	1-3
<ALIGN_AFTER>	1-4
<ALIGN_CHAR>	1-5
<ALIGN_NUMBER>	1-8
<AMPERSAND>	1-10
<APPENDIX>	1-11
<BACKSLASH>	1-13
<BOX>	1-14
<CALLOUT>	1-15
<CALLOUT_REF>	1-16
<CALLOUTS>	1-17
<CENTER_LINE>	1-19
<CHAPTER>	1-21
<CHEAD>	1-22
<CHECK_FOR_INCLUSION>	1-23
<CO>	1-24
<CODE_EXAMPLE>	1-25
<COMMENT>	1-29
<CONDITION>	1-31
<CONTENTS_FILE>	1-34
<COPYRIGHT_DATE>	1-36
<COPYRIGHT_PAGE>	1-37
<CP>	1-38
<CPAREN>	1-39
<DATE>	1-40
<DEFINE_BOOK_NAME>	1-42
<DEFINE_SYMBOL>	1-44
<DEFINITION_LIST>	1-46
<DEFINITION_LIST_HEAD>	1-48
<DEFLIST_DEF>	1-49
<DEFLIST_ITEM>	1-50
<DELAYED>	1-51
<DOCTYPE>	1-53
<DOUBLE_QUOTE>	1-54
<ELEMENT>	1-56
<ELLIPSIS>	1-58
<EMPHASIS>	1-60
<ENDCOPYRIGHT_PAGE>	1-62
<ENDPART_PAGE>	1-63
<ENDTITLE_PAGE>	1-64

## Contents

<EXAMPLE>	1-65
<EXAMPLE_ATTRIBUTES>	1-71
<EXAMPLE_FILE>	1-75
<EXAMPLE_SPACE>	1-76
<FCMD>	1-78
<FIGURE>	1-81
<FIGURE_ATTRIBUTES>	1-87
<FIGURE_FILE>	1-89
<FIGURE_SPACE>	1-93
<FILE_SPEC>	1-96
<FINAL_CLEANUP>	1-98
<FOOTNOTE>	1-101
<FOOTNOTE_TEXT>	1-104
<FOOTREF>	1-106
<FORMAT>	1-108
<FPARM>	1-109
<FPARMS>	1-110
<FRONT_MATTER>	1-111
<GDEF>	1-113
<GLOSSARY>	1-114
<GREF>	1-116
<GTERM>	1-117
<HEAD>	1-118
<HEAD <sub>n</sub> >	1-119
<HELLIPSIS>	1-122
<HYPHENATE>	1-123
<ICON>	1-124
<ICON_FILE>	1-127
<ICON_TEXT>	1-129
<INCLUDE>	1-130
<INCLUDES_FILE>	1-131
<INDEX_FILE>	1-133
<INTERACTIVE>	1-135
<KEEP>	1-138
<KEYWORD>	1-139
<LANGUAGE>	1-140
<LE>	1-143
<LINE>	1-144
<LINE_ART>	1-147
<LIST>	1-149
<LITERAL>	1-157
<LOWERCASE>	1-159
<MARK>	1-160
<MATH>	1-163
<MATH_CHAR>	1-178
<MCS>	1-186
<NESTED_TABLE_BREAK>	1-189
<NEWTERM>	1-191
<NOTE>	1-192
<OPAREN>	1-193
<ORDER_NUMBER>	1-194
<P>	1-195
<PAGE>	1-197

<PARENDCHAR>	1-199
<PART>	1-200
<PART_PAGE>	1-202
<PREFACE>	1-203
<PREFACE_SECTION>	1-205
<PRINT_DATE>	1-206
<PROFILE>	1-207
<QUOTE>	1-209
<REFERENCE>	1-211
<REVISION>	1-215
<REVISION_INFO>	1-217
<RIGHT_LINE>	1-218
<RULE>	1-220
<S>	1-221
<SAMPLE_TEXT>	1-223
<SET_APPENDIX_LETTER>	1-224
<SET_CHAPTER_NUMBER>	1-225
<SET_CONDITION>	1-226
<SET_FIGURE_FILE_SPACING_DEFAULT>	1-228
<SET_TABLE_ROW_BREAK_DEFAULT>	1-230
<SINGLE_QUOTE>	1-232
<SPAN>	1-233
<SPECIAL_CHAR>	1-236
<SUBHEADn>	1-238
<TABLE>	1-239
<TABLE_ATTRIBUTES>	1-242
<TABLE_FILE>	1-246
<TABLE_HEADS>	1-248
<TABLE_KEY>	1-250
<TABLE_KEYREF>	1-253
<TABLE_ROW>	1-254
<TABLE_ROW_BREAK>	1-256
<TABLE_SETUP>	1-259
<TABLE_SPACE>	1-261
<TABLE_UNIT>	1-263
<TABLE_UNIT_HEADS>	1-265
<TAG>	1-267
<TITLE>	1-268
<TITLE_PAGE>	1-269
<U>	1-270
<UNDERLINE>	1-272
<UPDATE_RANGE>	1-274
<UPPERCASE>	1-277
<USER_I_MESSAGE>	1-278
<USER_W_MESSAGE>	1-280
<VALID_BREAK>	1-282
<VALID_TABLE_ROW_BREAK>	1-284
<VARIABLE>	1-285
<VBAR>	1-286
<X>	1-287
<XS>	1-294
<XSUBENTRY>	1-296
<Y>	1-298

## Contents

---

<b>APPENDIX A</b>	<b>VAX DOCUMENT COMMAND SUMMARY</b>		<b>A-1</b>
	DOCUMENT[/FORMATTER]	A-2	
	DOCUMENT/GENERATE_SYMBOL	A-17	

---

<b>APPENDIX B</b>	<b>MESSAGES</b>		<b>B-1</b>
-------------------	-----------------	--	------------

---

<b>B.1</b>	<b>DOCUMENT COMMAND MESSAGES</b>		<b>B-2</b>
<b>B.2</b>	<b>TAG TRANSLATOR MESSAGES</b>		<b>B-9</b>
<b>B.3</b>	<b>TEXT PROCESSOR MESSAGES</b>		<b>B-59</b>
<b>B.4</b>	<b>DEVICE CONVERTER MESSAGES</b>		<b>B-82</b>
<b>B.5</b>	<b>INDEX FACILITY MESSAGES</b>		<b>B-95</b>

---

## INDEX

---

## TABLES

---

<b>1</b>	<b>Conventions Used in this Manual</b>		<b>viii</b>
<b>1-1</b>	<b>&lt;MATH&gt; Expressions</b>		<b>1-166</b>
<b>1-2</b>	<b>Tags for Mathematical Functions</b>		<b>1-171</b>
<b>1-3</b>	<b>&lt;MATH_CHAR&gt; Keywords and Symbols</b>		<b>1-178</b>
<b>1-4</b>	<b>Element Types and Default Output of Symbol Names</b>		<b>1-212</b>
<b>1-5</b>	<b>Interaction of &lt;X&gt; Tag Arguments and Command Line Indexing Options</b>		<b>1-290</b>
<b>A-1</b>	<b>Default File Types</b>		<b>A-3</b>

---

# Preface

---

## Document Structure

This book describes how to use the VAX DOCUMENT **global** tags. Global tags are SDML tags that you can use in any **doctype**, with a few exceptions. *VAX DOCUMENT Using Doctypes and Related Tags* describes the rest of the VAX DOCUMENT tags: the **doctype-specific** tags. Doctype-specific tags are tags restricted for use in certain doctypes.

For example, the <LIST> tag is a global tag that you can use in any doctype. The <SALUTATION> tag, however, is a doctype-specific tag that you can use only in the LETTER doctype.

- Chapter 1 provides a description of all global tags.
- Appendix A is a summary of the VAX DOCUMENT commands.
- Appendix B lists and explains the messages you see when processing source files.

---

## Intended Audience

This book is intended for writers, editors, and general users who wish to produce technical manuals, brochures, business letters, overhead slides, or online documentation using VAX DOCUMENT. Familiarity with a text editor is presumed, as is a basic knowledge of the VMS operating system.

---

## Associated Documents

This manual is part of the VAX DOCUMENT documentation set that includes the following books:

- *VAX DOCUMENT Producing Online and Printed Documentation*
- *VAX DOCUMENT Using Doctypes and Related Tags*
- *VAX DOCUMENT Designing Doctypes*
- *VAX DOCUMENT Tags Quick Reference*
- *VAX DOCUMENT Quick Reference Card*
- *VAX DOCUMENT Graphics Editor User's Guide*
- *VAX DOCUMENT Installation Guide*

---

## Conventions

Chapter 1 contains, in a fixed order, a discussion of each global tag. First, the name of the tag is followed by a brief overview that describes the purpose of the tag. Following the overview is a syntax section that displays the syntax of the tag: any required or optional arguments. Following the syntax section is an arguments section that describes any

## Preface

arguments, any related tags, any restrictions on the use of the tag, and any required terminating tags.

**Note:** The related tags, restrictions, and required terminator sections are omitted if there is no relevant information.

The category of “related tag” is defined broadly. A tag is related to the tag under discussion if one of the following criteria is met:

- It is required for use of the tag under discussion.
- It marks a text element of the same kind as the tag under discussion.
- It is commonly used with the tag under discussion.
- It is a tag with a similar function to the tag under discussion.

Next is a description section. The description expands the overview and presents more detailed information on using the tag.

The discussion of a tag concludes with an examples section, with at least one example or a reference to an example. The example shows how to code the tag in an SDML file, and, when possible, shows what the formatted result is when the file is processed for printing.

Each output example is introduced by a form of the sentence, “This example produces the following output:”. The examples were processed using a local doctype based on one of the six SOFTWARE doctypes, SOFTWARE.REFERENCE. Your output of these examples may vary, however, depending on the doctype you use and on whether any doctype modifications have been made to your local installation of VAX DOCUMENT.

Table 1 lists the typographical conventions used in this manual.

**Table 1 Conventions Used in this Manual**

Convention	Meaning
.	In examples, a vertical ellipsis represents the omission of data that the system displays in response to a command or to data you enter.
...	In examples, a horizontal ellipsis indicates that you can enter additional parameters, values, or other information. In tag syntax, a horizontal ellipsis indicates that arguments to the tag have been omitted.
<b>TERM</b>	A term that appears in bold type is defined in the glossary in the <i>VAX DOCUMENT Producing Online and Printed Documentation</i> .
<TAG>(argument)	Parentheses enclose an argument to a tag. A lowercase word as an argument to a tag indicates that a user-specified argument must be entered.
<TAG>[(argument)]	Brackets indicate that the enclosed argument to the tag is optional.



**Table 1 (Cont.) Conventions Used in this Manual**

Convention	Meaning
<TAG>[(argument-1 [ \argument-2])]	This tag syntax indicates that both arguments are optional. Only if you use <i>argument-1</i> , however, do you have the choice of using <i>argument-2</i> .
<TAG>[[([argument-1] \ [argument-2])]	This tag syntax indicates that both arguments are optional. You can use either argument as the first argument.
<TAG> [( { argument-1 KEYWORD-1 [ \KEYWORD-2 ] } )]	This tag syntax indicates that all arguments are optional. The braces indicate a choice between <i>argument-1</i> and KEYWORD-1. You must choose your first argument. If you choose KEYWORD-1, you also have the option of indicating KEYWORD-2 as the second argument.
<TAG>(KEYWORD)	An uppercase word within an argument to a tag indicates that the word is a keyword, and that you must enter the specific keyword.
<TAG>( \KEYWORD)	A keyword following a backslash in an argument to a tag indicates that that keyword must follow a backslash.
\	A backslash separates multiple arguments to a tag.



---

## New Features

The following list contains the new features of VAX DOCUMENT Version 2.0:

- The considerations for Bookreader output, if different from printed output, are noted for each affected tag.
- The <LANGUAGE> tag is a new tag for creating documentation in several languages.
- A new argument exists for the /DEVICE\_CONVERTER qualifier on the DOCUMENT command line: /DEVICE\_CONVERTER=DUPLEX. This argument creates a two-sided POSTSCRIPT document for printing on a duplex printer or for copying on a two-sided copier.
- Monospaced fonts defined for all the sizes are provided for text fonts in the POSTSCRIPT destination.
- The new attributes, BOTH and NOMASTER, exist for indexing with the <X> tag. BOTH specifies an index entry that you want to appear in both a single-document and master index. NOMASTER specifies an index entry that you want to appear only in a single-document index and not in a master index.
- You can refer to headings in your preface from other parts of your document. The headings in a preface, however, do not have numbers and do not appear in the table of contents.

The documentation set has been extensively revised. A list of the major changes in this manual follows:

- An improved index.
- More and improved examples for each tag.
- A clearer description of each tag.
- A glossary of terms exists in *VAX DOCUMENT Producing Online and Printed Documentation*.
- The first eight chapters of the *VAX DOCUMENT User Manual, Volume 1*, Version 1.1 is now condensed and part of *VAX DOCUMENT Producing Online and Printed Documentation*.



# 1

---

## Global Tag Descriptions

This chapter provides descriptions of all global tags. The tags are listed alphabetically. **Terminating tags**, which are tags that you must use to terminate a tag (such as <LIST> and its terminating tag <ENDLIST>), are not listed separately, except when the terminating tag has an argument. Each terminating tag is listed in the description of the tag it terminates. You can also find terminating tags listed in the index.

## <ABSTRACT>

---

## <ABSTRACT>

Begins a summary description of a document on the title page or part page of the document.

---

**SYNTAX**      <ABSTRACT>[(*heading*)]

---

**ARGUMENTS**      *heading*  
Specifies a heading for the abstract.

---

**related tags**

- <FRONT\_MATTER>
- <PART\_PAGE>
- <TITLE\_PAGE>

---

**restrictions**      Valid only in the context of a <TITLE\_PAGE> or <PART\_PAGE> tag.

---

**required terminator**      <ENDABSTRACT>

---

**DESCRIPTION**      The <ABSTRACT> tag begins a summary description of a document on the title page or part page of the document. A part page is a dividing page between major sections of a document.

---

**EXAMPLE**      See the example in the discussion of the <PART> tag.

---

## <ACCENT>

Supplies a freestanding acute, grave, or umlaut accent mark.

---

### SYNTAX

<ACCENT>( { ACUTE  
GRAVE  
UMLAUT } )

---

### ARGUMENTS

**ACUTE**  
**GRAVE**  
**UMLAUT**

Specifies one of these keywords to indicate the type of accent mark.

---

### related tags

- <MCS>
- <PARENDCHAR>

---

### restrictions

Invalid in math.

---

### DESCRIPTION

The <ACCENT> tag supplies a freestanding acute, grave, or umlaut accent mark. This tag has no effect for Bookreader output.

---

### EXAMPLE

The following example shows how to use the <ACCENT> tag. The <PARENDCHAR> tag in this example simply provides better spacing of the accent characters within the parentheses. See the <PARENDCHAR> tag description for more information.

```
<P>Accents used in the Digital Multinational Character Set include the
following:
<LIST>(UNNUMBERED)
<LE>Acute accents <PARENDCHAR>(<ACCENT>(ACUTE))
<LE>Grave accents <PARENDCHAR>(<ACCENT>(GRAVE))
<LE>Umlauts <PARENDCHAR>(<ACCENT>(UMLAUT))
<ENLIST>
```

This example produces the following output:

Accents used in the DEC Multinational Character Set include the following:

- Grave accents ( ` )
- Acute accents ( ´ )
- Umlauts ( ¨ )

## <ALIGN\_AFTER>

---

## <ALIGN\_AFTER>

Allows you to position text in a list after the text specified in the argument.

---

**SYNTAX**      <ALIGN\_AFTER>(text)

---

**ARGUMENTS**    *text*  
Specifies the string of text after which you want to align the text that follows.

---

**related tags**

- <ALIGN\_NUMBER>
- <ALIGN\_CHAR>
- <LINE>(INDENT)
- <LIST>(STACKED)

---

**restrictions**    Invalid in math.

---

**DESCRIPTION**    The <ALIGN\_AFTER> tag allows you to position text in a list after the text specified in the argument.

---

**EXAMPLE**        The following example shows how to create a stacked list where the text is aligned after the string "UN."

```
<LIST> (STACKED\BRACES)
<LE><ALIGN_AFTER> (UN) ALPHABETICAL
<LE>UNNUMBERED
<ENDLIST>
```

This example produces the following output:

```
{    ALPHABETICAL    }
{ UNNUMBERED        }
```



---

## <ALIGN\_CHAR>

Identifies a nonprinting character you want to use to align numeric information in text or within a column of a list or table.

---

### SYNTAX

<ALIGN\_CHAR>(character [\ DELTA])

---

### ARGUMENTS

#### *character*

Specifies the character to be used for alignment. The character must be one of the following special characters:

```
{ } ^ # $  
% * ~ @  
[ ] < >
```

#### **DELTA**

This is an optional keyword argument. It specifies that the character indicated for alignment be replaced in your output with a delta character to indicate spacing.

---

### related tags

- <ALIGN\_AFTER>
- <ALIGN\_NUMBER>
- <TABLE\_ROW>

---

### restrictions

Invalid in math, in any major section tag (such as <CHAPTER>, <APPENDIX>, <SECTION>, and <COMMAND\_SECTION>), and in any heading tag (such as <HEAD1>, <SUBHEAD1>, and <CHEAD>).

Do not nest an <ALIGN\_CHAR> tag in another <ALIGN\_CHAR> tag.

Invalid in monospaced examples. Aligned characters can be referenced in an example, but you must place the <ALIGN\_CHAR> and <ENDALIGN\_CHAR> tags outside the example.

---

### required terminator

<ENDALIGN\_CHAR>

---

### DESCRIPTION

The <ALIGN\_CHAR> tag identifies a nonprinting character you want to use to align numeric information in text or within a column of a list or table. This character signifies a space, so that the tag translator replaces each occurrence of the character with a space. The space is the same width as the numeric characters in the font that is active. (All numeric characters of a font have a uniform width.)

## <ALIGN\_CHAR>

Items in a column are aligned on the left by default. You may want to display a column of numbers aligned on the right, however. If so, use the <ALIGN\_CHAR> tag to define “#,” for example, as the alignment character. You make all the numeric entries an equal length by prefixing the shorter entries with the alignment character. Because the numeric characters are of uniform width, they will then align on the right.

You can also use the alignment character to add space within the number. For example, you could separate the digits of a number at the thousands position.

---

### EXAMPLES

The following example defines the number sign as the alignment character in a table.

```
1 <ALIGN_CHAR> (#)
  <TABLE>
  <TABLE_SETUP> (2\16)
  <TABLE_ROW> (0#123#456#781\01234567.89)
  <TABLE_ROW> (#####35#279\#####6.0)
  <TABLE_ROW> (#####4#341\###1429.857)
  <ENDTABLE>
  <ENDALIGN_CHAR>
```

This example produces the following output:

0 123 456 781	01234567.89
35 279	6.0
4 341	1429.857

The following example shows the table used in the previous example without using the alignment character.

```
2 <TABLE>
  <TABLE_SETUP> (2\16)
  <TABLE_ROW> (0123456789\01234567.89)
  <TABLE_ROW> (35279\6.0)
  <TABLE_ROW> (4361\1429.857)
  <ENDTABLE>
```

This example produces the following output:

0123456789	01234567.89
35279	6.0
4361	1429.857

The following example defines the number sign as the alignment character in a list.

```
3 <ALIGN_CHAR> (#)
  <LIST> (numbered)
  <LE> (0#123#456#781)
  <LE> (#####35#279)
  <LE> (#####4#341)
  <ENDLIST>
  <ENDALIGN_CHAR>
```

This example produces the following output:

```
1 0 123 456 781
2      35 279
3      4 361
```

The following example shows the list used in the previous example without using the alignment character.

```
4 <LIST>(numbered)
  <LE>0123456781
  <LE>35279
  <LE>4341
  <ENDLIST>
```

This example produces the following output:

```
1 0123456781
2 35279
3 4341
```

The following example shows how you can represent blank output characters in text. The <KEEP> tag ensures that the text specified in its argument will not be broken across lines. See the <KEEP> tag description for more information.

```
5 <P>The sum may be padded on the left with blanks, resulting in
  <ALIGN_CHAR>(#)
  <KEEP>(<SINGLE_QUOTE>###7.56<SINGLE_QUOTE>)
  <ENDALIGN_CHAR>
```

This example produces the following output:

The sum may be padded on the left with blanks, resulting in ' 7.56'

The following example shows how you can represent delta output characters in text.

```
6 <P>The sum may be padded on the left with delta characters, resulting in
  <ALIGN_CHAR>(#\DELTA)
  <KEEP>(<SINGLE_QUOTE>###7.56<SINGLE_QUOTE>)
  <ENDALIGN_CHAR>
```

This example produces the following output:

The sum may be padded on the left with delta characters, resulting in ' ΔΔΔ7.56'

## <ALIGN\_NUMBER>

---

## <ALIGN\_NUMBER>

Specifies a number that you want to be aligned with other numbers in the same column of a table.

---

**SYNTAX**      <ALIGN\_NUMBER>(number)

---

**ARGUMENTS**    *number*

Specifies a number, which can include commas and decimal points, that is to be aligned with other numbers in the same column. Fields in the number that are to be left blank may contain pound sign (#) characters representing blank numbers or semicolon (;) characters representing commas or decimal points.

---

**related tags**

- <ALIGN\_AFTER>
- <ALIGN\_CHAR>
- <TABLE\_ROW>

---

**restrictions**

Invalid in the context of a <MATH> tag.

---

**DESCRIPTION**

The <ALIGN\_NUMBER> tag specifies a number that you want to be aligned with other numbers in the same column of a table. Items in table columns are aligned on the left by default. To display a column of numbers aligned on the right or aligned around the decimal point, use the <ALIGN\_NUMBER> tag. Use the pound sign and semicolon characters to represent fields in the number that are to be blank in your output.

---

**EXAMPLES**

The following example shows a series of numbers aligned at the decimal point.

```
1 <table>
  <table_setup>(2\8)
  <table_row>(one\<align_number>(100,000,000.##))
  <table_row>(one\<align_number>(###;###;240.40))
  <table_row>(one\<align_number>(###;###;###.60))
  <table_row>(one\<align_number>(###;230,425))
<endtable>
```

This example produces the following output:

```
one      100,000,000.
one      240.40
one      .60
one      230,425
```

The following example is visually more elaborate, although the series of numbers are again aligned at the decimal point.

```
2 <table>
  <table_setup>(3\15\10)
  <table_heads>(Item\Net Revenue\Percentage)
  <table_row>(Knives\<align_number>(##;#20;435;##)\<align_number>(##10.44))
  <table_row>(Forks\<align_number>(1,432,064.23)\<align_number>(##45.0#))
  <table_row>(Spoons\<align_number>(##;###;245;##)\<align_number>(###.86))
<endtable>
```

This example produces the following output:

Item	Net Revenue	Percentage
Knives	20,435	% 10.44
Forks	1,432,064.23	% 45.0
Spoons	245	% .86

## <AMPERSAND>

---

## <AMPERSAND>

Specifies an ampersand (&) anywhere in your file. To specify an ampersand in an argument to a tag or in the context of the <MATH> tag, you must use this tag.

---

### SYNTAX <AMPERSAND>

---

**ARGUMENTS** *None.*

---

**related tags**

- The following tags label other characters that you must tag when they occur in an argument to a tag:

- <BACKSLASH>
  - <CPAREN>—required only if there is no opening parenthesis
  - <OPAREN>—required only if there is no closing parenthesis
  - <VBAR>

---

**restrictions**

You must use the <AMPERSAND> tag to specify an ampersand in an argument to a tag or in the context of the <MATH> tag.

---

### DESCRIPTION

The <AMPERSAND> tag specifies an ampersand (&) anywhere in your file. Because the ampersand has a special meaning to the tag translator when it occurs within an argument to a tag, you must use the <AMPERSAND> tag to output an ampersand within an argument to a tag or in the context of the <MATH> tag, or it may be misinterpreted by the tag translator. Outside of an argument to a tag, use the ampersand keyboard character rather than the tag.

---

### EXAMPLE

The following example shows how to use the <AMPERSAND> tag.

```
<SUBHEAD1>(Continuing the Line with <AMPERSAND>\cont_ampersand)
<P>Your BASIC statement may be formatted over two or more lines
by terminating all lines except the last with an ampersand (&).
```

This example produces the following output:

**Continuing the Line with &**

Your BASIC statement may be formatted over two or more lines by terminating all lines except the last with an ampersand (&).

---

## <APPENDIX>

Begins an appendix.

---

### SYNTAX

<APPENDIX> (*appendix title*[\ *symbol name*])

---

### ARGUMENTS

***appendix title***

Specifies the title for the appendix.

***symbol name***

This is an optional argument. It specifies the symbol name used in all cross-references to this appendix. The value of the symbol name in the cross-reference file is the letter assigned to the appendix.

If the appendix is part of a book, the appendix letter is determined by the position of the appendix with respect to other appendixes in the book.

If the file being processed is not currently part of a book, the first appendix in the source file is lettered A, and subsequent appendixes are sequentially lettered.

Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

### related tags

- <SET\_APPENDIX\_LETTER>

---

### restrictions

If you include the file containing this tag in a bookbuild, you must provide a symbol name.

---

### required terminator

<ENDAPPENDIX>

---

### DESCRIPTION

The <APPENDIX> tag begins an appendix. An appendix contains supplementary material at the end of a book and can contain any number of other tags.

DOCUMENT considers each appendix to be an **online topic**. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information about online topics.

## <APPENDIX>

---

### EXAMPLE

The following example shows how to code an appendix.

```
<APPENDIX> (<REFERENCE> (product_name) Command Summary\command_app)
.
.
.
<ENDAPPENDIX>
```

See Appendix A in this manual for a sample appendix.



---

## <BACKSLASH>

Specifies a backslash anywhere in your file. To specify a backslash in an argument to a tag, you must use this tag.

---

### SYNTAX <BACKSLASH>

---

**ARGUMENTS** *None.*

---

**related tags**

- The following tags label other characters that you must tag when they occur in an argument to a tag:

- <AMPERSAND>
  - <CPAREN>—required only if there is no opening parenthesis
  - <OPAREN>—required only if there is no closing parenthesis
  - <VBAR>

---

**restrictions**

You must use the <BACKSLASH> tag to specify a backslash in an argument to a tag.

---

### DESCRIPTION

The <BACKSLASH> tag specifies a backslash anywhere in your file. Because the literal backslash character is used to separate the arguments in the argument list to a tag, you must use the <BACKSLASH> tag to include a backslash as part of your argument. Outside of an argument to a tag, use the backslash keyboard character rather than the tag.

---

### EXAMPLE

The following example shows how to use the <BACKSLASH> tag.

```
<SUBHEAD1>(Using the Backslash (<BACKSLASH>)\using_backslash)  
<P>Use the backslash character (\) to separate BASIC statements  
on the same line.
```

This example produces the following output:

**Using the Backslash (\)**

Use the backslash character (\) to separate BASIC statements on the same line.

## <BOX>

---

## <BOX>

Produces a box that surrounds a user-specified character string.

---

**SYNTAX**      **<BOX>** (*label*)

---

**ARGUMENTS**    *label*  
Specifies the character string to be surrounded by the box.

---

**restrictions**    Invalid in the context of a <MATH> tag.  
The character string can be as many as 15 characters long. The box format is not available for all output devices.

---

**DESCRIPTION**    The <BOX> tag produces a box that surrounds a user-specified character string. The case of the text is output as you entered it. The font, however, that the <BOX> tag uses distinguishes the enclosed text from the rest of the text.

---

**EXAMPLE**        The following example shows how to use the <BOX> tag.

<P>You may need to enclose some text in a box, <BOX>(like this).

This example produces the following output:

You may need to enclose some text in a box, like this.

---

## <CALLOUT>

Identifies a number that is printed as a white letter in a black box or circle (depending on the output device) in an example. (The <CALLOUT> tag is identical to the <CO> tag.)

---

### SYNTAX

<CALLOUT>[(*number*)]

---

### ARGUMENTS

*number*

This is an optional argument. It specifies the number of the callout. If you do not specify a callout number, the callouts are numbered sequentially.

---

### related tags

- <CALLOUT\_REF>
- <CALLOUTS>
- <CO>
- <LIST>(CALLOUT)

---

### restrictions

Valid only in the context of a <CALLOUTS> tag.

---

### DESCRIPTION

The <CALLOUT> tag identifies a number that is printed as a white letter in a black box or circle (depending on the output device) in an example.

---

### EXAMPLES

See the examples in the discussion of the <CALLOUTS> tag.

## <CALLOUT\_REF>

---

## <CALLOUT\_REF>

Labels a reference in text to a callout number.

---

**SYNTAX**      **<CALLOUT\_REF>** (*callout number*)

---

**ARGUMENTS**      ***callout number***  
Specifies the number in the callout.

---

**related tags**

- <CALLOUT>
- <CALLOUTS>
- <CO>
- The following tags label other types of references:
  - <REFERENCE>(symbol name)
  - <GREF>(glossary term)

---

**restrictions**

Invalid in the context of a <MATH> and an <EXAMPLE> tag.

---

**DESCRIPTION**      The <CALLOUT\_REF> tag labels a reference in text to a callout number. You create the callout number by using the <CALLOUT> tag in an example or figure.

---

**EXAMPLE**      The following example shows how to use the <CALLOUT\_REF> tag.

```
<P>In <REFERENCE>(page_transitions_fig),  
callout <CALLOUT_REF>(1)  
labels the beginning of the process. Callout <CALLOUT_REF>(2) labels the  
intermediate stage.
```

This example produces the following output:

In Figure 3-3, callout ❶ labels the beginning of the process. Callout ❷ labels the intermediate stage.

---

## <CALLOUTS>

Labels the beginning of a series of callouts contained in an example and enables the use of the <CALLOUT> and <CO> tags.

---

### SYNTAX

<CALLOUTS>[[*callout number*] [\ *PREFIX*]]

---

### ARGUMENTS

#### *callout number*

This is an optional argument. It sets the number that begins numbering the callouts. If you do not specify a number, the numbering begins with one.

#### **PREFIX**

This is an optional keyword argument. It specifies that the callouts appear before the line they label, rather than at the end of the line. All callouts in any one example must precede the line they label or appear inside or at the end of the line they label. You cannot mix the location of the callouts.

This argument, if specified, must be the second argument, for example, <CALLOUTS>(\ PREFIX).

---

### related tags

- <CALLOUT>
- <CALLOUT\_REF>
- <CO>
- <LIST>(CALLOUT)

---

### restrictions

The <CALLOUTS> and <ENDCALLOUTS> tags must surround the examples within which the callouts are used.

---

### required terminator

<ENDCALLOUTS>

---

### DESCRIPTION

The <CALLOUTS> tag labels the beginning of a series of callouts contained in an example and enables the use of the <CALLOUT> and <CO> tags. See the <LIST> tag description for information about how to label callouts in a list.

## <CALLOUTS>

### EXAMPLES

The following example shows how to use the <CALLOUTS> tag.

```
1 <CALLOUTS>
  <CODE_EXAMPLE>
    DO WHILE (^EOF); <CALLOUT>
      READ FILE(INFILE) INTO(Y);
      PUT LIST(Y); <CALLOUT>
    END;
  <ENDCODE_EXAMPLE>
<ENDCALLOUTS>
<LIST>(CALLOUT)
<LE>This is a <KEYWORD>(DO) statement.
<LE>This <KEYWORD>(PUT) statement outputs the line read.
<ENDLIST>
```

This example produces the following output:

```
DO WHILE (^EOF); ❶
  READ FILE(INFILE) INTO(Y);
  PUT LIST(Y); ❷
END;
```

- ❶ This is a **DO** statement.
- ❷ This **PUT** statement outputs the line read.

The following example shows how you can set the number from which to begin numbering the callout sequence and how you can cause the callouts to precede the line to which they belong by using the PREFIX argument.

```
2 <CALLOUTS>(3\PREFIX)
  <CODE_EXAMPLE>
    <CALLOUT> DO WHILE (^EOF);
      READ FILE(INFILE) INTO(Y);
    <CALLOUT> PUT LIST(Y);
    END;
  <ENDCODE_EXAMPLE>
<ENDCALLOUTS>
```

This example produces the following output:

```
❸ DO WHILE (^EOF);
  READ FILE(INFILE) INTO(Y);
❹ PUT LIST(Y);
  END;
```

The following example shows how you can place callouts inside of lines and how you can use an argument to the <CALLOUT> tag to control the number of the callout.

```
3 <CALLOUTS>
  <CODE_EXAMPLE>
    DO WHILE <CALLOUT>(9) (^EOF);
      READ FILE(INFILE) INTO(Y);
      PUT <CALLOUT>(12) LIST(Y);
    END;
  <ENDCODE_EXAMPLE>
<ENDCALLOUTS>
```

This example produces the following output:

```
DO WHILE ❹ (^EOF);
  READ FILE(INFILE) INTO(Y);
  PUT ❷ LIST(Y);
  END;
```

---

## <CENTER\_LINE>

Specifies a line of text you want to center in the current text margin.

---

### SYNTAX

**<CENTER\_LINE>**(*text* [ { *\BIGSKIP*  
*\SMALLSKIP* } ])

---

### ARGUMENTS

***text***

Specifies a line of text you want to center.

***BIGSKIP***

***SMALLSKIP***

These are optional keyword arguments. They specify a set amount of vertical space to precede the element identified as a line or block of text. The actual amount of space created is determined by the doctype.

---

### related tags

- <CHHEAD>
- <LINE>
- <RIGHT\_LINE>

---

### restrictions

Invalid in monospaced examples, in the context of a <MATH> tag, and in arguments to tags that provide title or heading text.

Centered text must fit within the current text margin. If you specify text that is too wide, the text formatter issues a warning message, and you should examine your output.

---

### DESCRIPTION

The <CENTER\_LINE> tag specifies a line of text you want to center in the current margin.

## <CENTER\_LINE>

---

### EXAMPLE

The following example shows how to use both the BIGSKIP and SMALLSKIP keywords.

<P>Please include the following information:

<CENTER\_LINE>(Name\smallskip)  
<CENTER\_LINE>(Address\bigskip)  
<CENTER\_LINE>(Phone Number)

This example produces the following output:

Please include the following information:

Name

Address

Phone Number



---

## <CHAPTER>

Begins a chapter.

---

### SYNTAX

<CHAPTER> (*chapter title* [*\ symbol name*])

---

### ARGUMENTS

#### ***chapter title***

Specifies the name of the chapter. The doctype may also use this argument in the running title.

#### ***symbol name***

This is an optional argument. It specifies the symbol name used in all cross-references to this chapter. The value of the symbol name is the number assigned to the chapter. Each chapter is sequentially numbered.

Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

### related tags

- <REFERENCE>
- <SET\_CHAPTER\_NUMBER>

---

### restrictions

Invalid in the context of a <FRONT\_MATTER> tag.

If you include a file containing the <CHAPTER> tag in a bookbuild, including a bookbuild for Bookreader, you must provide a *symbol name* argument with the tag.

---

### DESCRIPTION

The <CHAPTER> tag begins a chapter. When printed, the chapter title and number are printed in a larger font. DOCUMENT automatically increments chapter numbers.

DOCUMENT considers each chapter to be an **online topic**. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information about online topics.

---

### EXAMPLE

The following example shows how to code a chapter.

```
<CHAPTER> (Global Tag Descriptions\tags_chap)
```

```
<P>This chapter includes descriptions of all global tags. The tags are listed alphabetically.
```

See the beginning of this chapter for a sample chapter.

## <CHEAD>

---

## <CHEAD>

Specifies an unnumbered centered heading.

---

### SYNTAX

**<CHEAD>** (*heading text*[\ *symbol name*])

---

### ARGUMENTS

#### ***heading text***

Specifies the text of the heading.

#### ***symbol name***

This is an optional argument. It specifies the name of the symbol used in all references to this heading. Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

### related tags

- <CENTER\_LINE>
  - <HEAD>
  - <HEADn>
  - <SUBHEADn>
- 

### DESCRIPTION

The <CHEAD> tag specifies an unnumbered centered heading. The heading does not appear in the table of contents. Use the <CHEAD> tag when the clarity of your document requires such a fine level of distinction.

---

### EXAMPLE

The following example shows how to use the <CHEAD> tag.

```
<CHEAD>(How to Use the <TAG>(thead) Tag\use_of_thead)
<P>The use of centered headings should be restricted to occasions when the
clarity of your document absolutely requires one.
```

This example produces the following output:

#### **How to Use the <CHEAD> Tag**

The use of centered headings should be restricted to occasions when the clarity of your document absolutely requires one.

---

## <CHECK\_FOR\_INCLUSION>

Marks a definitions file to ensure that the file is read only once.

---

### SYNTAX

<CHECK\_FOR\_INCLUSION> (*file label*)

---

### ARGUMENTS

***file label***

Specifies a string that uniquely identifies this file. This string is not the name of the file; rather, it sets an internal switch to indicate that the following input should be read only once. The string can contain up to 15 alphanumeric characters or underscores and must not begin with an underscore.

---

### restrictions

You must put the <CHECK\_FOR\_INCLUSION> tag at the top of the definitions file and the <ENDCHECK\_FOR\_INCLUSION> tag at the bottom of the file.

---

### required terminator

<ENDCHECK\_FOR\_INCLUSION>

---

### DESCRIPTION

The <CHECK\_FOR\_INCLUSION> tag marks a definitions file to ensure that the file is read only once, because to force the tag translator to process the definitions twice would be a waste of processing time. The tag is useful:

- For excluding the definitions file from being read when processing an individual element of a book, because the file has already been read for an entire bookbuild
- For excluding the definitions file from being read during a bookbuild if the source files already contain the same definitions

---

### EXAMPLE

The following example shows how you can use the <CHECK\_FOR\_INCLUSION> tag to exclude a file (in this example, the file MY\_DEFINITIONS\_FILE) from being read during a bookbuild.

```
<CHECK_FOR_INCLUSION>(MY_DEFINITIONS_FILE)
<DEFINE>(ansi_long\ANSI General Purpose Subset (BSR X3.74))
<DEFINE>(ansi_short\ANSI GPSubset)
.
.
.
<ENDCHECK_FOR_INCLUSION>
```

## <CO>

---

## <CO>

Identifies a number that is printed as a white letter in a black box or circle (depending on the output device) in an example. (The <CO> tag is identical to the <CALLOUT> tag.)

---

**SYNTAX**      <CO>[(*number*)]

---

**ARGUMENTS**    *number*

This is an optional argument. It specifies the number of the callout. If you do not specify a callout number, the callouts are numbered sequentially.

---

**related tags**

- <CALLOUT>
  - <CALLOUT\_REF>
  - <CALLOUTS>
  - <LIST>(CALLOUT)
- 

**restrictions**

Valid only in the context of a <CALLOUTS> tag.

---

**DESCRIPTION**    The <CO> tag identifies a number that is printed as a white letter in a black box or circle (depending on the output device) in an example.

---

**EXAMPLES**      See the examples in the discussion of the <CALLOUTS> tag.

---

## <CODE\_EXAMPLE>

Begins an example of code. Code consists of words or lines of instructions written in a programming language or a command language.

---

### SYNTAX

<CODE\_EXAMPLE> (*inline code example*)

or

<CODE\_EXAMPLE> [ ( { *KEEP*  
*WIDE* [ \ *MAXIMUM* ] } ) ]

code example text

.  
.  
.

<ENDCODE\_EXAMPLE>

---

### ARGUMENTS

#### *inline code example*

Specifies code you want to insert into your text.

If you do not use this argument, you must use the terminating tag <ENDCODE\_EXAMPLE>.

#### **KEEP**

This is an optional keyword argument. It specifies that the example is not to be broken across pages; that is, if the example does not fit on the current page, it is placed on the next page. If the example itself does not fit on an entire single page of output, it is broken anyway. This keyword has no effect for Bookreader output.

#### **WIDE**

This is an optional keyword argument. It specifies that the width of the example exceeds the document's default width for text. Depending on the **doctype**, this argument is interpreted as follows:

- If the doctype contains a left margin area that is normally used for headings, the example's width spans that area as well as the normal text area.
- If the document uses a multicolumn format, the example suspends the multicolumn output while the example is processed. The example is output across two columns and then multicolumn output is restored.

You must encase the code example in <EXAMPLE>, <FIGURE>, or <TABLE> tags in order to print the code example across two columns.

- Depending on the doctype, this argument provides a range of font sizes and font styles for examples.

## <CODE\_EXAMPLE>

### **MAXIMUM**

This is an optional keyword argument. It specifies that the text of the code example is adjusted to a smaller point size in order to fit within the boundaries of the left and right margins of the page. You can use this keyword in conjunction with WIDE to indicate that the example may require additional adjustment to fit within the boundaries of the page. You should use this keyword with discretion, and it may not be suitable in all doctypes.

---

#### related tags

- <INTERACTIVE>
- <VALID\_BREAK>

---

#### restrictions

Invalid in the context of a <FOOTNOTE> tag.

You cannot use indexing tags (<X> and <Y> tags) in code examples.

You cannot use tab characters to format code examples. You must use spaces.

You cannot use text element tags within a code example (for example, <P>, <LIST>, or <NOTE> tags).

If you use double hyphens within the <CODE\_EXAMPLE>(code) tag, the double hyphens are converted to a single en dash. This does not occur if you use double hyphens in the context of the <CODE\_EXAMPLE> and <ENDCODE\_EXAMPLE> tags.

---

#### required terminator

<ENDCODE\_EXAMPLE> —Required if you do not provide the *inline code example* argument to the <CODE\_EXAMPLE> tag.

---

## DESCRIPTION

The <CODE\_EXAMPLE> tag begins an example of code. Code consists of words or lines of instructions written in a programming language or a command language. The code that is begun with a <CODE\_EXAMPLE> tag is distinguished typographically in the output. The size of the example, whether it will be indented and how much it will be indented from the current left margin of text, is controlled by the doctype.

Using the <CODE\_EXAMPLE> tag makes the code example an **informal example**, in that the code example has no number or caption, cannot be referenced, and is not listed in the table of contents. If you enclose the code example within <EXAMPLE> and <ENDEXAMPLE> tags, however, the code example becomes a **formal example**; it then has a number and caption, can be referenced, and is listed in the table of contents.

By default, informal code examples do not **pop up** when viewed with Bookreader; formal code examples do. If you want an informal code example to pop up, use the <ONLINE\_POPUP> tag.

There are two types of code examples. The first is a short fragment that you may want included with the surrounding text. In this case, enter the code as an argument to the <CODE\_EXAMPLE> tag and do not use a terminating tag.

The second type of code example is one or more lines that you may want broken out of the text that surrounds it. In this case, enter the code between the <CODE\_EXAMPLE> and <ENDCODE\_EXAMPLE> tags. The character spaces and blank lines you enter to format the code are retained. You can also use the <ELLIPSIS> tag to achieve a vertical ellipsis showing that you have omitted some lines of code. If your code example is longer than a few lines, use the <VALID\_BREAK> tag to indicate the acceptable points for a page break.

---

## EXAMPLES

The following example shows a short code example that is run in with the surrounding text. The <VARIABLE> tag simply distinguishes a variable typographically. See the <VARIABLE> tag description for more information.

**1** <P>The <CODE\_EXAMPLE> (WHILE INLOOP) statement causes the following block to be repeated until the <VARIABLE>(INLOOP) variable is set to FALSE.

This example produces the following output:

The *WHILE INLOOP* statement causes the following block to be repeated until the *INLOOP* variable is set to FALSE.

The following example shows how to use the <CODE\_EXAMPLE> tag with a longer example that is broken out of the surrounding text. In this case, you need the <ENDCODE\_EXAMPLE> tag, because the code is not used as an argument to the <CODE\_EXAMPLE> tag.

**2** <P>The call frame is built on the stack by the following four instructions:  
<CODE\_EXAMPLE>  
    PUSHAB    B^SRVEXIT  
    PUSHL    FP  
    PUSHL    AP  
    CLRQ     -(SP)  
<ENDCODE\_EXAMPLE>

This example produces the following output:

The call frame is built on the stack by the following four instructions:

```
    PUSHAB    B^SRVEXIT  
    PUSHL    FP  
    PUSHL    AP  
    CLRQ     -(SP)
```

The following example shows a longer code example that uses the <ELLIPSIS> tag.

**3** <P>The instruction sequence listed here (patterned after code in module PROCSTRT) shows this second technique.  
<CODE\_EXAMPLE>  
PUSHL    executive-mode-PSL  
BSBB    DOREI  
<ELLIPSIS>  
PUSHL    user-mode-PSL  
BSBB    DOREI  
<ELLIPSIS>  
DOREI:   REI  
<ENDCODE\_EXAMPLE>

This example produces the following output:

## <CODE\_EXAMPLE>

The instruction sequence listed here (patterned after code in module PROCSTRT) shows this second technique.

```
PUSHL    executive-mode-PSL
BSBB     DOREI
.
.
.
PUSHL    user-mode-PSL
BSBB     DOREI
.
.
.
DOREI:   REI
```



---

## <COMMENT>

Marks a section of an SDML file that you do not want to appear in your output. Text marked by a <COMMENT> tag is ignored by the tag translator during processing.

---

### SYNTAX

<COMMENT>(*comment text*)

or

<COMMENT>  
comment text

·  
·  
·

<ENDCOMMENT>

---

### ARGUMENTS

*comment text*

Specifies the text you do not want to appear in your output. This is a required argument, unless you use the <COMMENT> and <ENDCOMMENT> tags with comment text between the two tags.

---

### restrictions

You cannot nest <COMMENT> tags.

---

### required terminator

<ENDCOMMENT> — Required if you do not specify an argument to the <COMMENT> tag.

---

### DESCRIPTION

The <COMMENT> tag marks a section of an SDML file that you do not want to appear in your output. Text marked by a <COMMENT> tag is ignored by the tag translator during processing.

Comments can provide useful reminders for your own use or use by writers who may modify your SDML file in the future. You can also use comments to exclude portions of text from the output file that you want to save for later inclusion. The text remains in the SDML file, but does not appear in the output.

## <COMMENT>

### EXAMPLE

The following example shows a code example with a comment embedded in the code. The tag translator ignores the text between the <COMMENT> and <ENDCOMMENT> tags.

```
<CODE_EXAMPLE>
;
; SECONDARY POOL COMMAND BUFFER BLOCKS
;
.=0
000000 C.CLK: .BLKW 1 ;LINK WORD
000002 C.CTCB: .BLKW 1 ;TCB ADDRESS OF TASK TO RECEIVE COMMAND
000004 C.CUCB: .BLKW 1 ;UCB ADDRESS IF RESPONSIBLE TERMINAL
000006 C.CCT: .BLKW 1 ;CHARACTER COUNT, EXCLUDING TRAILING CR
000010 C.CSTS: .BLKW 1 ;STATUS MASK
<COMMENT>
000012 C.CMCD: ;SYSTEM MESSAGE CODE
000012 C.CSO: .BLKW 1 ;STARTING OFFSET OF VALID COMMAND TEXT
<ENDCOMMENT>
000014 C.CTR: .BLKW 1 ;TERMINATING CHARACTER
000015 C.BLK: .BLKW 1 ;SIZE OF PACKET IN SEC POOL (32 WD.) BLOCKS
000016 C.CTXT: ;COMMAND TEXT, FOLLOWED BY CR
<ENDCODE_EXAMPLE>
```

This example produces the following output:

```

;
; SECONDARY POOL COMMAND BUFFER BLOCKS
;
.=0
000000 C.CLK: .BLKW 1 ;LINK WORD
000002 C.CTCB: .BLKW 1 ;TCB ADDRESS OF TASK TO RECEIVE COMMAND
000004 C.CUCB: .BLKW 1 ;UCB ADDRESS IF RESPONSIBLE TERMINAL
000006 C.CCT: .BLKW 1 ;CHARACTER COUNT, EXCLUDING TRAILING CR
000010 C.CSTS: .BLKW 1 ;STATUS MASK

000014 C.CTR: .BLKW 1 ;TERMINATING CHARACTER
000015 C.BLK: .BLKW 1 ;SIZE OF PACKET IN SEC POOL (32 WD.) BLOCKS
000016 C.CTXT: ;COMMAND TEXT, FOLLOWED BY CR
```

---

<CONDITION>

Marks a section of an SDML file that is not processed unless one of the arguments to the tag matches the argument in the related <SET\_CONDITION> tag or matches the condition you set with the /CONDITION qualifier on the DOCUMENT command line.

---

SYNTAX

<CONDITION>(*condition name-1* [\ *condition name-2*]  
[\ *condition name-3*])

conditional text

·  
·  
·

<ENDCONDITION>

---

ARGUMENTS

***condition name***

Specifies up to three condition names created with the <SET\_CONDITION> tag. At least one condition name is required.

---

related tags

- <SET\_CONDITION>

---

restrictions

You cannot nest <CONDITION> tags.

In monospaced examples, <CONDITION> and <ENDCONDITION> tags cannot occupy the same line as text for the example.

---

required terminator

<ENDCONDITION>

---

DESCRIPTION

The <CONDITION> tag marks a section of an SDML file that is not processed unless one of the arguments to the tag matches the argument in the related <SET\_CONDITION> tag or matches the condition you set with the /CONDITION qualifier on the DOCUMENT command line.

If the two tags' arguments are identical, the marked text is processed and included in the output file. If the arguments differ, the text is excluded from processing.

This tag can be very useful if you must produce two or more documents that differ in minor ways but contain much of the same text. It is efficient to maintain one SDML file and identify the portions of text that are unique to each document.

## <CONDITION>

Before processing the SDML file, you can specify the condition name for the desired document by using the <SET\_CONDITION> tag or the /CONDITION qualifier on the DOCUMENT command line. When the tag translator processes the file, it includes all portions of text tagged with that condition name argument and ignores all portions of text tagged with any other condition name. To obtain another version of the document, change the condition name in the argument to the <SET\_CONDITION> tag or change the /CONDITION qualifier and reprocess the file.

When you supply more than one condition name argument to the <CONDITION> tag, only one of the condition names needs to be present for the conditional text to be processed.

---

## EXAMPLES

The following example shows how you could organize a file if you were writing about holidays in general, and Christmas, Chanukah, and Passover in particular. Any of three different condition states could be set at the top of your file with the <SET\_CONDITION>(Christmas), <SET\_CONDITION>(Chanukah), or <SET\_CONDITION>(Passover) tags. The example contains four paragraphs, with each paragraph conditionalized differently.

If the file is processed with the <SET\_CONDITION>(Christmas) tag, the information on Christmas and religious holidays is processed. If the file is processed with the <SET\_CONDITION>(Chanukah) tag, the information on Chanukah and religious holidays is processed. If the file is processed with the <SET\_CONDITION>(Passover) tag, the information on Passover is processed.

The function of the <SET\_CONDITION> tag can be obtained by using the /CONDITION qualifier on the DOCUMENT command line instead. See *VAX DOCUMENT Producing Online and Printed Documentation* or Appendix A for more information.

```
1 <CONDITION>(Christmas)
  <P>Christmas, by convention, is celebrated on December 25th . . .
  <ENDCONDITION>

  <CONDITION>(Chanukah)
  <P>Chanukah is called the Festival of Lights . . .
  <ENDCONDITION>

  <CONDITION>(Passover)
  <P>Passover is usually celebrated . . .
  <ENDCONDITION>

  <CONDITION>(Christmas\Chanukah)
  <HEAD1>(Religious Holidays)
  <P>This paragraph contains general information about several religious
  holidays . . .
  <ENDCONDITION>
```

The following example shows how you can omit one of two clauses from processing based on the condition name specified in the <SET\_CONDITION> tag. The section of text that is identified with <CONDITION>(VMS) would be processed and placed in the output file, whereas the other text would be omitted.

```
2 <SET_CONDITION> (VMS)
  .
  .
When the
<CONDITION> (VMS) VAX/VMS<ENDCONDITION>
<CONDITION> (RSX) RSX11M/RSX11M-PLUS<ENDCONDITION>
command language interpreter translates the logical name . . .
```

This example produces the following output:

When the VAX/VMS command language interpreter translates the logical name . . .

The following example shows how you can differentiate text that is applicable to the two condition names (VMS and RSX) from text that is applicable to the condition name RSTS. By supplying both the VMS and RSX arguments to the first <CONDITION> tag, you ensure that the text is included in the output if either the VMS or RSX condition name is set. (Other text in the same SDML file might be conditional for only one of these condition names.)

```
3 <SET_CONDITION> (RSX)
  <P>When RSX . . .
  .
  .
<CONDITION> (VMS\RSX)
  <P>When the operating system . . .
  .
  .
<ENDCONDITION>
<CONDITION> (RSTS)
  <P>When RSTS . . .
  .
  .
<ENDCONDITION>
```

## <CONTENTS\_FILE>

---

## <CONTENTS\_FILE>

Specifies the position in a source file where a table of contents output file will be included; it does not produce a table of contents.

---

### SYNTAX

<CONTENTS\_FILE>[(*file spec*)]

---

### ARGUMENTS

#### *file spec*

This is an optional argument. It specifies the name of the table of contents file. Use this argument if you want to include a table of contents file with a name other than the default file name.

The default file name is *file spec\_contents.DVI\_device*. If you specify a file type, be sure that the device name matches the destination device you specify on the DOCUMENT command line. For example, if you specify *file spec\_contents.DVI\_PS*, be sure you specify PS as the destination on the command line.

---

### related tags

- <FRONT\_MATTER>
- <PROFILE>

---

### restrictions

Valid only in the context of a <FRONT\_MATTER> or <PROFILE> tag. This tag must occur within front matter or a profile in order for processing to continue.

---

### DESCRIPTION

The <CONTENTS\_FILE> tag specifies the position in a source file where a table of contents output file will be included; it does not produce a table of contents. A contents file is produced when you specify the qualifier /CONTENTS on the DOCUMENT command line.

It is recommended that you place the <CONTENTS\_FILE> tag in the front matter file immediately before the <PREFACE> tag. This ensures correct placement of the table of contents in both printed and Bookreader output.

A contents file always receives the filetype *.DVI\_device*, where *device* is the type of output device you specified on the DOCUMENT command line. To generate a table of contents from an individual file that contains a <CONTENTS\_FILE> tag, specify the /CONTENTS qualifier on the DOCUMENT command line.

If you specify the /CONTENTS qualifier on the DOCUMENT command line without using the <CONTENTS\_FILE> tag in your front matter file, a table of contents is generated separately from your document. For more information on generating a table of contents, see *VAX DOCUMENT Producing Online and Printed Documentation*.

Because a table of contents is required for a book you create for Bookreader, a table of contents file is generated automatically if you do not use a <CONTENTS\_FILE> tag or the /CONTENTS qualifier on the DOCUMENT command line. If you do use the <CONTENTS\_FILE> tag in a Bookreader bookbuild, however, the tag must not precede the <TITLE\_PAGE> tag.

---

**EXAMPLE**

To see the result of the <CONTENTS\_FILE> tag, refer to the table of contents in this manual. See the example in the <FRONT\_MATTER> tag description for an example of placing the <CONTENTS\_FILE> tag before the <PREFACE> tag in the front matter of a book.

## <COPYRIGHT\_DATE>

---

## <COPYRIGHT\_DATE>

Inserts a copyright date line on the copyright page along with other system-specific copyright information.

---

### SYNTAX

**<COPYRIGHT\_DATE>**(*date*[\ *owner*])

---

### ARGUMENTS

#### ***date***

Specifies official printing date information for the book.

#### ***owner***

This is an optional argument. It specifies the owner of the copyright.

**Note:** For Bookreader output, the arguments you specify are used to generate the copyright notice on the title page, not on the copyright page.

---

### related tags

- <COPYRIGHT\_PAGE>
- <FRONT\_MATTER>

---

### restrictions

Valid only in the context of a <COPYRIGHT\_PAGE> tag, which you can use only within the front matter of a book.

---

### DESCRIPTION

The <COPYRIGHT\_DATE> tag inserts a copyright date line on the copyright page along with other system-specific copyright information.

---

### EXAMPLE

The following example shows how to use the <COPYRIGHT\_PAGE> tag to produce a full page of output.

```
<FRONT_MATTER>(front)
<COPYRIGHT_PAGE>
<COPYRIGHT_DATE>(1990\Smith Publishing, Inc.)
<ENDCOPYRIGHT_PAGE>
<ENDFRONT_MATTER>
```

This example produces the following output:

```
Copyright ©1990 Smith Publishing, Inc.
All Rights Reserved.
```



---

## <COPYRIGHT\_PAGE>

Begins a copyright page and enables copyright page tags.

---

### SYNTAX <COPYRIGHT\_PAGE>

---

**ARGUMENTS** *None.*

---

**related tags**

- <COPYRIGHT\_DATE>
- <FRONT\_MATTER>
- <PRINT\_DATE>

---

**restrictions** Valid only in the front matter of a book.

---

**required terminator** <ENDCOPYRIGHT\_PAGE>

---

**DESCRIPTION** The <COPYRIGHT\_PAGE> tag begins a copyright page and enables copyright page tags. The following tags are enabled by the <COPYRIGHT\_PAGE> tag:

- <PRINT\_DATE>
- <COPYRIGHT\_DATE>

DOCUMENT considers a copyright page to be an **online topic**. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information about online topics.

---

**EXAMPLE** The following example shows the order in which you can use the copyright page tags. The output of this example is a separate copyright page containing the print and copyright dates in the front matter of the book.

```
<FRONT_MATTER>(front)
<COPYRIGHT_PAGE>
<PRINT_DATE>(March 1987)
<COPYRIGHT_DATE>(1987)
<ENDCOPYRIGHT_PAGE>
<ENDFRONT_MATTER>
```

To see an example of all the front matter tags in their correct order, see the example in the <FRONT\_MATTER> tag description.

## <CP>

---

## <CP>

Marks the continuation of a paragraph that is interrupted by another text element, such as a figure, list, or a table.

---

### SYNTAX

<CP>

---

### ARGUMENTS

*None.*

---

### related tags

- <P>

---

### restrictions

Invalid in the context of a <MATH> tag.

---

### DESCRIPTION

The <CP> tag marks the continuation of a paragraph that is interrupted by another text element, such as a figure, list, or a table. The <P> and <CP> tags may have no visible difference in effect. In a doctype in which all new paragraphs begin flush left, the formatted results achieved by a <CP> tag and by a <P> tag are identical. In a doctype in which paragraphs are indented, however, the continued paragraph may still begin flush left.

In some instances, the <CP> tag keeps text that follows a list or monospaced example from being left alone at the top of the following page, which more closely attaches the continued paragraph to the previous text element.

---

### EXAMPLE

The following example shows how to use the <CP> tag. Assume that the doctype is one in which paragraphs are indented.

```
<P>Each time you log in, the system automatically executes
two types of login command procedures:
<LIST>(UNNUMBERED)
<LE>A system login command procedure
<LE>Your personal login command procedure
<ENDLIST>
<CP>These login procedures are described in the following sections.
```

This example produces the following output:

Each time you log in, the system automatically executes two types of login command procedures:

- A system login command procedure
- Your personal login command procedure

These login procedures are described in the following sections.

---

## <CPAREN>

Supplies a closing parenthesis anywhere in your file. You must use this tag, however, to specify a closing parenthesis within an argument to a tag, so that the closing parenthesis is not interpreted as the closing parenthesis of the tag's argument.

---

### SYNTAX <CPAREN>

---

**ARGUMENTS** *None.*

**related tags**

- The following tags label other characters that you must tag when they occur in an argument to a tag:

- <AMPERSAND>
  - <BACKSLASH>
  - <OPAREN>—required only if there is no closing parenthesis
  - <VBAR>

---

### DESCRIPTION

The <CPAREN> tag supplies a closing parenthesis anywhere in your file. You must use this tag, however, to specify a closing parenthesis within an argument to a tag, so that the closing parenthesis is not interpreted as the closing parenthesis of the tag's argument. This tag is only beneficial (in terms of keystrokes and control of the output), then, as an unmatched closing parenthesis in an argument to a tag.

An unmatched parenthesis in an argument can cause errors when processed, because the parentheses are used to determine the beginning and ending of an argument list. The <CPAREN> tag inserts the closing parenthesis character, but it is not evaluated as a closing parenthesis.

---

### EXAMPLE

The following example shows how to use the <CPAREN> tag. The <PARENDCHAR> tag in this example simply provides better spacing of the right parenthesis within the parentheses. See the <PARENDCHAR> tag description for more information.

```
<SUBHEAD1>(Using a Closing Parenthesis  
<PARENDCHAR>(<CPAREN>) in an Argument to a Tag\closing_paren)
```

This example produces the following output:

**Using a Closing Parenthesis ( ) in an Argument to a Tag**

## <DATE>

---

## <DATE>

Produces either the current system or user-specified date and time.

---

### SYNTAX

**<DATE>**[( { *FULL*  
*date text* } )]

---

### ARGUMENTS

#### **FULL**

This is an optional keyword argument. It specifies a full VMS date and time string in the format "dd-mmm-yyyy hh:mm:ss.hh." If you do not specify an argument, only the date may be produced in the format "Month day, year." (The precise format varies in different doctypes.)

#### ***date text***

This is an optional argument. It specifies text that you provide for the current date. If you do not specify an argument, only the date may be produced in the format "Month day, year." (The precise format varies in different doctypes.)

The following table specifies for all standard doctypes the output format of the <DATE> tag without arguments.

Doctype	Format Description	<DATE> Tag Output
REPORT	Day-Month-Year	6-January-1990
MILSPEC	Day Month Year	6 January 1990
All others	Month Day, Year	January 6, 1990

---

### restrictions

Invalid in the context of a <MATH> tag.

---

### DESCRIPTION

The <DATE> tag produces either the current system or user-specified date and time.

Note that even if you specify the <DATE>(FULL) tag multiple times in a source file, it will always produce the same value. You cannot use it, therefore, for timing information.

---

### EXAMPLES

The following example shows how to use the <DATE> tag.

**1** <DATE>

This example produces the following output: January 28, 1991

The following example shows how to use the <DATE>(FULL) tag.

**2** <DATE> (FULL)

This example produces the following output: 28-JAN-1991 11:03:30.58

The following example shows how to use the <DATE>(date text) tag.

**3** <DATE>(February 26, 1990 A.D.)

This example produces the following output: February 26, 1990 A.D.

## <DEFINE\_BOOK\_NAME>

---

## <DEFINE\_BOOK\_NAME>

Defines the title of a book and associates a user-defined symbol name to the cross-reference file.

---

### SYNTAX

**<DEFINE\_BOOK\_NAME>** (*symbol name* \ *title*)

---

### ARGUMENTS

#### ***symbol name***

Specifies the symbol that is associated with the title of the book. Symbol names must not exceed 31 characters and must only include alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

#### ***title***

Specifies the exact text of the book's title.

---

### related tags

- <DEFINE\_SYMBOL>
  - <DELAYED>
  - <REFERENCE>
- 

### restrictions

If you want to nest a tag in the *title* argument, you must use the <DELAYED> tag to surround the nested tag. This causes the nested tag to be preserved until it is about to be output using the <REFERENCE> tag.

---

### DESCRIPTION

The <DEFINE\_BOOK\_NAME> tag defines the title of a book and associates a user-defined symbol name to the cross-reference file. When you subsequently reference the title with the <REFERENCE> tag, supplying the same symbol name as an argument to the <REFERENCE> tag, the book title is retrieved from the cross-reference file and is substituted for the reference. Depending on the doctype, the title may be output with emphasized text (for example, italicized text).

Note that the *symbol name* argument is the first argument to the tag. In text element tags, the symbol name is always the second argument. Only the <DEFINE\_BOOK\_NAME> and <DEFINE\_SYMBOL> tags take a symbol name as a first argument. Placing the symbol name as the first argument to these tags makes it easy to keep track of your symbol names when you list them in a local definitions file, where they are aligned and easy to find.

---

**EXAMPLE**

The following example shows how to use the <DEFINE\_BOOK\_NAME> tag for defining the symbol name of the book, *Book of Games, Volume 2*, as *games\_book*.

```
<DEFINE_BOOK_NAME>(games_book\Book of Games, Volume 2)
```

You can use the <REFERENCE>(games\_book) tag anywhere in the document to refer to the book by name. For example, you might say the following:

```
<P>The authors suggest that you read the <REFERENCE>(games_book) before you try any of the more complex games.
```

This example produces the following output:

The authors suggest that you read the *Book of Games, Volume 2* before you try any of the more complex games.

## <DEFINE\_SYMBOL>

---

## <DEFINE\_SYMBOL>

Associates a string of text with a user-defined symbol, so that you can reference the text by this symbol throughout the document.

---

### SYNTAX

<DEFINE\_SYMBOL>(symbol name \ text string)

---

### ARGUMENTS

#### **symbol name**

Specifies the name assigned to the symbol. Symbol names must not exceed 31 characters and must only include alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

#### **text string**

Specifies the text to be referenced by the symbol name. Throughout the document, you can specify the symbol name in the <REFERENCE> tag, and this text string is then substituted for the symbol name during processing.

---

### related tags

- <DEFINE\_BOOK\_NAME>
  - <DELAYED>
  - <REFERENCE>
- 

### restrictions

If you want to nest a tag in the *text string* argument, you must use the <DELAYED> tag to surround the nested tag. This causes the nested tag to be preserved until it is about to be output using the <REFERENCE> tag.

---

### DESCRIPTION

The <DEFINE\_SYMBOL> tag associates a string of text with a user-defined symbol, so that you can reference the text by this symbol throughout the document. It adds the symbol name and the text to the symbol table. When you subsequently reference this text by using the <REFERENCE> tag and by specifying the same symbol name as its argument, the current value of the symbol name is retrieved from the table and the text is substituted for the reference.

For more information on the use of the symbol table, see *VAX DOCUMENT Producing Online and Printed Documentation*.

**Note:** The symbol name argument to the <DEFINE\_SYMBOL> tag must be on the same line as the tag; spaces and carriage returns are significant and are interpreted as part of the symbol name.

Note that the *symbol name* argument is the first argument to the tag. In text element tags, the symbol name is always the second argument. Only the <DEFINE\_BOOK\_NAME> and <DEFINE\_SYMBOL> tags take a symbol name as a first argument. Placing the symbol name as the first argument to these tags makes it easy to keep track of your symbol names when you list them in a local definitions file, where they are aligned and easy to find.



## EXAMPLES

The following example shows how you can define the phrase, “RTL routine Set Logical Name, LIB\$SET\_LOGICAL,” as the symbol name, “set\_logical,” and then refer to it in text by that symbol name. By using the symbol name in the places you want the phrase to occur, you can save keystrokes and ensure that you never have a typing error in that phrase.

```
1 <define_symbol>(set_logical\RTL routine Set Logical Name, LIB$SET_LOGICAL,)  
<P>The <REFERENCE>(set_logical) requests the calling process’s CLI to  
define or redefine a supervisor-mode logical name.
```

This example produces the following output:

The RTL routine Set Logical Name, LIB\$SET\_LOGICAL, requests the calling process’s CLI to define or redefine a supervisor-mode logical name.

The following example uses the <DELAYED> tag within the argument to the <DEFINE\_SYMBOL> tag, causing the <DELAYED> tag to be evaluated by the tag translator after the <DEFINE\_SYMBOL> tag.

The symbol TEMP\_CHART\_EX is associated with the text string that is enclosed by the <DELAYED> and <ENDDelayed> tags. None of the enclosed tags will have been evaluated.

```
2 <define_symbol>(temp_chart_ex\<delayed>  
<example>(Temperature Chart)  
<code_example>  
    Centigrade    0    Fahrenheit  32  
    Centigrade 100    Fahrenheit 212  
<endcode_example>  
<endexample>  
<enddelayed>)
```

Subsequently, you can refer to the symbol name with a <REFERENCE> tag, as in the following example:

```
<REFERENCE>(temp_chart_ex)
```

Using this reference results in the following output:

### Example n–n Temperature Chart

---

Centigrade	0	Fahrenheit	32
Centigrade	100	Fahrenheit	212

---

## <DEFINITION\_LIST>

---

## <DEFINITION\_LIST>

Begins a definition list, which includes an item followed by its definition.

---

### SYNTAX <DEFINITION\_LIST>

---

**ARGUMENTS** *None.*

---

**related tags**

- <DEFINITION\_LIST\_HEAD>
  - <DEFLIST\_DEF>
  - <DEFLIST\_ITEM>
- 

**required terminator**

<ENDDEFINITION\_LIST>

---

### DESCRIPTION

The <DEFINITION\_LIST> tag begins a definition list, which includes an item, introduced by the <DEFLIST\_ITEM> tag, followed by its definition, introduced by the <DEFLIST\_DEF> tag. The definition list differs from a list created with the <LIST> tag in that the definition list items are not numbered or called out in any way.

You can give the definition list a heading by using the <DEFINITION\_LIST\_HEAD> tag. Each list can contain one or more items, using the <DEFLIST\_ITEM> tag, and each item can have a definition, using the <DEFLIST\_DEF> tag.

---

### EXAMPLE

The following example shows how to use the <DEFINITION\_LIST> tag.

```
<definition_list>
<deflist_item>(Hargreaves, James\d. 1778.)
<deflist_def>English inventor.
<deflist_item>(Harris, Joel Chandler\Born 1848.\Died 1908.)
<deflist_def>American author.
<enddefinition_list>
```

This example produces the following output:

***Hargreaves, James  
d. 1778.***  
English inventor.

**Harris, Joel Chandler**

**Born 1848.**

**Died 1908.**

American author.

## <DEFINITION\_LIST\_HEAD>

---

## <DEFINITION\_LIST\_HEAD>

Supplies the heading for a list of defined items you specify with the <DEFINITION\_LIST> tag.

---

**SYNTAX**            <DEFINITION\_LIST\_HEAD>(heading text)

---

**ARGUMENTS**        *heading text*  
Specifies the text for a heading that precedes the definition list.

---

**related tags**

- <DEFINITION\_LIST>
- <DEFLIST\_DEF>
- <DEFLIST\_ITEM>

---

**restrictions**        Valid only in the context of a <DEFINITION\_LIST> tag.

---

**DESCRIPTION**        The <DEFINITION\_LIST\_HEAD> tag supplies the heading for a list of defined items you specify with the <DEFINITION\_LIST> tag.

---

**EXAMPLE**            The following example shows how to use the <DEFINITION\_LIST\_HEAD> tag.

```
<definition_list>
<definition_list_head>(Worldwide Associates)
<deflist_item>(IAAF)
<deflist_def>
International Amateur Athletic Federation
<deflist_item>(IAEA)
<deflist_def>
International Atomic Energy Agency
<enddefinition_list>
```

This example produces the following output:

---

<b>WORLDWIDE ASSOCIATES</b>	<b>IAAF</b> International Amateur Athletic Federation
	<b>IAEA</b> International Atomic Energy Agency

---

## <DEFLIST\_DEF>

Introduces an item's definition in a definition list.

---

**SYNTAX**            <DEFLIST\_DEF>

---

**ARGUMENTS**        *None.*

---

**related tags**

- <DEFINITION\_LIST>
- <DEFINITION\_LIST\_HEAD>
- <DEFLIST\_ITEM>

---

**restrictions**        Valid only in the context of a <DEFINITION\_LIST> tag.

---

**DESCRIPTION**        The <DEFLIST\_DEF> tag introduces an item's definition in a definition list. Specify the item by using the <DEFLIST\_ITEM> tag and specify the definition list by using the <DEFINITION\_LIST> tag.

---

**EXAMPLE**            See the example in the discussion of the <DEFINITION\_LIST> tag.

## <DEFLIST\_ITEM>

---

## <DEFLIST\_ITEM>

Marks an item you want to define in a definition list.

---

**SYNTAX**      <DEFLIST\_ITEM>(item-1 [\ item-2 ... [\ item-7]])

---

**ARGUMENTS**    *item*  
Specifies the item you want to define. You can specify up to seven items.

---

**related tags**

- <DEFINITION\_LIST>
- <DEFINITION\_LIST\_HEAD>
- <DEFLIST\_DEF>

---

**restrictions**    Valid only in the context of a <DEFINITION\_LIST> tag.

---

**DESCRIPTION**    The <DEFLIST\_ITEM> tag marks an item you want to define in a definition list. A definition list contains paired entries, consisting of the item being defined, introduced by the <DEFLIST\_ITEM> tag, and the item's definition, introduced by the <DEFLIST\_DEF> tag.

---

**EXAMPLE**        See the example in the discussion of the <DEFINITION\_LIST> tag.

---

## <DELAYED>

Allows you to specify text that contains a tag in an argument to another tag. Execution of the <DELAYED> tag is delayed until the outer tag's text is processed.

---

**SYNTAX**      <DELAYED> (*delayed text*)  
   or

**<DELAYED>**  
**delayed text**

.

.

.

**<ENDDelayed>**

---

**ARGUMENTS**      *delayed text*

Specifies the text containing tags whose execution is delayed until the text is output.

---

**related tags**

- <DEFINE\_BOOK\_NAME>
- <DEFINE\_SYMBOL>

Any tag that takes a *symbol name* argument

---

**restrictions**

The <DELAYED> tag cannot be nested within itself.

You must use this tag if you want to nest a tag in the argument of a tag that also takes a symbol name argument.

---

**required terminator**

<ENDDelayed> —Required if you do not specify an argument to the <DELAYED> tag.

---

**DESCRIPTION**

The <DELAYED> tag allows you to specify text that contains a tag in an argument to another tag. Execution of the <DELAYED> tag is delayed until the outer tag's text is processed.

Using a tag within an argument to another tag is called **nesting** a tag in an outer tag's argument. Normally, the tag translator evaluates the nested tag first, then evaluates the outer tag.

## <DELAYED>

When you want the outer tag to be processed *before* the nested tag, you must use the <DELAYED> tag to surround the nested tag. For example, when the outer tag is the <DEFINE\_SYMBOL>(symbol name\text string) tag, you must use the <DELAYED> tag to surround the nested tag in the *text string* argument. This causes the nested tag to be preserved until it is about to be output using the <REFERENCE> tag.

---

### EXAMPLE

The following example shows how to use the <DELAYED> tag. The symbol TEMP\_CHART\_EX is associated with the text string that is enclosed by the <DELAYED> and <ENDDelayed> tags. None of the enclosed tags will have been evaluated.

```
<define_symbol>(temp_chart_ex\<delayed>
<example>(Temperature Chart)
<code_example>
    Centigrade 0    Fahrenheit 32
    Centigrade 100 Fahrenheit 212
<endcode_example>
<endexample>
<enddelayed>)
```

Subsequently, you can refer to the symbol name with a <REFERENCE> tag, as in the following example:

```
<REFERENCE>(temp_chart_ex)
```

Using this reference results in the following output:

#### Example n-n Temperature Chart

---

Centigrade	0	Fahrenheit	32
Centigrade	100	Fahrenheit	212

---



---

## <DOCTYPE>

Identifies the doctype for your file. This tag is for informational purposes only.

---

### SYNTAX

<DOCTYPE>(document type)

---

### ARGUMENTS

*document type*

Specifies the doctype you want to use for your file.

---

### DESCRIPTION

The <DOCTYPE> tag identifies the doctype for your file. This tag is for informational purposes only; it does not designate the doctype.

You can include this tag anywhere in your file for informational purposes. You do not have to include the <DOCTYPE> tag in your file in order to process your file, however. Instead, you must specify the doctype on the DOCUMENT command line, which allows you to process your file using different doctypes to determine which one suits your needs.

---

### EXAMPLE

The following example shows how you can identify a book's doctype as "REPORT". Refer to *VAX DOCUMENT Using Doctypes and Related Tags* for sample output of the various doctypes.

```
<DOCTYPE>(report)
```

## <DOUBLE\_QUOTE>

---

## <DOUBLE\_QUOTE>

Supplies a double quotation mark as it appears when it is typeset.

---

### SYNTAX <DOUBLE\_QUOTE>

---

**ARGUMENTS** *None.*

---

**related tags**

- <PARENDCHAR>
  - <QUOTE>
  - <SINGLE\_QUOTE>
- 

**DESCRIPTION** The <DOUBLE\_QUOTE> tag supplies a double quotation mark as it appears when it is typeset. This is different from the double quotation mark that your keyboard produces.

Your keyboard has two possible characters you can use for quotation marks: the double quote, ( " ) (ASCII 34) and the single quote ( ' ) (ASCII 39). In programming languages, a quoted string usually begins and ends with the same character, either *"a quoted string,"* or *'a quoted string.'* Using the double or single quotation marks from your keyboard is usually adequate for showing examples of user input or screen displays.

If you want your output to show a double quotation mark as it is appears when it is typeset, however, use the <DOUBLE\_QUOTE> tag.

If you want your output to follow typesetting conventions by using distinct opening and closing quotation marks, use the <QUOTE> tag instead of double quotation marks from your keyboard. You would use the <QUOTE> tag, for example, in citing a passage from another book. See the <QUOTE> tag description for more information.

---

**EXAMPLES** The following example shows how to use the <DOUBLE\_QUOTE> tag.

**i** <P>You can cause the translation of a symbol by using a double quotation mark (<DOUBLE\_QUOTE>) directly in front of it.

This example produces the following output:

You can cause the translation of a symbol by using a double quotation mark (") directly in front of it.

The following example shows what your output would be like without using the tag, but just entering the double quotation mark character from the keyboard.

- 2 <P>You can cause the translation of a symbol by using a double quotation mark (") directly in front of it.

This example produces the following output:

You can cause the translation of a symbol by using a double quotation mark (") directly in front of it.

## <ELEMENT>

---

## <ELEMENT>

Identifies a file that contains an element of a book.

---

**SYNTAX**      **<ELEMENT>** (*file spec*)

---

**ARGUMENTS**    ***file spec***  
Specifies a file. You must supply the file type (for example, .sdml).

---

**related tags**

- <INCLUDES\_FILE>
- <PROFILE>

---

**restrictions**      Valid only in the context of a <PROFILE> tag.

---

**DESCRIPTION**    The <ELEMENT> tag identifies a file that contains an element of a book. A profile of a book is required in order to build (process) a book. A book's profile contains an ordered list of the elements that comprise the book. In the profile, you label each of these elements with an <ELEMENT> tag. Only those files listed with <ELEMENT> tags are included in the book during the bookbuild. List those files in the profile according to the order in which they appear in the book.

A file specified as an element must begin with one of the following tags:

- <APPENDIX>
- <CHAPTER>
- <FRONT\_MATTER>
- <GLOSSARY>
- <PART>

For more information on creating a profile and bookbuilding, see *VAX DOCUMENT Producing Online and Printed Documentation*.

---

## EXAMPLE

Assume that a writer is writing a book that contains several chapters. One of the chapters is the file `My_Intro_Chap.SDML`. At the top of this file is included the `<CHAPTER>(Introduction\intro_chap)` tag to distinguish it as a chapter.

Along with the other <ELEMENT> tags for this book, the profile would then contain the following:

```
<ELEMENT>(My_Intro_Chap.SDML)
```

## **<ELEMENT>**

See the example in the <PROFILE> tag description to see how to arrange the <ELEMENT> tags in a profile.

## <ELLIPSIS>

---

## <ELLIPSIS>

Supplies vertical ellipsis points to show omitted material in text and tables.

---

### SYNTAX <ELLIPSIS>

---

**ARGUMENTS** *None.*

---

**related tags** • <HELLIPSIS>

---

**restrictions** Invalid in the context of a <MATH> tag.

Place the tag alone on a line in monospaced examples, or else the first “dot” of the ellipsis will be misaligned.

---

**DESCRIPTION** The <ELLIPSIS> tag supplies vertical ellipsis points to show omitted material in text and tables. The vertical ellipsis is positioned with respect to the current left margin.

---

**EXAMPLES** The following example shows how to use the <ELLIPSIS> tag.

**1** <P>The instruction sequence listed here (patterned after code in module PROCSTRT) shows this second technique.

```
<CODE_EXAMPLE>
PUSHL    executive-mode-PSL
BSBB     DOREI
<ELLIPSIS>
PUSHL    user-mode-PSL
BSBB     DOREI
<ELLIPSIS>
DOREI:   REI
<ENDCODE_EXAMPLE>
```

This example produces the following output:

The instruction sequence listed here (patterned after code in module PROCSTRT) shows this second technique.

```
    PUSHL    executive-mode-PSL
    BSBB     DOREI
    .
    .
    PUSHL    user-mode-PSL
    BSBB     DOREI
    .
    .
DOREI:   REI
```

The following example shows how you can use the <ELLIPSIS> tag in a table.

```

2 <TABLE>(Rules for Determining Expression Modes\express_modes_tab)
  <TABLE_ATTRIBUTES>(MULTIPAGE)
  <TABLE_SETUP>(2\43)
  <TABLE_HEADS>(Expression\Value Type)
  <TABLE_ROW>(Integer value\Integer)
  <TABLE_ROW>(<ELLIPSIS>\<ELLIPSIS>)
  <TABLE_ROW>(String value\String)
  <TABLE_ROW>(Integer lexical function\Integer)
  <TABLE_ROW>(String lexical function\<ELLIPSIS>)
  <TABLE_ROW>(Integer symbol\Integer)
  <TABLE_ROW>(String symbol\String)
  <TABLE_ROW>(Any value .AND. or .OR. any value\Integer)
  <TABLE_ROW>(Any value\Integer)
  <TABLE_ROW>(Any value\Integer)
<ENDTABLE>

```

This example produces the following output:

**Table n-n Rules for Determining Expression Modes**

<b>Expression</b>	<b>Value Type</b>
Integer value	Integer
.	.
.	.
.	.
String value	String
Integer lexical function	Integer
String lexical function	.
.	.
.	.
Integer symbol	Integer
String symbol	String
Any value .AND. or .OR. any value	Integer
Any value	Integer
Any value	Integer

## <EMPHASIS>

---

## <EMPHASIS>

Marks a word or phrase for distinctive typographical treatment.

---

### SYNTAX

**<EMPHASIS>**(text { \[*BOLD*] \[*ITALIC*] \[*SMALLCAPS*] \[*SMALL\_BOLDCAPS*] } )

---

### ARGUMENTS

#### **TEXT**

Specifies the text you want to emphasize.

#### **BOLD**

This is an optional keyword argument and must be the second argument if used. It specifies that the text be set in a bold font.

#### **ITALIC**

This is an optional keyword argument and must be the second argument if used. It specifies that the text be set in the current italic font. This is the default.

#### **SMALLCAPS**

This is an optional keyword argument and must be the second argument if used. It specifies that the text be set in small capital letters.

#### **SMALL\_BOLDCAPS**

This is an optional keyword argument and must be the second argument if used. It specifies that the text be set in small capital letters in a bold font.

---

### related tags

- <QUOTE>
- <UNDERLINE>

---

### DESCRIPTION

The <EMPHASIS> tag marks a word or phrase for distinctive typographical treatment. The default form of emphasis is to set the text in an italic font. Use one of the arguments to change the emphasis to another font.

---

### EXAMPLE

The following example shows how to use the <EMPHASIS> tag.

```
<P>An <EMPHASIS>(overuse) of the <TAG>(EMPHASIS) tag  
<EMPHASIS>(inevitably\bold) ALWAYS  
<EMPHASIS>(reduces\small_boldcaps) its  
<EMPHASIS>(effectiveness\smallcaps).
```

This example produces the following output:



**<EMPHASIS>**

An *overuse* of the <EMPHASIS> tag **inevitably** ALWAYS REDUCES its EFFECTIVENESS.

## <ENDCOPYRIGHT\_PAGE>

---

## <ENDCOPYRIGHT\_PAGE>

Terminates the copyright page and optionally provides text that is output at the bottom of the copyright page.

---

### SYNTAX

**<ENDCOPYRIGHT\_PAGE>***[(identification)]*

---

### ARGUMENTS

#### *identification*

This argument is optional. It specifies a local convention for information on the copyright page.

---

### related tags

- <COPYRIGHT\_PAGE>
- <FRONT\_MATTER>
- <PRINT\_DATE>

---

### restrictions

Use only to terminate a <COPYRIGHT\_PAGE> tag. You can only use the copyright page in the context of the <FRONT\_MATTER> tag.

---

### DESCRIPTION

The <ENDCOPYRIGHT\_PAGE> tag terminates the copyright page and optionally provides text that is output at the bottom of the copyright page.

---

### EXAMPLE

The following example shows the order in which to use the copyright page tags. The output of this example is a separate copyright page in the front matter of the book, and the page contains the print and copyright dates. The argument to the <ENDCOPYRIGHT\_PAGE> tag, "ZK5352", is a local identification number that appears at the bottom of the copyright page.

```
<FRONT_MATTER>(front)
<COPYRIGHT_PAGE>
<PRINT_DATE>(November 1990)
<COPYRIGHT_DATE>(1990)
<ENDCOPYRIGHT_PAGE>(ZK5352)
<ENDFRONT_MATTER>
```

To see an example of all the front matter tags in their correct order, see the example in the <FRONT\_MATTER> tag description.

See the copyright page of this manual for a sample copyright page.

---

## <ENDPART\_PAGE>

Terminates a part page (which is a dividing page between major sections of a document) and optionally specifies paging attributes for text that follows.

---

**SYNTAX**      <ENDPART\_PAGE>[(*RENUMBER*)]

---

**ARGUMENTS**      *RENUMBER*

This is an optional keyword argument. It specifies that page numbering following the part page begins with 1. If you do not use this argument, page numbering continues following the part page with the next odd-numbered page.

Using this argument has no effect on Bookreader output.

---

**related tags**      • <PART\_PAGE>

---

**restrictions**      Use only to terminate a <PART\_PAGE> tag.

---

**DESCRIPTION**      The <ENDPART\_PAGE> tag terminates a part page and optionally specifies paging attributes for text that follows.

---

**EXAMPLE**      See the example in the discussion of the <PART> tag.

## <ENDTITLE\_PAGE>

---

## <ENDTITLE\_PAGE>

Terminates a title page and optionally specifies text to appear at the bottom of the title page.

---

**SYNTAX**            <ENDTITLE\_PAGE>[(*text*)]

---

**ARGUMENTS**        *text*  
This is an optional argument. It specifies that a line of text appear at the bottom of the title page.

---

**related tags**        • <PART\_PAGE>

---

**restrictions**        Use only to terminate a <TITLE\_PAGE> tag.

---

**DESCRIPTION**        The <ENDTITLE\_PAGE> tag terminates a title page and optionally specifies text to appear at the bottom of the title page.

---

**EXAMPLE**            The following example shows how to code a title page of a short document. See the example in the discussion of the <FRONT\_MATTER> tag for an example of using this tag in the front matter of a book.

```
<TITLE_PAGE>  
<TITLE>(How to Code your SDML Source Files)  
<ENDTITLE_PAGE>(This is for informational purposes only.)
```

---

## <EXAMPLE>

Labels the beginning of an example.

---

### SYNTAX

<EXAMPLE>[(*example caption*[\ *symbol name*])]

---

### ARGUMENTS

#### *example caption*

This is an optional argument. It specifies the text of the caption to be associated with the example. The example caption and the associated example number will be included in the table of contents of the document.

#### *symbol name*

This is an optional argument. It specifies the symbolic identifier to be associated with the example. The symbol name is assigned a numeric value, which is the current example number. The symbol and its value are placed in the symbol table.

Symbol names must not exceed 31 characters and must only include alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

### related tags

- <CODE\_EXAMPLE>
- <EXAMPLE\_ATTRIBUTES>
- <EXAMPLE\_FILE>
- <EXAMPLE\_SPACE>
- <INTERACTIVE>
- <REFERENCE>
- <S>
- <SAMPLE\_TEXT>
- <U>
- <VALID\_BREAK>

---

### restrictions

Cannot contain any text element tags, such as <CHAPTER>, <FIGURE>, <MATH>, <LIST>, <P>, or <TABLE>.

Invalid in the context of a <FOOTNOTE> tag.

Can only contain text that you label with <CODE\_EXAMPLE>, <INTERACTIVE>, and <SAMPLE\_TEXT> tags. Use a <P> tag within the context of the <SAMPLE\_TEXT> tag to provide paragraph format of the sample text.

If you use an <EXAMPLE\_FILE> tag in the context of an <EXAMPLE> tag, that file included by the <EXAMPLE\_FILE> tag must contain either a code example (using the <CODE\_EXAMPLE> tag) or an interactive example (using the <INTERACTIVE> tag).

## <EXAMPLE>

required  
terminator

<ENDEXAMPLE>

### DESCRIPTION

The <EXAMPLE> tag labels the beginning of a **formal example** or **informal example**. A formal example is an example that has a number and a caption, can be referenced, and is listed in the table of contents. An informal example does not have a number or a caption, cannot be referenced, and is not listed in the table of contents. Moreover, you create an informal example with the <CODE\_EXAMPLE> tag. See the <CODE\_EXAMPLE> tag description for more information.

You can create a formal example in a number of ways:

- By including a specified file in your example by using the <EXAMPLE\_FILE> tag
- By leaving space for an example to be pasted in during final production by using the <EXAMPLE\_SPACE> tag
- By using text labeled with <CODE\_EXAMPLE>, <INTERACTIVE>, <S>, <SAMPLE\_TEXT>, and <U> tags

You cannot use tables, lists, paragraphs, and other text elements in a formal example. You can, however, use a <P> tag within the context of a <SAMPLE\_TEXT> tag to provide paragraph format of the sample text. See the <EXAMPLE\_SEQUENCE> tag description in *VAX DOCUMENT Using Doctypes and Related Tags* for information about how to construct an example sequence that can include a variety of text elements.

If the body of the example spans more than a single page of text, the example caption is repeated on each page on which the example continues.

If the example is more than a page, page breaks are handled automatically by the text processor. You can control the page breaking by using <PAGE> tags to specify explicit page breaks, or <VALID\_BREAK> tags within the example to specify good breaking points. If you explicitly break a page by using the <PAGE> or <VALID\_BREAK> tags, the example caption is continued on the following page, as it is if the page is broken automatically by the text processor.

When a **floating** figure or floating example precedes a multipage table (all tables not marked with the KEEP keyword are multipage), a succession of short pages may occur before the page on which the table begins. This is because a multipage table forces any previous floating figures or examples to be output before the table begins.

If this situation occurs, code the preceding floating figures or examples with the KEEP keyword so that they will be kept with the text preceding them, resulting in better-balanced pages.

See the description of the FLOAT keyword argument to the <EXAMPLE\_ATTRIBUTES> tag for information about floating figures or examples.

Note the following when processing a book for Bookreader display:

- You must use the *symbol name* argument to process a formal example for Bookreader display.
- DOCUMENT automatically creates **pop-up windows** for all formal examples.
- When introducing and cross-referencing a formal example, you must use the <REFERENCE> tag to create a **hotspot**. Hotspots are regions in the windows you **click on** with your mouse to access cross-referenced topics.
- You can use the <ONLINE\_POPUP> tag if you want to make informal examples **pop up**.

## EXAMPLES

The following example shows a one-page example that uses the WIDE argument of the <EXAMPLE\_ATTRIBUTE> tag. The contents of the example are included using the <EXAMPLE\_FILE> tag. The file that is included must contain either a code example or interactive example.

```

❏ <EXAMPLE>(VAXcluster Multi-file Summary\multi_file_exam)
   <EXAMPLE_ATTRIBUTES>(WIDE)
   <EXAMPLE_FILE>(monitor_multi_file_summary.sdml)
   <ENDEXAMPLE>

```

This example produces the following output:

### Example n-n VAXcluster Multi-file Summary

```

+-----+          VAX/VMS Monitor Utility
| AVE |          TIME IN PROCESSOR MODES
+-----+          MULTI-FILE SUMMARY

```

Node:	MOE	CURLEY	LARRY	Row	Row	Row
From:	15-APR-1984 18:17	15-APR-1984 18:17	15-APR-1984 18:17	Sum	Average	Minimum
To:	15-APR-1984 20:17	15-APR-1984 20:17	15-APR-1984 20:17			
Interrupt Stack	6.51	0.50	6.25	13.2	4.4	0.50
Kernel Mode	25.73	0.42	12.43	38.5	12.8	0.42
Executive Mode	9.46	0.95	1.81	12.2	4.0	0.95
Supervisor Mode	1.97	0.00	0.16	2.1	0.7	0.00
User Mode	13.24	5.33	56.14	74.7	24.9	5.33
Compatibility Mode	0.00	0.07	0.00	0.0	0.0	0.00
Idle Time	23.61	0.02	92.63	116.2	38.7	0.02

```

+-----+          VAX/VMS Monitor Utility
| AVE |          PAGE MANAGEMENT STATISTICS
+-----+          MULTI-FILE SUMMARY

```

Node:	MOE	CURLEY	LARRY	Row	Row	Row
From:	15-APR-1984 18:17	15-APR-1984 18:17	15-APR-1984 18:17	Sum	Average	Minimum
To:	15-APR-1984 20:17	15-APR-1984 20:17	15-APR-1984 20:17			
Page Fault Rate	36.73	8.81	0.49	46.0	15.3	0.49
Page Read Rate	14.28	4.71	0.00	19.0	6.3	0.00
Page Read I/O Rate	1.20	0.70	0.00	1.9	0.6	0.00
Page Write Rate	0.00	0.00	0.00	0.0	0.0	0.00
Page Write I/O Rate	0.00	0.00	0.00	0.0	0.0	0.00

Example n-n (continued on next page)

## <EXAMPLE>

### Example n-n (Cont.) VAXcluster Multi-file Summary

---

Free List Fault Rate	8.60	1.40	0.24	10.2	3.4	0.24
Modified List Fault Rate	5.83	2.29	0.00	8.1	2.7	0.00
Demand Zero Fault Rate	12.96	1.68	0.24	14.8	4.9	0.24
Global Valid Fault Rate	8.10	2.69	0.00	10.8	3.6	0.00
Wrt In Progress Fault Rate	0.00	0.00	0.00	0.0	0.0	0.00
System Fault Rate	4.92	0.53	0.18	5.6	1.8	0.18
Free List Size	7586.30	8630.14	9665.06	25881.5	8627.1	7586.30
Modified List Size	87.69	324.07	32.12	443.8	147.9	32.12

---

The following example shows how to use the <CODE\_EXAMPLE> and <ENDCODE\_EXAMPLE> tags in an informal example.

```
2 <EXAMPLE>
  <CODE_EXAMPLE>

  $!
  $! The next two lines redefine the cmd. line prompt to be the node
  $! name's initial.
  $!
  $ NODE :== 'F$LOGICAL("SYS$NODE")'
  $ SET PROMPT="'F$EXTRACT(1,1,NODE)':"
  $!

  <ENDCODE_EXAMPLE>
  <ENDEXAMPLE>
```

This example produces the following output:

```
$!
$! The next two lines redefine the cmd. line prompt to be the node
$! name's initial.
$!
$ NODE :== 'F$LOGICAL("SYS$NODE")'
$ SET PROMPT="'F$EXTRACT(1,1,NODE)':"
$!
```

The following example shows how to use the <CODE\_EXAMPLE> and <ENDCODE\_EXAMPLE> tags in a formal example.

```
3 <EXAMPLE> (Login Commands\login_exam)
  <CODE_EXAMPLE>

  $!
  $! The next two lines redefine the cmd. line prompt to be the node
  $! name's initial.
  $!
  $ NODE :== 'F$LOGICAL("SYS$NODE")'
  $ SET PROMPT="'F$EXTRACT(1,1,NODE)':"
  $!

  <ENDCODE_EXAMPLE>
  <ENDEXAMPLE>
```

This example produces the following output:



**Example n-n Login Commands**

---

```

$!
$!   The next two lines redefine the cmd. line prompt to be the node
$!   name's initial.
$!
$ NODE :=='F$LOGICAL("SYS$NODE")'
$ SET PROMPT="'F$EXTRACT(1,1,NODE)':"
$!

```

---

The following example shows how to use the <INTERACTIVE>, <ENDINTERACTIVE>, <S>, and <U> tags within an example. Note that you should specify whatever space follows the system prompt within the <S> tag.

4 <P>If you do not specify the full command line, DCL will prompt you for the missing information. For example, if you do not specify an input file and an output file when you enter the COPY command, you will be prompted as follows:

```

<EXAMPLE>(Using the COPY Command\using_copy_command)
<INTERACTIVE>
<S>($ )<U>(COPY)
<S>($ _From: )<U>(INTERACT.SDML)
<S>($ _To: )<U>(NEWFILE.SDML)
<ENDINTERACTIVE>
<ENDEXAMPLE>

```

This example produces the following output:

If you do not specify the full command line, DCL will prompt you for the missing information. For example, if you do not specify an input file and an output file when you enter the COPY command, you will be prompted as follows:

**Example n-n Using the COPY Command**

---

```

$ COPY
$ _From: INTERACT.SDML
$ _To: NEWFILE.SDML

```

---

The following example shows how to use the <SAMPLE\_TEXT> tag within the <EXAMPLE> and <ENDEXAMPLE> tags to get specially formatted sample text within a formal example.

5 <EXAMPLE>(Some Sample Text\sample\_text)

```

<SAMPLE_TEXT>
<P>This paragraph shows how to use the <SAMPLE_TEXT> tag within the
<EXAMPLE> and <ENDEXAMPLE> tags to get specially formatted sample
text within a formal example.
<ENDSAMPLE_TEXT>
<ENDEXAMPLE>

```

This example produces the following output:

**<EXAMPLE>**

**Example n-n Some Sample Text**

---

This paragraph shows how to use the <SAMPLE\_TEXT> tag within the <EXAMPLE> and <ENDEXAMPLE> tags to get specially formatted sample text within a formal example.

---

---

## <EXAMPLE\_ATTRIBUTES>

Specifies attributes for the current example.

---

### SYNTAX

**<EXAMPLE\_ATTRIBUTES>**( { *FLOAT*  
                  *KEEP*  
                  *MULTIPAGE* }  
[ \ *WIDE* ] )

---

### ARGUMENTS

#### **FLOAT**

Specifies whether the location of a one-page example is allowed to float. *FLOAT* indicates that if there is not enough room on the current page for the example, the text processor will fill the current page with the text from the source file that follows the <EXAMPLE> tag sequence, and place the example at the top of the next page of output.

Float is the default for an example that has a caption and does not specify *MULTIPAGE* or *KEEP*. *FLOAT* has no effect for Bookreader output.

#### **KEEP**

Specifies whether the example is to be kept with the text that immediately precedes it. *KEEP* has no effect for Bookreader output.

#### **MULTIPAGE**

Specifies that the example has multiple elements, and that each element is allowed to break at the start of a new page. This argument is required if the example elements require more than one page. When an example is continued, the example caption is automatically repeated at the beginning of each new page of output. *MULTIPAGE* has no effect for Bookreader output.

#### **WIDE**

This is an optional keyword argument. It specifies that the width of the example exceeds the document's default width for text. You can use *WIDE* with *FLOAT*, *KEEP*, or *MULTIPAGE*, or you can use *WIDE* as the only argument. Depending on the doctype, this argument is interpreted as follows:

- If the doctype contains a left margin area that is normally used for headings, the example's width will span that area as well as the normal text area.
- If the document uses a multicolumn format, the example suspends multicolumn output while the example is processed. The example is output across two columns and then multicolumn output is restored.
- Depending on the doctype, this argument provides a range of font sizes and font styles for examples.

## <EXAMPLE\_ATTRIBUTES>

---

### related tags

- <EXAMPLE>
  - <EXAMPLE\_FILE>
  - <EXAMPLE\_SPACE>
  - <REFERENCE>
- 

### restrictions

Valid only in the context of an <EXAMPLE> tag.

---

### DESCRIPTION

The <EXAMPLE\_ATTRIBUTES> tag specifies attributes for the current example. By default, if an example does not fit on the current output page, it is placed at the top of the next page. The text that follows the example in the SDML file is used to fill the current output page. Thus, the example's position in the text may change. It is said to **float**. Formal examples that float, then, may appear out of place. Use the <EXAMPLE\_ATTRIBUTES>(KEEP) tag to force the formal example to be kept with the text that immediately precedes it.

If you think the example will be longer than a page, use the MULTIPAGE keyword argument to indicate that the example requires more than one page of output. The example will not be allowed to float.

If the example is formal and you want to align the caption with the example (that has been extended into the left column), use the WIDE argument with both the <EXAMPLE\_ATTRIBUTES> and <CODE\_EXAMPLE> tags.

---

### EXAMPLE

This example shows how to use the MULTIPAGE and WIDE arguments to the <EXAMPLE\_ATTRIBUTES> tag. Note the following:

- The <VALID\_BREAK> tags in the <CODE\_EXAMPLE> provide the text formatter with information about suitable places to break pages.
- Because the example is wide, the WIDE argument is specified not only for the <EXAMPLE\_ATTRIBUTES> tag, but also for the <CODE\_EXAMPLE> tag within the example.

## <EXAMPLE\_ATTRIBUTES>

```
<EXAMPLE> (Command Procedure for Adding a User\adduserex)
<EXAMPLE_ATTRIBUTES> (MULTIPAGE\WIDE)
<CODE_EXAMPLE> (WIDE)
$ !
$ !   ADDUSER.COM -- Adds a new user to the system authorization file
$ !
$ USERDISK = "WRKD$:"                ! Default disk for new users
$ UAF = "$AUTHORIZE"
<VALID_BREAK>
$ ON CONTROLY THEN GOTO CLEANUP
$ ON WARNING THEN GOTO CLEANUP
$ OLDDIR = F$LOGICAL("SYS$DISK") + F$DIRECTORY()'
$ PREVPRIV = F$SETPRV("SYSPRV")
$ IF .NOT. F$PRIVILEGE("SYSPRV") THEN GOTO NOPRIV
$ SET DEFAULT SYS$SYSTEM
$ !
$ !   Request account information
$ !
$ INQUIRE USERNAME "Username"
$ INQUIRE FULLNAME "Full name"
$ SET TERMINAL/NOECHO
$ INQUIRE PASSWORD "Password ['Username']"
$ SET TERMINAL/ECHO
$ IF PASSWORD .EQS. "" THEN PASSWORD = USERNAME
$GET_GRP:
$ INQUIRE GRP "UIC Group Number"
$ IF GRP .EQS. "" THEN GRP = "*"
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "Determine the UIC from the following listing:"
$ WRITE SYS$OUTPUT ""
$ UAF SHOW ['GRP',*]/BRIEF
$ INQUIRE UIC
$ IF UIC .EQS. "" THEN GOTO GET_GRP
$ IF F$LOCATE("[",UIC) .EQ. F$LENGTH(UIC) .AND. -
    F$LOCATE("<",UIC) .EQ. F$LENGTH(UIC) THEN UIC = "[" + UIC + "]"
$ INQUIRE ACCOUNT "Account Name [VMS]"
$ IF ACCOUNT .EQS. "" THEN ACCOUNT = "VMS"
$ INQUIRE PRIVS "Privileges [NONE]"
$ IF PRIVS .NES. "" THEN PRIVS = "/PRIV=(" + PRIVS + ")"
$ USERDIR = F$EXTRACT(0,9,USERNAME)
$ INQUIRE TMP "Login Directory ['USERDIR']"
$ IF TMP .NES. "" THEN USERDIR = TMP
$ INQUIRE TMP "Login Device ['USERDISK']"
$ IF TMP .NES. "" THEN USERDISK = TMP
$ DQUOTA = 0
$ IF F$SEARCH("''USERDISK'[0,0]QUOTA.SYS") .EQS. "" THEN GOTO NQO
$ DQUOTA = 1
<VALID_BREAK>

$ !
$ !   Restore prior working environment
$ !
$CLEANUP:
$ SET TERMINAL/ECHO
$ PREVPRIV = F$SETPRV(PREVPRIV)
$ SET DEFAULT 'OLDDIR'
$ EXIT
$ !
$ !   In case proper privileges are not set
$ !
s$NOPRIV:
$ WRITE SYS$OUTPUT "You need SETPRV or SYSPRV privilege to run this procedure"
$ GOTO CLEANUP
<ENCODE_EXAMPLE>
<ENDEXAMPLE>
```

This example produces the following output:

## <EXAMPLE\_ATTRIBUTES>

### Example n-n Command Procedure for Adding a User

---

```
$ !
$ !   ADDUSER.COM -- Adds a new user to the system authorization file
$ !
$ USERDISK = "WRKD$:"           ! Default disk for new users
$ UAF = "$AUTHORIZE"
$ ON CONTROLY THEN GOTO CLEANUP
$ ON WARNING THEN GOTO CLEANUP
$ OLDDIR = F$LOGICAL("SYS$DISK") + F$DIRECTORY()'
$ PREVPRIV = F$SETPRV("SYSPRV")
$ IF .NOT. F$PRIVILEGE("SYSPRV") THEN GOTO NOPRIV
$ SET DEFAULT SYS$SYSTEM
$ !
$ !   Request account information
$ !
$ INQUIRE USERNAME "Username"
$ INQUIRE FULLNAME "Full name"
$ SET TERMINAL/NOECHO
$ INQUIRE PASSWORD "Password ['Username']"
$ SET TERMINAL/ECHO
$ IF PASSWORD .EQS. "" THEN PASSWORD = USERNAME
$GET_GRP:
$ INQUIRE GRP "UIC Group Number"
$ IF GRP .EQS. "" THEN GRP = "*"
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "Determine the UIC from the following listing:"
$ WRITE SYS$OUTPUT ""
$ UAF SHOW ['GRP',*]/BRIEF
$ INQUIRE UIC
$ IF UIC .EQS. "" THEN GOTO GET_GRP
$ IF F$LOCATE("[",UIC) .EQ. F$LENGTH(UIC) .AND. -
    F$LOCATE("<",UIC) .EQ. F$LENGTH(UIC) THEN UIC = "[" + UIC + "]"
$ INQUIRE ACCOUNT "Account Name [VMS]"
$ IF ACCOUNT .EQS. "" THEN ACCOUNT = "VMS"
$ INQUIRE PRIVS "Privileges [NONE]"
$ IF PRIVS .NES. "" THEN PRIVS = "/PRIV=( " + PRIVS + " )"
$ USERDIR = F$EXTRACT(0,9,USERNAME)
$ INQUIRE TMP "Login Directory ['USERDIR']"
$ IF TMP .NES. "" THEN USERDIR = TMP
$ INQUIRE TMP "Login Device ['USERDISK']"
$ IF TMP .NES. "" THEN USERDISK = TMP
$ DQUOTA = 0
$ IF F$SEARCH("''USERDISK'[0,0]QUOTA.SYS") .EQS. "" THEN GOTO NQO
$ DQUOTA = 1
$ !
$ !   Restore prior working environment
$ !
$CLEANUP:
$ SET TERMINAL/ECHO
$ PREVPRIV = F$SETPRV(PREVPRIV)
$ SET DEFAULT 'OLDDIR'
$ EXIT
$ !
$ !   In case proper privileges are not set
$ !
$NOPRIV:
$ WRITE SYS$OUTPUT "You need SETPRV or SYSPRV privilege to run this procedure"
$ GOTO CLEANUP
```

---

---

## <EXAMPLE\_FILE>

Causes a separate file containing an example to be included in the source file.

---

**SYNTAX**      <EXAMPLE\_FILE> (*file spec*)

---

**ARGUMENTS**      *file spec*  
Specifies the file containing the example. The file can be a logical name. See the <INCLUDES\_FILE> tag description for information on how to define a logical name.

- 
- related tags**
- <EXAMPLE>
  - <EXAMPLE\_ATTRIBUTES>
  - <EXAMPLE\_SPACE>
  - <REFERENCE>

---

**restrictions**      Valid only in the context of an <EXAMPLE> tag.

---

**DESCRIPTION**      The <EXAMPLE\_FILE> tag causes a separate file containing an example to be included in the source file.

The included file should be an SDML file containing a complete coded example, perhaps using <CODE\_EXAMPLE> or <INTERACTIVE> tags. If the <EXAMPLE> tag specifies the WIDE argument, the included file must specify the WIDE argument in the appropriate tags. For more information on correctly coding examples and example files, see *VAX DOCUMENT Producing Online and Printed Documentation*.

You can give the <EXAMPLE\_FILE> tag a logical name for an argument. Define the logical name by first entering an <INCLUDES\_FILE> tag in the profile. The information in that tag is used to make the proper logical name assignment during processing.

---

**EXAMPLE**      See the example in the discussion of the <EXAMPLE> tag.

## <EXAMPLE\_SPACE>

---

## <EXAMPLE\_SPACE>

Leaves space for an example to be pasted in during final production.

---

### SYNTAX

**<EXAMPLE\_SPACE>**( { *value*  
*FULL\_PAGE* } [ \ *text* ])

---

### ARGUMENTS

#### ***value***

Specifies the size of the space in picas, a scale used by typesetters. There are approximately 6 picas in an inch (and 12 points in a pica). Thus, if the example is 4 inches high, you should specify 24 picas. The value must not exceed page length limitations for the current doctype. If you omit the value, a default value of 2 picas is used.

#### ***FULL\_PAGE***

This keyword specifies that a full blank page be reserved for the example. For Bookreader output, a small amount (smaller than a printed page or a Bookreader page) of blank space is left.

#### ***text***

This is an optional argument. It specifies the text to be printed in the center of the space. (For example, an art file number, a file name that contains the example, or some other note.)

---

### related tags

- <EXAMPLE>
- <EXAMPLE\_ATTRIBUTES>
- <EXAMPLE\_FILE>
- <REFERENCE>

---

### restrictions

Valid only in the context of an <EXAMPLE> tag.

The *value* argument must not exceed page length limitations.

---

### DESCRIPTION

The <EXAMPLE\_SPACE> tag, leaves space for an example to be pasted in during final production. Any text you supply as the second argument is printed in the center of the space.

---

### EXAMPLE

This example shows how to use the <EXAMPLE\_SPACE> tag.

```
<EXAMPLE>(Completing Form DD-214\annotated_dd214_exam)
<EXAMPLE_ATTRIBUTES>(wide)
<EXAMPLE_SPACE>(12\Photo-reduced copy of the end of DD-214 here.)
<ENDEXAMPLE>
```

This example produces the following output:



**Example n-n Completing Form DD-214**

---

---

## <FCMD>

---

## <FCMD>

Specifies the keyword portion of a command/parameter pair in a format section.

---

### SYNTAX

**<FCMD>**(*keyword* [*\ parameters*])

---

### ARGUMENTS

#### ***keyword***

Specifies the keyword portion of the command/parameter pair.

#### ***parameters***

This is an optional argument. It specifies the command parameters, if any.

Parameters specified in this argument differ in your output from arguments specified using the <FPARMS> tag. If you do specify the <FCMD> tag with a null second argument, it is recommended that you explicitly declare the absence of parameters using the <FPARMS> tag, as follows:

```
<FCMD>(KEYWORD) <FPARMS>()
```

If you do not specify a second argument, and if the <FPARMS> tag is not specified, the tag translator issues a warning message that it has no explicit declaration of parameters.

---

### related tags

- <FORMAT>
  - <FPARM>
  - <FPARMS>
- 

### restrictions

Valid only in the context of a <FORMAT> tag.

---

### DESCRIPTION

The <FCMD>tag specifies the keyword portion of a command/parameter pair in a format section. You can specify the keyword's argument list as the second argument to this tag. The <FCMD>tag should be used only in the context of a format section. A format section is established by <FORMAT> and <ENDFORMAT> tags.

If the text of the *parameters* argument does not fit on one line, the text formatter chooses suitable line breaks based on the presence of spaces.

---

## EXAMPLES

The following example specifies a single command keyword. The <FPARMS> tag is explicitly specified as null. The output that this example produces is shown in Output Sample 1.

1

```
<FORMAT>  
<FCMD>(exit) <FPARMS>()  
<ENDFORMAT>
```

The following example specifies the command keyword and its parameters using both the <FCMD> and <FPARMS> tags. The output that this example produces is shown in Output Sample 2.

2

```
<FORMAT>  
<FCMD>(append) <FPARMS>(input-file-spec output-file-spec)  
<ENDFORMAT>
```

The following example specifies both the *keyword* and the *parameters* arguments to the <FCMD> tag. The output that this example produces is shown in Output Sample 3.

Notice the difference between Output Sample 2 and 3. In Sample 2, space is left between the keyword and the parameters. In Sample 3, no space is left. The formatting is produced by the syntax used in the <FCMD> tag.

3

```
<FORMAT>  
<FCMD>(f$element\ (element-number,delimiter,string))  
<ENDFORMAT>
```

The following example shows how to control the interpretation of keywords tagged with the <KEYWORD> tag when they are mixed in with the parameters of a format specification. The output that this example produces is shown in Output Sample 4.

4

```
<FORMAT>  
<FCMD>(set protection\[=code]<keyword>(/DEFAULT))  
<ENDFORMAT>
```

The following example specifies the command keyword and its parameters using the <FCMD>, <FPARMS>, and <FPARM> tags. The output that this example produces is shown in Output Sample 5.

5

```
<FORMAT>  
<FCMD>(append) <FPARMS>(one parameter) <FPARM>(two parameters) <FPARM>(three parameters)  
<ENDFORMAT>
```

---

**OUTPUT**            **exit**  
**SAMPLE 1**

---

**OUTPUT**            **append**    *input-file-spec output-file-spec*  
**SAMPLE 2**

<FCMD>

---

**OUTPUT**            **f\$element**(*element-number,delimiter,string*)  
**SAMPLE 3**

---

**OUTPUT**            **set protection**[=*code*]/**DEFAULT**  
**SAMPLE 4**

---

**OUTPUT**            **append**    *one parameter*  
**SAMPLE 5**                            *two parameters*  
    *three parameters*

---

## <FIGURE>

Labels the beginning of a figure.

---

**FORMAT**      <FIGURE>[(*figure caption*[\ *symbol name*])]

---

**ARGUMENTS**      *figure caption*

This is an optional argument. It specifies the text of the caption to be associated with the figure. The figure caption and the associated figure number will be included in the table of contents for the document. If the body of the figure spans more than a single page of text, the figure caption is repeated on each page on which the figure continues.

***symbol name***

This is an optional argument. It specifies the symbolic identifier to be associated with the example. The symbol name is assigned a numeric value, which becomes the current figure number. The symbol name and its value are placed in the symbol table.

Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores in them. Do not begin a symbol name with an underscore.

---

**related tags**

- <FIGURE\_ATTRIBUTES>
- <FIGURE\_FILE>
- <FIGURE\_SPACE>
- <ICON>
- <ICON\_FILE>
- <LINE\_ART>
- <REFERENCE>

---

**restrictions**

Invalid in the context of a <FOOTNOTE> tag.

---

**required terminator**

<ENDFIGURE>

---

**DESCRIPTION**

The <FIGURE> tag labels the beginning of a figure. A **formal figure** has a number and a caption, can be cross-referenced, and is listed in the table of contents. An **informal figure** does not have a number or a caption, cannot be cross-referenced, and is not listed in the table of contents.

## <FIGURE>

A formal figure is sometimes long. If you think one of your figures will be longer than a page, label it as a multipage figure in the <FIGURE\_ATTRIBUTES> tag. See the <FIGURE\_ATTRIBUTES> tag description for more information.

Note the following when processing a book for Bookreader display:

- You must use the *symbol name* argument to process a formal figure for Bookreader display.
- DOCUMENT automatically creates **pop-up windows** for all formal figures.
- When introducing and cross-referencing a formal figure, you must use the <REFERENCE> tag to create a **hotspot**. Hotspots are regions in the windows you **click on** with your mouse to access cross-referenced topics.
- You can use the <ONLINE\_POPUP> tag if you want to make informal figures **pop up**.

When you print an informal or formal figure, or process an informal figure for Bookreader display, the figure is indented in the same way the regular text is for the particular **doctype** you are using.

---

## EXAMPLES

In the following example, the <FIGURE\_ATTRIBUTES> tag contains a KEEP argument, to specify that the figure should be kept with the immediately preceding text. The <LINE\_ART> tag also has a KEEP argument, to specify that the figure should not break across pages.

The figures you are able to create using your terminal keyboard may be adequate for draft purposes, but may not be acceptable in the context of typeset output.

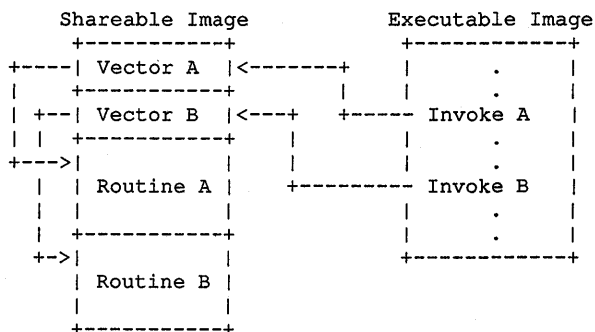
1 <FIGURE>(Example of Terminal-Created Art\transfer\_vector\_exam)  
<FIGURE\_ATTRIBUTES>(KEEP)  
<LINE\_ART>(KEEP)

```
          Shareable Image          Executable Image
          +-----+                +-----+
+----| Vector A |<-----+      | . . |
| | +---| Vector B |<----+      +----- Invoke A |
| | +-----+                | . |
+---->| | Routine A | +----- Invoke B |
| | +-----+                | . |
+-->| | Routine B |          +-----+
| | +-----+
+-----+
```

<ENDLINE\_ART>  
<ENDFIGURE>

This example produces the following output:

Figure n-n Example of Terminal-Created Art



The following example shows how to code a figure that is allowed to extend over several pages. The MULTIPAGE argument to the <FIGURE\_ATTRIBUTES> tag tells the text formatter that the figure is more than one page. Note that the text formatter does not break a large figure and adjust its portions on multiple pages; rather, MULTIPAGE indicates that the text formatter outputs the text, "Cont'd on next page", to indicate that the figure continues. The figure that you create should fit on one page. If it does not, the remainder of the figure does not print, but the text, "Cont'd on next page" still appears. For Bookreader output, the words "Cont'd on next page" do not appear.

```

2 <FIGURE>(Digital Multinational Character Set\char_set_fig)
  <FIGURE_ATTRIBUTES>(Multipage)
  <FIGURE_FILE>(LN03\character_set.SIX\27)
  <FIGURE_FILE>(PS\character_set.PS\27)
  <FIGURE_FILE>(BOOKREADER\character_set.BRF\54)
  <FIGURE_FILE>(LN03\character_set2.SIX\27)
  <FIGURE_FILE>(PS\character_set2.PS\27)
  <FIGURE_FILE>(BOOKREADER\character_set2.BRF\54)
  <ENDFIGURE>

```

This example produces the following output:

<FIGURE>

Figure n-n Digital Multinational Character Set

Column	0	1	2	3	4	5	6	7	
Bits	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	
Row	b4 b3 b2 b1	b4 b3 b2 b1	b4 b3 b2 b1	b4 b3 b2 b1	b4 b3 b2 b1	b4 b3 b2 b1	b4 b3 b2 b1	b4 b3 b2 b1	
0	0 0 0 0 NUL	0 0 0 0 DLE	20 16 10 SP	40 32 20 0	60 48 30 @	100 64 40 P	120 80 50 ,	140 96 60 p	160 112 70 0
1	0 0 0 1 SOH	1 1 1 1 DC1	21 17 11 !	41 33 21 1	61 49 31 A	101 65 41 Q	121 81 51 a	141 97 61 q	161 113 71 1
2	0 0 1 0 STX	2 2 2 2 DC2	22 18 12 "	42 34 22 2	62 50 32 B	102 66 42 R	122 82 52 b	142 98 62 r	162 114 72 2
3	0 0 1 1 ETX	3 3 3 3 DC3	23 19 13 #	43 35 23 3	63 51 33 C	103 67 43 S	123 83 53 c	143 99 63 s	163 115 73 3
4	0 1 0 0 EOT	4 4 4 4 DC4	24 20 14 \$	44 36 24 4	64 52 34 D	104 68 44 T	124 84 54 d	144 100 64 t	164 116 74 4
5	0 1 0 1 ENQ	5 5 5 5 NAK	25 21 15 %	45 37 25 5	65 53 35 E	105 69 45 U	125 85 55 e	145 101 65 u	165 117 75 5
6	0 1 1 0 ACK	6 6 6 6 SYN	26 22 16 &	46 38 26 6	66 54 36 F	106 70 46 V	126 86 56 f	146 102 66 v	166 118 76 6
7	0 1 1 1 BEL	7 7 7 7 ETB	27 23 17 '	47 39 27 7	67 55 37 G	107 71 47 W	127 87 57 g	147 103 67 w	167 119 77 7
8	1 0 0 0 BS	10 10 10 CAN	30 24 18 (	50 40 30 8	70 56 38 H	110 72 48 X	130 88 58 h	150 104 68 x	170 120 78 8
9	1 0 0 1 HT	11 11 11 EM	31 25 19 )	51 41 29 9	71 57 39 I	111 73 49 Y	131 89 59 i	151 105 69 y	171 121 79 9
10	1 0 1 0 LF	12 12 12 SUB	32 26 20 *	52 42 30 :	72 58 38 J	112 74 50 Z	132 90 60 j	152 106 70 z	172 122 80 A
11	1 0 1 1 VT	13 13 13 ESC	33 27 21 +	53 43 31 ;	73 59 39 K	113 75 51 [	133 91 61 k	153 107 71 {	173 123 81 B
12	1 1 0 0 FF	14 14 14 FS	34 28 22 ,	54 44 32 <	74 60 40 L	114 76 52 \	134 92 62 l	154 108 72 	174 124 82 C
13	1 1 0 1 CR	15 15 15 GS	35 29 23 -	55 45 33 =	75 61 41 M	115 77 53 ]	135 93 63 m	155 109 73 }	175 125 83 D
14	1 1 1 0 SO	16 16 16 RS	36 30 24 .	56 46 34 >	76 62 42 N	116 78 54 ^	136 94 64 n	156 110 74 ~	176 126 84 E
15	1 1 1 1 SI	17 17 17 US	37 31 25 /	57 47 35 ?	77 63 43 O	117 79 55 _	137 95 65 o	157 111 75 DEL	177 127 85 F

Key

Character ESC 33 Octal, 27 Decimal, 1B Hex

ZK-0347A-GE

Figure n-n (continued on next page)



Figure n-n (Cont.) Digital Multinational Character Set

8	9	10	11	12	13	14	15	Column	Row			
1 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1	b8 b7 b6 b5 b4 b3 b2 b1	Bits			
200 128 80	DCS 144 90	240 160 A0	°	260 176 80	Á	300 192 C0	à	320 208 D0	340 224 E0	360 240 F0	0 0 0 0	0
201 129 81	PU1 145 91	í	±	261 177 81	À	301 193 C1	á	321 209 D1	341 225 E1	361 241 F1	0 0 0 1	1
202 130 82	PU2 146 92	ç	²	262 178 82	Ã	302 194 C2	â	322 210 D2	342 226 E2	362 242 F2	0 0 1 0	2
203 131 83	STS 147 93	£	³	263 179 83	Ä	303 195 C3	ã	323 211 D3	343 227 E3	363 242 F3	0 0 1 1	3
204 132 84	IND CCH 148 94			264 180 84	Å	304 196 C4	ä	324 212 D4	344 228 E4	364 244 F4	0 1 0 0	4
205 133 85	NEL MW 149 95	¥	µ	265 181 85	Ä	305 197 C5	å	325 213 D5	345 229 E5	365 245 F5	0 1 0 1	5
206 134 86	SSA SPA 150 96		¶	266 182 86	Æ	306 198 C6	æ	326 214 D6	346 230 E6	366 246 F6	0 1 1 0	6
207 135 87	ESA EPA 151 97	§	·	267 183 87	Ç	307 199 C7	ç	327 215 D7	347 231 E7	367 247 F7	0 1 1 1	7
210 136 88	HTS	¸		270 184 88	È	310 200 C8	è	330 216 D8	350 232 E8	370 248 F8	1 0 0 0	8
211 137 89	HTS	©	¹	271 185 89	É	311 201 C9	é	331 217 D9	351 233 E9	371 249 F9	1 0 0 1	9
212 138 8A	VTS	ª	º	272 186 8A	Ê	312 202 CA	ê	332 218 DA	352 234 EA	372 250 FA	1 0 1 0	10
213 139 8B	PLD CSI 155 98	«	»	273 187 8B	Ë	313 203 CB	ë	333 219 DB	353 235 EB	373 251 FB	1 0 1 1	11
214 140 8C	PLU ST 156 9C	¼	¼	274 188 8C	Ì	314 204 CC	ì	334 220 DC	354 236 EC	374 252 FC	1 1 0 0	12
215 141 8D	RI OSC 157 9D	½	½	275 189 8D	Í	315 205 CD	í	335 221 DD	355 237 ED	375 253 FD	1 1 0 1	13
216 142 8E	SS2 PM 158 9E	¾	¾	276 190 8E	Î	316 206 CE	î	336 222 DE	356 238 EE	376 254 FE	1 1 1 0	14
217 143 8F	SS3 APC 159 9F	¿	¿	277 191 8F	Ï	317 207 CF	ï	337 223 DF	357 239 EF	377 255 FF	1 1 1 1	15

ZK-0348A-GE

The following example shows how to code an informal figure that appears within text without floating, without a number, and without a caption.

```

<P> A VT200 is different from a VT300 terminal.
The following figures illustrate an obvious difference in hardware between
the two types of terminals.

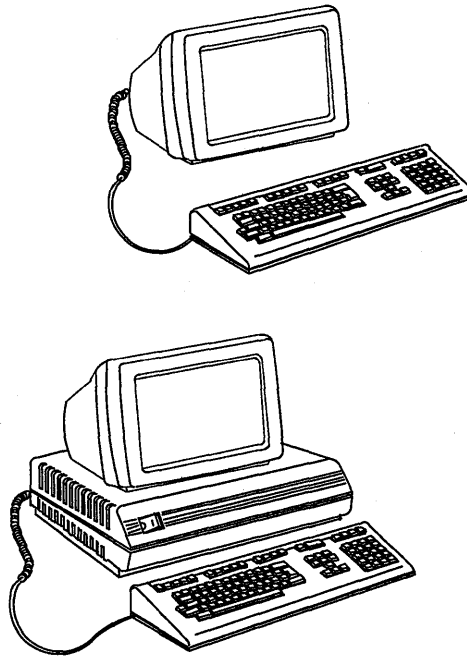
<FIGURE>
<FIGURE_FILE>(LN03\VT_Terminals.SIX\21)
<FIGURE_FILE>(PS\VT_Terminals.PS\21)
<FIGURE_FILE>(BOOKREADER\VT_Terminals.BRF\21)
<ENDFIGURE>
<P>
The first figure shows that ...

```

This example produces the following output:

<FIGURE>

A VT200 is different from a VT300 terminal. The following figures illustrate an obvious difference in hardware between the two types of terminals.



ZK-9570-GE

The first figure shows that . . .

---

## <FIGURE\_ATTRIBUTES>

Specifies the placement of a figure on the page.

---

### FORMAT

<FIGURE\_ATTRIBUTES> ( { *FLOAT*  
*KEEP*  
*MULTIPAGE* } [ \ *WIDE* ] )

---

### ARGUMENTS

#### **FLOAT**

This keyword specifies whether the location of a one-page figure is allowed to float if there is not enough room on the current page for the figure. *FLOAT* instructs the text processor to fill the current page with the text from the source file that follows the <FIGURE> tag sequence, and to place the figure at the top of the next page of output.

Float is the default for a figure that has a caption and does not specify *MULTIPAGE* or *KEEP*. *FLOAT* has no effect for Bookreader output.

#### **KEEP**

This keyword specifies that the figure be kept with the immediately preceding text. This is the default for a figure without a caption. *KEEP* has no effect for Bookreader output.

#### **MULTIPAGE**

This keyword specifies that the figure has multiple elements, and that each element is allowed to break at the start of a new page. This argument is required if the figure elements require more than one page. When a figure is continued, the figure caption and figure column heads, if any, are automatically repeated at the beginning of each new page of output. *MULTIPAGE* has no effect for Bookreader output.

#### **WIDE**

This is an optional keyword argument. It specifies that the width of the figure exceeds the document's default width for text. You can use *WIDE* with *FLOAT*, *KEEP*, or *MULTIPAGE*, or you can use *WIDE* as the only argument. Depending on the doctype, this argument is interpreted as follows:

- If the doctype contains a left margin area that is normally used for headings, the figure's width spans that area as well as the normal text area.
- If the document uses a multicolumn format, the figure suspends multicolumn output while the figure is processed. The figure is output across two columns and then multicolumn output is restored.
- Depending on the doctype, this argument provides a range of font sizes and font styles for figures.

**Note:** If you use a <FIGURE\_ATTRIBUTES> tag and specify the *WIDE* argument to that tag, which makes the caption align at the

## <FIGURE\_ATTRIBUTES>

leftmost position of the image area, you still must use the **WIDE** argument to the <FIGURE\_FILE> tag to align the figure itself at the leftmost position of the image area.

---

### related tags

- <FIGURE>
  - <FIGURE\_FILE>
  - <FIGURE\_SPACE>
- 

### restrictions

Valid only in the context of a <FIGURE> tag.

---

## DESCRIPTION

The <FIGURE\_ATTRIBUTES> tag specifies the placement of a figure on the page. Use the tag in the context of a **formal figure** to adjust the pagination and placement of the figure. See the <FIGURE> tag description for more information about formal and **informal figures**.

If the figure is formal and you need a wide caption that extends into the leftmost position of the image area, use the **WIDE** argument. You still must use the **WIDE** argument to the <FIGURE\_FILE> tag to align the figure itself at the leftmost position of the image area.

A formal figure is sometimes long. If you think one of your figures will be longer than a page, use the **MULTIPAGE** argument. Your figure will then be placed on the current page and continued on following pages, as space permits.

A multipage figure can contain one or more elements (that is, one or more <FIGURE\_FILE>, <FIGURE\_SPACE>, and <CODE\_EXAMPLE> tags).

If your multipage figure (consisting of a single <CODE\_EXAMPLE>) contains no blank lines, you might want to insert <VALID\_BREAK> tags in the figure. The <VALID\_BREAK> tags specify reasonable page breaking points. See the description of the <VALID\_BREAK> tag for more information.

A one-page (or smaller) formal figure's position in the output file can change. If the figure does not completely fit on the current page, the entire figure is placed at the top of the next page. The text that follows the example in the SDML file then is used to fill the current output page. When the figure's position can change, the figure is said to *float*. Formal figures that float, then, may appear out of place. Use the <FIGURE\_ATTRIBUTES>(KEEP) tag to force the figure to be kept with the text that immediately precedes it.

Floating is the default for a formal figure that fits on one page. If you want an informal figure to float, you must specify the **FLOAT** argument if your figure fits on one page. A multipage figure never floats.

---

## EXAMPLE

See the examples in the discussion of the <FIGURE> tag.

---

## <FIGURE\_FILE>

Includes a graphics file in your output file if the output device has graphics capability.

---

### FORMAT

<FIGURE\_FILE> (*target device* { \ *file spec* }  
                                  { \ *SPACE* }  
                                  { \ *vertical size* } [ \ *WIDE* ]  
                                  { \ *FULL\_PAGE* }

---

### ARGUMENTS

#### **target device**

Specifies a keyword indicating the output device for the graphics file. Keywords are provided both for devices that do (LN03, PS, and Bookreader) and do not (LINE, MAIL, and TERMINAL) support graphics. Each keyword allows you to insert the necessary amount of white space for the specific output device; keywords for devices that do not support graphics allow you to insert blank space in place of a figure. The following table lists the output devices and expected output for each keyword.

---

Keyword	Device	Output
BOOKREADER	DECwindows screen	The specified graphics file is output online using Bookreader.
LN03	LN03 Laser Printer	The specified sixel graphics file is output as a figure.
PS	PRINTSERVER 40 or LN03R SCRIPTPRINTER	The specified PostSCRIPT graphics file <sup>1</sup> is output as a figure.
LINE MAIL TERMINAL	Line Printer	If specified, blank space is output with the <i>file spec</i> argument written in that blank space. Use only one of these three keywords to indicate a monospaced destination.

---

<sup>1</sup>PostSCRIPT graphics files must conform to the Encapsulated PostSCRIPT File Format published by Adobe Systems, Inc.

---

If you specify the <FIGURE\_FILE> tag for a given device and subsequently process the file on another output device, no output will appear in the position of the <FIGURE\_FILE> tag.

#### **file spec**

Specifies the graphics file.

## <FIGURE\_FILE>

### **SPACE**

This keyword lets you reserve blank space in the output file for a figure. You can use it if you do not use the *file spec* argument. Use SPACE to reserve space for art that will be pasted in at a later date or when you expect to process a file for more than one output device, but do not have graphics files for all devices.

### **vertical size**

Specifies the vertical size of the printed graphic in picas (6 picas equal an inch, and 12 points equal a pica). This argument may be a nonnegative integer or decimal number, including zero.

### **FULL\_PAGE**

You can use FULL\_PAGE if you do not use the *vertical size* argument. It specifies that a full blank page be reserved for the figure. For Bookreader output, a small amount (smaller than a printed page or a Bookreader page) of blank space is left.

### **WIDE**

This is an optional keyword argument. If you use it, it must come last in the argument list. WIDE specifies that the graphic be aligned at the leftmost position of the image area (if your doctype uses a wide left gutter).

**Note:** If you use a <FIGURE\_FILE> tag and specify the WIDE argument to that tag, which makes the figure itself align at the leftmost position of the image area, you still must use the WIDE argument to the <FIGURE\_ATTRIBUTES> tag to align the caption at the leftmost position of the image area.

## related tags

- 
- <FIGURE>
  - <FIGURE\_ATTRIBUTES>
  - <ICON>
  - <ICON\_FILE>
  - <LINE\_ART>
  - <REFERENCE>
  - <SET\_FIGURE\_FILE\_SPACING\_DEFAULT>
- 

## restrictions

Valid only in the context of a <FIGURE> tag.

You must use an LN03 or POSTSCRIPT laser printer to print graphics files or use Bookreader to view graphics files online.

If you use the <FIGURE\_FILE> tag in a file that you process for multiple output devices, you must supply a tag for each device, using the SPACE argument for devices for which you do not have graphics.

---

## DESCRIPTION

The <FIGURE\_FILE> tag includes a graphics file in your output file if the output device has graphics capability.

To ensure that your figure is the right size for inclusion with your output, follow this procedure:

- 1 Create and print your graphics file full size (100 percent).
- 2 Measure it.
- 3 If the figure at 100 percent is the desired size and will fit on the page, incorporate that measurement into the <FIGURE\_FILE> tag. Otherwise, use the Graphics Editor to reduce the figure to the desired size.
- 4 Process the SDML file that the Graphics Editor produces and print the output. See the *VAX DOCUMENT Graphics Editor User's Guide* for information on using the Graphics Editor.
- 5 Evaluate the result. If it is not satisfactory, change either of the following:
  - One or both keywords specified in the <FIGURE\_FILE> tag to correct the alignment, spacing, or both, for the figure
  - The graphics file itself to correct the size, content, or both

Because there is an interaction between changing the contents or size of the graphics file and specifying its placement in the <FIGURE\_FILE> tag, satisfactory results may require several loops through the process, much like coding complicated tables. Refer to *VAX DOCUMENT Producing Online and Printed Documentation* for information on coding tables.

If you plan to process your file for more than a single destination, you can use multiple <FIGURE\_FILE> tags to ensure that the appropriate figure will be included for the appropriate destination; an example of this coding is given in the following section. If you use multiple <FIGURE\_FILE> tags in this way, you should specify only a single <FIGURE\_FILE> tag for all the monospaced destination keywords (MAIL, TERMINAL, or LINE).

---

## EXAMPLES

The following example shows how to code a captioned, numbered figure. The figure would appear in the table of contents.

```
1 <FIGURE>(The Front View)
  <FIGURE_FILE>(LN03\Frontpanel.SIX\6)
  <ENDFIGURE>
```

The following example shows how to code an uncaptioned, unnumbered figure. The figure would not appear in the table of contents.

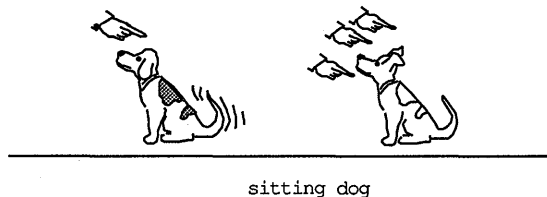
## <FIGURE\_FILE>

2 <FIGURE>  
<FIGURE\_FILE> (LN03\Example\_Figure.PS\10)  
<ENDFIGURE>

The following example shows how to use multiple <FIGURE\_FILE> tags so that your figure will process for multiple destinations. If you process this file for a POSTSCRIPT destination, the file included is DOG\_TRAINING.PS. If you process this file for an LN03 laser printer destination, the file included is DOG\_TRAINING.SIX. If you process this file for a line printer destination, no file is included; instead, 10 blank lines are reserved for the graphic. If you process this file for Bookreader, the file included is DOG\_TRAINING.BRF.

3 <FIGURE>  
<FIGURE\_FILE> (LN03\Dog\_Training.SIX\10)  
<FIGURE\_FILE> (PS\Dog\_Training.PS\10)  
<FIGURE\_FILE> (LINE\SPACE\10)  
<FIGURE\_FILE> (BOOKREADER\Dog\_Training.BRF\10)  
<ENDFIGURE>

This example produces the following output:



ZK-9571-GE



---

## <FIGURE\_SPACE>

Marks the space required for a figure that will be pasted in during final production.

---

### FORMAT

<FIGURE\_SPACE>( { *value*  
                  *FULL\_PAGE* } [*\ text*])

---

### ARGUMENTS

#### *value*

Specifies the amount of vertical space to be left on the page. Specify the value in picas, a scale used by typesetters. There are 6 picas in an inch (and 12 points in a pica). Thus, if the figure to be pasted in is 4 inches high, you should specify 24. If you do not specify a value, a default value of 2 is used.

#### *FULL\_PAGE*

This keyword specifies that a full blank page be reserved for the figure if you do not use the *value* argument. For Bookreader output, a small amount (smaller than a printed page or a Bookreader page) of blank space is left.

#### *text*

This is an optional argument. It specifies the text that describes the status of the figure, an art file number, or the words "To Be Set." The text is output in the middle of the space left for the figure.

---

### related tags

- <FIGURE>
- <FIGURE\_ATTRIBUTES>
- <FIGURE\_FILE>
- <FIGURE\_SPACE>
- <LINE\_ART>
- <REFERENCE>

---

### restrictions

Valid only in the context of a <FIGURE> tag.

The *value* argument must not exceed page length limitations.

---

### DESCRIPTION

The <FIGURE\_SPACE> tag marks the space required for a figure that will be pasted in during final production.

If you specify some descriptive text in the second argument, that text is output in the middle of the space left for the figure.

## <FIGURE\_SPACE>

---

### EXAMPLES

The following example shows how to leave space for a figure that is 18 picas long. Notice that 18 picas (or 3 inches or 216 points) of space are left for the figure.

**1** <FIGURE>(Illustration of a Computer Terminal\terminal\_fig)  
<FIGURE\_SPACE>(18\An illustration of a computer terminal.)  
<ENDFIGURE>

This example produces the following output:

**Figure n-n Illustration of a Computer Terminal**

---

An illustration of a computer terminal.

---

The following example shows how to leave space for a figure that will take one page.

**2** <FIGURE>(Illustration of a Garage\garage\_fig)  
<FIGURE\_SPACE>(FULL\_PAGE\An illustration of a garage.)  
<ENDFIGURE>

This example produces output like that found on the following page.

Figure n-n Illustration of a Garage

---

An illustration of a garage.

---

DECLIT AA VAX JT84C  
VAX DOCUMENT using global  
tags

## <FILE\_SPEC>

---

## <FILE\_SPEC>

Allows you to use a file specification that contains angle brackets as an argument to a tag without VAX DOCUMENT interpreting that file specification as a tag.

---

**FORMAT**            <FILE\_SPEC> (*file specification*)

---

**ARGUMENTS**        *file specification*  
Specifies the file specification that contains angle brackets.

---

**related tags**

- <ELEMENT>
- <EXAMPLE\_FILE>
- <FIGURE\_FILE>
- <ICON\_FILE>
- <INCLUDE>
- <INCLUDES\_FILE>
- <TABLE\_FILE>

---

**DESCRIPTION**     The <FILE\_SPEC> tag allows you to use a file specification that contains angle brackets as an argument to a tag that lets you include external files without VAX DOCUMENT interpreting that file specification as a tag. A complete list of tags that let you include external files is found in the “Related Tags” section of this tag description.

Some keyboards do not provide square brackets, ([) and (]), so it is common for the directory portion of a file specification to be delimited by angle brackets. Because VAX DOCUMENT would normally interpret text within angle brackets as an SDML tag, the <FILE\_SPEC> tag converts any angle brackets in its argument into the corresponding square brackets.

You do not have to use the <FILE\_SPEC> tag to specify logical names in a file specification whose equivalence strings contain angle brackets.

---

**EXAMPLE**            The following example shows how to use the <FILE\_SPEC> tag.

```
<figure_file>(ln03<file_spec>(mydisk:<figure>diagram.six)\20)
```

## <FILE\_SPEC>

The directory name in this file specification would normally be seen as a <FIGURE> tag, which would lead to erroneous results. Because the file specification is used as an argument to the <FILE\_SPEC> tag, the angle brackets are converted to square brackets before the apparent <FIGURE> tag is recognized. Thus, the example is equivalent to the following:

```
<figure_file>(ln03\mydisk:[figure]diagram.six\20)
```

## <FINAL\_CLEANUP>

---

## <FINAL\_CLEANUP>

Provides explicit formatting instructions to the text formatter to be used for final formatting and clean up.

---

### FORMAT

**<FINAL\_CLEANUP>** ( { *COLUMN\_BREAK*  
*LINE\_BREAK*  
*PAGE\_BREAK*  
*SPECIAL\_BREAK* } )

---

### ARGUMENTS

#### ***COLUMN\_BREAK***

Specifies that a new column of text be started at the place where the <FINAL\_CLEANUP> tag occurs. This argument is only valid in a two-column doctype: the ARTICLE or REPORT.TWOCOL doctypes, described in *VAX DOCUMENT Using Doctypes and Related Tags*.

#### ***LINE\_BREAK***

Specifies that the following text begin a new line of output.

#### ***PAGE\_BREAK***

Specifies that the following text begin a new page of output.

#### ***SPECIAL\_BREAK***

Specifies a special break when vertical spacing appears to be lost. In some circumstances, the output of a two-column page may have had some of its vertical spacing lost due to the text formatter processing; for example, a heading tag may not have any space before it. You can correct this by placing the <FINAL\_CLEANUP>(SPECIAL\_BREAK) tag in your file between the words that are output on the final line of the first column.

Use the SPECIAL\_BREAK argument only if there is a spacing problem and only after you are ready to give your document a final format check, because changes to the source file may help the text formatter resolve the spacing problem.

---

### related tags

- <COLUMN> — In the ARTICLE and REPORT.TWOCOL doctypes, described in *VAX DOCUMENT Using Doctypes and Related Tags*.
- <LINE>
- <PAGE>

---

### restrictions

Should be used only for final production formatting.

If you use the <FINAL\_CLEANUP>(PAGE\_BREAK) tag in a table to force a page break, the table headings are not repeated on the page after the break. You must repeat the <TABLE\_HEADS> tag immediately after the <FINAL\_CLEANUP>(PAGE\_BREAK) tag to restore the default table headings. Headings

are repeated on each page of a table only when DOCUMENT is allowed to break the table across pages.

In doctypes with **right-justified** text (that is, text aligned against the right margin), the <FINAL\_CLEANUP>(LINE\_BREAK) tag must be on the same line of the text you want broken.

---

## DESCRIPTION

The <FINAL\_CLEANUP> tag provides explicit formatting instructions to the text formatter to be used for final formatting and clean up. The <FINAL\_CLEANUP> tag is not a generic markup tag; it explicitly instructs the text formatter to change aspects of the page makeup.

In a single-column doctype, the <FINAL\_CLEANUP> tag enables you to specify line breaks and page breaks using the LINE\_BREAK and PAGE\_BREAK arguments. In a two-column doctype, you can use the COLUMN\_BREAK argument to start a new column of text. Use the SPECIAL\_BREAK argument only rarely at the final stage of production to affect the final format of your document.

When you are working with the ARTICLE or REPORT.TWOCOL doctypes, you may need to make some final adjustments when your text is complete. The text formatter makes formatting decisions based on your source file. It is more difficult for the text formatter to create a well-formatted two-column page than a well-formatted one-column page. Therefore, the output of these doctypes may need additional attention, using the <FINAL\_CLEANUP> tag, during the last stage of production.

Using the <FINAL\_CLEANUP> tag has no effect for Bookreader output.

---

## EXAMPLES

The following example shows how final production might use the <FINAL\_CLEANUP>(LINE\_BREAK) tag in the front matter of a book to make it look better.

```
❶ <FRONT_MATTER>(front)
    .
    .
    .
    <REVISION_INFO>(Revision/Update Information:\This revised manual
    supersedes
    the <final_cleanup>(line_break)<emphasis>(VAX DOCUMENT User Manual, Volume 1 Version 1.1)
    <oparen>Order Number AA--JT84B--TE<cparen>.)
    .
    .
    .
```

## <FINAL\_CLEANUP>

The following example shows how final production might use the <FINAL\_CLEANUP>(PAGE\_BREAK) tag, because the information in these specific headings might look better or fit better on separate pages.

```
2 <HEAD1>(Survey Results)
  .
  .
  .
  <FINAL_CLEANUP>(page_break)
  <HEAD2>(Brand X)
  .
  .
  .
```



---

## <FOOTNOTE>

Creates a footnote in text and tables. In text, this tag specifies the footnote character and footnote text, and places the footnote text at the bottom of the page or in a **pop-up window** for Bookreader output. In a table, this tag specifies the footnote character and footnote text; the <FOOTREF> tag outputs the footnote character. Footnotes in tables appear at the bottom of the table.

---

**FORMAT**            <FOOTNOTE>(char \ *footnote text*)

---

### ARGUMENTS

#### ***char***

Specifies the footnote character. The character can be a single character, a number, or one of the following keywords denoting special characters associated with footnote references:

DAG (†)  
DDAG (‡)  
R (®)  
S (§)  
TM (™)

When placing a footnote in a table, you must use a number or one of the keywords as the footnote character.

If you use more than one of the keywords allowed for table footnotes, you should declare them in the following order to ensure that the footnotes print in the correct order at the bottom of the page or in the pop-up window for Bookreader output:

- 1 TM
- 2 R
- 3 S
- 4 DAG
- 5 DDAG

#### ***footnote text***

Specifies the text of the footnote.

---

### related tags

- <FOOTNOTE\_TEXT>
- <FOOTREF>

---

### restrictions

Invalid in the context of a <MATH> tag.

## <FOOTNOTE>

The following restrictions apply to footnotes in text:

- Do not place the following tags within an argument to the <FOOTNOTE> tag: <CODE\_EXAMPLE>, <EXAMPLE>, <FIGURE>, <HEADn>, <MATH>, <NOTE>, or <TABLE>.
- Do not place more than four footnotes within a stacked list or a monospaced example.

The following restrictions apply to footnotes in a table:

- Use the <FOOTNOTE> and <FOOTREF> tag combination in a table. It is the <FOOTREF> tag that causes the footnote character to appear. Place the <FOOTREF> tag in the argument to the <TABLE\_ROW> tag at the end of the argument you want to footnote.
- Declare all table footnotes at the top of the table using a <FOOTNOTE> tag just after the <TABLE\_SETUP> tag. Do not specify <FOOTNOTE> tags in a nested table.
- Do not specify more than twelve footnotes in a table. If you use a nonnumeric footnote character, there can be no more than seven numeric footnotes.
- Footnotes in the body of a multipage table appear at the bottom of a page only if they are called out on that page.
- Footnotes called out in the heading of a multipage table appear at the bottom of each page of the table.

---

## DESCRIPTION

The <FOOTNOTE> tag creates a footnote in text and tables. In text, this tag specifies the footnote character, which appears in the text at the place where the tag is located, and footnote text. The footnote text appears at the bottom of the page or in a **pop-up** window for Bookreader output (once you click on the superscript footnote reference, which is a **hotspot**). In a table, this tag specifies the footnote character and footnote text; the <FOOTREF> tag outputs the footnote character. Footnotes in tables appear at the bottom of the table for both printed and Bookreader output.

For footnotes in a heading, on a copyright page, or on a title page, use the <FOOTNOTE\_TEXT> and <FOOTREF> tag combination. See the <FOOTNOTE\_TEXT> and <FOOTREF> tag descriptions for more information.

---

## EXAMPLES

The following example shows how to place a footnote in text.

**1** <P>The <TAG>(footnote) tag produces output that looks like this.<FOOTNOTE>(1\Note how footnote text appears at the bottom of the page or in a pop-up window for Bookreader output.)

This example produces the following output:

The <FOOTNOTE> tag produces output that looks like this.<sup>1</sup>

The following example shows how to use a footnote in a table.

---

<sup>1</sup> Note how footnote text appears at the bottom of the page or in a pop-up window for Bookreader output.

```

2 <TABLE>(Rules for Determining Expression Modes\express_modes_tab)
  <TABLE_ATTRIBUTES>(MULTIPAGE)
  <TABLE_SETUP>(2\43)
  <FOOTNOTE>(1\A footnote in a table.)
  <TABLE_HEADS>(Expression\Value Type)
  <TABLE_ROW>(Integer value\Integer)
  <TABLE_ROW>(String value\String<FOOTREF>(1))
  <TABLE_ROW>(Integer lexical function\Integer)
  <TABLE_ROW>(String lexical function\String)
  <TABLE_ROW>(Integer symbol<FOOTREF>(1)\Integer)
  <TABLE_ROW>(String symbol\String)
  <TABLE_ROW>(Any value .AND. or .OR. any value\Integer)
  <TABLE_ROW>(Any value\Integer)
  <TABLE_ROW>(Any value\Integer)
<ENDTABLE>

```

This example produces the following output:

**Table n.n Rules for Determining Expression Modes**

Expression	Value Type
Integer value	Integer
String value	String <sup>1</sup>
Integer lexical function	Integer
String lexical function	String
Integer symbol <sup>1</sup>	Integer
String symbol	String
Any value .AND. or .OR. any value	Integer
Any value	Integer
Any value	Integer

<sup>1</sup>A footnote in a table.

The following example shows how to use the <FOOTNOTE> and <FOOTREF> tags in text to produce multiple references to the same footnote.

```

3 <P>The macro format for a $GETDVI request is:<FOOTNOTE>(1\The
  eighth (last)
  argument is not used; it is reserved for future use.)
  <CODE_EXAMPLE>
  $GETDVI [efn],[chan],[devnam],itmlst,[iosb],[astadr],[astprm]
  <ENDCODE_EXAMPLE>
  <P>The high-level language format for a $GETDVI request is:<FOOTREF>(1)
  <CODE_EXAMPLE>
  SYS$GETDVI([efn],[chan],[devnam],itmlst<FOOTREF>(1),[iosb],[astadr],[astprm])
  <ENDCODE_EXAMPLE>

```

This example produces the following output:

The macro format for a \$GETDVI request is:<sup>1</sup>

\$GETDVI [efn],[chan],[devnam],itmlst,[iosb],[astadr],[astprm]

The high-level language format for a \$GETDVI request is:<sup>1</sup>

SYS\$GETDVI([efn],[chan],[devnam],itmlst<sup>1</sup>,[iosb],[astadr],[astprm])

<sup>1</sup> The eighth (last) argument is not used; it is reserved for future use.

## <FOOTNOTE\_TEXT>

---

## <FOOTNOTE\_TEXT>

Specifies the text of a footnote in a heading, copyright page, or title page.

---

**FORMAT**            <FOOTNOTE\_TEXT>(char | text)

---

**ARGUMENTS**        *char*  
Specifies the footnote character. The character can be a single character or number, or it can be one of the following keywords denoting special characters associated with footnote references:

DAG (†)  
DDAG (‡)  
R (®)  
S (§)  
TM (™)

*text*  
Specifies the text associated with the footnote.

---

**related tags**

- <FOOTNOTE>
- <FOOTREF>
- <SPECIAL\_CHAR>

---

**restrictions**

Do not use the <FOOTNOTE\_TEXT> tag for table footnotes; use the <FOOTNOTE> tag.

Must follow the related <FOOTREF> tag in the source file.

You must use the <SPECIAL\_CHAR> tag to output a special character that you want to appear in the *text* argument to the <FOOTNOTE\_TEXT> tag.

---

**DESCRIPTION**     The <FOOTNOTE\_TEXT> tag specifies the text of a footnote in a heading, copyright page, or title page. The footnote text appears at the bottom of the page or in a **pop-up window** for Bookreader output. The <FOOTNOTE\_TEXT> tag must follow the <FOOTREF> tag.

Use the <FOOTREF> tag to output a footnote in the argument to a <HEADn> tag and the <COMMAND\_SECTION\_HEAD> and <SUBCOMMAND\_SECTION\_HEAD> tags of the SOFTWARE doctype, and to output footnotes on a copyright page or a title page. In all other cases, you must use the <FOOTNOTE> tag.

---

## EXAMPLES

The following example shows how you might use a footnote on the title page of a document.

**1** <FRONT\_MATTER>  
<TITLE\_PAGE>  
<TITLE>(Guide to Growing Premium Fruit Bearing Plants)  
  
<ABSTRACT>(July 1990)  
This document describes how to cultivate LEMON<FOOTREF>(TM) trees.  
<ENDABSTRACT>  
  
<FOOTNOTE\_TEXT>(TM\LEMON is a trademark of Lemon Growers, Inc.)  
<ENDTITLE\_PAGE>

The following example shows how to use the <FOOTNOTE\_TEXT> and <FOOTREF> tags in a heading.

**2** <HEAD1>(Introduction to The News Today<footref>(TM))  
<footnote\_text>(TM\The News Today  
is a trademark of the American Television  
Society.)

This example produces the following output:

n.n

---

## Introduction to The News Today™

The following example shows how to use multiple arguments to the <FOOTREF> tag and how to use the <SPECIAL\_CHAR> tag in footnote text.

**3** <HEAD1>(Using PostScript<FOOTREF>(1\2) Printers\ps\_printers)  
<FOOTNOTE\_TEXT>(1\PostScript <SPECIAL\_CHAR>(TRADEMARK\_SYMBOL) is a trademark  
of Adobe Systems, Inc.)  
<FOOTNOTE\_TEXT>(2\For more information about Digital's printers . . . )

This example produces the following output:

n.n

---

## Using PostScript<sup>1,2</sup> Printers

---

™ The News Today is a trademark of the American Television Society.

<sup>1</sup> PostScript™ is a trademark of Adobe Systems, Inc.

<sup>2</sup> For more information about Digital's printers . . .

## <FOOTREF>

---

## <FOOTREF>

Creates a footnote character in a heading, copyright page, or title page. Also used to specify a previously defined footnote in text more than once, and along with the <FOOTNOTE> tag, to produce a footnote in a table.

---

**FORMAT**            <FOOTREF>(char-1 [\ char-2...[\ char-9]])

---

**ARGUMENTS**        *char-1...9*

Specifies up to nine characters for a footnote. A character can be a single character or number, or it can be one of the following keywords denoting special characters associated with footnote references:

DAG (†)  
DDAG (‡)  
R (®)  
S (§)  
TM (™)

In a table, the reference must be to a number or to one of the special characters.

---

**related tags**

- <FOOTNOTE>
- <FOOTNOTE\_TEXT>

---

**restrictions**

Invalid in the context of a <MATH> tag.

You can use only one <FOOTREF> tag per heading or your Bookreader bookbuild will fail.

If you want to generate a special character in the context of footnotes, use the <FOOTREF> tag. In other contexts, use the <SPECIAL\_CHAR> tag, or you may encounter Bookreader bookbuilding problems.

When using this tag in an argument list, you must place the tag at the end of the argument you want to footnote; you cannot embed it in text. The end of an argument is indicated by a backslash (\) and the end of an argument list is indicated by a closing parenthesis ()).

---

**DESCRIPTION**

The <FOOTREF> tag creates a footnote character in a heading, copyright page, or title page. It is also used to specify a previously defined footnote in text more than once, and along with the <FOOTNOTE> tag, to produce a footnote in a table.

The <FOOTREF> tag causes as many as nine superscripted characters to appear where you place the tag in your file.

You must use the <FOOTREF> tag:

- To specify footnotes in the argument to a <HEADn> tag and <COMMAND\_SECTION\_HEAD> and <SUBCOMMAND\_SECTION\_HEAD> tags of the SOFTWARE doctype
- To specify footnotes on a copyright page and title page
- To specify the same footnote in text more than once
- To specify a footnote in a table

To footnote information in a heading, use a combination of two tags: <FOOTREF> and <FOOTNOTE\_TEXT>. Place the <FOOTREF> tag in the argument to the heading to refer to the <FOOTNOTE\_TEXT> tag. Place the <FOOTNOTE\_TEXT> tag on the line directly following the <HEADn> tag.

For Bookreader output, you can use only one <FOOTREF> tag per heading. Also, the character or number that the <FOOTREF> tag produces is a **hotspot**.

To use a footnote in a table, use a combination of tags: <FOOTNOTE> and <FOOTREF>. Define the <FOOTNOTE> tag just after the <TABLE\_SETUP> tag. Then use the <FOOTREF> tag to label the footnote references within the argument to a <TABLE\_ROW> tag; do not define the footnote within a table row. The <FOOTREF> tag causes the text of the footnote to appear at the bottom of the table for both printed and Bookreader output. For a printed document, if the table is longer than a single page, and if the <FOOTREF> tag appears multiple times, the footnote text appears at the bottom of the table on each page containing the reference.

If you have a footnote in text that you want to refer to more than once, use a <FOOTNOTE> tag directly after the text to label the first occurrence of the footnote and the <FOOTREF> tag to label the subsequent references to that note. The <FOOTREF> tag always refers to the most recent occurrence of the <FOOTNOTE> tag that uses the same argument number or character.

**Note: The text formatter does not automatically repeat text footnotes, except for multipage tables, if references are output on more than one page. You must reenter the footnote for it to appear on another page.**

---

## EXAMPLES

See the examples in the discussion of the <FOOTNOTE> and <FOOTNOTE\_TEXT> tags.

## <FORMAT>

---

## <FORMAT>

Enables the <FCMD>, <FPARM>, and <FPARMS> tags to distinguish formatted command keywords and parameters.

---

### FORMAT

**<FORMAT>**[( { *heading text* [ \ *WIDE* ] } )]  
*WIDE*

---

### ARGUMENTS

#### ***heading text***

This is an optional argument. It specifies a heading.

#### ***WIDE***

This is an optional keyword argument. It specifies, for doctypes in which the left margin is indented, that the body of the formatted text be extended into the margin.

---

### related tags

- <FCMD>
- <FPARM>
- <FPARMS>

---

### restrictions

Invalid in the context of <EXAMPLE>, <FIGURE>, <MATH>, and <TABLE> tags.

---

### required terminator

<ENDFORMAT>

---

### DESCRIPTION

The <FORMAT> tag enables the <FCMD>, <FPARM>, and <FPARMS> tags to distinguish formatted command keywords and parameters.

---

### EXAMPLE

See the examples in the discussion of the <FCMD> tag.



---

## <FPARM>

Specifies a parameter to be formatted following the <FPARMS> tag, aligned under the parameters portion of a keyword/parameters pair.

---

**FORMAT**            <FPARM> (*parameter*)

---

**ARGUMENTS**        *parameters*  
Lists additional command parameters, if any.

---

**related tags**

- <FCMD>
- <FORMAT>
- <FPARMS>

---

**restrictions**        Valid only in the context of a <FORMAT> tag.

---

**DESCRIPTION**        The <FPARM> tag specifies a parameter to be formatted following the <FPARMS> tag, aligned under the parameters portion of a keyword/parameters pair.

---

**EXAMPLE**            See the examples in the discussion of the <FCMD> tag.

## <FPARMS>

---

## <FPARMS>

Specifies the parameter portion of a command/parameter pair in a format section.

---

**FORMAT**      <FPARMS> (*parameters*)

---

**ARGUMENTS**      *parameters*  
Lists the command parameters, if any. If there are no parameters, you must specify the argument as null: <FPARMS> ().

---

**related tags**

- <FCMD>
- <FORMAT>
- <FPARM>

---

**restrictions**      Valid only in the context of a <FORMAT> tag.

---

**DESCRIPTION**      The <FPARMS> tag specifies the parameter portion of a formatted command/parameter pair in a format section.

---

**EXAMPLE**      See the examples in the discussion of the <FCMD> tag.

---

---

## <FRONT\_MATTER>

Begins the front matter of a book.

---

**FORMAT**      <FRONT\_MATTER>[(*symbol name*)]

---

**ARGUMENTS**      *symbol name*

This is an optional argument. It specifies the term that you assign to the front matter. A symbol name argument is required if the front matter is part of a bookbuild.

Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

**related tags**

- See the list of tags in the description section.

---

**required terminator**

<ENDFRONT\_MATTER>

---

## DESCRIPTION

The <FRONT\_MATTER> tag begins the front matter of a book. You must use front matter to create a book for Bookreader. The following tags are used to create the front matter:

- <ABSTRACT>
- <CONTENTS\_FILE>
- <COPYRIGHT\_DATE>
- <COPYRIGHT\_PAGE>
- <FRONT\_MATTER>
- <ORDER\_NUMBER>
- <PREFACE>
- <PREFACE\_SECTION>
- <PRINT\_DATE>
- <REVISION\_INFO>
- <TITLE>
- <TITLE\_PAGE>

## <FRONT\_MATTER>

---

### EXAMPLE

The following example shows the order of all the front matter tags and how you can use them in a file that contains the front matter of a book.

```
<FRONT_MATTER>(front)
<TITLE_PAGE>
<DEFINE_BOOK_NAME>(using_global_tags\Using Global Tags)
<TITLE>(<REFERENCE>(using_global_tags))
<ORDER_NUMBER>(xx-12345)
<ABSTRACT>
This book describes all the global tags.
<ENDABSTRACT>
<REVISION_INFO>(Revision/Update Information:\This is a new manual.)
<ENDTITLE_PAGE>
<COPYRIGHT_PAGE>
<PRINT_DATE>(March 1987)
<COPYRIGHT_DATE>(1987)
<ENDCOPYRIGHT_PAGE>
<CONTENTS_FILE>
<PREFACE>(11)
<PREFACE_SECTION>(The changes to your system)
<ENDPREFACE>
<ENDFRONT_MATTER>
```

---

## <GDEF>

Labels the text that defines a term in a glossary or anywhere in a document.

---

**FORMAT**      <GDEF>

---

**ARGUMENTS**      *None.*

---

**related tags**

- <GLOSSARY>
- <GREF>
- <GTERM>

---

**DESCRIPTION**      The <GDEF> tag labels the text that defines a term in a glossary or anywhere in a document. Use the <GDEF> tag with the <GTERM> tag, which labels the term that you want to define. You have the options of using the <GLOSSARY> and <ENDGLOSSARY> tags to establish the beginning and ending format of a glossary, or simply using the <GTERM> and <GDEF> tags to create glossary entries anywhere in your document.

---

**EXAMPLE**      See the example in the discussion of the <GLOSSARY> tag.

## <GLOSSARY>

---

## <GLOSSARY>

Formats a glossary of terms in a document or book.

---

**FORMAT**      **<GLOSSARY>**[(*text* [\ *symbol name*])]

---

**ARGUMENTS**    *text*

This is an optional argument. It specifies any text string you want to label the start of the glossary. If you do not specify any text, the default is "Glossary."

*symbol name*

This is an optional argument. It specifies the term that you use to reference the glossary.

Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

**related tags**

- <GDEF>
- <GTERM>

---

**restrictions**

The symbol name argument is required if you include this file in a bookbuild.

---

**required terminator**

<ENDGLOSSARY>

---

**DESCRIPTION**

The <GLOSSARY> tag formats a glossary of terms in a document or book. Follow this with a list of <GTERM> tags, with the glossary terms defined as arguments to the <GTERM> tags. You must follow the term with the definition, which you label with a <GDEF> tag. For a printed book, the glossary appears in the table of contents with the beginning page number.

If you are creating a book for Bookreader, because the entire glossary is an **online topic**, you have the option of displaying in the table of contents the beginning and ending glossary term that comprises each online topic of information. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information about setting the online topic level for a glossary.

---

## EXAMPLES

The following example shows how to use the <GLOSSARY> tags to form a separate glossary.

```
1 <GLOSSARY\gloss_chap>
  <GTERM>(habitat)
  <GDEF>An area or natural environment.
  <GTERM>(habitual)
  <GDEF>Acting according to habit.
  <GTERM>(hack)
  <GDEF>A worn-out horse.
  <GTERM>(hackneyed)
  <GDEF>Trite, banal.
  <ENDGLOSSARY>
```

This example produces the following output:

---

## Glossary

**habitat:** An area or natural environment.

**habitual:** Acting according to habit.

**hack:** A worn-out horse.

**hackneyed:** Trite, banal.

The following example shows how to code glossary entries that you want anywhere in your document, rather than in a separate glossary.

```
2 <GTERM>(habitat)
  <GDEF>(An area or natural environment.)
  <GTERM>(habitual)
  <GDEF>(Acting according to habit.)
```

This example produces the following output:

**habitat:** An area or natural environment.

**habitual:** Acting according to habit.

## <GREF>

---

## <GREF>

Marks a cross-reference to a term within a glossary.

---

**FORMAT**      **<GREF>**(*glossary term*)

---

**ARGUMENTS**      *glossary term*  
Specifies the term referred to.

---

### related tags

- <GDEF>
  - <GLOSSARY>
  - <GTERM>
  - The following tags label other types of cross-references:
    - <REFERENCE>
    - <CALLOUT\_REF>
- 

**restrictions**      Invalid in the context of a <MATH> tag.

---

**DESCRIPTION**      The <GREF> tag marks a cross-reference from anywhere in your document, including the glossary, to a term within a glossary. The *glossary term* argument appears in an italic typeface in some doctypes.

---

**EXAMPLE**      The following example shows how to use the <GREF> tag to refer to the glossary term “cat”.

```
<GTERM>(cat)
<GDEF>(A carnivorous mammal.)
<GTERM>(cow)
<GDEF>(The mature female of cattle.)
<GTERM>(lion)
<GDEF>(A large <GREF>(cat).)
```

This example produces the following output:

**cat:** A carnivorous mammal.

**cow:** The mature female of cattle.

**lion:** A large *cat*.



---

## <GTERM>

Labels a term you want to define in a glossary or anywhere in a document.

---

**FORMAT**      <GTERM>(term)

---

**ARGUMENTS**      *term*  
Specifies the term you want to define in the glossary or anywhere in a document.

---

**related tags**

- <GDEF>
- <GLOSSARY>
- <GREF>

---

**required terminator**      <GDEF>

---

**DESCRIPTION**      The <GTERM> tag labels a term you want to define in a glossary or anywhere in a document. You must follow the <GTERM> tag with a <GDEF> tag, which labels the text that defines the term. To create glossary entries, use the <GTERM> and <GDEF> tags in the context of <GLOSSARY> and <ENDGLOSSARY> tags.

---

**EXAMPLE**      See the example in the discussion of the <GLOSSARY> tag.

## <HEAD>

---

## <HEAD>

Creates an unnumbered heading on the left side of the page.

---

**FORMAT**      **<HEAD>** (*heading text* [ \ *symbol name*])

---

**ARGUMENTS**    ***heading text***  
Specifies the text of the heading.

***symbol name***

This is an optional argument. It specifies the name of the symbol used in all references to this heading. Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

**related tags**

- <CHEAD>
  - <CENTER\_LINE>
  - <HEADn>
  - <SUBHEADn>
- 

**DESCRIPTION**    The <HEAD> tag creates an unnumbered heading on the left side of the page. The heading does not appear in the table of contents.

---

**EXAMPLE**        The following example shows how to use the <HEAD> tag.

```
<HEAD>(How to Write a Letter\how_to)
<P>
```

The first thing that you should do in writing a letter is to get a clean piece of paper and a well-sharpened pencil.

This example produces the following output:

**How to Write a Letter**

The first thing that you should do in writing a letter is to get a clean piece of paper and a well-sharpened pencil.

---

## <HEADn>

Marks a heading of the level you specify (<HEAD1> through <HEAD6> in all *doctypes/bold* except MILSPEC; in MILSPEC, <HEAD1> through <HEAD20>).

---

### FORMAT

<HEADn>(heading text [ \ symbol name])

---

### ARGUMENTS

#### **heading text**

Specifies the text of the heading. If the doctype you are using produces headings that are all capital letters in your output, those letters will appear that way regardless of how you enter them in your input file. You should, however, use uppercase and lowercases letters according to your local conventions in order to obtain the proper capitalization of the heading in the table of contents and in cross-references.

#### **symbol name**

This is an optional argument. It specifies the name of the symbol used in all references to this heading. Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

### related tags

- <CHEAD>
- <HEAD>
- <SUBHEAD1>
- <SUBHEAD2>

---

### restrictions

Invalid in the context of a <FOOTNOTE> tag.

You must use the *symbol name* argument when processing a heading for Bookreader display.

---

### DESCRIPTION

The <HEADn> tag marks a heading of the level you specify (<HEAD1> through <HEAD6> in all *doctypes/bold* except MILSPEC; in MILSPEC, <HEAD1> through <HEAD20>). Each of the six heading tags does the following:

- Outputs the heading text supplied by its first argument
- Automatically numbers the heading
- Resets all the counters for lower heading levels (if any)
- Specifies the symbol name with which cross-references to that heading are made

Entries for each of the headings may appear in the table of contents, depending on the doctype.

## <HEADn>

The proper choice of the heading level depends on an understanding of the logical structure of the document you are writing. A <HEAD2> tag will always be logically subordinate to a <HEAD1> tag. The same is true for the relationship between <HEAD3> and <HEAD2> tags, <HEAD4> and <HEAD3> tags, and so on.

**Note:** If you use the <HEADn> tag in a preface, the headings are output without numbers. You can refer to the headings from other parts of your document. The headings, however, do not appear in the table of contents.

DOCUMENT considers each first-level heading to be an **online topic**. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information about online topics.

---

### EXAMPLES

The following <HEAD1> tag labels a first-level heading.

**1**    <HEAD1>(Running Tasks\tasks)  
      <P>This section explains how the running tasks ...

The following <HEAD2> tag labels a second-level heading.

**2**    <HEAD2>(More Running Tasks\more\_tasks)  
      <P>This section explains more specifically how ...

The following <HEAD3> tag labels a third-level heading.

**3**    <HEAD3>(Final Running Tasks\final\_tasks)  
      <P>This section explains the final running ...

Examples 1, 2, and 3, shown together for comparison, produce the following output:

---

## 1.1 Running Tasks

This section explains how the running tasks ...

---

### 1.2.1 Specific Running Tasks

This section explains more specifically how ...

---

#### 1.2.1.1 Final Running Tasks

This section explains the final running ...

You could make a cross-reference to the second heading, for example, by using the `<REFERENCE>(more_tasks)` tag. Because that heading is the second level of the first section in the document, the reference would be output as Section 1.2.1.

To control the output of the reference, use one of the formats listed in the following table:

<b>Reference</b>	<b>Output</b>
<code>&lt;REFERENCE&gt;(tasks)</code>	Section 1.2
<code>&lt;REFERENCE&gt;(tasks\value)</code>	1.2
<code>&lt;REFERENCE&gt;(tasks\text)</code>	Running Tasks
<code>&lt;REFERENCE&gt;(tasks\full)</code>	Section 1.2, Running Tasks

## <HELLIPSIS>

---

## <HELLIPSIS>

Places horizontal ellipsis points on a line.

---

**FORMAT**      **<HELLIPSIS>**

---

**ARGUMENTS**      *None.*

---

**related tags**      • `<ELLIPSIS>`

---

**restrictions**      Invalid in the context of a `<MATH>` tag.

---

**DESCRIPTION**      The `<HELLIPSIS>` tag places horizontal ellipsis points on a line. Often, it is used to label omitted material.

---

**EXAMPLE**      The following example shows how to use the `<HELLIPSIS>` tag.

`<P>A horizontal ellipsis may provide an indefinite ending <HELLIPSIS>`

This example produces the following output:

A horizontal ellipsis may provide an indefinite ending . . .

---

## <HYPHENATE>

Provides information about legal hyphenation of a word.

---

**FORMAT**      <HYPHENATE>(part1 \ part2[ \ part3...[ \ part9]])

---

**ARGUMENTS**    *part1...part9*  
Specifies the word of text and up to nine valid hyphenation points. Each argument to the tag specifies a portion of the word. Use the backslash to indicate the hyphenation points.

---

**related tags**

- <FINAL\_CLEANUP>(LINE\_BREAK)
- <KEEP>
- <LINE>

---

**restrictions**    Invalid in the context of a <MATH> and an <EXAMPLE> tag.

---

**DESCRIPTION**    The <HYPHENATE> tag provides information about legal hyphenation of a word. This tag does not force a term to be hyphenated; it merely provides the text formatter with information about legal places to hyphenate the word.

Use the <HYPHENATE> tag when you are not satisfied with line breaks within paragraphs in your final output, and you determine that the line breaks are caused because a word is not being hyphenated. This tag is also useful when a word or term (usually a technical term that is not commonly used) is not being hyphenated correctly.

---

**EXAMPLE**      The following example shows how to use the <HYPHENATE> tag.

```
<p>Among the more common literary devices used are  
<hyphenate>(on\o\mat\o\poe\ia) and anthropomorphism.
```

This example produces the following output:

Among the more common literary devices used are onomato-  
poeia and anthropomorphism.

## <ICON>

---

## <ICON>

Allows you to print a small graphic with explanatory text printed to the right or left of the graphic.

---

### FORMAT <ICON>

---

### ARGUMENTS *None.*

---

### related tags

- <FIGURE\_FILE>
  - <ICON\_FILE>
  - <ICON\_TEXT>
- 

### restrictions

Use the <ICON> tag for graphics that have a length and width of approximately 2 inches. Use the <FIGURE\_FILE> tag for larger graphics.  
The output device must support graphics.

---

### required terminator

<ENDICON>

---

### DESCRIPTION

The <ICON> tag allows you to print a small graphic with explanatory text printed to the right or left of the graphic.

---

### EXAMPLES

The following example shows how to code a graphic on the right, with text on the left.

```
1 <ICON>
  <ICON_FILE>(LN03\small_art.six\1.5\2.0\RIGHT)
  <ICON_TEXT>(The text accompanying the
small piece of art. The text can be smaller or larger
than the graphic; the <tag>(ICON) tags make the necessary
adjustments for the output.)
<ENDICON>
```



The following example shows one graphic placed on the left, with text on the right, followed by a second graphic placed on the right, with text on the left.

2

```

<ICON>
<ICON_FILE>(LN03\RUNNING SHOES.SIX\13\6)
<ICON_FILE>(PS\RUNNING SHOES.PS\13\6)
<ICON_FILE>(LINE\RUNNING SHOES\13\6)
<ICON_FILE>(BOOKREADER\RUNNING SHOES.BRF\13\6)
<ICON_TEXT>(The image at the left is of a typical pair of running shoes,
shown here to illustrate the placement of an icon with text on the right.)
<ENDICON>
<ICON>
<ICON_FILE>(LN03\FORKLIFT.SIX\10\14\RIGHT)
<ICON_FILE>(PS\FORKLIFT.PS\10\14\RIGHT)
<ICON_FILE>(LINE\FORKLIFT\10\14\RIGHT)
<ICON_FILE>(BOOKREADER\FORKLIFT.BRF\10\14\RIGHT)
<ICON_TEXT>(The image at the right is of a forklift, shown here to illustrate
the placement of an icon with text on the left.)
<ENDICON>

```

This example produces the following output:

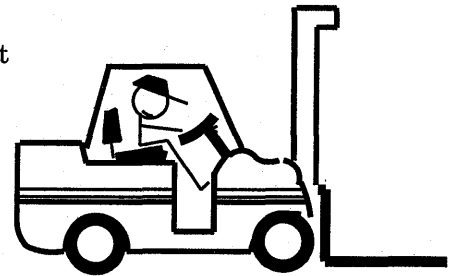


running  
a program

ZK-9557-GE

The image at the left is of a typical pair of running shoes, shown here to illustrate the placement of an icon with text on the right.

The image at the right is of a forklift, shown here to illustrate the placement of an icon with text on the left.



ZK-9558-GE

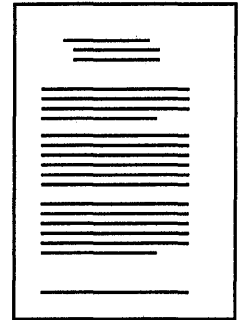
## <ICON>

The following example shows a graphic placed on the right, with wrapping text on the left.

```
3 <ICON>
<ICON_FILE>(LN03\TEXT_PAGE.SIX\11\7.5\RIGHT)
<ICON_FILE>(PS\TEXT_PAGE.PS\11\7.5\RIGHT)
<ICON_FILE>(LINE\TEXT_PAGE\11\7.5\RIGHT)
<ICON_FILE>(BOOKREADER\TEXT_PAGE.BRF\11\7.5\RIGHT)
<ICON_TEXT>(The image at the right is a page of text, shown here to illustrate
wrapping text. The image at the right is page of text, shown here to illustrate
wrapping text. The image at the right is a page of text, shown here to illustrate
wrapping text. The image at the right is a page of text, shown here to illustrate
wrapping text. The image at the right is a page of text, shown here to illustrate
wrapping text. The image at the right is a page of text, shown here to illustrate
wrapping text. The image at the right is a page of text, shown here to illustrate
wrapping text. The image at the right is a page of text, shown here to illustrate
wrapping text. The image at the right is a page of text, shown here to illustrate
wrapping text. The image at the right is a page of text, shown here to illustrate
wrapping text.\WRAP)
<ENDICON>
```

This example produces the following output:

The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text. The image at the right is a page of text, shown here to illustrate wrapping text.



ZK-9741-GE

---

## <ICON\_FILE>

Specifies a graphics file that accompanies text within the <ICON> and <ENDICON> tags.

---

### FORMAT

<ICON\_FILE> (*target device* { \ *file spec* }  
                  \ *vertical size*  
                  \ *horizontal size*  
                  [ \ *RIGHT* ] )

---

### ARGUMENTS

#### *target device*

Specifies a keyword indicating the output device for the graphics file. Keywords are provided both for devices that do (LN03, PS, and Bookreader) and do not (LINE, MAIL, and TERMINAL) support graphics. Each keyword allows you to insert the necessary amount of white space for the specific output device; keywords for devices that do not support graphics allow you to insert blank space in place of an icon. The following table lists the output devices and expected output for each keyword.

Keyword	Device	Output
BOOKREADER	DECwindows screen	The specified graphics file is output online using Bookreader.
LN03	LN03 Laser Printer	The specified sixel graphics file is output as an icon.
PS	PRINTSERVER 40 or LN03R SCRIPTPRINTER	The specified PostSCRIPT graphics file <sup>1</sup> is output as an icon.
LINE MAIL TERMINAL	Line Printer	If specified, blank space is output with the file spec argument written in that blank space. Use only one of these three keywords to indicate a monospaced destination.

---

<sup>1</sup>PostSCRIPT graphics files must conform to the Encapsulated PostSCRIPT File Format published by Adobe Systems, Inc.

#### *file spec*

Specifies the graphics file. No default file type is supplied.

#### **SPACE**

This keyword lets you reserve blank space in the output file for an icon. You can use it if you do not use the *file spec* argument. Use SPACE to reserve space for art that will be pasted in at a later date, or when you expect to process a file for more than one output device, but do not have graphics files for all devices.

## <ICON\_FILE>

### ***vertical size***

Specifies the vertical size of the printed graphic in picas (there are 6 picas in an inch and 12 points in a pica). This argument must be a nonnegative integer or decimal number, including zero.

### ***horizontal size***

Specifies the width of the printed graphic in picas (there are 6 picas to an inch and 12 points in a pica). This argument must be a nonnegative integer or decimal number, including zero.

### ***RIGHT***

This is an optional keyword argument. If you use it, it must come last in the argument list. RIGHT specifies that the graphic image be placed to the right of the text. If you do not specify this argument, the image is placed on the left of the text specified in the <ICON\_TEXT> tag.

---

#### **related tags**

- <FIGURE\_FILE>
- <ICON>
- <ICON\_TEXT>

---

#### **restrictions**

Valid only in the context of an <ICON> tag.  
The output device must support graphics.

---

## **DESCRIPTION**

The <ICON\_FILE> tag specifies a graphics file that accompanies text within the <ICON> and <ENDICON> tags. The specified file should be a graphics file suitable for printing on the specified output device.

If you plan to process your file for more than a single destination, you can use multiple <ICON\_FILE> tags to ensure that the appropriate icon will be included for the appropriate destination. An example of this coding is provided in the examples in the <ICON> tag description. If you use multiple <ICON\_FILE> tags in this way, you should specify only a single <ICON\_FILE> tag for all the monospaced destination keywords (MAIL, TERMINAL, or LINE).

---

## **EXAMPLE**

See the examples in the description of the <ICON> tag.

---

## <ICON\_TEXT>

Labels the text that accompanies a graphic image that you include with the <ICON\_FILE> tag.

---

**FORMAT**      <ICON\_TEXT>(text [ \ WRAP])

---

**ARGUMENTS**    *text*  
Specifies the text that accompanies the graphic. The text can be as long as you want and can include paragraphs and lists.

**WRAP**  
This is an optional keyword argument. It specifies that the text wrap around the graphic if the text is longer than the graphic.

---

**related tags**

- <ICON>
- <ICON\_FILE>

---

**restrictions**    Valid only in the context of an <ICON> tag.  
If you use the WRAP argument, the text cannot contain any <LIST> or <P> tags.

---

**DESCRIPTION**    The <ICON\_TEXT> tag labels the text that accompanies a graphic image that you include with the <ICON\_FILE> tag. The text appears along the side of the icon or wraps around the icon if you specify the WRAP argument and the text is long enough.

Both the <ICON\_TEXT> and <ICON\_FILE> tags must occur in the context of the <ICON> and <END\_ICON> tags.

---

**EXAMPLE**        See the examples in the description of the <ICON> tag.

## <INCLUDE>

---

## <INCLUDE>

Specifies the position in a document where you want to include a file for processing.

---

**FORMAT**      **<INCLUDE>** (*file spec*)

---

**ARGUMENTS**      ***file spec***  
Specifies the file you want to include.

If you specify a logical name instead, and the source file is an element of a book, you can define the logical name using an `<INCLUDES_FILE>` tag in the profile of the book. If the source file is not an element of a book, or if the profile does not contain the `<INCLUDES_FILE>` tag, be sure to define the logical name before processing the file.

---

**related tags**

- `<EXAMPLE_FILE>`
- `<TABLE_FILE>`

---

**restrictions**      Invalid in an argument to a tag.

In a file that contains a book element that you want to process through a bookbuild, make sure to place the book element tag (for example, `<CHAPTER>`) as the first tag in the file. Place the `<INCLUDE>` tag after the book element tag.

---

**DESCRIPTION**      The `<INCLUDE>` tag specifies the position in a document where you want to include a file for processing.

---

**EXAMPLE**      The following example shows how to include a file that does the following:

- Contains text that might be repeated multiple times in a source file
- Is used in more than one document

```
<INCLUDE>(doc_local_templates:boilerplate.sdml)
```

If you did not use the file specification `doc_local_templates:boilerplate.sdml`, but instead used a logical name, the logical name could be equated to the file specification by using the `<INCLUDES_FILE>` tag in the profile. See the example in the `<INCLUDES_FILE>` tag description.

---

## <INCLUDES\_FILE>

Equates a logical name with a file specification during processing of a profile.

---

**FORMAT**            <INCLUDES\_FILE> (*logical name \ file spec*)

---

**ARGUMENTS**        *logical name*  
Specifies the logical name for the included file. (You can use this name as the argument to an <INCLUDE> tag.)

*file spec*  
Specifies the file specification into which the logical name translates.

- 
- related tags**
- <ELEMENT>
  - <INCLUDE>
  - <PROFILE>

---

**restrictions**        Valid only in the context of a <PROFILE> tag.

---

**DESCRIPTION**        The <INCLUDES\_FILE> tag equates a logical name with a file specification during processing of a profile. You can place the <INCLUDES\_FILE> tag in your profile to define a logical name for a file whose contents are included in one of your element files. The first argument to this tag is the logical name for the included file. (You use this name as the argument to the <INCLUDE> tag.) The second argument specifies the actual file specification into which the logical name translates.

You do not need to use the <INCLUDES\_FILE> tag if either of the following is true:

- Your element file does not contain any <INCLUDE> tags.
- The <INCLUDE> tags in your element file have complete file specifications as arguments.

The <INCLUDES\_FILE> tags follow the <ELEMENT> tag for which they are needed. During a bookbuild, VAX DOCUMENT establishes the logical name definition for each <INCLUDES\_FILE> tag before it actually reads and processes the book element file names in the preceding <ELEMENT> tag. The logical name remains defined during the processing of later book elements.

When a book element is processed by itself (when you use the /PROFILE qualifier on the command line), VAX DOCUMENT again establishes the logical name definitions that were specified by the <INCLUDES\_FILE> tag in the profile.

## <INCLUDES\_FILE>

---

### EXAMPLE

The following example shows how to use the <INCLUDES\_FILE> tag to define a logical name for the file MY\_VERY\_LONG\_TABLE.SDML, which is kept in the directory [mydirectory]. In the book element file ERROR\_CHAP.SDML, you include the table with the <INCLUDE>(error\_msg\_tab) tag.

```
<PROFILE>
.
.
.
<ELEMENT>(error_chap.sdml)
<INCLUDES_FILE>(error_msg_tab\mydisk:[mydirectory]my_very_long_table.sdml)
.
.
.
<ELEMENT>(tags_chap.sdml)
<ENDPROFILE>
```



---

## <INDEX\_FILE>

Specifies the position in a book or document where you want to include an index file in the output.

---

**FORMAT**      <INDEX\_FILE>[(*file spec*)]

---

**ARGUMENTS**      *file spec*

This is an optional argument. It specifies the name of the index file. Use this argument if you want to include an index file with a name other than the default file name.

The default file name is *file spec\_index.DVI\_device*. If you specify a file type, be sure that the device name matches the destination device you specify on the DOCUMENT command line. For example, if you specify *file spec\_index.DVI\_PS*, be sure you specify PS as the destination on the command line.

---

**related tags**

- <CONTENTS\_FILE>
- <PROFILE>

---

**DESCRIPTION**

The <INDEX\_FILE> tag Specifies the position in a book or document where you want to include an index file in the output. This tag does not produce an index, but instead indicates placement of the index file. An index file is produced when you specify the /INDEX qualifier on the DOCUMENT command line. See Appendix A for information about using the /INDEX qualifier on the command line.

Using the <INDEX\_FILE> tag is highly recommended (but not required) for Bookreader output, because an index is helpful for locating information in a book you create for Bookreader.

It is recommended that you place the <INDEX\_FILE> tag in your profile. This ensures correct placement of the index in both printed and Bookreader output.

An index file always receives the file type *.DVI\_device*, where *device* is the type of output device you specified on the DOCUMENT command line. To generate an index from an individual file that contains an <INDEX\_FILE> tag, specify the /INDEX qualifier on the DOCUMENT command line.

If you specify the /INDEX qualifier on the DOCUMENT command line without using the <INDEX\_FILE> tag in your profile, an index is generated separately from your document. For more information on generating an index, see *VAX DOCUMENT Producing Online and Printed Documentation*.

## <INDEX\_FILE>

---

### EXAMPLE

See the index in this manual for a sample index. The <INDEX\_FILE> tag is placed in the profile file after the last appendix file. See the <PROFILE> tag description for an example of an <INDEX\_FILE> tag in a profile.

---

## <INTERACTIVE>

Begins an example dialog between the user and system and enables the <S> and <U> tags to distinguish system text from user text.

---

### FORMAT

<INTERACTIVE>(code)  
or  
<INTERACTIVE>[[ [KEEP]\ [WIDE] ]]  
                  \ [MAXIMUM]]  
interactive code or text  
.  
.  
.  
  
<ENDINTERACTIVE>

---

### ARGUMENTS

#### **code**

Specifies a code fragment you want to insert into your text.

If you do not specify this argument, you must use <ENDINTERACTIVE> as the terminating tag.

#### **KEEP**

This is an optional keyword argument. It specifies that the example is not to be broken across pages; that is, if the example does not fit on the current page, it will be placed on the next page. If the example itself does not fit on an entire single page of output, it will be broken anyway. This keyword has no effect for Bookreader output.

#### **WIDE**

This is an optional keyword argument. It specifies that the width of the example exceeds the document's default width for text, and, depending on the **doctype**, is interpreted as follows:

- If the doctype contains a left margin area that is normally used for headings, the example's width spans that area as well as the normal text area.
- If the document uses a multicolumn format, multicolumn output is suspended while the example is processed. The example is output and multicolumn output is then restored.
- Depending on the doctype, this argument provides a range of font sizes and font styles for examples.

## <INTERACTIVE>

### **MAXIMUM**

This is an optional keyword argument. It specifies that the text of the code example is adjusted to a smaller point size in order to fit within the boundaries of the left and right margins of the page. You can use this keyword in conjunction with **WIDE** to indicate that the example may require additional adjustment to fit within the boundaries of the page. You should use this keyword with discretion, and it may not be suitable in all doctypes.

**Note:** You can use the **KEEP**, **WIDE**, and **MAXIMUM** keywords together or separately and in any order in the argument list.

---

#### related tags

- <S>
- <U>
- <VALID\_BREAK>

---

#### restrictions

You cannot use indexing tags (<X> and <Y> tags) in the context of an <INTERACTIVE> tag.

You cannot use tab characters to format interactive examples; use spaces instead.

You cannot use text element tags, such as <P>, <LIST>, or <NOTE>, in the context of an <INTERACTIVE> tag.

---

#### required terminator

<ENDINTERACTIVE> — Required if you do not specify an argument to the <INTERACTIVE> tag.

---

## DESCRIPTION

The <INTERACTIVE> tag begins an example dialog between the user and system and enables the <S> and <U> tags to distinguish system text from user text. If your interactive example is longer than a few lines, use the <VALID\_BREAK> tag to indicate the acceptable points for a page break within that example.

---

## EXAMPLE

This example shows how to use the tags to create an interactive example. Note that you should specify whatever space follows the system prompt within the <S> tag.

```
<P>If you do not specify the full command line, DCL will prompt
you for the missing information. For example, if you do not specify an
input file and an output file when you enter the COPY command, you will
be prompted as follows:
<INTERACTIVE>
<S>($ )<U>(COPY)
<S>($ _From: )<U>(INTERACT.SDML)
<S>($ _To: )<U>(NEWFILE.SDML)
<ENDINTERACTIVE>
```

This example produces the following output:

If you do not specify the full command line, DCL will prompt you for the missing information. For example, if you do not specify an input file and an output file when you enter the COPY command, you will be prompted as follows:

```
$ COPY
$_From: INTERACT.SDML
$_To: NEWFILE.SDML
```

## <KEEP>

---

## <KEEP>

Specifies that a string of text should not be broken across a line of output.

---

**FORMAT**      **<KEEP>**(*text*)

---

**ARGUMENTS**    *text*  
Specifies the text string you want kept on the same line of output.

---

**related tags**

- You can use the following tags in conjunction with the <KEEP> tag to allow you to specify formatting attributes:

- <DEFINE\_SYMBOL>
- <EMPHASIS>
- <HYPHENATE>

---

**DESCRIPTION**    The <KEEP> tag specifies that a string of text should not be broken across a line of output. Use this tag to specify a string of text you want kept on the same line of output.

---

**EXAMPLES**      In the following example, the file specification is specified as an argument to the <KEEP> tag so that it will not be broken across a line.

**1**    <P>  
The complete file specification is: <KEEP>(DISK\$:[SMITH.TRIPS.EXPENSES]MILEAGE.TXT)

This example produces the following output:

The complete file specification is:  
DISK\$:[SMITH.TRIPS.EXPENSES]MILEAGE.TXT

The following example defines a text element symbol name using an en dash and specifies that the line should not break between the en dash and the 11 when you refer to the symbol.

**2**    <define\_symbol>(vax11\<delayed><keep>(vax--11)<enddelayed>)

You refer to the symbol with the <REFERENCE>(vax11) tag, which produces the following output:

VAX-11

---

## <KEYWORD>

Labels a word that you want to distinguish typographically.

---

**FORMAT**      <KEYWORD>(word)

---

**ARGUMENTS**    *word*  
Specifies the word you want to distinguish.

---

**related tags**

- <NEWTERM>
- <SPECIAL\_CHAR>
- <VARIABLE>

---

**DESCRIPTION**    The <KEYWORD> tag labels a word that you want to distinguish typographically. The default action of the <KEYWORD> tag outputs the keyword in boldface. As formatted output, the <KEYWORD> tag may appear similar to the <EMPHASIS> tag with the BOLD argument. By using separate tags to label different kinds of information, however, you are free to change the format of any one kind of information without affecting the others.

What constitutes a keyword is something about which both editor and writer must agree. Use the <KEYWORD> tag consistently within a document and across a document set.

---

**EXAMPLE**            The following example shows how to use the <KEYWORD> tag.

<P>A <KEYWORD>(field) is a set of contiguous bytes in a logical record.

This example produces the following output:

A **field** is a set of contiguous bytes in a logical record.

## <LANGUAGE>

---

## <LANGUAGE>

Causes text that is automatically generated by VAX DOCUMENT (such as "Example", "Figure", "Table", and so on) to be output in the language you specify. It also causes text to be hyphenated and index entries to be sorted according to the rules of the language you specify.

---

### FORMAT

<LANGUAGE> ( *CANADIAN*  
*DANISH*  
*DUTCH*  
*ENGLISH*  
*FINNISH*  
*FRENCH*  
*GERMAN*  
*ITALIAN*  
*NORWEGIAN*  
*PORTUGUESE*  
*SPANISH*  
*SWEDISH* )

---

### ARGUMENTS

#### *language-x*

Specifies the language. English is the default language. Note that CANADIAN specifies the French Canadian language.

---

### restrictions

You can use this tag only once in a file; subsequent occurrences of the tag are ignored.

You must place this tag at the beginning of your file or immediately after the <PROFILE> tag in a profile.

---

### DESCRIPTION

The <LANGUAGE> tag causes text that is automatically generated by VAX DOCUMENT (such as "Example", "Figure", "Table", and so on) to be output in the language you specify. It also causes text to be hyphenated and index entries to be sorted according to the rules of the language you specify.

To use the default foreign language text, use the <LANGUAGE> tag with the argument of your choice. You can change the default foreign language text as follows:

- 1 To modify the output for individual use only, copy DOC\$STANDARD\_FORMATS:TEX\$LANGUAGE\_MODULES.TEX into a directory and define the logical name DOC\$PERSONAL\_FORMATS to point to that directory. For example:



```
$ DEFINE DOC$PERSONAL_FORMATS DISK1:[MYDIR]
```

To modify the output for all users on the system, copy DOC\$STANDARD\_FORMATS:TEX\$LANGUAGE\_MODULES.TEX into a directory and define the logical name DOC\$LOCAL\_FORMATS to point to that directory.

- 2 Edit this local copy of TEX\$LANGUAGE\_MODULES.TEX by uncommenting the line that inputs the language you want to change. In the following example, the line that inputs the Danish language file has been uncommented, so that a local copy of TEX\$DANISH\_MODULE.TEX will be used instead of the default version.

```
%\input doc$$dormats:tex$canadian_module.tex  
\input doc$$dormats:tex$danish_module.tex % enables your Danish changes  
\input doc$$dormats:tex$dutch_module.tex
```

- 3 The text for the supported languages is contained in the following files. Copy the file you wish to change into a local directory and edit the file for local language changes.

```
TEX$CANADIAN_MODULE.TEX  
TEX$DANISH_MODULE.TEX  
TEX$DUTCH_MODULE.TEX  
TEX$ENGLISH_MODULE.TEX  
TEX$FINNISH_MODULE.TEX  
TEX$FRENCH_MODULE.TEX  
TEX$GERMAN_MODULE.TEX  
TEX$ITALIAN_MODULE.TEX  
TEX$NORWEGIAN_MODULE.TEX  
TEX$PORTUGUESE_MODULE.TEX  
TEX$SPANISH_MODULE.TEX  
TEX$SWEDISH_MODULE.TEX
```

- 4 You can modify any of the text strings by entering new text within the braces for a specific macro.

The hyphenation patterns for the supported languages are contained in the file TEX\$EXCEPTIONS.TEX. The default version of this file, which resides in DOC\$STANDARD\_FORMATS, is invoked when the text formatter begins processing. You can modify the hyphenation of any word as follows:

- To change the hyphenation for individual use only, copy DOC\$STANDARD\_FORMATS:TEX\$EXCEPTIONS.TEX into a directory, define the logical name DOC\$PERSONAL\_FORMATS to point to that directory, and edit the file.
- To change the hyphenation for all users on the system, copy DOC\$STANDARD\_FORMATS:TEX\$EXCEPTIONS.TEX into a directory, define the logical name DOC\$LOCAL\_FORMATS to point to that directory, and edit the file.

## <LANGUAGE>

VAX DOCUMENT searches for a copy of `TEX$LANGUAGE_MODULES.TEX` and `TEX$EXCEPTIONS.TEX` in the order listed below, and uses the first copy of each file that it encounters:

- 1 `DOC$PERSONAL_FORMATS`—for individual use
- 2 `DOC$LOCAL_FORMATS`—for local use
- 3 `DOC$STANDARD_FORMATS`—the default version

---

## EXAMPLES

The following example shows how to code a file using the `<LANGUAGE>` tag.

1 `<LANGUAGE>` (FRENCH)  
`<CHAPTER>` (Le Tour De France)  
`<P>` La plus longue course de bicyclettes du monde a lieu en France chaque année au début de l'été: c'est le Tour de France.

The following example is part of the English language section of `TEX$EXCEPTIONS.TEX`, to which you add the hyphenation exception in the format shown.

2

```
*****  
% English  
*****  
  
\gdef\ENGLexceptions{  
\hyphenation{  
as-so-ciate  
as-so-ciates  
dec-li-na-tion  
oblig-a-tory  
}  
}
```

---

<LE>

Labels a list element.

---

**FORMAT**            <LE>[(*callout number*)]

---

**ARGUMENTS**        *callout number*  
This is an optional argument. It sets the callout number identifying the list element.

---

**related tags**        • <LIST>

---

**restrictions**        Valid only in the context of a <LIST> tag.  
You can use the *callout number* argument of the <LE> tag only in the context of a <LIST>(CALLOUT) tag.

---

**DESCRIPTION**      The <LE> tag labels a list element.

---

**EXAMPLES**            See the examples in the discussion of the <LIST> tag.

## <LINE>

---

## <LINE>

Specifies that the text that follows this tag be placed on a new line of output.

---

### FORMAT

$$\langle \text{LINE} \rangle [ ( \left\{ \begin{array}{l} \text{INDENT} [ \ \backslash \text{unit number} ] \\ \text{BIGSKIP} \\ \text{SMALLSKIP} \end{array} \right\} ) ]$$

---

### ARGUMENTS

#### **INDENT**

This is an optional keyword argument. It specifies that the next line or block of text be indented from the preceding text.

#### ***unit number***

If you use **INDENT**, you have the option of specifying a *unit number* from 2 to 9 indicating the number of units that the text is to be indented. The default indent is 1 unit. The size of these units is determined separately for each doctype.

#### **BIGSKIP**

#### **SMALLSKIP**

These are optional keyword arguments. They specify that a set amount of vertical space precede the element identified as a line or block of text. The actual amount of space created is determined by the doctype.

---

### related tags

- <CENTER\_LINE>
- <FINAL\_CLEANUP>
- <RIGHT\_LINE>

---

### restrictions

Invalid in the context of an <EXAMPLE> and a <MATH> tag.

---

### DESCRIPTION

The <LINE> tag specifies that the text that follows this tag be placed on a new line of output. The result of using the <LINE> tag differs according to whether you use it in the context of a <P> tag, a <TABLE\_ROW> tag, or a <FORMAT> tag:

- When you specify the <LINE> tag in the context of a paragraph of text or a list element, the <LINE> tag ends the current paragraph or list element and then starts a new block of text. By default, the block of text is not preceded by any extra vertical space, nor is it indented.
- When you specify the <LINE> tag in the context of a <TABLE\_ROW> or a <FORMAT> tag, the <LINE> tag causes the following text to begin on a new line of output, but does not modify the current paragraph.

Be sure that you want a line break in all formatting cases, including Bookreader output. Do not use the <LINE> tag in a paragraph simply to cause the text formatter to break a line within the paragraph. Breaking a line to override a specific line break is a final formatting instruction; use the <FINAL\_CLEANUP>(LINE\_BREAK) tag for this.

---

## EXAMPLES

The following example shows how to use the <LINE> tag in a list.

**1** <LIST>(NUMBERED)  
<le>ITEM  
<LINE>This item specifies...

This example produces the following output:

**1** ITEM  
This item specifies . . .

The following example shows how to use the <LINE> tag to create an indented line with a small skip.

**2** <P>This is a normal paragraph.  
<line>(INDENT\1\SMALLSKIP)  
This is a block paragraph, indented with a small skip.

This example produces the following output:

This is a normal paragraph.  
This is a block paragraph, indented with a small skip.

The following example shows the misuse of the <LINE> tag for formatting within a paragraph.

**3** <P>Elaborate designs are inappropriate for software manuals or for any serious or formal books because they do not reflect the intention of the writer. The purpose of any book design is<LINE> to clarify what the author is conveying, to translate the text attractively as print on a page, to communicate the message visually in harmony with the ideas.

Although the correct output may result on a given output device for a particular run of a file, the output may also be formatted incorrectly, as follows:

Elaborate designs are inappropriate for software manuals or for any serious or formal books because they do not reflect the intention of the writer. The purpose of any book design is to clarify what the author is conveying, to translate the text attractively as print on a page, to communicate the message visually in harmony with the ideas.

The following example shows how to use the <LINE> tag in a three-column table.

## <LINE>

```
4 <TABLE>(Card Reader Errors: Causes and Corrective Actions\cardread_tab)
<TABLE_ATTRIBUTES>(wide\multipage)
<TABLE_SETUP>(3\12\21)
<TABLE_HEADS>(Error\Causes\Corrective Action)
<TABLE_ROW>(READ CHECK\Card edges torn <LINE> Punch in column 0
or 81\Remove the faulty card from the output stacker, duplicate the card, place
it in the input hopper, and press the <EMPHASIS>(RESET\bold) button.)
<TABLE_ROW_BREAK>(FIRST)
<TABLE_ROW>(PICK CHECK\Damage to leading edge <LINE> Torn webs
<LINE> Cards
stapled together\Remove the card from the input hopper, duplicate
the faulty card, place the card back in the input
hopper, and press the <EMPHASIS>(RESET\bold) button.)
<TABLE_ROW_BREAK>(LAST)
<TABLE_ROW>(STACK CHECK\Jam in the card track <LINE> Badly mutilated card
\Correct the jam and/or remove the mutilated card from the output stacker,
duplicate the card, place it in the input hopper, and press the
<EMPHASIS>(RESET\bold) button.)
<TABLE_ROW>(HOPPER CHECK\Input hopper empty <LINE> Output stacker full\Load
the input hopper. <LINE> Unload the output stacker.)
<ENDTABLE>
```

This example produces the following output:

**Table n-n Card Reader Errors: Causes and Corrective Actions**

Error	Causes	Corrective Action
READ CHECK	Card edges torn Punch in column 0 or 81	Remove the faulty card from the output stacker, duplicate the card, place it in the input hopper, and press the <b>RESET</b> button.
PICK CHECK	Damage to leading edge Torn webs Cards stapled together	Remove the card from the input hopper, duplicate the faulty card, place the card back in the input hopper, and press the <b>RESET</b> button.
STACK CHECK	Jam in the card track Badly mutilated card	Correct the jam and/or remove the mutilated card from the output stacker, duplicate the card, place it in the input hopper, and press the <b>RESET</b> button.
HOPPER CHECK	Input hopper empty Output stacker full	Load the input hopper. Unload the output stacker.

The following example shows how to use the <LINE> tag in the context of a <FORMAT> tag.

```
5 <FORMAT>(MY COMMAND)
<FCMD>( )
<FPARMS>(one parameter<line>two<line>three)
<ENDFORMAT>
```

This example produces the following output:

---

**MY COMMAND**    *one parameter*  
                  *two*  
                  *three*

---

## <LINE\_ART>

Labels a rough sketch produced at the terminal keyboard for draft output, to give some idea of what the final figure will look like.

---

### FORMAT

<LINE\_ART>[[([KEEP] \ [WIDE] \ [MAXIMUM])]]

---

### ARGUMENTS

#### **KEEP**

This is an optional keyword argument. It specifies that the line art not be broken across pages; that is, if the line art does not fit on the current page, it will be placed on the next page. If the line art itself does not fit on a single page of output, it will be broken anyway. This keyword has no effect for Bookreader output.

#### **WIDE**

This is an optional keyword argument. It specifies that the width of the line art exceeds the document's default width for text. Depending on the **doctype**, this argument is interpreted as follows:

- If the doctype contains a left margin area that is normally used for headings, the line art's width spans that area as well as the normal text area.
- If the document uses a multicolumn format, the line art suspends the multicolumn output while it is being processed. The line art is then output, and then multicolumn output is restored.
- Depending on the doctype, this argument provides a range of font sizes and font styles for line art.

#### **MAXIMUM**

This is an optional keyword argument. It specifies that the text of the code example is adjusted to a smaller point size in order to fit within the boundaries of the left and right margins of the page. You can use this keyword in conjunction with **WIDE** to indicate that the example may require additional adjustment to fit within the boundaries of the page. You should use this keyword with discretion, and it may not be suitable in all doctypes.

**Note:** You can use the **KEEP**, **WIDE**, and **MAXIMUM** keywords together or separately and in any order in the argument list.

---

### related tags

- <FIGURE>
- <VALID\_BREAK>

---

### required terminator

<ENDLINE\_ART>

## <LINE\_ART>

---

### DESCRIPTION

The <LINE\_ART> tag labels a rough sketch produced at the terminal keyboard for draft output, to give some idea of what the final figure will look like.

Note that the results of keyboard drawing may be adequate for draft purposes, but may not be acceptable for laser printer output.

When you print an informal or formal figure (within which is the line art) or process an informal figure for Bookreader display, the figure is indented in the same way the regular text is for the particular **doctype** you are using.

---

### EXAMPLE

See the first example in the discussion of the <FIGURE> tag.



---

## <LIST>

Begins a list. You specify the type of list by using a keyword argument to the tag.

---

### FORMAT

<LIST>(keyword[\ attribute-1 [\ attribute-2]])

---

### ARGUMENTS

#### **KEYWORD**

Specifies the type of list. The following keywords indicate the type of list.

ALPHABETIC	The list element identifiers are alphabetic letters.
CALLOUT	The list element identifiers are reverse-print callout numbers (on supported output devices), for example ⑤.
NUMBERED	The list element identifiers are Arabic numerals.
ROMAN	The list element identifiers are Roman numerals.
SIMPLE	There are no list element identifiers.
STACKED	Individual list elements do not have identifiers, but the entire list is stacked within the specified set of delimiters (braces, brackets, double brackets, single or double vertical rules).
UNNUMBERED	List element identifiers are special characters.

#### **ATTRIBUTES**

This is an optional argument. It specifies the attributes of the list element keywords. You can specify more than one attribute.

BRACES	For a STACKED list. Each attribute for a STACKED list specifies the delimiter to be used to surround the stacked elements. If you do not specify a keyword, list elements are stacked without a surrounding delimiter.
BRACKETS	
DOUBLE_BRACKETS	
DOUBLE_VERTICAL_RULE	
VERTICAL_RULE	
<i>char</i>	For an UNNUMBERED list. Specifies a single character or a tag that results in a single printed character of output that is used to indicate list elements. If you do not specify <i>char</i> , the bullet character is used.
<i>start-letter</i>	For an ALPHABETIC list. Specifies the alphabetic letter to use for the first item in the sequence. Subsequent items are automatically incremented.
<i>start-number</i>	For a NUMBERED or ROMAN list. Specifies the number to be assigned to the first list element. Subsequent list elements are automatically incremented.

## <LIST>

UPPERCASE

For an ALPHABETIC or ROMAN list. Specifies that the list element identifiers (alphabetic letters or Roman numerals) be printed in uppercase. By default, Alphabetic and Roman list element identifiers are printed in lowercase.

---

### related tags

- <ALIGN\_AFTER>
- <ALIGN\_CHAR>
- <LE>

---

### restrictions

Valid only in the context of a <P> or a <TABLE> tag.

You can nest <LIST> tags that use the following keywords: ALPHABETIC, NUMBERED, ROMAN, SIMPLE, or UNNUMBERED. (Nesting a list means that an element of a list may itself contain the beginning of another list.) None of these list types are compatible with the <LIST>(STACKED) tag.

---

### required terminator

<ENDLIST>

---

## DESCRIPTION

The <LIST> tag begins a list. You specify the type of list by using a keyword argument to the tag.

The <LIST>(ALPHABETIC), <LIST>(NUMBERED), and <LIST>(ROMAN) tags begin a list whose elements have a particular sequence or priority. The <LIST>(SIMPLE) or <LIST>(UNNUMBERED) tags begin a list that has no inherent order or priority. Alphabetic lists are useful when nested within numbered lists. Use a numbered list to indicate a particular sequence or priority within the list elements.

The <LIST>(SIMPLE) tag labels a simple list with no enumerator or special character preceding each list element. It terminates the current paragraph and indents the list, so that the list is seen as a separate entity from the text above and below it.

The <LIST>(UNNUMBERED) tag labels a list that has no particular order or priority within the list element. Normally a bullet is the character used before each list element. You can specify another character by using it as the second argument to the <LIST>(UNNUMBERED) tag; for example, <LIST>(UNNUMBERED\+).

Use the <LIST>(STACKED) tag to begin a list whose elements are left-justified on successive lines. The block of text can have large braces or brackets placed on each side. The block of text is then centered vertically so that it aligns with text to the left or right on the same line. The <LIST>(STACKED) tag is especially useful for showing syntactic elements with a <FORMAT> tag.

---

## EXAMPLES

The following is an example of nested lists. The first nested list has no particular sequence. The order of the second nested list is reflected in the use of the <LIST>(ALPHABETIC) tag.

```
1 <LIST>(NUMBERED)
  <LE>Review the document
  <LE>Find out these numbers:
    <LIST>(UNNUMBERED)
      <LE>number of copies
      <LE>department number
      <LE>order number
    <ENDLIST>
  <LE>For revisions of documents not already in the library, have
  the writer rename the file according to its new order number.
  <LE>Open a library for the book if none exists.
    <LIST>(ALPHABETIC)
      <LE>Decide with the writer about timing:
        <LIST>(UNNUMBERED)
          <LE>Must be done before final production.
          <LE>Should not be done until text is stable.
        <ENDLIST>
      <LE>Cooperate with the business department on paperwork.
      <LE>Help writers name new element files correctly.
      <LE>Verify that all files are of the same file type.
      <LE>For a revision, be sure files are renamed with the new file name.
    <ENDLIST>
  <ENDLIST>
```

This example produces the following output:

- 1 Review the document
- 2 Find out these numbers:
  - number of copies
  - department number
  - order number
- 3 For revisions of documents not already in the library, have the writer rename the file according to its new order number.
- 4 Open a library for the book if none exists.
  - a. Decide with the writer about timing:
    - Must be done before final production.
    - Should not be done until text is stable.
  - b. Cooperate with the business department on paperwork.
  - c. Help writers name new element files correctly.
  - d. Verify that all files are of the same file type.
  - e. For a revision, be sure files are renamed using the new file name.

The following is an example of an alphabetic list that begins with the letter “e”.

## <LIST>

**2** <P>Items e and f describe the \$GETJPI AST activity:  
<LIST>(ALPHABETIC\e)  
<LE>An ACB is constructed for a special kernel AST.  
<LE>When the special kernel mode AST routine executes in the context of the target process, the requested information is moved into the system buffer. The ACB is then reset to deliver a special kernel mode AST back to the requesting process.  
<ENDLIST>

This example produces the following output:

Items e and f describe the \$GETJPI AST activity:

- e.** An ACB is constructed for a special kernel AST.
- f.** When the special kernel mode AST routine executes in the context of the target process, the requested information is moved into the system buffer. The ACB is then reset to deliver a special kernel mode AST back to the requesting process.

The following is an example of a list with callouts.

**3** <P>At this point, you can log in to the system as the system manager by performing the following steps at the console terminal:  
<LIST>(CALLOUT)  
<LE>Press RETURN.  
<LE>In response to the system's request for your user name, type SYSTEM.  
<LE>In response to the system's request for your password, type MANAGER.  
<ENDLIST>

This example produces the following output:

At this point, you can log in to the system as the system manager by performing the following steps at the console terminal:

- 1** Press RETURN.
- 2** In response to the system's request for your user name, type SYSTEM.
- 3** In response to the system's request for your password, type MANAGER.

The following is an example of a numbered list.

**4** <P>At this point, you can log in to the system as the system manager by performing the following steps at the console terminal:  
<LIST>(NUMBERED)  
<LE>Press RETURN.  
<LE>In response to the system's request for your user name, type SYSTEM.  
<LE>In response to the system's request for your password, type MANAGER.  
<ENDLIST>

This example produces the following output:

At this point, you can log in to the system as the system manager by performing the following steps at the console terminal:

- 1** Press RETURN.
- 2** In response to the system's request for your user name, type SYSTEM.
- 3** In response to the system's request for your password, type MANAGER.

The following is an example of a numbered list that begins with the number "6".

```
5 <P>Items six and seven describe two main principles of a generic
markup language:
<LIST>(NUMBERED\6)
<LE>Descriptive markup predominates and is distinguished from processing
instructions.
<LE>Markup is formally defined for each type of document.
<ENDLIST>
```

This example produces the following output:

Items six and seven describe two main principles of a generic markup language:

- 6 Descriptive markup predominates and is distinguished from processing instructions.
- 7 Markup is formally defined for each type of document.

The following is an example of a list that begins with the Roman numeral "iv".

```
6 <P>Items six and seven describe two main principles of a generic
markup language:
<LIST>(ROMAN\4)
<LE>Descriptive markup predominates and is distinguished from processing
instructions.
<LE>Markup is formally defined for each type of document.
<ENDLIST>
```

This example produces the following output:

Items six and seven describe two main principles of a generic markup language:

- iv Descriptive markup predominates and is distinguished from processing instructions.
- v Markup is formally defined for each type of document.

The following is an example of a simple list.

```
7 <P>The following items are needed:
<LIST>(SIMPLE)
<LE>bread
<LE>milk
<LE>cheese
<LE>cereal
<LE>fruit
<ENDLIST>
```

This example produces the following output:

The following items are needed:

- bread
- milk
- cheese
- cereal
- fruit

## <LIST>

The following is an example of two stacked lists with braces around the list elements. The tags are indented to help in visually checking the matching of the <LIST>(STACKED) and <ENDLIST> tags. This indentation has no effect on the output, however.

```
8 <P>
ON <LIST>(stacked\braces)
  <LE>ANYCONDITION
  <LE>ENDFILE(reference)
  <LE>ENDPAGE(reference)
  <LE>FINISH
  <LE>KEY(reference)
  <LE>UNDEFINEDFILE(reference)
  <LE>ERROR
  <LE>FIXEDOVERFLOW
  <LE>OVERFLOW
  <LE>UNDERFLOW
  <LE>VAXCONDITION(expression)
  <LE>ZERODIVIDE
<ENDLIST>

<LIST>(stacked\braces)
  <LE>statement
  <LE>begin-block
<ENDLIST>
```

This example produces the following output:

```
ON { ANYCONDITION
    ENDFILE(reference)
    ENDPAGE(reference)
    FINISH
    KEY(reference)
    UNDEFINEDFILE(reference)
    ERROR
    FIXEDOVERFLOW
    OVERFLOW
    UNDERFLOW
    VAXCONDITION(expression)
    ZERODIVIDE } { statement
                  }
                  { begin-block }
```

The following is an example of nested and stacked lists with brackets.

```
9 <P>
OPEN FILE (reference) [TITLE(expression)]
<P>
<LIST>(stacked\brackets)
  <LE>[STREAM] <LIST>(stacked\brackets)
    <LE>[INPUT]
    <LE>OUTPUT [LINESIZE] [PRINT [PAGESIZE(integer)] ]
  <ENDLIST>
  <LE>RECORD <LIST>(stacked\brackets)
    <LE>[INPUT]
    <LE>OUTPUT
    <LE>UPDATE
  <ENDLIST>
  <LIST>(stacked\brackets)
    <LE>DIRECT
    <LE>[SEQ[UENTIAL]]
  <ENDLIST>
  [KEYED]
<ENDLIST>
```

This example produces the following output:

```
OPEN FILE (reference) [TITLE(expression)]
```

```

[ [STREAM] [ [INPUT]
  OUTPUT [LINESIZE] [PRINT [PAGESIZE(integer)] ] ] ]
[ RECORD [ [INPUT]
  OUTPUT ] [ DIRECT
  UPDATE ] [ [SEQ[UENTIAL]] ] [KEYED] ] ]

```

The following is an example of stacked lists with braces used within a table. The <LIST> tags are indented simply as an aid for visually checking the nesting of tags.

```

10 <TABLE>
<TABLE_SETUP>(2\20)
<TABLE_HEADS>(Items\Description)
<TABLE_ROW>(<LIST>(stacked\braces)
  <LE>one
  <LE>two
  <LE>three
  <ENDLIST>\Stacked in a table.)
<TABLE_ROW>(Another item\<LIST>(stacked\braces)
  <LE>one
  <LE>two
  <LE>three
  <ENDLIST>)
<TABLE_ROW>(This is the last item\
<ENDTABLE>

```

This example produces the following output:

Items	Description
{ one two three }	Stacked in a table.
Another item	{ one two three }
This is the last item	

The following is an example of an unnumbered list.

```

11 <P>To create a system that more closely suits the requirements of
your site, you can do any of the following:
<LIST>(UNNUMBERED)
<LE>Select a default bootstrap command procedure
<LE>Modify system parameters for special hardware configuration needs or
special workload requirements
<LE>Perform other site-specific modifications
<ENDLIST>

```

This example produces the following output:

To create a system that more closely suits the requirements of your site, you can do any of the following:

- Select a default bootstrap command procedure
- Modify system parameters for special hardware configuration needs or special workload requirements
- Perform other site-specific modifications

## <LIST>

The following is an example of how you can specify a character other than a bullet to label each list element.

```
12 <P>To create a system that more closely suits the requirements of
your site, you can do any of the following:
<LIST>(UNNUMBERED\+)
<LE>Select a default bootstrap command procedure
<LE>Modify system parameters for special hardware configuration needs or
special workload requirements
<LE>Perform other site-specific modifications
<ENDLIST>
```

This example produces the following output:

To create a system that more closely suits the requirements of your site, you can do any of the following:

- + Select a default bootstrap command procedure
- + Modify system parameters for special hardware configuration needs or special workload requirements
- + Perform other site-specific modifications





## <LITERAL>

The following example shows how to use the <LITERAL>(literal text) tag.

- 2 <P> I've experimented with my own method for noting questionable phrases, by placing <LITERAL><NOTE\_PROBLEM> near the phrase.

This example produces the same output as the first example.

---

## <LOWERCASE>

Labels text that you want to appear as lowercase in the final output.

---

**FORMAT**      <LOWERCASE>(*text*)

---

**ARGUMENTS**    *text*  
Specifies the text to appear in lowercase.

---

**related tags**      • <UPPERCASE>

---

**DESCRIPTION**    The <LOWERCASE> tag labels text that you want to appear as lowercase in the final output. In your book, you may have a text element, such as a heading, that normally appears in uppercase. For example, in some doctypes, first-level headings use all uppercase letters. If you need to overcome the default case in one of your tags and ensure that the result in the final output appears in lowercase, use this tag.

---

**EXAMPLE**            The following example shows how to use the <LOWERCASE> tag. Assume that the doctype being used causes the <HEAD2> tag to output a heading that is in uppercase, no matter what the case of the text passed to it.

```
<HEAD2>(HeRe iS aN ExAmPlE oF <LOWERCASE>(lOwErCaSe) tExT)
```

The example produces the following output:

HERE IS AN EXAMPLE OF lowercase TEXT

The default to uppercase letters causes the heading to be uniformly uppercase, with the exception of the text used in the argument to the <LOWERCASE> tag.

## <MARK>

---

## <MARK>

Indicates the beginning of new or modified information.

---

### FORMAT

<MARK>

---

### ARGUMENTS

*None.*

---

### related tags

- <REVISION>
- <UPDATE\_RANGE>

---

### restrictions

Valid only in the context of a <REVISION> tag.

Both the <MARK> and <ENDMARK> tags must be within the code for a stacked list (using the <LIST>(stacked) tag) if you want change bars in a stacked list.

<MARK> tags within the arguments to the following tags must be terminated within the argument by the <ENDMARK> tag.

- <APPENDIX>, <CHAPTER>, <EXAMPLE>, <FCMD>, <FIGURE>, <FPARMS>, <FOOTNOTE>, <GLOSSARY>, <HEADn>, <REVISION\_INFO>, <TABLE>, and <TITLE> when used within the context of a <PART\_PAGE> tag

Using <MARK> and <ENDMARK> within the arguments to all these tags, (except for <FCMD>, <FPARMS>, <FOOTNOTE>, and <REVISION\_INFO>), create change bars in the table of contents.

- The following SOFTWARE doctype tags: <COMMAND>, <COMMAND\_SECTION>, <ROUTINE>, <ROUTINE\_SECTION>, <STATEMENT>, <STATEMENT\_SECTION>, <SUBCOMMAND>, <SUBCOMMAND\_SECTION\_HEAD>, <SDML\_TAG>, and <TAG\_SECTION>

---

### required terminator

<ENDMARK>

---

### DESCRIPTION

The <MARK> tag indicates the beginning of new or modified information. The <MARK> tag, along with the <ENDMARK> tag, must be enabled by the <REVISION> tag in order to produce vertical bars in the margin of the document (and optionally, in the table of contents—see the RESTRICTIONS section), as well as in MAIL, TERMINAL, and LINE output.

Make sure to place the <MARK> and <ENDMARK> tags next to the text they mark, without preceding or following the text with other tags. The text formatter interprets the <MARK> tag based on the last text character encountered before the <MARK> tag.

---

## EXAMPLE

In the following example, the first paragraph is the original and the second paragraph is the modified version that shows how to use the <MARK> and <ENDMARK> tags.

```
<P>
The following characters are legal in MACRO-11 source programs:
<LIST>(UNNUMBERED)
<LE>The letters A through Z. Both upper- and lowercase letters are
acceptable, although, upon input, lowercase letters are converted to
uppercase.
<LE>The digits 0 through 9.
<LE>The characters period <PARENDCHAR>(.) and dollar sign
<PARENDCHAR>($). These characters are reserved for use as Digital
Equipment Corporation system program symbols.
<ENDLIST>
```

```
<REVISION>
```

```
.
.
.
```

```
<P>
The following characters are legal in MACRO-11 source programs:
<LIST>(UNNUMBERED)
<LE>The letters A through Z. Both upper- and lowercase letters are
acceptable, although, upon input, lowercase letters are converted to
uppercase.
```

```
<MARK>
```

```
<LE>Characters in the Digital Multinational Character Set (MCS). A chart
showing the MCS is located in <REFERENCE>(mcs_app),
with a list of directives that
support the MCS.
<ENDMARK>
```

```
<LE>The digits 0 through 9.
<LE>The characters period <PARENDCHAR>(.) and dollar sign
<PARENDCHAR>($). These characters are reserved for use as Digital
Equipment Corporation system program symbols.
<ENDLIST>
```

The output for the original paragraph and the modified paragraph follows:

The following characters are legal in MACRO-11 source programs:

- The letters A through Z. Both upper- and lowercase letters are acceptable, although, upon input, lowercase letters are converted to uppercase.
- The digits 0 through 9.
- The characters period (.) and dollar sign (\$). These characters are reserved for use as Digital Equipment Corporation system program symbols.

The following characters are legal in MACRO-11 source programs:

- The letters A through Z. Both upper- and lowercase letters are acceptable, although, upon input, lowercase letters are converted to uppercase.

## <MARK>

- Characters in the Digital Multinational Character Set (MCS). A chart showing the MCS is located in Appendix A, with a list of directives that support the MCS.
- The digits 0 through 9.
- The characters period (.) and dollar sign (\$). These characters are reserved for use as Digital Equipment Corporation system program symbols.

---

## <MATH>

Labels a short mathematical expression or the beginning of an extended mathematical example.

---

### FORMAT

**<MATH>**( { *math expression*  
**DISPLAY**[ \ *symbol name* ] } )

---

### ARGUMENTS

#### ***math expression***

Specifies a mathematical expression that you want to include in the text of a sentence or paragraph.

#### ***DISPLAY***

This keyword specifies that a mathematical equation or expression that is longer than one line is to be set off from the surrounding text. If you use **DISPLAY**, you must use the **<ENDMATH>** tag to terminate the display.

#### ***symbol name***

This is an optional argument. It specifies a symbol name by which a mathematical equation that is longer than one line is to be referenced. If you specify this argument, the equation is assigned a number and the number is printed to the right of the equation.

Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

### related tags

- **<CODE\_EXAMPLE>**
- **<MATH\_CHAR>**
- **<REFERENCE>**

---

### restrictions

You cannot use mathematical examples that are longer than one line in the context of a **<TABLE>** or an **<EXAMPLE>** tag.

Invalid in the context of a **<HEADn>** tag.

Invalid in the context of a **<FOOTNOTE>** tag.

Invalid in the context of a **<FORMAT>** tag.

You cannot use tab characters and carriage returns for space in the context of a **<MATH>** tag.

The set of tags valid within a mathematical expression is limited to those listed in the tables in the following **DESCRIPTION** section.

---

### required terminator

**<ENDMATH>** —Required if you specify the **DISPLAY** argument to the **<MATH>** tag.

## <MATH>

---

### DESCRIPTION

The <MATH> tag labels a short mathematical expression or the beginning of an extended mathematical example. Within an argument to the <MATH> tag or within the bounds of the <MATH>(DISPLAY) . . . <ENDMATH> tags, you can specify simple or complex mathematical expressions, according to the rules outlined below. The context in which you enter text and tags within a mathematical expression is severely restricted.

In a mathematical expression, all formatting is controlled by the text processor. When you use the DISPLAY keyword, the output will be offset from the surrounding text, but you do not have any control over its positioning. Furthermore, blank spaces and carriage returns are ignored; the text processor assumes that all text strings are mathematical variable names and makes all decisions regarding the output formatting.

#### Parentheses

Most of the tags valid in math do not accept arguments. Parentheses, however, are frequently used in mathematical expressions. You must be careful, therefore, to use a space between the tag and the parentheses. For example:

```
<MATH>(A <minus> (B<PLUS>C))
```

**Note:** At times, it may be helpful to you to use the <OPAREN> and <CPAREN> tags in place of the opening and closing parenthesis in math equations. This way, you and the text formatter can determine what you mean to be a literal parenthesis as opposed to the opening or closing parenthesis of an argument.

#### Simple Expressions

The following tags are used to indicate simple binary operations. In the context of a mathematical expression, you can use either the tag name or the function's symbol.

<TIMES>	*	Multiplication
<PLUS>	+	Addition
<MINUS>	-	Subtraction
<DIVIDED_BY>	/	Division
<EQUALS>	=	Equality

For example, the following are equivalent:

```
<MATH>(total = A + B - C * D / E)
<MATH>(total <EQUALS> A <PLUS> B <MINUS> C <TIMES> D <DIVIDED_BY> E)
```

Both produce the following output:  $total = A + B - C * D / E$

**Note:** Using any nonalphanumeric characters (such as @, #, \$, tabs, and so on) other than those specified above in the context of math may produce unpredictable results, including errors from the text formatter.



Operators are evaluated using the normal mathematical rules for precedence of operators. You can control the evaluation by using the <GROUP> tag, much as you would use parentheses in a mathematical expression. Note the use of the <GROUP> tag in the second math expression in the following code fragment:

```
<MATH>(total = A +B-C*D<OVER>(e) )
<P>
<MATH>(total = <GROUP><GROUP>(A +B-C*D)<OVER>(e) )
```

This example produces the following output:

$$\frac{total=A+B-C*D}{e}$$

$$total = \frac{A+B-C*D}{e}$$

The effect of the <GROUP> tag is supplied by default for fractional expressions when you use the <FRACTION> tag. For example:

```
<MATH>(display)
total = <FRACTION>(A + B - C * D\E)
<ENDMATH>
```

### Variable Names

The text formatter assumes that variable names are primarily alphabetic letters or special characters represented by the <MATH\_CHAR> tag. If you need to specify any variable names in expressions that contain special characters, you must use the <VARIABLE> or <TEXT> tags. For example:

```
<MATH><variable>(event_flag) = 1)
```

If you require multiword variable names, you must use the <SP> tag to indicate spacing. For example:

```
<MATH>(display)
<VARIABLE>(SUCCESS<sp>RATE) <equals>
<group>(
<VARIABLE>(TOTAL<sp>HITS) <over> (<VARIABLE>(TOTAL<sp>HITS) <plus>
<VARIABLE>(TOTAL<sp> MISSES)
)
)
<times> 100
<ENDMATH>
```

This example produces the following output:

$$SUCCESS RATE = \frac{TOTAL HITS}{TOTAL HITS + TOTAL MISSES} * 100$$

You can provide special annotation for variable names by using the following tags:

Tag	Output
<BAR_CHAR>(x)	$\bar{x}$
<DOT_CHAR>(x)	$\dot{x}$
<HAT>(xyz)	$\widehat{xyz}$
<HAT_CHAR>(x)	$\hat{x}$

## <MATH>

Tag	Output
<OVERLINE>(var)	$\overline{var}$
<TILDE>(xyz)	$\widetilde{xyz}$
<TILDE_CHAR>(x)	$\tilde{x}$
<UNDERLINE>(var)	$\underline{var}$
<VECTOR>(x)	$\vec{x}$

### Summary of Tags

Table 1–1 summarizes the tags that are valid in mathematical expressions.

**Table 1–1 <MATH> Expressions**

Tag	Operation and Output
<AMPERSAND>	Specifies a literal ampersand character in a math expression.
<ATOP>(expression)	Stacks an expression, as in $\frac{a}{b}$ .
<BACKSLASH>	Specifies a literal backslash character in a math expression.
<CAL>(letter)	Specifies a calligraphic uppercase letter, as in $\mathcal{A}$ .
<CASES> <CASE_ROW> <ENDCASES>	Specifies a case construction. See “Matrixes and Cases” in this tag description.
<CDOTS>	Specifies centered dots, as in $\dots$ .
<CHOOSE>(expression1\expression2)	Parenthetical notation, as in $\binom{a}{b}$ .
<DDOTS>	Specifies diagonal dots, as in $\ddots$ .
<DIVIDED_BY>	Division, as in $a/b$ .
<DOTS>	Specifies horizontal dots, as in $\dots$ .
<DOT_TIMES>	Multiplication, as in $a \cdot b$ .
<EQUALIGN> <ELINE> <ENDEQUALIGN>	Starts an alignment section, so that expressions on the right and left side of a character (usually an equal sign) are balanced. See “Aligning Expressions” in this tag description.
<EQUALS>	Equality, as in $a = b$ .
<FRACTION>(numerator\denominator)	Specifies a fraction, as in $\frac{a}{b}$ .
<FUNC>(expression)	Specifies a function, as in $f(a)$ .
<GROUP>(expression)	Provides control over the order of operation. See example four in the “Examples” section in this tag description.

**Table 1–1 (Cont.) <MATH> Expressions**

Tag	Operation and Output
<INTEGRAL>	Specifies an integral, as in $\int_0^{\frac{\pi}{2}}$ . See “Operators With and Without Limits” in this tag description.
<INTEGRAL_LIMITS>	Specifies an integral that places superscripts and subscripts above and below the sign, rather than to the right. See “Operators With and Without Limits” in this tag description.
<INTEGRAL_NOLIMITS>	Specifies an integral that places superscripts and subscripts to the right of the sign, rather than above and below. See “Operators With and Without Limits” in this tag description.
<MATRIX> <MATRIX_ROW> <ENDMATRIX>	Specifies a matrix. See “Matrixes and Cases” in this tag description.
<MINUS>	Minus, as in $a - b$ .
<MOD>	Modulo, as in $n \bmod p$ .
<LBAR>	Begins an expression delimited with vertical bars. See “Delimited Expressions” in this tag description.
<LBRACE>	Begins an expression delimited with curly braces. See “Delimited Expressions” in this tag description.
<LBRACKET>	Begins an expression delimited with square brackets. See “Delimited Expressions” in this tag description.
<LCEIL>	Begins an expression delimited with ceiling characters ( $\lceil$ ). See “Delimited Expressions” in this tag description.
<LFLOOR>	Begins an expression delimited with floor characters ( $\lfloor$ ). See “Delimited Expressions” in this tag description.
<LPAREN>	Begins an expression delimited with parentheses. See “Delimited Expressions” in this tag description.
<OVER>(expression)	Division, as in $\frac{a}{b}$ .
<PI>	The pi character, $\pi$ .
<PLUS>	Plus, as in $a + b$ .
<PMOD>(expression)	Parenthetical mod, as in $a \pmod{p}$ .
<PROD>	Specifies a product, as in $\prod_0^{\frac{\pi}{2}}$ . See “Operators With and Without Limits” in this tag description.

**Table 1-1 (Cont.) <MATH> Expressions**

<b>Tag</b>	<b>Operation and Output</b>
<PROD_LIMITS>	Specifies a product that places superscripts and subscripts above and below the sign, rather than to the right. See "Operators With and Without Limits" in this tag description.
<PROD_NOLIMITS>	Specifies a product that places superscripts and subscripts to the right of the sign, rather than above and below. See "Operators With and Without Limits" in this tag description.
<RBAR>	Ends an expression delimited with vertical bars. See "Delimited Expressions" in this tag description.
<RBRACE>	Ends an expression delimited with curly braces. See "Delimited Expressions" in this tag description.
<RBRACKET>	Ends an expression delimited with square brackets. See "Delimited Expressions" in this tag description.
<RCEIL>	Ends an expression delimited with ceiling characters ( $\lceil$ ). See "Delimited Expressions" in this tag description.
<RFLOOR>	Ends an expression delimited with floor characters ( $\lfloor$ ). See "Delimited Expressions" in this tag description.
<RPAREN>	Ends an expression delimited with parentheses. See "Delimited Expressions" in this tag description.
<SP>	Provides space in a variable name or expression.
<SQRT>(expression)	The square root, as in $\sqrt{a}$ .
<SUBSCRIPT>(expression)	Subscription, as in $a_b$ .
<SUM>	Specifies summation, as in $a \sum b$ . See "Operators With and Without Limits" in this tag description.
<SUM_LIMITS>	Specifies summation that places superscripts and subscripts above and below the sign, rather than to the right. See "Operators With and Without Limits" in this tag description.
<SUM_NOLIMITS>	Specifies summation that places superscripts and subscripts to the right of the sign, rather than above and below. See "Operators With and Without Limits" in this tag description.

**Table 1–1 (Cont.) <MATH> Expressions**

Tag	Operation and Output
<SUPERSCRIPT>(expression)	Exponentiation, as in $a^b$ . See “Operators With and Without Limits” in this tag description.
<TEXT>	Specifies text in a math expression.
<TIMES>	Multiplication, as in $a * b$ .
<TO>	Indicates progression, as in $1 \rightarrow 10$ .
<VARIABLE>	Specifies a variable name that contains nonalphanumeric characters.
<VDOTS>	Specifies vertical dots, as in $\therefore$ .
<VECTOR>(var)	A vector, as in $\vec{x}$ .
<X_TIMES>	Multiplication, as in $a \times b$ .

**Operators With and Without Limits**

When you specify <SUBSCRIPT> and <SUPERSCRIPT> tags, the subscript or superscript applies to the immediately preceding variable or expression. You can specify these tags in any order, as shown in the following example:

```
<MATH>(a<subscript>(2)<superscript>(n-1))
```

This example produces the following output:  $a_2^{n-1}$

In extended mathematical expressions, the <SUPERSCRIPT> and <SUBSCRIPT> tags produce differing results in conjunction with the <INTEGRAL>, <PROD>, and <SUM> tags and their complementary tags—<INTEGRAL\_LIMITS> and <INTEGRAL\_NOLIMITS>, <PROD\_LIMITS> and <PROD\_NOLIMITS>, and <SUM\_LIMITS> and <SUM\_NOLIMITS>.

The <INTEGRAL>, <PROD>, and <SUM> tags place their subscripts and superscripts with regard to whether the tags are used in math text mode (used as an argument to the <MATH> tag) or in math display mode (used between the <MATH>(DISPLAY) and <ENDMATH> tags).

In math text mode, the <INTEGRAL>, <PROD>, and <SUM> tags all place the superscript and subscript to the right. Code these tags as follows:

```
<MATH>( <integral><subscript>(n=1)<superscript>( <pi><over>(2) ) <sp> )
<MATH>( <prod><subscript>(n=1)<superscript>( <pi><over>(2) ) <sp> )
<math>( <sum><subscript>(n=1)<superscript>( <pi><over>(2) ) )
```

This example produces the following output:  $\int_{n=1}^{\frac{\pi}{2}} \prod_{n=1}^{\frac{\pi}{2}} \sum_{n=1}^{\frac{\pi}{2}}$

In math display mode, the <PROD> and <SUM> tags place the superscript and subscript above and below their respective signs; however, the <INTEGRAL> tag places superscripts and subscripts to the right. Code these tags as follows:

```
<MATH>(display)
<integral><subscript>(n=1)<superscript>( <pi><over>(2) ) <sp>
  <prod><subscript>(n=1)<superscript>( <pi><over>(2) ) <sp>
  <sum><subscript>(n=1)<superscript>( <pi><over>(2) )
<ENDMATH>
```

## <MATH>

This example produces the following output:

$$\int_{n=1}^{\frac{\pi}{2}} \prod_{n=1}^{\frac{\pi}{2}} \sum_{n=1}^{\frac{\pi}{2}}$$

The <INTEGRAL\_NOLIMITS>, <PROD\_NOLIMITS>, and <SUM\_NOLIMITS> tags always place the subscripts and superscripts to the right of the signs, regardless of the math mode. Code these tags as follows:

```
<MATH>(display)
<integral_nolimits><subscript>(n=1)<superscript>( <pi><over>(2) ) <sp>
  <prod_nolimits><subscript>(n=1)<superscript>( <pi><over>(2) ) <sp>
    <sum_nolimits><subscript>(n=1)<superscript>( <pi><over>(2) )
<ENDMATH>
```

This example produces the following output:

$$\int_{n=1}^{\frac{\pi}{2}} \prod_{n=1}^{\frac{\pi}{2}} \sum_{n=1}^{\frac{\pi}{2}}$$

The <INTEGRAL\_LIMITS>, <PROD\_LIMITS>, and <SUM\_LIMITS> tags always place the subscripts and superscripts above and below the signs, regardless of the math mode. Code these tags as follows:

```
<MATH>(display)
<integral_limits><subscript>(n=1)<superscript>( <pi><over>(2) ) <sp>
  <prod_limits><subscript>(n=1)<superscript>( <pi><over>(2) ) <sp>
    <sum_limits><subscript>(n=1)<superscript>( <pi><over>(2) )
<ENDMATH>
```

This example produces the following output:

$$\int_{n=1}^{\frac{\pi}{2}} \prod_{n=1}^{\frac{\pi}{2}} \sum_{n=1}^{\frac{\pi}{2}}$$

### Mathematical Functions

In addition to the operations and special functions listed in Table 1-1, you can specify mathematical functions using any of the tags listed in Table 1-2. These tags all let you specify the tag with or without an argument. If you specify an argument, place it in parentheses following the function name. For example:

```
<MATH>(<SIN>(d))
```

This produces:  $\sin(d)$ .

**Table 1–2 Tags for Mathematical Functions**

<b>Tag</b>	<b>Function</b>
<ARCCOS>	arccos
<ARCSIN>	arcsin
<ARCTAN>	arctan
<ARG>	arg
<COS>	cos
<COSH>	cosh
<COT>	cot
<COTH>	coth
<CSC>	csc
<DEG>	deg
<DET>	det
<DIM>	dim
<EXP>	exp
<GCD>	gcd
<HOM>	hom
<INF>	inf
<KER>	ker
<LG>	lg
<LIM>	lim
<LIMINF>	lim inf
<LIMSUP>	lim sup
<LN>	ln
<LOG>	log
<MAX>	max
<MIN>	min
<MOD>	mod
<PMOD>	(mod $n$ ) <sup>1</sup>
<PR>	Pr
<SEC>	sec
<SIN>	sin
<SINH>	sinh
<SUP>	sup
<TAN>	tan
<TANH>	tanh

<sup>1</sup>Using the <PMOD> tag adds a space before the output of the tag, which is why the output of the tag is not aligned in the "Function" column of this table.

An example of using some of the tags from the previous table follows:

## <MATH>

```
<LIST> (UNNUMBERED)
<le><math>(
<sin>2<math_char>(theta)
  <equals>2<sin><math_char>(theta)<cos><math_char>(theta)
)
<le><math>(
O(n <log>n <log><log>n)
)
<le><math>(
<pr>(X >x) = <exp>(-x/<math_char>(mu))
)
<le><math>(
  <max><subscript>(1<math_char>(geq)n<math_char>(geq)m)
  <log><subscript>(2)P<subscript>(n)
)
<le><math>(
<lim><subscript>(x<to>0)<group>(
  <sin>x<over>(x)
<equals>1
)
<endmath>
<ENDLIST>
```

These examples produce the following output:

- $\sin 2\theta = 2 \sin \theta \cos \theta$
- $O(n \log n \log \log n)$
- $\Pr(X > x) = \exp(-x/\mu)$
- $\max_{1 \geq n \geq m} \log_2 P_n$
- $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$

### Aligning Expressions

To align math expressions on the right and left side of an alignment point (usually an equal sign) so that the expressions are balanced, use the following tags:

- <EQUALIGN>—begins an alignment section and enables the <ELINE> tag.
- <ELINE>—takes an argument representing the left and right side of the equation. The first character in the second argument is the character upon which the alignment is based.
- <ENDEQUALIGN>—terminates the alignment section.

For example:

```
<math>
<EQUALIGN>
<eline>(H(f)\=
<GROUP>( <LBAR><GROUP>( <FRACTION>(E(x)\E(0))) <RBAR>
<SUBSCRIPT>(E(0)=e<SUPERSCRIP>(j<MATH_CHAR>(OMEGA)t)))
<Eline>( \= <GROUP>(e<SUPERSCRIP>(-<MATH_CHAR>(GAMMA)x)))
<endequalign>
<ENDMATH>
```



This example produces the following output:

$$H(f) = \left. \frac{E(x)}{E(0)} \right|_{E(0)=e^{j\omega t}}$$

$$= e^{-\gamma x}$$

### Delimited Expressions

To produce delimited expressions in mathematics, you must use one of the following pairs of tags:

- **<LBAR>** and **<RBAR>**—for opening and closing vertical bars
- **<LBRACE>** and **<RBRACE>**—for opening and closing curly braces
- **<LBRACKET>** and **<RBRACKET>**—for opening and closing square brackets
- **<LCEIL>** and **<RCEIL>**—for opening and closing ceiling delimiters
- **<LFLOOR>** and **<RFLOOR>**—for opening and closing floor delimiters
- **<LPAREN>** and **<RPAREN>**—for opening and closing parentheses

The text formatter automatically assumes that text within these pairs is to be grouped, and it sizes the delimiters automatically.

For example:

```
<MATH>(display)
<group>(
  C<subscript>(dg)
  ) =
  <fraction>( <math_char>(partial)Q<subscript>(d) \
              <math_char>(partial)V<subscript>(g) )
=
-C<subscript>(ox)
  [0.5
    +
    <lparen>f<subscript>(0) DVG -
      <group>(2f<subscript>(0) V<subscript>(com) <over>(f<subscript>(1))
      )
    <rparen>
    <group>(1 <over>(f<subscript>(1) <superscript>(2)))
  ]
<ENDMATH>
```

This example produces the following output:

$$C_{dg} = \frac{\partial Q_d}{\partial V_g} = -C_{ox} \left[ 0.5 + \left( f_0 DVG - \frac{2f_0 V_{com}}{f_1} \right) \frac{1}{f_1^2} \right]$$

## <MATH>

### Matrixes and Cases

You can construct matrixes and case constructs using the formats provided with <MATRIX> and <CASES> tags.

The <MATRIX> tag has the following format:

$$\langle \text{MATRIX} \rangle \left( \begin{array}{c} \text{BRACES} \\ \text{BRACKETS} \\ \text{VERTICAL\_RULE} \end{array} \right)$$

The keywords BRACES, BRACKETS, and VERTICAL\_RULE override the default matrix delimiter, parentheses.

When you construct a matrix, you must specify each row in the matrix using the <MATRIX\_ROW> tag. You can specify a maximum of nine columns for the row. Terminate the matrix with the <ENDMATRIX> tag. For example:

```
<matrix> (brackets)
<matrix_row> (A)
<matrix_row> (B)
<matrix_row> (C)
<matrix_row> (D)
<endmatrix>
```

This simple, one-column matrix produces the following output:

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}$$

A more complex example shows how to code a multicolumn matrix:

```
<math> (display)
<det> <matrix> (vertical_rule)
  <matrix_row> (c<subscript>(0) \c<subscript>(1) \
    c<subscript>(2) \<dots> \c<subscript>(n))
  <matrix_row> (c<subscript>(1) \c<subscript>(2) \
    c<subscript>(3) \<dots> \c<subscript>(n+1))
  <matrix_row> (c<subscript>(2) \c<subscript>(3) \
    c<subscript>(4) \<dots> \c<subscript>(n+2))
  <matrix_row> (<vdots> \<vdots> \<vdots> \<vdots>)
  <matrix_row> (c<subscript>(n) \c<subscript>(n+1) \
    c<subscript>(n+2) \<dots> \c<subscript>(2n))
  <endmatrix> > 0.
<endmath>
```

This produces the following output:

$$\det \begin{vmatrix} c_0 & c_1 & c_2 & \dots & c_n \\ c_1 & c_2 & c_3 & \dots & c_{n+1} \\ c_2 & c_3 & c_4 & \dots & c_{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_n & c_{n+1} & c_{n+2} & \dots & c_{2n} \end{vmatrix} > 0.$$

The following example shows how to code a multicolumn matrix with fractions:

```
<math>(display)
<matrix>(brackets)
<matrix_row>(4<over>(4<minus>6)\2<over>(6<minus>4))
<matrix_row>(3<over>(6<minus>4)\1<over>(4<minus>6))
<endmatrix> =
<matrix>(brackets)
<matrix_row>(4<over>( <minus>2)\2<over>(2))
<matrix_row>(3<over>(2)\1<over>( <minus>2))
<endmatrix> =
<matrix>(brackets)
<matrix_row>( <minus>2<sp>1)
<matrix_row>(1.5\ <minus>.5)
<endmatrix>
<endmath>
```

This produces the following output:

$$\begin{bmatrix} \frac{4}{4-6} & \frac{2}{6-4} \\ \frac{3}{6-4} & \frac{1}{4-6} \end{bmatrix} = \begin{bmatrix} \frac{4}{-2} & \frac{2}{2} \\ \frac{3}{2} & \frac{1}{-2} \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}$$

The **<CASES>** tag is similar to the **<MATRIX>** tag, but it produces only a large left-hand brace; there is no closing delimiter. The following example shows how to use the **<CASES>** tag and its related tags:

```
<MATH>(display)
<cases>
<case_row>(1/3\if\0> x\;)
<case_row>(2/3\if\3< x\;)
<case_row>(0\elsewhere.)
<endcases>
<ENDMATH>
```

This example produces the following output:

$$\begin{cases} 1/3 & \text{if } 0 > x; \\ 2/3 & \text{if } 3 < x; \\ 0 & \text{elsewhere.} \end{cases}$$

---

## EXAMPLES

The following example illustrates a simple mathematical expression used in the context of a sentence.

- i** The area of a circle is calculated using the formula **<MATH>(a<equals><pi>r<superscript>(2))**.

This example produces the following output:

The circumference of a circle is calculated using the formula  $c = \pi r^2$ .

The following example shows how to use the **<MATH>(DISPLAY)** tag. Note that the parenthetical expression following the first **<MINUS>** tag must have a space in front of it; otherwise, the expression will be interpreted as an argument to the **<MINUS>** tag.

## <MATH>

2 `<MATH>(DISPLAY)`  
`vsize <equals> psize <minus>(<minus>topglue) <minus> topdepth <minus> footerglue`  
`<ENDMATH>`

This example produces the following output:

$$vsize = psize - (-topglue) - topdepth - footerglue$$

The following example shows how to use the <MATH> tag in a list.

3 `<LIST>(NUMBERED)`  
`<le><MATH>(1<over>(2))`  
`<le><MATH>(n+1<over>(3))`  
`<le><MATH>(<choose>(N+1)\3)`  
`<le><MATH>(<sum><subscript>(n=1) <superscript>(3) Z<subscript>(n) <superscript>(2))`  
`<ENDLIST>`

This example produces the following output:

- 1  $\frac{1}{2}$
- 2  $\frac{n+1}{3}$
- 3  $\binom{N+1}{3}$
- 4  $\sum_{n=1}^3 Z_n^2$

The following example illustrates how to use the <GROUP> tag to indicate the order of operation.

4 `<MATH>(DISPLAY)`  
`<GROUP>{`  
`a<over>(x<plus>y<superscript>(3))`  
`}`  
`<EQUALS> <SQRT>(<times><pi>)`  
`<ENDMATH>`

This example produces the following output:

$$\frac{a}{x + y^3} = \sqrt{*}\pi$$

Note what the output would be if you did not use the <GROUP> tag:

$$\frac{a}{x + y^3} = \sqrt{*}\pi$$

In the following example, the characters representing the mathematical operations are used directly.

5 `<MATH>(total = A + B - C * D / E)`

This example produces the following output:

$$total = A + B - C * D / E.$$

Note that this is equivalent to the following:

`<MATH>(total <EQUALS> A <PLUS> B <MINUS> C <TIMES> D <DIVIDED_BY> E)`

The following example shows how to use the <MATH> tag to generate true minus signs in your SDML files.

6 <MATH>(--)  
This begins a comment line.

This example produces the following output:

--This begins a comment line.

The following example shows how to use the <REFERENCE> tag to refer to an equation.

7 <MATH>(DISPLAY\math\_equation)  
math = numbers + letters  
<ENDMATH>  
<p>As <REFERENCE>(math\_equation) shows, the relationship between letters and math must include numbers.

This example produces the following output:

$$math = numbers + letters \tag{1-1}$$

As (1-1) shows, the relationship between letters and math must include numbers.

## <MATH\_CHAR>

---

## <MATH\_CHAR>

Creates a special mathematical symbol.

---

**FORMAT**      **<MATH\_CHAR>** (*keyword*)

---

**ARGUMENTS**    ***keyword***  
Indicates the special symbol you want to access. The keywords, and the symbols they produce, are listed in Table 1–3.

---

**related tags**

- <MATH>
- <MCS>
- <SPECIAL\_CHAR>

---

**restrictions**    Invalid in headings, that is, within an argument to a <HEADn> tag.  
The characters produced using the <MATH\_CHAR> tag are sized only for normal text sizes. Therefore, these characters do not produce good visual results in text that requires a different font size, like header levels, text in the OVERHEADS doctype, and so on.

---

**DESCRIPTION**    The <MATH\_CHAR> tag creates a special mathematical symbol. Table 1–3 summarizes the keywords and special symbols you can access with the <MATH\_CHAR> tag.

**Table 1–3 <MATH\_CHAR> Keywords and Symbols**

Keyword	Symbol
<b>The Greek Letters</b>	
ALPHA	$\alpha$
BETA	$\beta$
CHI	$\chi$
DELTA	$\delta$
EPSILON	$\epsilon$
ETA	$\eta$
GAMMA	$\gamma$
IOTA	$\iota$
KAPPA	$\kappa$
LAMBDA	$\lambda$
MU	$\mu$

**Table 1–3 (Cont.) <MATH\_CHAR> Keywords and Symbols**

<b>Keyword</b>	<b>Symbol</b>
<b>The Greek Letters</b>	
NU	$\nu$
OMEGA	$\omega$
OMICRON	$o$
PHI	$\phi$
PI	$\pi$
PSI	$\psi$
RHO	$\rho$
SIGMA	$\sigma$
TAU	$\tau$
THETA	$\theta$
UPSILON	$\upsilon$
VAREPSILON	$\varepsilon$
VARPHI	$\varphi$
VARPI	$\varpi$
VARRHO	$\varrho$
VARSIGMA	$\varsigma$
VARTHETA	$\vartheta$
XI	$\xi$
ZETA	$\zeta$
<b>The Uppercase Greek Letters</b>	
UPPERCASE_ALPHA	A
UPPERCASE_BETA	B
UPPERCASE_CHI	X
UPPERCASE_DELTA	$\Delta$
UPPERCASE_EPSILON	E
UPPERCASE_ETA	H
UPPERCASE_GAMMA	$\Gamma$
UPPERCASE_IOTA	I
UPPERCASE_KAPPA	K
UPPERCASE_LAMBDA	$\Lambda$
UPPERCASE_MU	M
UPPERCASE_NU	N
UPPERCASE_OMEGA	$\Omega$
UPPERCASE_OMICRON	O
UPPERCASE_PHI	$\Phi$

## <MATH\_CHAR>

Table 1-3 (Cont.) <MATH\_CHAR> Keywords and Symbols

Keyword	Symbol
<b>The Uppercase Greek Letters</b>	
UPPERCASE_PI	$\Pi$
UPPERCASE_PSI	$\Psi$
UPPERCASE_RHO	$\rho$
UPPERCASE_SIGMA	$\Sigma$
UPPERCASE_TAU	$\tau$
UPPERCASE_THETA	$\Theta$
UPPERCASE_UPSILON	$\Upsilon$
UPPERCASE_XI	$\Xi$
UPPERCASE_ZETA	$\text{Z}$
<b>The Ordinal Operators</b>	
ALEPH	$\aleph$
ANGLE	$\angle$
BACKSLASH	$\backslash$
BOT	$\perp$
CLUBSUIT	$\clubsuit$
DIAMONDSUIT	$\diamond$
DOUBLE_VERT	$\parallel$
ELL	$\ell$
EMPTYSET	$\emptyset$
EXISTS	$\exists$
FLAT	$\flat$
FORALL	$\forall$
HBAR	$\hbar$
HEARTSUIT	$\heartsuit$
IM	$\Im$
IMATH	$\imath$
INFTY	$\infty$
JMATH	$\jmath$
NABLA	$\nabla$
NATURAL	$\natural$
NEG	$\neg$
PARTIAL	$\partial$
PRIME	$\prime$
RE	$\Re$
SHARP	$\sharp$



Table 1-3 (Cont.) <MATH\_CHAR> Keywords and Symbols

Keyword	Symbol
<b>The Ordinal Operators</b>	
SPADESUIT	♠
SURD	√
TOP	⤴
TRIANGLE	△
WP	∅
<b>Binary Operators</b>	
AMALG	∩
AST	*
BIGCIRC	○
BIGTRIANGLEDOWN	▽
BIGTRIANGLEUP	△
BULLET	•
CAP	∩
CDOT	·
CIRC	◦
CUP	∪
DAGGER	†
DDAGGER	‡
DIAMOND	◇
DIV	÷
MP	≠
ODOT	⊙
OMINUS	⊖
OPLUS	⊕
OSLASH	⊘
OTIMES	⊗

# <MATH\_CHAR>

Table 1-3 (Cont.) <MATH\_CHAR> Keywords and Symbols

Keyword	Symbol
<b>Binary Operators</b>	
PM	$\pm$
SETMINUS	$\setminus$
SQCAP	$\sqcap$
SQCUP	$\sqcup$
STAR	$*$
TIMES	$\times$
TRIANGLELEFT	$\triangleleft$
TRIanglerIGHT	$\triangleright$
UPLUS	$\oplus$
VEE	$\vee$
WEDGE	$\wedge$
WR	$\wr$
<b>Relational Operators</b>	
APPROX	$\approx$
ASYMP	$\asymp$
BOWTIE	$\bowtie$
CONG	$\cong$
DASHV	$\dashv$
DOTEQ	$\doteq$
EQUIV	$\equiv$
FROWN	$\frown$
GEQ	$\geq$
GG	$\gg$
IN	$\in$
LL	$\ll$
LEQ	$\leq$
MID	$\mid$
MODELS	$\models$
NI	$\ni$
NOT_APPROX	$\not\approx$
NOT_ASYMP	$\not\asymp$
NOT_CONG	$\not\cong$
NOT_EQUIV	$\not\equiv$
NOT_EQ	$\neq$
NOT_GEQ	$\ngeq$

Table 1-3 (Cont.) <MATH\_CHAR> Keywords and Symbols

Keyword	Symbol
<b>Relational Operators</b>	
NOT_GT	$\nrightarrow$
NOT_LEQ	$\nlessdot$
NOT_LT	$\nless$
NOT_PRECEQ	$\nlessdot$
NOT_PREC	$\nless$
NOT_SIMEQ	$\napprox$
NOT_SIM	$\napprox$
NOT_SQSUBSETEQ	$\nsubseteq$
NOT_SQSUPSETEQ	$\nsupseteq$
NOT_SUBSET	$\not\subset$
NOTSUBSETEQ	$\not\subseteq$
NOT_SUCCEQ	$\nlessdot$
NOT_SUCC	$\nless$
NOT_SUPSETEQ	$\nsupseteq$
NOT_SUPSET	$\not\supseteq$
PARALLEL	$\parallel$
PERP	$\perp$
PRECEQ	$\lessdot$
PREC	$\less$
PROPTO	$\propto$
SIMEQ	$\approx$
SIM	$\sim$
SMILE	$\smile$
SQSUBSETEQ	$\subseteq$
SQSUPSETEQ	$\supseteq$
SUBSETEQ	$\subset$
SUBSET	$\subset$
SUCCEQ	$\lessdot$
SUCC	$\less$
SUPSETEQ	$\supseteq$
SUPSET	$\supseteq$
VDASH	$\vdash$

# <MATH\_CHAR>

Table 1–3 (Cont.) <MATH\_CHAR> Keywords and Symbols

Keyword	Symbol
<b>Arrows</b>	
DOUBLE_DOWNARROW	$\Downarrow$
DOUBLE_LEFTARROW	$\Leftrightarrow$
DOUBLE_LEFTRIGHTARROW	$\Leftrightarrow$
DOUBLE_LONGLEFTARROW	$\Leftrightarrow$
DOUBLE_LONGLEFTRIGHTARROW	$\Leftrightarrow$
DOUBLE_LONGRIGHTARROW	$\Rightarrow$
DOUBLE_RIGHTARROW	$\Rightarrow$
DOUBLE_UPARROW	$\Uparrow$
DOUBLE_UPDOWNARROW	$\Updownarrow$
DOWNARROW	$\downarrow$
HOOKLEFTARROW	$\hookleftarrow$
HOOKRIGHTARROW	$\hookrightarrow$
LEFTARROW	$\leftarrow$
LEFTHARPOONDOWN	$\lrcorner$
LEFTHARPOONUP	$\lrcorner$
LEFTRIGHTARROW	$\leftrightarrow$
LONGLEFTARROW	$\longleftarrow$
LONGLEFTRIGHTARROW	$\longleftrightarrow$
LONGMAPSTO	$\mapsto$
LONGRIGHTARROW	$\longrightarrow$
MAPSTO	$\mapsto$
NEARROW	$\nearrow$
NWARROW	$\nwarrow$
RIGHTARROW	$\rightarrow$
RIGHTHARPOONDOWN	$\rharpoonright$
RIGHTHARPOONUP	$\rharpoonright$
RIGHTLEFTHARPOONS	$\rightleftharpoons$
SEARROW	$\searrow$
SWARROW	$\swarrow$
UPARROW	$\uparrow$
UPDOWNARROW	$\updownarrow$

Table 1-3 (Cont.) <MATH\_CHAR> Keywords and Symbols

Keyword	Symbol
<b>Delimiters</b>	
LANGLE	<
RANGLE	>
LBRACE	{
RBRACE	}
LBRACK	[
RBRACK	]
LCEIL	⌈
RCEIL	⌋
LFLOOR	⌊
RFLOOR	⌋

**EXAMPLE**

The following example lists math expressions using the <MATH\_CHAR> tag.

```

<LIST> (SIMPLE)
<le><MATH> (x=y>z)
<le><MATH> (x:=y)
<le><MATH> (x<math_char>(leq)y<math_char>(not_eq)z)
<le><MATH> (x<math_char>(sim)y<math_char>(simeq)z)
<le><MATH> (x<math_char>(equiv)y<math_char>(not_equiv)z)
<le><MATH> (x<math_char>(subset)y<math_char>(subseteq)z)
<ENDLIST>

```

This example produces the following output:

$$\begin{aligned}
&x = y > z \\
&x := y \\
&x \leq y \neq z \\
&x \sim y \simeq z \\
&x \equiv y \neq z \\
&x \subset y \subseteq z
\end{aligned}$$

<MCS>

---

<MCS>

Labels a character in the Digital Multinational Character Set.

---

**FORMAT**      <MCS>(character)

---

**ARGUMENTS**    *character*

Specifies the character you want. The character may be any one of the following:

---

Argument	Character
spanish_inverted_exclamation	¡
cents	¢
british_pound	£
japanese_yen	¥
section_sign	§
general_currency	¤
copyright	©
feminine_ordinal	ª
double_open_angle_brackets	«
degree	°
plus_or_minus	±
superscript2	²
superscript3	³
micro	µ
pilcrow	¶
raised_period	·
superscript1	¹
masculine_ordinal	º
double_close_angle_brackets	»
one_fourth	¼
one_half	½
spanish_inverted_question	¿
cap_a_grave	À
cap_a_acute	Á
cap_a_circumflex	Â
cap_a_tilde	Ã
cap_a_umlaut	Ä
cap_a_ring	Å

Argument	Character
cap_ae	Æ
cap_c_cedilla	Ç
cap_e_grave	È
cap_e_acute	É
cap_e_circumflex	Ê
cap_e_umlaut	Ë
cap_i_grave	Ì
cap_i_acute	Í
cap_i_circumflex	Î
cap_i_umlaut	Ï
cap_n_tilde	Ñ
cap_o_grave	Ò
cap_o_acute	Ó
cap_o_circumflex	Ô
cap_o_tilde	Õ
cap_o_umlaut	Ö
cap_oe	Œ
cap_o_slash	Ø
cap_u_grave	Ù
cap_u_acute	Ú
cap_u_circumflex	Û
cap_u_umlaut	Ü
cap_y_umlaut	ÿ
german_ss	ß
small_a_grave	à
small_a_acute	á
small_a_circumflex	â
small_a_tilde	ã
small_a_umlaut	ä
small_a_ring	å
small_ae	æ
small_c_cedilla	ç
small_e_grave	è
small_e_acute	é
small_e_circumflex	ê
small_e_umlaut	ë
small_i_grave	ì
small_i_acute	í
small_i_circumflex	î

## <MCS>

Argument	Character
small_i_umlaut	ï
small_n_tilde	ñ
small_o_grave	ò
small_o_acute	ó
small_o_circumflex	ô
small_o_tilde	õ
small_o_umlaut	ö
small_oe	œ
small_o_slash	ø
small_u_grave	ù
small_u_acute	ú
small_u_circumflex	û
small_u_umlaut	ü
small_y_umlaut	ÿ

### related tags

- <MATH\_CHAR>
- <SPECIAL\_CHAR>

### restrictions

Invalid in the context of a <MATH> tag.

## DESCRIPTION

The <MCS> tag labels a character in the Digital Multinational Character Set. Use this tag when the terminal you are using does not accept or display these special characters.

## EXAMPLE

The following example shows how to use the <MCS> tag.

<P>The Digital Multinational Character Set includes the currency sign for the Japanese yen (<MCS>(japanese\_yen)).

This example produces the following output:

The Digital Multinational Character Set includes the currency sign for the Japanese yen (¥).



---

## <NESTED\_TABLE\_BREAK>

Marks a place where a **nested table** may be broken across pages.

---

**FORMAT**            <NESTED\_TABLE\_BREAK>

---

**ARGUMENTS**        *None.*

---

**related tags**

- <TABLE>
- <TABLE\_FILE>

---

**restrictions**

Valid only within a nested table.

Invalid between paragraphs in a single table row.

Breakpoints for long tables nested inside multipage tables have the following limitations:

- The first level table must not use the **KEEP** argument in the <TABLE\_ATTRIBUTES> tag.
- The first level table must only have two or three columns, which you set up using the <TABLE\_SETUP> tag.

---

**DESCRIPTION**

The <NESTED\_TABLE\_BREAK> tag marks a place where a nested table may be broken across pages. A nested table is a table that is coded within another table. You create a nested table by placing the appropriate table tags within an argument to the <TABLE\_ROW> tag of the outer table. Place the <NESTED\_TABLE\_BREAK> tag between table rows of a nested table. This marks a place that the nested table may be broken across pages. This tag has no effect for Bookreader output.

If you place a <NESTED\_TABLE\_BREAK> tag in a nested table that contains headings, the headings are not repeated on subsequent pages. Also, a rule may be output at the point that you include the <NESTED\_TABLE\_BREAK> tag, thereby appearing to end the nested table that continues on the next page.

---

**EXAMPLE**            The following example shows how to use the <NESTED\_TABLE\_BREAK> tag.

## <NESTED\_TABLE\_BREAK>

```
<TABLE>(Table Caption\tab_log_name)
<TABLE_SETUP>(2\10)
<TABLE_HEADS>(First Head\Second Head)
<TABLE_ROW>(Item Here\Definition for Item here.)
<TABLE_ROW>(Item Here\Definition that tells about nested table.
    <COMMENT>(*nested table begins*)
    <TABLE>
    <TABLE_SETUP>(2\15)
    <TABLE_ROW>(Item\Text of item description here)
    <TABLE_ROW>(Item\Text of item description here)
    .
    .
    <TABLE_ROW>(Item\Text of item description here)
    <TABLE_ROW>(Item\Text of item description here)
    <NESTED_TABLE_BREAK>
    <TABLE_ROW>(Item\Text of item description here)
    .
    .
    <TABLE_ROW>(Item\Text of item description here)
    <NESTED_TABLE_BREAK>
    <TABLE_ROW>(Item\Text of item description here)
    .
    .
    <TABLE_ROW>(Item\Text of item description here)
    .
    .
    <TABLE_ROW>(Item Here\Definition for Item here.)
    <TABLE_ROW>(Item Here\Definition for Item here.)
    <ENDTABLE>
    <COMMENT>(*nested table ends*)
)
<TABLE_ROW>(Item Here\Definition for Item here.)
<TABLE_ROW>(Item Here\Definition for Item here.)
<TABLE_ROW>(Item Here\Definition for Item here.)
<ENDTABLE>
```

---

## <NEWTERM>

Labels a term introduced into the text for the first time. The term is italicized in output.

---

**FORMAT**      <NEWTERM> (*term*)

---

**ARGUMENTS**    *term*  
Specifies the new term you just introduced.

---

**related tags**    • <KEYWORD>  
                  • <VARIABLE>

---

**restrictions**    Invalid in the context of a <MATH> tag.

---

**DESCRIPTION**    The <NEWTERM> tag labels a term introduced into the text for the first time. The term is italicized in output.

---

**EXAMPLE**        The following example shows how to use the <NEWTERM> tag.

<P>To begin a session at the terminal, you must first <NEWTERM>(log in).  
Logging in consists of identifying yourself as  
an authorized user.

This example produces the following output:

To begin a session at the terminal, you must first *log in*. Logging in  
consists of identifying yourself as an authorized user.



---

## <OPAREN>

Supplies an opening parenthesis anywhere in your file. You must use this tag, however, to specify an opening parenthesis within an argument to a tag, so that the opening parenthesis is not interpreted as the opening parenthesis of the tag's argument.

---

**FORMAT**      <OPAREN>

---

**ARGUMENTS**      *None.*

---

**related tags**

- The following tags label other characters that you must tag when they occur in an argument to a tag:

- <AMPERSAND>
  - <BACKSLASH>
  - <CPAREN>—required only if there is no opening parenthesis
  - <VBAR>

---

**DESCRIPTION**

The <OPAREN> tag supplies an opening parenthesis anywhere in your file. You must use this tag, however, to specify a opening parenthesis within an argument to a tag, so that the opening parenthesis is not interpreted as the opening parenthesis of the tag's argument.

Using the tag, then, is only beneficial (in terms of keystrokes and control of the output) as an unmatched opening parenthesis in an argument to a tag. An unmatched parenthesis in an argument can cause errors when processed, because parentheses are used to determine the beginning and ending of an argument list.

---

**EXAMPLE**

The following example shows how to use the <OPAREN> tag.

```
<SUBHEAD1>(Using an Opening Parenthesis <PARENDCHEAR>  
(<OPAREN>) in an Argument to a Tag\oparen)
```

This example produces the following output:

**Using an Opening Parenthesis (()) in an Argument to a Tag**

## <ORDER\_NUMBER>

---

## <ORDER\_NUMBER>

Labels the order number or part number that appears on the title page of a book.

---

**FORMAT**      <ORDER\_NUMBER>(*number*)

---

**ARGUMENTS**    *number*  
Specifies the order number for the book.

---

**restrictions**    Valid only in the context of a <TITLE\_PAGE> tag.

---

**DESCRIPTION**    The <ORDER\_NUMBER> tag labels the order number or part number that appears on the title page of a book. This tag has no effect for Bookreader output.

---

**EXAMPLE**        See the example in the discussion of the <FRONT\_MATTER> tag.

---

<P>

Marks the beginning of a new paragraph.

---

**FORMAT**            <P>

---

**ARGUMENTS**        *None.*

---

**related tags**

- <CENTER\_LINE>
- <CP>
- <LINE>
- <RIGHT\_LINE>

---

**DESCRIPTION**

The <P> tag marks a new paragraph. An internal counter keeps track of the context in which the new paragraph begins. This allows you to freely use <P> tags inside lists and in other contexts where you want to start a new paragraph, while maintaining the same logical level of discussion.

A <P> tag is expected after every heading. If you do not immediately start a new paragraph after a heading, you must label the beginning of another text element, such as a list.

A <P> tag also specifies that the text that follows it is an **online chunk**. If you code an index entry within the paragraph, when you **click on that** entry in the Bookreader's index, the paragraph in which you coded the entry appears.

---

**EXAMPLES**            The following example shows how to use the <P> tag.

**1**    <P>  
Here is a sentence or two in a paragraph. The following paragraph will show the <TAG>(P) tag at work separating paragraphs.  
<P>Here is the second paragraph so that you can observe the relationship it has to the first.

This example produces the following output:

Here is a sentence or two in a paragraph. The following paragraph will show the <P> tag at work separating paragraphs.

Here is the second paragraph so that you can observe the relationship it has to the first.

**<P>**

The following example shows how to use a <P> tag within a list.

```
2 <list>(unnumbered)
  <le>Oranges
  <le>Apples
  <p>Note that there are several types of apples<hellipsis>
  <le>Bananas
<endlist>
```

This example produces the following output:

- Oranges
- Apples
- Note that there are several types of apples ...
- Bananas



---

## <PAGE>

Breaks a page of text, forcing the text that follows the tag to begin on a new page.

---

### FORMAT

<PAGE>[[ { *EVEN* }  
          { *ODD* } ]]

---

### ARGUMENTS

*EVEN*  
*ODD*

These are optional keyword arguments. They specify that the next page of output have an even or an odd page number. If you use *EVEN*, and depending on where you put the <PAGE>(EVEN) tag, you will have two consecutive pages of output numbered 10 and 12, for example. If you use *ODD*, and depending on where you put the <PAGE>(ODD) tag, you will have two consecutive pages of output numbered 13 and 15, for example. There is no blank page of output between the pages.

---

### related tags

- <FINAL\_CLEANUP>(PAGE\_BREAK)
- <ONLINE\_CHUNK>

---

### restrictions

Invalid in the context of a <MATH> tag.

The arguments *EVEN* and *ODD* are invalid in tables.

If you use this tag in a table to force a page break, the table headings are not repeated on the page after the break. You must repeat the <TABLE\_HEADS> tag immediately after the <PAGE> tag to restore the default table headings. Headings are repeated on each page of a table only when DOCUMENT is allowed to break the table across pages.

---

### DESCRIPTION

The <PAGE> tag breaks a page of text, forcing the text that follows the tag to begin on a new page. Using the <PAGE> tag has no effect for Bookreader output.

**Note:** Use this tag only in special cases, where a page must be broken at that point. This tag is not for general use, such as for overall pagination; this can jeopardize the output format and device independence of your generically coded source file.

## <PAGE>

---

### EXAMPLE

The following example shows how to code a short document in which the information might be easier to locate when specific headings start on new pages.

```
<HEAD1>(Survey Results)
```

```
.
```

```
.
```

```
<PAGE>
```

```
<HEAD2>(Brand X)
```

```
.
```

```
.
```

```
<PAGE>
```

```
<HEAD2>(Brand Z)
```

```
.
```

```
.
```

```
.
```

---

## <PARENDCHAR>

Supplies parentheses around a character you specify, resulting in better spacing of the character between the parentheses.

---

**FORMAT**            <PARENDCHAR>(char)

---

**ARGUMENTS**        *char*  
Specifies the character within parentheses. You specify only the character; the parentheses are added during processing.

---

**restrictions**        Invalid in the context of a <MATH> tag.

---

**DESCRIPTION**      The <PARENDCHAR> tag supplies parentheses around a character you specify, resulting in better spacing of the character within the parentheses. Frequently, writers use the name for a special character followed by that character in parentheses. For instance, in discussing wildcard characters, it is usual to say, “The percent sign (%) is the wildcard for a single character.”

A problem in using proportionally spaced fonts is that single characters surrounded by parentheses can look crowded, like this: “The percent sign (%) is the wildcard for a single character.” Compare the percent sign in parentheses in this paragraph to the percent sign in parentheses in the previous paragraph; you can see the slight crowding of the percent sign in this paragraph.

Typographers put a small amount of space, called a “thin space,” between the parentheses and the single character to achieve a balance. When you use the <PARENDCHAR> tag, that thin space is added for you.

Use the <PARENDCHAR> tag to label characters that are small or that are more vertical than horizontal in shape. The parentheses are added during processing.

---

**EXAMPLE**            The following example shows how to use the <PARENDCHAR> tag.

<P>The tilde <PARENDCHAR>(~) is used to prevent the word following it from becoming a main entry in the index.

This example produces the following output:

The tilde (~) is used to prevent the word following it from becoming a main entry in the index.

Without the <PARENDCHAR> tag, your output would look like this:

The tilde (~) is used to prevent the word following it from becoming a main entry in the index.

## <PART>

---

## <PART>

Labels the start of a major division within a document and starts it on a new page.

---

**FORMAT**      `<PART>[(part title[\ symbol name])]`

---

**ARGUMENTS**      *part title*

This is an optional argument for printed output when you process a single file, but required if you include the file in a bookbuild, including a Bookreader bookbuild. This argument specifies the title for this part of your document.

*symbol name*

This is an optional argument for printed output when you process a single file, but required if you include the file in a bookbuild, including a Bookreader bookbuild. This argument specifies the term that you assign to this part and then use to reference the part throughout your document.

Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

**related tags**

- `<PART_PAGE>`

---

**DESCRIPTION**

The `<PART>` tag labels the start of a major division within a document and starts it on a new page. Parts may consist of the following:

- A set of chapters or appendixes whose contents are logically related. The chapters or appendixes are numbered sequentially from the start of the document to the end, regardless of the parts; that is, the chapters or appendixes within each part do not begin at “1”.
- A collection of units of logically related information, for example, reference items in the SOFTWARE doctype.

Like chapters, parts are automatically numbered. If you specify a *symbol name* argument, a symbol table entry is created for the part. The *part title* and *symbol name* arguments are required for Bookreader output.

The `<PART_PAGE>` tag creates a divider page for a part. You can specify a title for the part that is printed on the part page, a running title for the part (same or different from the title) that is printed on each page of the part, and an abstract printed on the part page to describe the information in the part.

By default, the part page is assigned the next odd page number in the current numbering sequence (except for Bookreader output, where there are no page numbers) in a document that does not have chapters. If you specify the RENUMBER argument to the <ENDPART\_PAGE> tag, the part page is assigned the number 1, and the next printed page of output is assigned the number 3. See the <ENDPART\_PAGE> tag description for more information.

Using the RENUMBER argument and the <RUNNING\_TITLE> tag have no effect for Bookreader output.

DOCUMENT considers each part to be an **online topic**. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information about online topics.

---

## EXAMPLE

The following example shows how to use the <PART> tag.

```
<part>(Introduction to Global Tags\intro_part)
<part_page>
<title>(<reference>(intro_part)\<reference>(intro_part\text))
<abstract>
This part contains introductory chapters. <reference>(ref_part)
contains more detailed reference information.
<endabstract>
<running_title>(Introduction to Global Tags)
<endpart_page>(RENUMBER)
```

This example produces the following output on the part page if this is the first part specified in the document:

- The <PART\_PAGE> tag creates a new page of output. The page is assigned the next odd page number in the numbering sequence.
- The title: “PART I Introduction to Global Tags” will be printed at the top of the part page. The first <REFERENCE> tag within the <TITLE> tag generates “PART I”; the second <REFERENCE> tag generates “Introduction to Global Tags”, as a result of using the *text* argument. You need to use the two <REFERENCE> tags to create both the part number and the part title. The use of uppercase Roman numerals to display the part number depends on the doctype you use.
- The abstract text is printed.
- The running title, “Introduction to Global Tags” is printed on each page of the part.
- Using the RENUMBER argument to the <ENDPART\_PAGE> tag assigns the part page to be number 1, and the next printed page is assigned to be page 3. If you do not use the RENUMBER argument, the part page is assigned the next odd page number in the current numbering sequence in a document that does not have chapters.

## <PART\_PAGE>

---

## <PART\_PAGE>

Inserts a divider page for a new part of a document.

---

### FORMAT      <PART\_PAGE>

---

**ARGUMENTS**      *None.*

---

**related tags**

- <ABSTRACT>
  - <PART>
  - <RUNNING\_TITLE>
  - <TITLE>
- 

**required  
terminator**

<ENDPART\_PAGE>

---

### DESCRIPTION

The <PART> tag inserts a divider page for a new part of a document. You can use the <TITLE>, <RUNNING\_TITLE>, and <ABSTRACT> tags to provide information on the part page and to specify a running title for that part of your document.

The <RUNNING\_TITLE> tag has no effect for Bookreader output.

---

### EXAMPLE

See the example in the discussion of the <PART> tag.

---

## <PREFACE>

Labels the beginning of a preface.

---

### FORMAT

<PREFACE>[(*page number* [\ *symbol name*])]

---

### ARGUMENTS

#### *page number*

This is an optional argument. It specifies the page number on which you want the preface to begin. You must specify the number with an Arabic numeral (the number in the formatted result will be a lowercase Roman numeral). If you do not specify a number, the default page on which the preface begins is Roman numeral five (v). This argument has no effect for Bookreader output.

#### *symbol name*

This is an optional argument. It specifies the name of the symbol used in all references to this heading. Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

### related tags

- <PREFACE\_SECTION>

---

### restrictions

Valid only in the context of a <FRONT\_MATTER> tag.

---

### required terminator

<ENDPREFACE>

---

## DESCRIPTION

The <PREFACE> tag labels the beginning of a preface. If you want the preface of your document placed after the table of contents, which usually begins on page (Roman numeral) iii, you must specify the starting page number for the preface for printed output; the *page number* argument has no effect for Bookreader output. You cannot know what this number will be until the document is nearly completed and you know what the page count is for the table of contents.

The preface always begins on a right-hand page following the table of contents; thus, it always begins with an odd page number, by default the Roman numeral five (v). You must adjust the page numbering of the preface, however, if the table of contents is longer than you expected (that is, longer than three pages).

If the table of contents ends on an even page (therefore, a left-hand page), you add one to the preface's page number, which starts the preface on the right-hand page. For example, if the table of contents ends on page 8, you would use the following tag to specify that the preface begin on page 9:

<PREFACE> (9)

## <PREFACE>

If the table of contents ends on an odd page (therefore, a right-hand page), you add two to that page number, which starts the preface on the right-hand page (and in this case, leaving a blank left-hand page). For example, if the table of contents ends on page 9, you would use the following tag to specify that the preface begin on page 11:

```
<PREFACE> (11)
```

You cannot reference any headings from the preface; sections you code within the preface with <HEAD<sub>n</sub>> tags are unnumbered and do not produce table of contents entries.

If your preface includes a section that you want to appear in the table of contents, you must code it as follows:

```
<PREFACE_SECTION> (New Section)
```

DOCUMENT considers the preface to be an **online topic**. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information about online topics.

---

### EXAMPLE

See the example in the discussion of the <FRONT\_MATTER> tag.



---

## <PREFACE\_SECTION>

Creates a major section in the preface of a book to provide information, such as a summary of changes to the book.

---

**FORMAT**            <PREFACE\_SECTION>(title [\ symbol name])

---

**ARGUMENTS**        *title*  
Specifies the title you give to the preface section.

*symbol name*  
This is an optional argument. It specifies the name of the symbol used in all references to this heading. Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

**related tags**        • <PREFACE>

---

**restrictions**        Valid only in the context of a <FRONT\_MATTER> tag.

---

**DESCRIPTION**        The <PREFACE\_SECTION> tag creates a major section in the preface of a book to provide information, such as a summary of changes to the book.  
  
Note that because <HEAD<sub>n</sub>> tags in a preface do not produce table of contents entries, you must code sections of your preface that you want to appear in the table of contents as follows:

<PREFACE\_SECTION>(New Section)

---

**EXAMPLE**            See the example in the discussion of the <FRONT\_MATTER> tag.

## <PRINT\_DATE>

---

## <PRINT\_DATE>

Inserts a print date line on the copyright page.

---

**FORMAT**      <PRINT\_DATE>(date)

---

**ARGUMENTS**    *date*  
Specifies official printing date information for the book.

---

**restrictions**    Valid only in the context of a <COPYRIGHT\_PAGE> tag in the front matter of a document.

---

**DESCRIPTION**    The <PRINT\_DATE> tag inserts a print date line on the copyright page.

---

**EXAMPLE**        See the example in the discussion of the <FRONT\_MATTER> tag.

---

## <PROFILE>

Indicates that the source file is a profile that generates a bookbuild.

---

### FORMAT <PROFILE>

---

**ARGUMENTS** *None.*

---

**related tags**

- <ELEMENT>
- <INCLUDES\_FILE>

---

**required terminator** <ENDPROFILE>

---

### DESCRIPTION

The <PROFILE> tag indicates that the source file is a profile that generates a bookbuild. A profile of a book is required in order to build (process) a book. Between the <PROFILE> and <ENDPROFILE> tags, you introduce each element of the book with an <ELEMENT> tag.

Only those files listed with <ELEMENT> tags are included in the book during the bookbuild. List them in the profile in the order in which they appear in the book.

Within a profile file, never include a tag that produces text. Also, do not place an <INCLUDE> tag in a profile.

There are several optional tags that you can include between the <PROFILE> and <ENDPROFILE> tags, other than <ELEMENT> tags. These include the following:

- <COMMENT> and <ENDCOMMENT>
- <CONDITION> and <ENDCONDITION>
- <CONTENTS\_FILE>
- <INCLUDES\_FILE>
- <INDEX\_FILE>
- <REVISION>
- <SET\_CONDITION>
- <SET\_ONLINE\_TOPIC>

See *VAX DOCUMENT Producing Online and Printed Documentation* for more information on bookbuilding.

## <PROFILE>

---

### EXAMPLE

The following example shows the profile of a book that contains a table of contents, four chapters, an appendix, and an index. The profile shown in this example must be in a file by itself. You should call the file by a name that indicates it is a profile, such as "COMPUTER\_PROFILE.SDML."

Each of the files named in the <ELEMENT> tags should begin with one of the book element tags, such as <CHAPTER>(Introduction\introduction\_chap).

```
<PROFILE>      <COMMENT>(***Profile for How to Use a Computer***)
<CONTENTS_FILE> <COMMENT>(***insert table of contents here***)
<ELEMENT>(Mydisk:[Mydirectory]front_matter.sdml)
<ELEMENT>(Mydisk:[Mydirectory]intro_chap.sdml)
<ELEMENT>(Mydisk:[Mydirectory]applications_chap.sdml)
<ELEMENT>(Mydisk:[Mydirectory]tools_chap.sdml)
<ELEMENT>(Mydisk:[Mydirectory]conclusion_chap.sdml)
<ELEMENT>(Mydisk:[Mydirectory]questions_app.sdml)

<INDEX_FILE> <COMMENT>(***insert index here***)
<ENDPROFILE>
```



## <QUOTE>

---

### EXAMPLES

The following example shows how to use the <QUOTE> tag to obtain distinct opening and closing quotation marks in your output.

1 <P>The symbol B will be defined to be the string <QUOTE>(April showers.)

This example produces the following output:

The symbol B will be defined to be the string “April showers.”

The following example shows the same output as the first example using the keyboard quotation marks instead of using the <QUOTE> tag.

2 <P>The symbol B will be defined to be the string "April showers."

This example produces the following output:

The symbol B will be defined to be the string "April showers."

The following example shows how to code some text if you want a punctuation mark outside the quotation marks. Put the punctuation mark outside of the closing parenthesis of the argument to the <QUOTE> tag or after the <ENDQUOTE> tag.

3 <P>Did Abraham Lincoln write, <QUOTE> I claim not to have controlled events, but confess plainly that events have controlled me <ENDQUOTE>?

This example produces the following output:

Did Abraham Lincoln write, “ I claim not to have controlled events, but confess plainly that events have controlled me ”?

---

## <REFERENCE>

Makes a reference to a symbol name in a book element or text element. When processed, the <REFERENCE> tag is replaced with the current value of the symbol name that is stored in the cross-reference file.

---

### FORMAT

<REFERENCE>(symbol name [ { \ FULL  
                                  \ TEXT  
                                  \ VALUE } ])

---

### ARGUMENTS

#### **symbol name**

Specifies the name of a symbol assigned in a book element or text element tag (for example, <HEADn> or <TABLE>). Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

#### **FULL**

This is an optional keyword argument. It outputs all information associated with the symbol, including the value, type, and text.

#### **TEXT**

This is an optional keyword argument. It outputs only the text associated with the symbol table entry, that is, the caption (for figures, tables, and examples), or the heading text (for chapter, section, and heading-level tags).

#### **VALUE**

This is an optional keyword argument. It outputs only the current value of the symbol, for example, the heading level number or the table number, without any related text.

If you do not specify any of the keywords, the current value of the symbol and the text related to the kind of symbol it is, for example, Figure 3-2 or Table 3-2, is output. This is the default.

---

### related tags

- <HOTSPOT>—See *VAX DOCUMENT Using Doctypes and Related Tags*

---

### DESCRIPTION

The <REFERENCE> tag makes a reference to a symbol name in a book element or text element. When processed, the <REFERENCE> tag is replaced with the current value of the symbol name that is stored in the cross-reference file.

A second argument to the tag controls the exact output of the reference. When you do not specify a second argument, the output defaults according to whether the symbol refers to a heading, a figure, a table, a chapter, and so on. Table 1-4 summarizes the default output for each type of text or book element.

## <REFERENCE>

Table 1-4 Element Types and Default Output of Symbol Names

Text Element	Second Argument to <REFERENCE>			
	(default)	VALUE	TEXT	FULL
BOOK TITLE	The <i>title</i> argument specified in <DEFINE_BOOK_NAME>.			
EXAMPLE	"Example" number	number	caption	"Example" number, caption
FIGURE	"Figure" number	number	caption	"Figure" number, caption
HEAD <sub>n</sub>	"Section" number	number	caption	"Section" number, caption
TABLE	"Table" number	number	caption	"Table" number, caption
TEXT STRING	The <i>text string</i> argument specified in <DEFINE_SYMBOL>.			

Book Elements	(default)	VALUE	TEXT	FULL
APPENDIX	"Appendix" letter	letter	title	"Appendix" letter, title
CHAPTER	"Chapter" number	number	title	"Chapter" number, title
PART	"Part" number	number	title	"Part" number, title
SECTION	"Section" number	number	title	"Section" number, title

See *VAX DOCUMENT Producing Online and Printed Documentation* for more information on symbol names and cross-referencing.

For a book you create for Bookreader, you must use the <REFERENCE> tag to allow DOCUMENT to automatically cross-reference the symbol names you assign to the tag.

The following tags require a symbol name for a book you create for Bookreader:

- <APPENDIX>
- <CHAPTER>
- <CHEAD>
- <EXAMPLE>
- <FIGURE>
- <HEAD>
- <HEAD<sub>n</sub>>
- <PREFACE>
- <PREFACE\_SECTION>
- <SUBHEAD<sub>n</sub>>
- <TABLE>
- <COMMAND>



- <ROUTINE>
- <SDML\_TAG>
- <STATEMENT>
- <SUBCOMMAND>

The last five tags are listed in *VAX DOCUMENT Using Doctypes and Related Tags*.

The following tags accept a symbol name, but do not require one, for both printed books and books you create for Bookreader:

- <FRONT\_MATTER>
- <GLOSSARY>
- <PART>
- <MATH>

The following tags, listed in *VAX DOCUMENT Using Doctypes and Related Tags*, accept a symbol name for a printed book; the tags are ignored for books you create for Bookreader:

- <REF\_NOTE>
- <SECTION>

When you use the <REFERENCE> tag to refer to a formal example, figure, or table, the words that appear in the **hotspot** are directly related to the way you code the <REFERENCE> tag. For example, suppose you enter a <REFERENCE>(symbol name\VALUE) tag when referring to a hotspot, but you physically type in an identifying reference before the tag, like “See Figure <REFERENCE>(symbol name\VALUE)”. The word that you used as the identifying reference will appear outside of the hotspot; for example, “1-1” would appear in the hotspot, but “Figure” would not.

---

## EXAMPLES

The following example shows how to use the <REFERENCE> and <TABLE> tags to make a default reference to the symbol name of a table. The symbol name “lognames\_tab” is defined in the <TABLE> tag and then referred to with the <REFERENCE> tag.

**1** <REFERENCE>(lognames\_tab) shows ten default system logical names.  
<TABLE>(System Logical Names\lognames\_tab)

·  
·  
·

This example produces the following output:

Table 1-4 shows ten default system logical names.

For Bookreader output, “Table 1-4” is a hotspot: Table 1-4.

## <REFERENCE>

The following example shows how to use the <REFERENCE> and <TABLE> tags to refer to the table in full, that is, to refer to the number and name of the table. The symbol name “lognames\_tab” is defined in the <TABLE> tag and then referred to with the <REFERENCE> tag.

2 <REFERENCE>(lognames\_tab\full) shows ten default system logical names.  
<TABLE>(System Logical Names\lognames\_tab)

.  
.

This example produces the following output:

Table 1-4, System Logical Names, shows ten default system logical names.

For Bookreader output, “Table 1-4, System Logical Names” is a hotspot:

[Table 1-4, System Logical Names](#)

The following example shows how to use the <REFERENCE> and <HEAD1> tags to refer to the text of the heading. The symbol name “assigning\_attributes” is defined in the <HEAD1> tag and then \ referred to with the <REFERENCE> tag.

3 The <REFERENCE>(assigning\_attributes\text) section describes the method to assign color to the objects on your screen.  
<HEAD1>(Assigning Color Attributes\assigning\_attributes)

This example produces the following output:

The Assigning Color Attributes section describes the method to assign color to the objects on your screen.

For Bookreader output, “Assigning Color Attributes” is a hotspot:

[Assigning Color Attributes](#)

The following example shows how to use the <REFERENCE> tag to refer to a chapter number without outputting any related information, such as the chapter title.

4 <P>See Chapters <REFERENCE>(lognames\_chap\value) and <REFERENCE>(filespec\_chap\value) to see . . .

This example produces the following output:

See Chapters 3 and 4 to see . . .

For Bookreader output, “3” and “4” are hotspots: [3](#) and [4](#).

---

## <REVISION>

Indicates that the document contains either new or modified information.

---

**FORMAT**      <REVISION>[(UPDATE [\ update info])]

---

**ARGUMENTS**      **UPDATE**

This is an optional keyword argument, but you must specify it if you are producing the document for an update. If you specify UPDATE, the file must contain the <UPDATE\_RANGE> and <ENDUPDATE\_RANGE> tags to indicate the pages to be processed. If no <UPDATE\_RANGE> tags are present, no output is produced. The UPDATE argument has no effect for Bookreader output.

**update info**

This is an optional argument. It specifies information that is related to the system version and the date of the update. The text appears on the bottom of each page of output in the update. The *update info* argument has no effect for Bookreader output.

---

**related tags**

- <MARK>
- <UPDATE\_RANGE>

---

**restrictions**

Invalid in the two multicolumn doctypes, ARTICLE and REPORT.TWOCOL.

---

**DESCRIPTION**

The <REVISION> tag indicates that the document contains either new or modified information. It enables the <MARK>, <ENDMARK>, <UPDATE\_RANGE>, and <ENDUPDATE\_RANGE> tags. By default, these tags are defined for all doctypes to be non-operational; that is, if a file is processed without the <REVISION> tag, the related tags produce no output.

You can produce update pages only for a printed document; you cannot produce them for Bookreader.

When a file that contains the <REVISION>(UPDATE) tag is processed, the table of contents and index are handled as follows: if you specify /CONTENTS and /INDEX on the command line, you receive a table of contents and an index even if you do not specify the <CONTENTS\_FILE> and <INDEX\_FILE> tags within the <UPDATE\_RANGE> and <ENDUPDATE\_RANGE> tags.

**Note:** If the device converter encounters errors on point pages created using the <REVISION>(update) tag, it signals them as occurring on the original page, not the point page. For example, an error on point page 11-2.1 is signalled as occurring on page 11-2.

## <REVISION>

---

### EXAMPLES

The following example shows how to use the <REVISION> tag in a document that is extensively revised.

```
1 <REVISION>
  .
  .
  .
<P>
The following characters are legal in MACRO-11 source programs:
<LIST>(UNNUMBERED)
<LE>The letters A through Z. Both upper- and lowercase letters are
acceptable, although, upon input, lowercase letters are converted to
uppercase.
<MARK>
<LE>Characters in the Digital Multinational Character Set (MCS). A chart
showing the MCS is located in <REFERENCE>(mcs_app),
with a list of directives that support the MCS.
<LE>The digits 0 through 9.
<ENDMARK>
<LE>The characters period <PARENDCHAR>(.) and dollar sign
<PARENDCHAR>($). These characters are reserved for use as Digital
Equipment Corporation system program symbols.
<ENDLIST>
```

This example produces the following output:

The following characters are legal in MACRO-11 source programs:

- The letters A through Z. Both upper- and lowercase letters are acceptable, although, upon input, lowercase letters are converted to uppercase.
- Characters in the Digital Multinational Character Set (MCS). A chart showing the MCS is located in Appendix A, with a list of directives that support the MCS.
- The digits 0 through 9.
- The characters period (.) and dollar sign (\$). These characters are reserved for use as Digital Equipment Corporation system program symbols.

The following example shows how to use the <REVISION> tag to update a manual. See the <UPDATE\_RANGE> tag description for more information on identifying a section of updated material.

```
2 <REVISION>(UPDATE\November 1990)
  .
  .
  .
<UPDATE_RANGE>(3\10)
```

---

## <REVISION\_INFO>

Labels a section on a title page that provides information on what previous books have been superseded by the current one.

---

**FORMAT**      <REVISION\_INFO> (*[title text] \ info*)

---

**ARGUMENTS**    *title text*  
This is an optional argument. It specifies heading information. If you do not specify this argument, the default text “Revision/Update Information:” is supplied.

*info*  
Specifies revision and update information.

---

**related tags**      • <FRONT\_MATTER>

---

**restrictions**      Valid only in the context of a <TITLE\_PAGE> tag.

---

**DESCRIPTION**    The <REVISION\_INFO> tag labels a section on a title page that provides information on what previous books have been superseded by the current one.

---

**EXAMPLE**          See the example in the discussion of the <FRONT\_MATTER> tag.

## <RIGHT\_LINE>

---

## <RIGHT\_LINE>

Specifies a line of text that is to be aligned against the right-hand margin of the page.

---

### FORMAT

**<RIGHT\_LINE>**(*text* [ { *\BIGSKIP*  
*\SMALLSKIP* } ] )

---

### ARGUMENTS

***text***

Specifies the line of text to be aligned against in the right-hand margin of the page.

***BIGSKIP***

***SMALLSKIP***

These are optional keyword arguments. They specify that a set amount of vertical space precede the element identified as a line or block of text. The actual amount of space inserted is determined by the doctype.

---

### related tags

- <CENTER\_LINE>
- <LINE>

---

### restrictions

Invalid in the context of an <EXAMPLE> and a <MATH> tag.

The aligned text must fit within the right-hand margin. If you specify text that is too wide, the text formatter issues a warning message, and you should examine your output.

---

### DESCRIPTION

The <RIGHT\_LINE> tag specifies a line of text that is to be aligned against the right-hand margin of the page. This text is said to be **right-justified**.

**Note:** When you use the <RIGHT\_LINE> tag as the first tag in a table column, it leaves extra space in that column.

---

### EXAMPLES

The following example shows how to use the <RIGHT\_LINE>(SMALLSKIP) tag.

```
¶ <P>Please include the following information:  
<RIGHT_LINE>(Name)  
<RIGHT_LINE>(Address\smallskip)  
<RIGHT_LINE>(Phone Number)
```

This example produces the following output:

Please include the following information:

Name  
Address  
Phone Number

The following example shows how to use the <RIGHT\_LINE>(BIGSKIP) tag.

2 <P>Please include the following information:  
<RIGHT\_LINE>(Name)  
<RIGHT\_LINE>(Address\bigskip)  
<RIGHT\_LINE>(Phone Number)

This example produces the following output:

Please include the following information:

Name  
Address  
Phone Number

## <RULE>

---

## <RULE>

Outputs a horizontal rule in a table.

---

**FORMAT**      **<RULE>**

---

**ARGUMENTS**      *None.*

---

**related tags**

- <SPAN>
  - <TABLE\_HEADS>
  - <TABLE\_ROW>
  - <TABLE\_UNIT\_HEADS>
- 

**restrictions**

Valid only in an argument to a <TABLE\_ROW>, <TABLE\_HEADS>, or <TABLE\_UNIT\_HEADS> tag; it must immediately follow the argument text under which the rule is to be placed.

---

**DESCRIPTION**

The <RULE> tag outputs a horizontal rule in a table. You place a <RULE> tag inside an argument to a <TABLE\_HEADS>, <TABLE\_ROW>, or <TABLE\_UNIT\_HEADS> tag.

The horizontal length of the rule does not correspond to the dimensions of the table. It equals the width of the table column or the spanned columns, if the argument also is preceded by a <SPAN> tag. See the <SPAN> tag description for more information.

---

**EXAMPLE**

See the example in the discussion of the <TABLE\_UNIT> tag.



---

<S>

Labels the system portion of a dialog between user and system in an interactive example.

---

**FORMAT**            <S>(text)

---

**ARGUMENTS**        *text*  
Specifies the text of the system message.

---

**related tags**        • <INTERACTIVE>  
                          • <U>

---

**restrictions**        Invalid in the context of a <MATH> tag.

---

**DESCRIPTION**        The <S> tag labels the system portion of a dialog between user and system in an interactive example. An example containing this type of dialog must have both parts identified in order to differentiate the two types of text in the source code, the output, or both.

In the Software doctype, you can also use the <S> and <U> tags to differentiate the system and user text inside of examples created with the <EXAMPLE\_SEQUENCE> and <EXI> tags. For more information on this doctype, refer to *VAX DOCUMENT Using Dotypes and Related Tags*.

---

**EXAMPLES**            The following example shows how to use the <S> tag.

**i**    <P>The system prompt <S>(\$) indicates you can enter a command.

This example produces the following output:

The system prompt \$ indicates you can enter a command.

The following example shows how to code dialog between both the system and the user. In a dialog, the <S> and <U> tags must be used between <INTERACTIVE> and <ENDINTERACTIVE> tags. Note that you should specify, within the argument to the <S> tag, whatever space follows the system prompt.

<S>

2 <P>The following example of VAXMAIL contains messages from both the system and a user of the system:

```
<INTERACTIVE>
<U>(mail)
<S>(MAIL> )<U>(send)
<S>(To: )<U>(nodename::Courtney)
<S>(%MAIL-E-NOSUCHUSR, no such user COURTNEY at node NODENAME)
<ENDINTERACTIVE>
```

This example produces the following output:

The following example of VAXMAIL contains messages from both the system and a user of the system:

```
mail
MAIL> send
To: nodename::Courtney
%MAIL-E-NOSUCHUSR, no such user COURTNEY at node NODENAME
```

---

## <SAMPLE\_TEXT>

Typographically distinguishes an extract of text.

---

**FORMAT**            <SAMPLE\_TEXT>

---

**ARGUMENTS**        *None.*

---

**related tags**        • <CENTER\_LINE>  
                          • <P>

---

**restrictions**        The <SAMPLE\_TEXT> tag does not provide paragraph formatting. If you want to show a paragraph within the sample text, you must label it with a <P> tag.

---

**required terminator**    <ENDSAMPLE\_TEXT>

---

**DESCRIPTION**        The <SAMPLE\_TEXT> tag typographically distinguishes an extract of text. The sample text is indented from the normal text margins.

---

**EXAMPLE**            The following example shows how to use the <SAMPLE\_TEXT> tag.

```
<P>The language is perpetually in flux: it is a living stream,
shifting, changing, receiving new strength from a thousand tributaries. . .
<SAMPLE_TEXT>
<P>Taken from Strunk, William Jr. and White, E.B.: The Elements of Style.
Macmillan Publishing Co., Inc., 1979.
<ENDSAMPLE_TEXT>
```

This example produces the following output:

The language is perpetually in flux: it is a living stream, shifting, changing, receiving new strength from a thousand tributaries. . .

Taken from Strunk, William Jr. and White, E.B.: The Elements of Style. Macmillan Publishing Co., Inc., 1979.

## <SET\_APPENDIX\_LETTER>

---

## <SET\_APPENDIX\_LETTER>

Overrides the default appendix letter assigned to an appendix by VAX DOCUMENT.

---

**FORMAT**            <SET\_APPENDIX\_LETTER> (*appendix letter*)

---

**ARGUMENTS**        *appendix letter*  
Specifies the letter of the appendix. This argument must be a letter from A to Z.

---

**related tags**

- <APPENDIX>
- <SET\_CHAPTER\_NUMBER>
- The MILSPEC doctype <SET\_APPENDIX\_NUMBER> tag

---

**DESCRIPTION**     The <SET\_APPENDIX\_LETTER> tag overrides the default appendix letter assigned to an appendix by VAX DOCUMENT. The tag is useful only in an element build to override the default appendix letter created by VAX DOCUMENT; it is ignored in a bookbuild. The <SET\_APPENDIX\_LETTER> tag resets the current appendix letter and resets the lettering for all following appendixes. For example, if you specified C as the argument, the <APPENDIX> tag generates Appendix C. The next <APPENDIX> tag generates Appendix D, and so on.

Place the <SET\_APPENDIX\_LETTER> tag in your SDML file before the <APPENDIX> tags you want it to affect, because the <SET\_APPENDIX\_LETTER> tag affects only the <APPENDIX> tags that follow it.

You can use the <SET\_APPENDIX\_LETTER> tag multiple times in an SDML file.

---

**EXAMPLE**            The following example shows how to use the <SET\_APPENDIX\_LETTER> tag. The appendix "Error Messages" is explicitly set to C using the <SET\_APPENDIX\_LETTER> tag. This causes any subsequent appendixes to be numbered beginning with the letter D, unless you use another <SET\_APPENDIX\_LETTER> tag to reset the current appendix letter.

```
<SET_APPENDIX_LETTER>(C)
<APPENDIX>(Error Messages\error_msg_app)
<p>
The following error messages...
```

---

## <SET\_CHAPTER\_NUMBER>

Overrides the default chapter number assigned to a chapter by VAX DOCUMENT.

---

**FORMAT**            <SET\_CHAPTER\_NUMBER>(chapter number)

---

**ARGUMENTS**        *chapter number*  
Specifies the number of the chapter. This argument must be a positive integer.

---

**related tags**

- <CHAPTER>
- <SET\_APPENDIX\_LETTER>

---

**DESCRIPTION**     The <SET\_CHAPTER\_NUMBER> tag overrides the default chapter number assigned to a chapter by VAX DOCUMENT. The tag is useful only in an element build to override the default chapter number created by VAX DOCUMENT; it is ignored in a bookbuild. The <SET\_CHAPTER\_NUMBER> tag resets the current chapter number and the numbering for all following chapters. For example, if you specify 13 as the chapter number, the next <CHAPTER> tag generates Chapter 13. The following <CHAPTER> tag generates Chapter 14, and so on.

Place the <SET\_CHAPTER\_NUMBER> tag in your SDML file before the <CHAPTER> tags you want it to affect, because the <SET\_CHAPTER\_NUMBER> tag affects only the <CHAPTER> tags that follow it.

You can use the <SET\_CHAPTER\_NUMBER> tag multiple times in an SDML file.

---

**EXAMPLE**            The following example shows how to use the <SET\_CHAPTER\_NUMBER> tag. The chapter is set to 13 using the <SET\_CHAPTER\_NUMBER> tag. This causes any subsequent chapters to be numbered beginning with number 14, unless you use another <SET\_CHAPTER\_NUMBER> tag to reset the current chapter number.

```
<SET_CHAPTER_NUMBER>(13)
<CHAPTER>(Supported Devices\sup_dev_chap)
<p>
The primary supported devices...
```

## <SET\_CONDITION>

---

## <SET\_CONDITION>

Creates or removes a condition in your SDML file.

---

### FORMAT

**<SET\_CONDITION>**(*condition name*[ \ *REMOVE*])

---

### ARGUMENTS

#### ***condition name***

Specifies a name you use to conditionalize a portion of your SDML file. This name is limited to 28 characters.

#### ***REMOVE***

This is an optional keyword argument. It removes the condition name.

---

### related tags

- <CONDITION>

---

### DESCRIPTION

The <SET\_CONDITION> tag creates or removes a condition in your SDML file. You specify conditions with the <CONDITION> tag.

For a complete explanation of creating conditional text, refer to *VAX DOCUMENT Producing Online and Printed Documentation*.

Normally, you create a condition at the front of your input file and it remains in effect for the rest of the file. If you want to process a portion of the input file without the condition, you can remove it by repeating the <SET\_CONDITION> tag with the REMOVE argument.

You can use the /CONDITION qualifier on the DOCUMENT command line instead of placing the <SET\_CONDITION> tag into your file. For example, you might use the command line qualifier /CONDITION=local instead of putting the <SET\_CONDITION>(local) tag in your input file.

---

### EXAMPLE

The following example shows how to use the <SET\_CONDITION> tag at the beginning of your file in order to establish the condition. In this case, the condition, set by the <SET\_CONDITION>(Christmas) tag, is "Christmas". This means that only the information on Christmas, designated with the <CONDITION> tag, is processed.

This example also shows how to use the <SET\_CONDITION> tag with the REMOVE argument in order to remove the condition you set with the <SET\_CONDITION> tag. Removing the condition applies for the rest of your file. In this case, the first instance of Christmas information is processed, but all instances of Christmas information after the <SET\_CONDITION>(Christmas\REMOVE) tag are not processed.

## <SET\_CONDITION>

```
<SET_CONDITION>(Christmas)
<CONDITION>(Christmas)
<P>Christmas, by convention, is celebrated on December 25th...
<ENDCONDITION>

<CONDITION>(Chanukah)
<P>Chanukah is called the Festival of Lights...
<ENDCONDITION>

<CONDITION>(Passover)
<P>Passover is usually celebrated...
<ENDCONDITION>

<SET_CONDITION>(Christmas\REMOVE)

<CONDITION>(Christmas\Chanukah)
<HEAD1>(Religious Holidays)
<P>This paragraph contains general information about several religious
holidays...
<ENDCONDITION>
```





## <SET\_FIGURE\_FILE\_SPACING\_DEFAULT>

---

### EXAMPLES

The following example shows how to set the figure file spacing default to 0 picas before a figure and 1.24 picas after a figure. All subsequent graphics files included using the <FIGURE\_FILE> tag will also have this amount of white space around the figure.

**1** <SET\_FIGURE\_FILE\_SPACING\_DEFAULT>(0\1.24)  
<FIGURE>(Graphic File Inclusion)  
<FIGURE\_FILE>(LN03\MY\_FILES:HOUSE.SIX\13.76)  
<ENDFIGURE>

In the following example, the figure file spacing is set to 10 picas before the figure and 0 picas after it. Using the second <SET\_FIGURE\_FILE\_SPACING\_DEFAULT> tag resets the amount of spacing for any subsequent figures to 2 picas before the figure and 2 picas after the figure.

**2** <SET\_FIGURE\_FILE\_SPACING\_DEFAULT>(10\0)  
<FIGURE>(Graphic File Inclusion)  
<FIGURE\_FILE>(LN03\MY\_FILES:HOUSE.SIX\13.76)  
<ENDFIGURE>  
<SET\_FIGURE\_FILE\_SPACING\_DEFAULT>(2\2)

## <SET\_TABLE\_ROW\_BREAK\_DEFAULT>

---

## <SET\_TABLE\_ROW\_BREAK\_DEFAULT>

Overrides the default value for a multipage table's first valid break.

---

**FORMAT**            <SET\_TABLE\_ROW\_BREAK\_DEFAULT>(number of rows)

---

**ARGUMENTS**        *number of rows*

Specifies the number of default table rows that must be on the first page of a multipage table. This argument must be a positive integer.

You can specify the <SET\_TABLE\_ROW\_BREAK\_DEFAULT> tag anywhere in an SDML file. The value you specify using this tag will affect subsequent tables until the next occurrence of the tag or until the end of the file.

---

**related tags**

- <TABLE\_ROW\_BREAK>

---

**DESCRIPTION**

The <SET\_TABLE\_ROW\_BREAK\_DEFAULT> tag overrides the default value for a multipage table's first valid break. By default, tables are considered to be multipage; that is, if there is not enough room on the current page for a table, the table is continued onto subsequent pages, with any captions and headings repeated at the top of each new page. When the text formatter chooses places in the table at which to insert page breaks, it normally breaks the table between <TABLE\_ROW> tags. By default, it assumes that it is correct to break a table between any two table rows after the first table row.

You can override this default behavior in the following ways:

- Specify the <SET\_TABLE\_ROW\_BREAK\_DEFAULT> tag to provide a default number of rows that must be on the first page of the table. For example, if you specify the <SET\_TABLE\_ROW\_BREAK\_DEFAULT>(3) tag, any subsequent table will not be broken until after the third row.
- Specify the CONTROLLED argument to the <TABLE\_ATTRIBUTES> tag, which indicates that you are going to specify, explicitly, the range in which the table will be allowed to break. You then must use the <TABLE\_ROW\_BREAK>(FIRST) and <TABLE\_ROW\_BREAK>(LAST) tags to indicate the first and last allowable page break points. Between these two tags, the table may be broken between any two <TABLE\_ROW> tags. See the example in the description of the <TABLE\_ATTRIBUTES> tag.
- Use the KEEP argument to the <TABLE\_ATTRIBUTES> tag to indicate that the table must not be broken across pages, unless it is longer than a page. See the example in the description of the <TABLE\_ATTRIBUTES> tag.
- Use the <VALID\_TABLE\_ROW\_BREAK> tag to indicate a place within a long table row that is an allowable break point. See the example in the description of the <VALID\_BREAK> tag.

## <SET\_TABLE\_ROW\_BREAK\_DEFAULT>

- Use the <NESTED\_TABLE\_BREAK> tag to indicate a place in a nested table (a table within a table) that is an allowable break point.

This tag has no effect for Bookreader output.

---

### EXAMPLE

In the following example, the table consists of short, one-line items. The <SET\_TABLE\_ROW\_BREAK\_DEFAULT> tag indicates that this table, and any following it, must not be broken until after the fifth row.

```
<SET_TABLE_ROW_BREAK_DEFAULT>(5)
<TABLE>
<TABLE_ATTRIBUTES>(SINGLE_SPACED)
<TABLE_SETUP>(2\5)
<TABLE_HEADS>(Code\Numeric Value)
<TABLE_ROW>(A\1)
<TABLE_ROW>(B\2)
.
.
.
<TABLE_ROW>(Z\26)
<ENDTABLE>
```

## <SINGLE\_QUOTE>

---

## <SINGLE\_QUOTE>

Supplies a single quotation mark as it appears on a keyboard.

---

### FORMAT <SINGLE\_QUOTE>

---

**ARGUMENTS** *None.*

---

**related tags**

- <DOUBLE\_QUOTE>
- <PARENDCHAR>
- <QUOTE>

---

**DESCRIPTION** The <SINGLE\_QUOTE> tag supplies a single quotation mark as it appears on a keyboard. This is different from the single quotation mark that appears on your terminal screen.

If you use the single quotation mark from your keyboard instead of the tag, your output will contain an unmatched closing single quotation mark—one half of what should be a matched pair according to typesetting conventions.

---

**EXAMPLE** The following example shows how to use the <SINGLE\_QUOTE> tag. The <PARENDCHAR> tag simply provides better spacing of the single quote within the parentheses.

```
<P>Notice the difference between using the  
<tag>(single_quote) tag to output a single quotation mark,  
for example, <PARENDCHAR><SINGLE_QUOTE>, and using the  
keyboard single quotation mark, for example, <PARENDCHAR>(').
```

This example produces the following output:

Notice the difference between using the <SINGLE\_QUOTE> tag to output a single quotation mark, for example, ( ' ), and using the keyboard single quotation mark, for example, ( ' ).

---

**<SPAN>**

Specifies that the accompanying argument in a table row or a table head spans more than one table column.

---

**FORMAT**      **<SPAN>**(*number of columns*[\ *LEFT*])

---

**ARGUMENTS**      ***number of columns***

Specifies the number of table columns that the text following the <SPAN> tag spans. This argument must be a whole number and must be no larger than the number of table columns remaining in the table row, including the table column in which you placed the <SPAN> tag itself.

***LEFT***

This is an optional keyword argument. It specifies that the text of the argument be aligned to the left-most position of the spanned columns. If you do not use this argument, the text of the argument is centered in the spanned columns.

---

**related tags**

- <RULE>
- <TABLE\_HEADS>
- <TABLE\_UNIT\_HEADS>

---

**restrictions**

Valid only in an argument to a <TABLE\_ROW>, <TABLE\_HEADS>, or <TABLE\_UNIT\_HEADS> tag.

Must immediately precede the argument text.

---

**DESCRIPTION**

The <SPAN> tag specifies that the accompanying argument in a table row or a table head spans more than one table column. The text is displayed without regard to the gutter that separates the columns. The text is centered in the spanned columns, unless you specify the *LEFT* argument.

If the text length exceeds the width of the spanned columns, it is broken at a word boundary and displayed on additional table lines. Any additional lines of text are centered or aligned against the left margin to agree with the alignment of text in the first line.

Use the <RULE> tag if you want to place a rule beneath a spanned heading created using the <SPAN> tag.

## <SPAN>

### EXAMPLES

The following example shows how to use the <SPAN> tag to center the table heading over three columns. It also shows how to use the LEFT keyword to position a table unit heading to the left-most position of the spanned columns. Note the use of the <RULE> tag in this example.

```
1 <TABLE>
  <TABLE_SETUP>(3\12\12)
  <TABLE_HEADS>(<SPAN>(3)Types of Ancient Weaponry)
  <TABLE_UNIT>
  <TABLE_UNIT_HEADS>(Polearms)
  <TABLE_ROW>(Spear\Javelin\Halberd)
  <ENDTABLE_UNIT>
  <TABLE_UNIT>
  <TABLE_UNIT_HEADS>(Maces\<SPAN>(2\LEFT)Swords<RULE>)
  <TABLE_UNIT_HEADS>( \Short swords\Long swords)
  <TABLE_ROW>(Great Mace\Gladius\Great Sword)
  <ENDTABLE_UNIT>
<ENDTABLE>
```

This example produces the following output:

Types of Ancient Weaponry		
<b>Polearms</b>		
Spear	Javelin	Halberd
<b>Maces</b>	<b>Swords</b>	
	<b>Short swords</b>	<b>Long swords</b>
Great Mace	Gladius	Great Sword

In the following example, notice how using the <SPAN> tag in this way centers "Small Fixture" correctly.

```
2 <table>
  <table_setup>(5\9\9\9\9)
  <table_row>(\<span>(2)Small Fixture\<span>(2)Large Fixture)
  <table_row>(Component\Minimum\Maximum\Minimum\Maximum)
<endtable>
```

Notice, however, that "Large Fixture" may be too far to the right, as in the following example:

	Small Fixture		Large Fixture	
Component	Minimum	Maximum	Minimum	Maximum

This occurs because the <SPAN> tag attempts to span the last column to the margin. To fix this problem, code the table to specify an additional column and specify a width for the next to last column, thereby limiting the distance that is spanned. The following is an example of such a correction:

```
<table>
<table_setup>(6\8\8\8\8)
<table_row>(\<span>(2)Small Fixture\<span>(2)Large Fixture)
<table_row>(Component\Minimum\Maximum\Minimum\Maximum)
<endtable>
```

This example produces the following output:

	Small Fixture		Large Fixture	
Component	Minimum	Maximum	Minimum	Maximum

## <SPECIAL\_CHAR>

---

## <SPECIAL\_CHAR>

Provides access to special characters that are not available on the terminal keyboard.

---

### FORMAT

**<SPECIAL\_CHAR>** (*keyword*)

---

### ARGUMENTS

#### **keyword**

Specifies a keyword associated with the special character you want to produce in your output file. The character is always a non-math character. The following are valid keywords:

Keyword	Character
CLOSE_DOUBLE_BRACKET	]]
DAGGER	†
DOUBLE_DAGGER	‡
DOUBLE_HYPHEN	--
FULL_DIAMOND	◆
OPEN_DOUBLE_BRACKET	[[
REGISTERED_SYMBOL	®
SECTION_SIGN	§
TRADEMARK_SYMBOL	™

---

### related tags

- <MATH\_CHAR>
- <MCS>

---

### DESCRIPTION

The <SPECIAL\_CHAR> tag provides access to special characters that are not available on the terminal keyboard.

You can use the <SUBSCRIPT> and <SUPERSCRIPIT> tags with the <SPECIAL\_CHAR> tag to get subscripted and superscripted characters. For example, you can enter the following code to get a superscripted registered symbol:

```
<SUPERSCRIPIT> (<SPECIAL_CHAR> (REGISTERED_SYMBOL))
```



---

**EXAMPLE**

In the following example, the special character ♦ is used in an example of terminal output to show that text wraps at the end of a line.

```
%COPY-I-COPIED, $DISK1:[MYDIRECTORY]MYFILE.PLI;5  
<tag>(special_char) (FULL DIAMOND)  
copied to $DISK1:[YOURDIRECTORY]YOURFILE.PLI;5
```

This example produces the following output:

```
%COPY-I-COPIED, $DISK1:[MYDIRECTORY]MYFILE.PLI;5  
♦  
copied to $DISK1:[YOURDIRECTORY]YOURFILE.PLI;5
```

## <SUBHEADn>

---

## <SUBHEADn>

Marks an unnumbered subsidiary heading.

---

**FORMAT**            <SUBHEADn> (*heading text* [ \ *symbol name*])

---

**ARGUMENTS**        *heading text*  
Specifies the text of the subsidiary heading.

***symbol name***

This is an optional argument. It specifies the name of the symbol used in all references to this heading. Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

**related tags**

- <CHEAD>
- <HEAD1> through <HEAD6>

---

## DESCRIPTION

The <SUBHEAD1> and <SUBHEAD2> tags mark an unnumbered subsidiary heading. You are allowed only two levels of subheadings. Each reflects a logical hierarchy within the structure of the text.

These subheadings are not numbered and do not appear in the table of contents. They cannot be readily used for cross-references and you should use them only when the clarity of your exposition requires such a fine level of distinction.

In some doctypes, the distinction is made between the <SUBHEAD1> and <SUBHEAD2> tags by making the <SUBHEAD2> tag a run-in heading. This means that the following text begins on the same line as the subheading. The doctype controls the text formatter's placement of the text following the <SUBHEAD2> tag. Whether or not you expect the heading to be run-in, always tag the text element following the heading (for example, always mark a paragraph with a <P> tag).

---

## EXAMPLE

The following example shows how to use the <SUBHEADn> tag.

```
<SUBHEAD1>(Using a Subhead\use_of_subhead)
<P>The use of subheads should be restricted to occasions when the clarity
of your exposition absolutely requires one.
```

This example produces the following output:

**Using a Subhead**

The use of subheads should be restricted to occasions when the clarity of your exposition absolutely requires one.

---

## <TABLE>

Begins a sequence of columnar data.

---

**FORMAT**      <TABLE>[(*table caption*[\ *symbol name*])]

---

### **ARGUMENTS**      *table caption*

This is an optional argument. It specifies the text of the caption to be associated with the table. Using this argument indicates that the table is a *formal table*, because the table has a number and a caption and is listed in the table of contents. If you do not specify a *table caption* argument, the table is considered an *informal table*, because it has no number or caption and is not listed in the table of contents.

If the body of the table spans more than a single page of text, the table caption is repeated on each page on which the table continues.

### *symbol name*

This is an optional argument. It specifies the symbolic identifier to be associated with the table. The symbol is assigned a numeric value, which is the current table number. The symbol and its value are placed in the symbol table.

Symbol names must not exceed 31 characters and must only contain alphabetic letters, numbers, or underscores. Do not begin a symbol name with an underscore.

---

### **related tags**

- <ALIGN\_CHAR>
- <ALIGN\_NUMBER>
- <NESTED\_TABLE\_BREAK>
- <REFERENCE>
- <RULE>
- <SPAN>
- <TABLE\_ATTRIBUTES>
- <TABLE\_FILE>
- <TABLE\_HEADS>
- <TABLE\_KEY>
- <TABLE\_KEYREF>
- <TABLE\_ROW>
- <TABLE\_ROW\_BREAK>
- <TABLE\_SETUP>
- <TABLE\_SPACE>

## <TABLE>

- <TABLE\_UNIT>
- <TABLE\_UNIT\_HEADS>
- <VALID\_TABLE\_ROW\_BREAK>

---

### restrictions

Invalid in the context of an <EXAMPLE>, <FOOTNOTE>, <MATH>, or <FIGURE> tag.

---

### required terminator

<ENDTABLE>

---

## DESCRIPTION

The <TABLE> tag begins a sequence of columnar data. A **formal table** has a number and a caption, can be cross-referenced, and is listed in the table of contents. An **informal table** does not have a number or a caption, cannot be cross-referenced, and is not listed in the table of contents.

Note the following when processing a book for Bookreader display:

- You must use the *symbol name* argument to process a formal table for Bookreader display.
- DOCUMENT automatically creates **pop-up windows** for all formal tables.
- When introducing and cross-referencing a formal table, you must use the <REFERENCE> tag to create a **hotspot**. Hotspots are regions in the windows you **click on** with your mouse to access cross-referenced topics.
- You can use the <ONLINE\_POPUP> tag if you want to make informal tables **pop up**.
- Remember that coding a table for printed output may not be suitable for Bookreader output, because the Bookreader's window is narrower than a printed page and the Bookreader's font size is bigger.

---

## EXAMPLES

The following example shows how to produce a two-column table.

```
1 <TABLE>(Rules for Determining Expression Modes\express_modes_tab)
  <TABLE_ATTRIBUTES>(MULTIPAGE)
  <TABLE_SETUP>(2\33)
  <TABLE_HEADS>(Expression\Value Type)
  <TABLE_ROW>(String value\String)
  <TABLE_ROW>(Integer lexical function\Integer)
  <TABLE_ROW_BREAK>(first)
  <TABLE_ROW>(Integer symbol\Integer)
  <TABLE_ROW>(String symbol\String)
  <ENDTABLE>
```

This example produces the following output:

Table n-n Rules for Determining Expression Modes

Expression	Value Type
String value	String
Integer lexical function	Integer
Integer symbol	Integer
String symbol	String

The following example shows how to code a nested table.

```

2 <TABLE>(Food Combining Workshop\schedule_tab)
  <TABLE_SETUP>(2\10)
  <TABLE_HEADS>(Day\Schedule )
  <TABLE_ROW>(Monday\<TABLE>
    <TABLE_SETUP>(2\8)
    <TABLE_ROW>(Morning\Review Workshop Objectives)
    <TABLE_ROW>(Afternoon\Study Protein Groups)
    <TABLE_ROW>(Evening\Study Minerals)
    <ENDTABLE>)
  <TABLE_ROW>(Tuesday\<TABLE>
    <TABLE_SETUP>(2\8)
    <TABLE_ROW>(Morning\Study Fruit Groups)
    <TABLE_ROW>(Afternoon\Study Vegetable Groups)
    <TABLE_ROW>(Evening\Combining Food Groups)
    <ENDTABLE>)
  <TABLE_ROW>(Wednesday\<TABLE>
    <TABLE_SETUP>(2\8)
    <TABLE_ROW>(Morning\Poultry and Fish Diets)
    <TABLE_ROW>(Afternoon\Special Diets)
    <TABLE_ROW>(Evening\Closing Remarks)
    <ENDTABLE>)
  <ENDTABLE>

```

This example produces the following output:

Table n-n Food Combining Workshop

Day	Schedule
Monday	Morning Review Workshop Objectives Afternoon Study Protein Groups Evening Study Minerals
Tuesday	Morning Study Fruit Groups Afternoon Study Vegetable Groups Evening Combining Food Groups
Wednesday	Morning Poultry and Fish Diets Afternoon Special Diets Evening Closing Remarks



## <TABLE\_ATTRIBUTES>

Keyword	Description
WIDE	<p>Specifies that the width of the table exceeds the document's default width for text. Depending on the doctype, this argument is interpreted as follows:</p> <ul style="list-style-type: none"><li>• If the doctype contains a left margin area that is normally used for headings, the table's width spans that area as well as the normal text area.</li><li>• If the document uses a multicolumn format, the table suspends multicolumn output while the table is processed. The table is output across two columns and then multicolumn output is restored.</li></ul>

### related tags

- <TABLE>
- <TABLE\_SETUP>

### restrictions

- Valid only in the context of a <TABLE> tag.
- Must precede the <TABLE\_SETUP> tag.
- WIDE and MAXIMUM might not produce error-free output for monospaced output devices.

## DESCRIPTION

The <TABLE\_ATTRIBUTES> tag specifies special formatting for a table.

## EXAMPLES

The following example uses the KEEP and WIDE attributes.

```
1 <TABLE>(Equivalence Names for Default Process Logical Names\equiv_names_proc_tab)
  <TABLE_ATTRIBUTES>(WIDE\KEEP)
  <TABLE_SETUP>(4\8\10\12)
  <TABLE_HEADS>(Logical Name\Interactive\Batch\Command Procedure)
  <TABLE_ROW>(SYSSCOMMAND\Terminal\Disk\Terminal)
  <TABLE_ROW>(SYSSDISK\Disk\Disk\Disk)
  <TABLE_ROW>(SYSSINPUT\Terminal\Disk\Disk)
  <TABLE_ROW>(SYSSERROR\Terminal\Log file\Terminal)
  <TABLE_ROW>(SYSSLOGIN\Directory\Directory\Directory)
  <TABLE_ROW>(SYSSNET\\)
  <TABLE_ROW>(SYSSOUTPUT\Terminal\Log file\Terminal)
  <TABLE_ROW>(SYSSSCRATCH\Directory\Directory\Directory)
  <TABLE_ROW>(TT\Terminal\Null device\Terminal)
  <ENDTABLE>
```

This example produces the following output:

## <TABLE\_ATTRIBUTES>

**Table n-n Equivalence Names for Default Process Logical Names**

Logical Name	Interactive	Batch	Command Procedure
SY\$COMMAND	Terminal	Disk	Terminal
SY\$DISK	Disk	Disk	Disk
SY\$INPUT	Terminal	Disk	Disk
SY\$ERROR	Terminal	Log file	Terminal
SY\$LOGIN	Directory	Directory	Directory
SY\$NET			
SY\$OUTPUT	Terminal	Log file	Terminal
SY\$SCRATCH	Directory	Directory	Directory
TT	Terminal	Null device	Terminal

2

The same example, but with the MAXIMUM attribute, produces the following output:

**Table n-n Equivalence Names for Default Process Logical Names**

Logical Name	Interactive	Batch	Command Procedure
SY\$COMMAND	Terminal	Disk	Terminal
SY\$DISK	Disk	Disk	Disk
SY\$INPUT	Terminal	Disk	Disk
SY\$ERROR	Terminal	Log file	Terminal
SY\$LOGIN	Directory	Directory	Directory
SY\$NET			
SY\$OUTPUT	Terminal	Log file	Terminal
SY\$SCRATCH	Directory	Directory	Directory
TT	Terminal	Null device	Terminal

3

The same example, but with the SINGLE\_SPACED attribute, produces the following output:

**Table n-n Equivalence Names for Default Process Logical Names**

Logical Name	Interactive	Batch	Command Procedure
SY\$COMMAND	Terminal	Disk	Terminal
SY\$DISK	Disk	Disk	Disk
SY\$INPUT	Terminal	Disk	Disk
SY\$ERROR	Terminal	Log file	Terminal
SY\$LOGIN	Directory	Directory	Directory
SY\$NET			
SY\$OUTPUT	Terminal	Log file	Terminal
SY\$SCRATCH	Directory	Directory	Directory
TT	Terminal	Null device	Terminal



## <TABLE\_ATTRIBUTES>

The following example shows how to code a two-column table using the CONTROLLED attribute to the <TABLE\_ATTRIBUTES> tag, along with the <TABLE\_ROW\_BREAK>(FIRST) and <TABLE\_ROW\_BREAK>(LAST) tags, to override the settings created with the <SET\_TABLE\_ROW\_BREAK\_DEFAULT> tag.

Suppose this table is the first of five tables in a single SDML file. You have set the table row break default to be 3 (using the <SET\_TABLE\_ROW\_BREAK\_DEFAULT> tag). Only for this first table, moreover, you want to explicitly control the table page breaks (instead of the table breaks being allowed only after the third table row), so you allow them after the eighth and before the fourteenth rows using the <TABLE\_ATTRIBUTES>(CONTROLLED), <TABLE\_ROW\_BREAK>(FIRST), and <TABLE\_ROW\_BREAK>(LAST) tags. After this table ends, the <SET\_TABLE\_ROW\_BREAK\_DEFAULT>(3) tag takes effect again for the next four tables.

```
4 <SET_TABLE_ROW_BREAK_DEFAULT>(3)
  <TABLE>(Rules for Determining Expression Modes\express_modes_tab)
  <TABLE_ATTRIBUTES>(MULTIPAGE\CONTROLLED)
  <TABLE_SETUP>(2\43)
  <TABLE_HEADS>(Expression\Value Type)
  <TABLE_ROW>(Integer value\Integer)
  <TABLE_ROW>(String value\String)
  <TABLE_ROW>(Integer lexical function\Integer)
  <TABLE_ROW>(String lexical function\String)
  <TABLE_ROW>(Integer symbol\Integer)
  <TABLE_ROW>(String symbol\String)
  <TABLE_ROW>(Any value .AND. or .OR. any value\Integer)
  <TABLE_ROW>(Any value\Integer)

  <COMMENT>(Break after this)
  <TABLE_ROW_BREAK>(first)
  <TABLE_ROW>(Any value\Integer)
  <TABLE_ROW>(Integer value\Integer)
  <TABLE_ROW>(String value\String)
  <TABLE_ROW>(Integer lexical function\Integer)
  <TABLE_ROW>(String lexical function\String)
  <COMMENT>(Do not break after this)
  <TABLE_ROW_BREAK>(last)

  <TABLE_ROW>(Integer symbol\Integer)
  <TABLE_ROW>(String symbol\String)
  <TABLE_ROW>(Any value .AND. or .OR. any value\Integer)
  <TABLE_ROW>(Any value\Integer)
  <TABLE_ROW>(Any value\Integer)

<ENDTABLE>
```

## <TABLE\_FILE>

---

## <TABLE\_FILE>

Causes a separate file containing a formal table to be included in the SDML input file.

---

### FORMAT

**<TABLE\_FILE>**( { *file spec*  
                  { *logical name* } )

---

### ARGUMENTS

#### ***file spec***

Specifies the file specification of the file to be included.

#### ***logical name***

Specifies the logical name of the file to be included. You can define a logical name in your profile using an <INCLUDES\_FILE> tag. If you do not define the logical name in the profile with the <INCLUDES\_FILE> tag, be sure you define the logical name elsewhere before processing the file with VAX DOCUMENT.

---

### related tags

- <TABLE>
- 

### restrictions

Valid only in the context of a <TABLE> tag.

---

### DESCRIPTION

The <TABLE\_FILE> tag causes a separate file containing a formal table to be included in the SDML input file. (It is identical in action to the <INCLUDE> tag.) The included file should be an SDML file containing a completely coded table.

By keeping a table in a separate SDML file, you can include the table in more than one document, or in chapters of the same document, without having to reproduce the code. In each location, the table is given a different number.

---

### EXAMPLES

The following example shows how to use the <TABLE\_FILE> tag to include a wide table that is contained in a separate file, called STANDARD\_FILETYPES.SDML.

Note that the included table file will not process individually. It will only process when you include it in another SDML file that specifies the <TABLE> and <ENDTABLE> tags.

**1** <TABLE>(Default VMS File Types\stand\_filetypes\_tab)  
<TABLE\_ATTRIBUTES>(wide)  
<TABLE\_FILE>(standard\_filetypes.sdml)  
<ENDTABLE>

In the following example, the file ENDLIST.SDML contains all the tags required for the table.

**2** <REFERENCE>(command\_sum\_tab) lists all the commands.  
<TABLE\_FILE>(endlist.sdml)

The tags in ENDLIST.SDML are:

```
<TABLE>(Command Summary\command_sum_tab)
<TABLE_ATTRIBUTES>(Multipage)
<TABLE_SETUP>(2\18)
<TABLE_HEADS>(Command\Description)
<TABLE_ROW>( . . . )
.
.
.
<ENDTABLE>
```

This example assumes that the table is being used in only one document. Placing the table in a separate file allows you to process it individually, or in the context of the file in which it occurs.



This example produces the following output:

**Table n-n Rules for Determining Expression Modes**

<b>Expression</b>	<b>Value Type</b>
Integer value	Integer
String value	String

<b>Value Type</b>	<b>Expression</b>
Integer	Integer lexical function
String	String lexical function
Integer	Integer symbol
String	String symbol

## <TABLE\_KEY>

---

## <TABLE\_KEY>

Begins a key or legend for a table.

---

### FORMAT <TABLE\_KEY>

---

**ARGUMENTS** *None.*

---

**related tags**

- <FOOTNOTE>
- <TABLE>
- <TABLE\_KEYREF>
- <TABLE\_SETUP>

---

**restrictions**

Valid only in the context of a <TABLE> tag.

Must appear immediately after the <TABLE\_SETUP> tag, either before or after any <FOOTNOTE> tags.

When you use a <TABLE\_KEY> tag, the number of possible footnotes for the same table is reduced from 12 to 11.

The tags that you can use between the <TABLE\_KEY> and <ENDTABLE\_KEY> tags are restricted to the <P>, <LIST>, and <EMPHASIS> tags. For example, you cannot use a <TABLE\_ROW> tag within the table key.

Invalid in a nested table (a table within a table). The <TABLE\_KEY> tag must follow the <TABLE\_SETUP> tag for the outermost table.

---

**required terminator**

<ENDTABLE\_KEY>

---

### DESCRIPTION

The <TABLE\_KEY> tag begins a key or legend for a table. Abbreviations or special terms are often used in a table, either in the column headings or in the entries of the table. The table may then need a key or legend printed below it to explain the special terms. The <TABLE\_KEY> tag begins such a table key.

A table key differs from a table footnote in that the table key is not numbered and does not refer to a callout in the table. You can declare only one table key for a table, whereas up to 12 footnotes are possible for a table that does not include a table key, or up to 11 footnotes for a table that does include a table key.

Declare the table key immediately after the <TABLE\_SETUP> tag that begins the table and before the first tag that begins the table rows. This placement is identical to the placement for footnotes in a table. If footnotes are also present, they either can precede or follow the table key declaration.

Use the <TABLE\_KEYREF> tag to specify where you want the table key to be output. When the table is a multipage table and if you want the table key printed at the foot of any or all portions of the table, place the <TABLE\_KEYREF> tag anywhere in the argument list to the <TABLE\_HEADS> tag. For a book you create for Bookreader, the table key will always appear at the end of the table.

If you want the table key to be printed only on specified pages of a multipage table, place the <TABLE\_KEYREF> tag in any <TABLE\_ROW> tag that corresponds to the different pages. You may have to print your table before you can determine which <TABLE\_ROW> tags correspond to which pages.

---

**EXAMPLE**

The following example shows how to use the <TABLE\_KEY> tag.

```
<table>(Compatability of Lock Modes\lock_tab)
  <table_attributes>(wide\keep)
  <table_setup>(7\10\5\5\5\5)

  <table_key>
    <emphasis>(Key to Lock Modes\bold)
    <list>(simple)
      <le>NL---Null lock
      <le>CR---Concurrent read
      <le>CW---Concurrent write
      <le>PR---Protected read
      <le>PW---Protected write
      <le>EX---Exclusive lock
    <ENDLIST>
  <endtable_key>

  <table_heads>(Mode of Requested\
    <span>(5)Mode of Currently Granted Locks<rule>)
    <table_heads>(Lock<TABLE_KEYREF>\NL\CR\CW\PR\PW\EX)
    <table_row>(NL\Yes\Yes\Yes\Yes\Yes\Yes)
    <table_row>(CR\Yes\Yes\Yes\Yes\Yes\No)
    <table_row>(CW\Yes\Yes\Yes\No\No\No)
    <table_row>(PR\Yes\Yes\No\Yes\No\No)
    <table_row>(PW\Yes\Yes\No\No\No\No)
    <table_row>(EX\Yes\No\No\No\No\No)
  <endtable>
```

This example produces the following output:

## <TABLE\_KEY>

Table n-n Compatibility of Lock Modes

Mode of Requested Lock	Mode of Currently Granted Locks					
	NL	CR	CW	PR	PW	EX
NL	Yes	Yes	Yes	Yes	Yes	Yes
CR	Yes	Yes	Yes	Yes	Yes	No
CW	Yes	Yes	Yes	No	No	No
PR	Yes	Yes	No	Yes	No	No
PW	Yes	Yes	No	No	No	No
EX	Yes	No	No	No	No	No

### Key to Lock Modes

NL—Null lock  
CR—Concurrent read  
CW—Concurrent write  
PR—Protected read  
PW—Protected write  
EX—Exclusive lock



---

## <TABLE\_KEYREF>

Specifies that a table key be printed below the table (or any portion of a multipage table) in which this tag appears.

---

**FORMAT**            <TABLE\_KEYREF>

---

**ARGUMENTS**        *None.*

---

**related tags**        • <TABLE\_KEY>

---

**restrictions**        Valid only in the context of a table in which you specify the <TABLE\_KEY> tag.

Invalid in nested tables (a table within a table).

You can use this tag only at the end of an argument list and you cannot embed it in text. The end of an argument list is indicated by a backslash (\) or a closing parenthesis ()).

---

**DESCRIPTION**        The <TABLE\_KEYREF> tag specifies that a table key be printed below the table (or any portion of a multipage table) in which this tag appears. The table key is printed only if the <TABLE\_KEYREF> tag appears in the table.

If you want the table key printed at the foot of all portions of a multipage table, place the <TABLE\_KEYREF> tag anywhere in the argument list to the <TABLE\_HEADS> tag.

If you want the table key to be printed only on specified pages of a multipage table, place the <TABLE\_KEYREF> tag anywhere in the argument list to the <TABLE\_ROW> tag that corresponds to the different pages. You may have to print your table before you can determine which <TABLE\_ROW> tags correspond to which pages.

---

**EXAMPLE**            See the example in the discussion of the <TABLE\_KEY> tag.



```
<TABLE>(Rules for Determining Expression Modes\express_modes_tab)
  <TABLE_ATTRIBUTES>(WIDE\MULTIPAGE)
  <TABLE_SETUP>(2\43)
  <TABLE_HEADS>(Expression\Value Type)
  <TABLE_ROW>(Integer value\Integer)
  <TABLE_ROW>(String value\String)
  <TABLE_ROW>(Integer lexical function\Integer)
  <TABLE_ROW>(String lexical function\String)
  <TABLE_ROW>(Integer symbol\Integer)
  <TABLE_ROW>(String symbol\String)
  <TABLE_ROW>(Any value .AND. or .OR. any value\Integer)
  <TABLE_ROW>(Any value\Integer)
  <TABLE_ROW>(Any value\Integer)
<ENDTABLE>
```

This example produces the following output:

**Table n-n Rules for Determining Expression Modes**

<b>Expression</b>	<b>Value Type</b>
Integer value	Integer
String value	String
Integer lexical function	Integer
String lexical function	String
Integer symbol	Integer
String symbol	String
Any value .AND. or .OR. any value	Integer
Any value	Integer
Any value	Integer

## <TABLE\_ROW\_BREAK>

---

## <TABLE\_ROW\_BREAK>

Specifies the boundaries in which a long table can be broken onto a new page.

---

### FORMAT

**<TABLE\_ROW\_BREAK>**( { *FIRST* }  
                                  { *LAST* } )

---

### ARGUMENTS

***FIRST***  
***LAST***

If you use <TABLE\_ATTRIBUTES>(CONTROLLED), these keyword arguments specify the bounds within which a multipage table may be broken. After you specify the <TABLE\_ROW\_BREAK>(FIRST) tag, any row in the table up to the occurrence of the <TABLE\_ROW\_BREAK>(LAST) tag is treated as a suitable page-breaking point.

---

### related tags

- <NESTED\_TABLE\_BREAK>
- <TABLE>
- <TABLE\_ATTRIBUTES>(CONTROLLED)
- <TABLE\_UNIT>
- <VALID\_TABLE\_ROW\_BREAK>

---

### restrictions

Ignored in a table specified with the <TABLE\_ATTRIBUTES>(KEEP) tag and in a nested table.

Enabled only by the <TABLE\_ATTRIBUTES>(CONTROLLED) tag.

---

### DESCRIPTION

The <TABLE\_ROW\_BREAK> tag specifies the boundaries in which a long table can be broken onto a new page. You must, however, use the <TABLE\_ATTRIBUTES>(CONTROLLED) tag in order for the <TABLE\_ROW\_BREAK> tag to work. The <TABLE\_ROW\_BREAK> tag has no effect for Bookreader output.

Tables in VAX DOCUMENT are, by default, multipage tables. If you do not specify <TABLE\_ROW\_BREAK>(FIRST) and <TABLE\_ROW\_BREAK>(LAST) to provide the text formatter with reasonable guidelines to break tables across pages, the text formatter will nonetheless select page breaks. The page breaks selected will not always be suitable. Therefore, using the <TABLE\_ROW\_BREAK> tag is recommended.

If the table includes long table units and you want to allow them to break across pages, use the <TABLE\_ROW\_BREAK>(FIRST) and <TABLE\_ROW\_BREAK>(LAST) tags within the bounds of each <TABLE\_UNIT> and <ENDTABLE\_UNIT> tag.

## <TABLE\_ROW\_BREAK>

By default, the text formatter will not break a page within a table row. With some restrictions, you can specify valid page breaks within table rows using the <VALID\_TABLE\_ROW\_BREAK> tag. See the <VALID\_TABLE\_ROW\_BREAK> tag description for more information.

You can give, with some restrictions, a nested table (that is, a table within a table) valid page breaks using the <NESTED\_TABLE\_BREAK> tag. See the <NESTED\_TABLE\_BREAK> tag description for more information.

### EXAMPLE

The following example shows how to produce a two-column table using the <TABLE\_ROW\_BREAK>(FIRST) and <TABLE\_ROW\_BREAK>(LAST) tags. Note that you must use the CONTROLLED argument to the <TABLE\_ATTRIBUTES> tag to enable the <TABLE\_ROW\_BREAK>(FIRST) and <TABLE\_ROW\_BREAK>(LAST) tags.

```
<TABLE>(Rules for Determining Expression Modes\express_modes_tab)
  <TABLE_ATTRIBUTES>(WIDE\MULTIPAGE\CONTROLLED)
  <TABLE_SETUP>(2\43)
  <TABLE_HEADS>(Expression\Value Type)
  <TABLE_ROW>(Integer value\Integer)
  <TABLE_ROW>(String value\String)
  <TABLE_ROW>(Integer lexical function\Integer)

  <COMMENT>(break after this)
  <TABLE_ROW_BREAK>(first)
  <TABLE_ROW>(String lexical function\String)
  <TABLE_ROW>(Integer symbol\Integer)
  <COMMENT>(do not break after this)
  <TABLE_ROW_BREAK>(last)

  <TABLE_ROW>(String symbol\String)
  <TABLE_ROW>(Any value .AND. or .OR. any value\Integer)
  <TABLE_ROW>(Any value\Integer)
  <TABLE_ROW>(Any value\Integer)
  <TABLE_ROW>(Integer value\Integer)
  <TABLE_ROW>(String value\String)
  <TABLE_ROW>(Integer lexical function\Integer)
  <TABLE_ROW>(String lexical function\String)
  <TABLE_ROW>(Integer symbol\Integer)
  <TABLE_ROW>(String symbol\String)
  <TABLE_ROW>(Any value .AND. or .OR. any value\Integer)
  <TABLE_ROW>(Any value\Integer)
  <TABLE_ROW>(Any value\Integer)

<ENDTABLE>
```

This example produces the following output (except for Bookreader output, where the <TABLE\_ROW\_BREAK> tag has no effect):

**Table n–n Rules for Determining Expression Modes**

Expression	Value Type
Integer value	Integer
String value	String
Integer lexical function	Integer
String lexical function	String

## <TABLE\_ROW\_BREAK>

**Table n-n (Cont.) Rules for Determining Expression Modes**

<b>Expression</b>	<b>Value Type</b>
Integer symbol	Integer
String symbol	String
Any value .AND. or .OR. any value	Integer
Any value	Integer
Any value	Integer
Integer value	Integer
String value	String
Integer lexical function	Integer
String lexical function	String
Integer symbol	Integer
String symbol	String
Any value .AND. or .OR. any value	Integer
Any value	Integer
Any value	Integer

---

## <TABLE\_SETUP>

Declares up to 9 columns in a table and the approximate width to be assigned to each column.

---

**FORMAT**            <TABLE\_SETUP>(number \ width<sub>1</sub>[ . . . \ width<sub>number-1</sub>])

---

**ARGUMENTS**        *number*  
Specifies the number of columns in the table. Must be a numeric argument in the range of 2 through 9.

*width<sub>1</sub>*

*width<sub>number-1</sub>*

Specify the approximate widths of columns 1 through *number* - 1. Do not specify the width of the last column. This column occupies the space left between the second-to-last column and the right margin.

The width unit is measured in characters. This is an approximate measurement, however, because the width of each character depends on the fonts being used in the doctype. You may need to modify the *width* argument after processing the table and viewing the output.

---

**related tags**

- <TABLE>
- <TABLE\_ATTRIBUTES>

---

**restrictions**

- Valid only in the context of a <TABLE> tag.
- If you do not include the <TABLE\_SETUP> tag in a sequence of <TABLE> tags, no table is included in your output.
- Must appear after the <TABLE\_ATTRIBUTES> tag and before the <TABLE\_ROW> tags in the table tag sequence.

---

**DESCRIPTION**     The <TABLE\_SETUP> tag Declares up to 9 columns in a table and the approximate width to be assigned to each column.

In the *width* argument, define the width of each column except the last, which is supplied automatically.

Values you specify in the <TABLE\_SETUP> tag are checked during processing to determine whether the values exceed or approach the table's limit for the given doctype. Warning or error messages are issued as appropriate.

If the table column widths do not allow the table to fit within the page width, the text processor substitutes a smaller point size for the font in which the table is set. It then attempts again to fit the table within the text margin.

## <TABLE\_SETUP>

---

### EXAMPLE

The following example shows how to use the <TABLE\_SETUP> tag.

```
<TABLE>(Names of Cheeses from Around the World\cheeses_tab)
<TABLE_ATTRIBUTES>(wide\multipage)
<TABLE_SETUP>(4\14\14\14)
<TABLE_HEADS>(Cheese Name\Location of Production\Type\Color)
<TABLE_ROW>(Spreadable\Great Britain\Curdled\White)
<TABLE_ROW>(Smooth\Switzerland\Aged and Tasty\Off yellow)
<TABLE_ROW>(Chunky\American\Bland\Yellow)
<ENDTABLE>
```

This example produces the following output:

**Table n-n Names of Cheeses from Around the World**

---

<b>Cheese Name</b>	<b>Location of Production</b>	<b>Type</b>	<b>Color</b>
Spreadable	Great Britain	Curdled	White
Smooth	Switzerland	Aged and Tasty	Off yellow
Chunky	American	Bland	Yellow

---



---

## <TABLE\_SPACE>

Marks a blank space in a table where information will be pasted in during final production.

---

**FORMAT**            <TABLE\_SPACE>(value \ text)

---

**ARGUMENTS**        *value*  
Specifies the amount of vertical space (in picas) you want left on the page.

*text*  
Specifies text that describes the status of the table (for example, "To Be Set," an art file number, or some other note). The text is output in the middle of the space left for the table.

---

**restrictions**        You must use this tag as an argument to a <TABLE\_ROW> tag.

---

**DESCRIPTION**        The <TABLE\_SPACE> tag marks a blank space in a table where information will be pasted in during final production. Specify the *value* argument in picas. There are 6 picas in an inch (and 12 points in a pica). Thus, if the table you want pasted in is four inches high, you should specify 24 picas. If you do not specify a value, a default value of 8 picas is used.

If you specify some descriptive text in the second argument, that text is output in the middle of the space left for the table.

---

**EXAMPLE**            The following example shows how to code a table with the <TABLE\_SPACE> tag. Each <TABLE\_SPACE> tag reserves space inside of the table row for a diagram.

```
<TABLE>(Script Control Blocks\script_tab)
  <TABLE_ATTRIBUTES>(WIDE)
  <TABLE_SETUP>(2\22)
  <TABLE_HEADS>(Block ID\Layout)
  <TABLE_ROW>(SCBEG; starts the scripting sequence\<TABLE_SPACE>(4\SCBEG diagram))
  <TABLE_ROW>(SCMID; gives the body of the script\<TABLE_SPACE>(4\SCMIDdiagram))
  <TABLE_ROW>(SCEND; ends the scripting sequence\<TABLE_SPACE>(4\SCEND diagram))
<ENDTABLE>
```

This example produces the following output:

## <TABLE\_SPACE>

**Table n-n Script Control Blocks**

<b>Block ID</b>	<b>Layout</b>
SCBEG; starts the scripting sequence	SCBEG diagram
SCMID; gives the body of the script	SCMID diagram
SCEND; ends the scripting sequence	SCEND diagram

---

---

## <TABLE\_UNIT>

Begins a portion of a table containing rows that you want to group as a logical unit.

---

### FORMAT <TABLE\_UNIT>

---

**ARGUMENTS** *None.*

**related tags**

- <TABLE>
- <TABLE\_ROW\_BREAK>
- <TABLE\_UNIT\_HEADS>

---

**restrictions** Valid only in the context of a <TABLE> tag.  
You must not declare the <TABLE\_ATTRIBUTE> tag with the KEEP argument in the context of the <TABLE\_UNIT> tag.

---

**required terminator** <ENDTABLE\_UNIT>

---

**DESCRIPTION** The <TABLE\_UNIT> tag begins a portion of a table containing rows that you want to group as a logical unit. You can give the table unit a heading.  
If the table units are long and you want to allow them to break across pages, use the <TABLE\_ROW\_BREAK>(FIRST) and <TABLE\_ROW\_BREAK>(LAST) tags within the bounds of each <TABLE\_UNIT> and <ENDTABLE\_UNIT> tag.

---

**EXAMPLE** The following example shows how to use the <TABLE\_UNIT> tag.

```
<Table>(String Passing Techniques Used by the Run-Time Library\Str_Tech_Tab)
<TABLE_ATTRIBUTES>(WIDE\MULTIPAGE)
<Table_Setup>(4\20\7\10)
<Table_Heads>(\<SPAN>(3)String Descriptor Fields<RULE>)
<Table_Heads>(String Type\Class\Length\Pointer)
<Table_Unit>
<Table_Unit_Heads>(<SPAN>(4\LEFT)Input Argument to Procedures)
<Table_Row>(Input string passed by descriptor\Read\Read\Read)
<Endtable_Unit>
```

## <TABLE\_UNIT>

```

<Table_Unit>
<Table_Unit_Heads>(<SPAN>(4\LEFT)Output Argument from Procedures;
Called Procedure Assumes the Descriptor Class)
<Table_Row>(Output string passed by descriptor,
fixed-length\Ignored\Read\Read)
<Table_Row>(Output string passed by descriptor, dynamic\Ignored\Read, may be
modified\Read, may be modified)
<Endtable_Unit>

<Table_Unit>
<Table_Unit_Heads>(<SPAN>(4)Output Argument from Procedures, Calling Program
Specifies the Descriptor Class in the Descriptor)
<Table_Row>(Output string, fixed-length ---DSC$K_CLASS = S, Z, A, NCA, SD
\Read\Read\Read)
<Table_Row>(Output string, dynamic ---DSC$K_CLASS_D
\Read\Read, may be modified\Read, may be modified)
<Table_Row>(Output string, varying-length ---DSC$K_CLASS_VS
\Read\MAXSTRLEN is read; CURLLEN is modified\Read)
<Endtable_Unit>

<Endtable>

```

This example produces the following output:

**Table n-n String Passing Techniques Used by the Run-Time Library**

String Type	String Descriptor Fields		
	Class	Length	Pointer
<b>Input Argument to Procedures</b>			
Input string passed by descriptor	Read	Read	Read
<b>Output Argument from Procedures; Called Procedure Assumes the Descriptor Class</b>			
Output string passed by descriptor, fixed-length	Ignored	Read	Read
Output string passed by descriptor, dynamic	Ignored	Read, may be modified	Read, may be modified
<b>Output Argument from Procedures, Calling Program Specifies the Descriptor Class in the Descriptor</b>			
Output string, fixed-length —DSC\$K_CLASS = S, Z, A, NCA, SD	Read	Read	Read
Output string, dynamic —DSC\$K_CLASS_D	Read	Read, may be modified	Read, may be modified
Output string, varying-length —DSC\$K_CLASS_VS	Read	MAXSTRLEN is read; CURLLEN is modified	Read



## <TABLE\_UNIT\_HEADS>

When the heading text for a table column requires multiple lines, you can do the following:

- Supply the text as a long argument to a single <TABLE\_UNIT\_HEADS> tag. The text is automatically displayed on the required number of lines.
- Supply the text of the heading as shorter arguments to successive <TABLE\_UNIT\_HEADS> tags. In this way, you can control how the text is displayed on successive lines of the heading.

If a page break occurs in the table unit, all lines of the table unit heading are repeated at the top of the next page.

---

### EXAMPLE

See the example in the discussion of the <TABLE\_UNIT> tag.

---

## <TAG>

Labels a tag and its arguments.

---

**FORMAT**      **<TAG>**(*tag name* [*\ tag arg-1 . . . [\ tag arg-8]]*)

---

**ARGUMENTS**      ***tag name***  
Specifies the name of the tag.

***tag arg***  
This is an optional argument. It specifies an argument to the tag. The number of arguments you specify depends on the number of arguments available to the tag you are labeling.

---

**related tags**      • <LITERAL>

---

**restrictions**      Invalid in the context of a <MATH> tag.

---

**DESCRIPTION**      The <TAG> tag labels a tag and its arguments.

---

**EXAMPLES**      The following example shows how to use the <TAG> tag.

**1**      <P>You use the <TAG>(p) tag to begin a new paragraph.

This example produces the following output:

You use the <P> tag to begin a new paragraph.

The following example shows how to use the <TAG> tag with an argument.

**2**      Use the <TAG>(code\_example\WIDE) tag if your example has long lines.

This example produces the following output:

Use the <CODE\_EXAMPLE>(WIDE) tag if your example has long lines.

The following example shows the difference in output caused by the <TAG> tag and the <LITERAL> tag.

**3**      A writer can use the <tag>(line\_art) tag to label a rough sketch. A writer should not use the <literal><icon><endliteral> tag for that purpose.

This example produces the following output:

A writer can use the <LINE\_ART> tag to label a rough sketch. A writer should not use the <icon> tag for that purpose.

## <TITLE>

---

## <TITLE>

Labels the title used on either a title page or part page. (A part page is a divider page for a new part of a document.)

---

**FORMAT**            **<TITLE>** (*title line-1*[\ *title line-2*[\ *title line-3*]])

---

**ARGUMENTS**        ***title line-n***  
Specifies one to three separate lines of text for the title.

---

**related tags**

- <PART\_PAGE>
- <TITLE\_PAGE>

---

**restrictions**

Valid only in the context of a <PART\_PAGE> or a <TITLE\_PAGE> tag.

Must appear immediately after the <PART\_PAGE> or <TITLE\_PAGE> tag.

The <RIGHT\_LINE> and <CENTER\_LINE> tags are invalid within an argument to the <TITLE> tag.

Accepts only two *title line* arguments when used within a part page or in the SOFTWARE.BROCHURE doctype.

---

**DESCRIPTION**     The <TITLE> tag labels the title used on either a title page or part page. (A part page is a divider page for a new part of a document.)

On a part page, the <TITLE> tag accepts one or two title text arguments, and the title text is kept on the same line.

In the SOFTWARE.BROCHURE doctype, the <TITLE> tag accepts one or two title text arguments. If you specify only the first argument, the title text is placed at the top of the first page and at the bottom of each successive page; if you also specify the optional second argument, that title text is placed at the bottom of each page. The title text placed at the bottom of the page by the SOFTWARE.BROCHURE <TITLE> tag will be overridden by the text argument of any subsequent <CHAPTER> tags. Refer to *VAX DOCUMENT Using Dotypes and Related Tags* for more information on using the tag in the SOFTWARE.BROCHURE doctype.

---

**EXAMPLE**            See the example in the discussion of the <FRONT\_MATTER> tag.



---

## <TITLE\_PAGE>

Labels the beginning of a title page and enables the title page tags.

---

**FORMAT**            <TITLE\_PAGE>

---

**ARGUMENTS**        *None.*

---

**restrictions**        Valid only in the context of a <FRONT\_MATTER> tag.  
If you are using a preface, you must terminate the <TITLE\_PAGE> tag before you use the <PREFACE> tag. If you are using a copyright page, you must terminate the <TITLE\_PAGE> tag before you use the <COPYRIGHT\_PAGE> tag.

---

**required terminator**    <ENDTITLE\_PAGE>

---

**DESCRIPTION**        The <TITLE\_PAGE> tag labels the beginning of a title page and enables the following title page tags:

- <ABSTRACT>
- <ORDER\_NUMBER>
- <REVISION\_INFO>
- <TITLE>

DOCUMENT considers the title page to be an **online topic**. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information about online topics.

Your Bookreader document must contain a title page with the <TITLE\_PAGE> and <ENDTITLE\_PAGE> tags. The <FRONT\_MATTER> tag and the **LMF** (License Management Facility) tags must appear before the <TITLE\_PAGE> tag. Do not place any tags that generate text before the <TITLE\_PAGE> tag.

---

**EXAMPLE**            See the example in the discussion of the <FRONT\_MATTER> tag.

**<U>**

---

**<U>**

Labels the user portion of a dialog between user and system in an interactive example.

---

**FORMAT**            **<U>**(*text*)

---

**ARGUMENTS**        ***text***  
Specifies the text of the user input.

---

**related tags**

- `<INTERACTIVE>`
- `<S>`

---

**DESCRIPTION**     The `<U>` tag labels the user portion of a dialog between user and system in an interactive example. You must identify both parts of an example containing this type of dialog in order to differentiate the two types of text in the source code, the output, or both. In Bookreader output, the text that you label with the `<U>` tag appears in boldface.

In the SOFTWARE doctype, you can use the `<U>` and `<S>` tags to differentiate the system and user text inside of examples created with the `<EXAMPLE_SEQUENCE>` and `<EXI>` tags. For more information on this doctype, refer to *VAX DOCUMENT Using Doctypes and Related Tags*.

---

**EXAMPLES**            The following example shows how to use the `<U>` tag.

**1**     `<P>`The user prompt `<U>`(send) indicates you can send a command.

This example produces the following output:

The user prompt *send* indicates you can send a command.

The following example shows how to code dialog between both the system and the user. In a dialog, the `<S>` and `<U>` tags must be used between `<INTERACTIVE>` and `<ENDINTERACTIVE>` tags. Note that you should specify, within the argument to the `<S>` tag, whatever space follows the system prompt.

**2**     `<P>`The following example of VAXMAIL contains messages from both the system and a user of the system:  
`<INTERACTIVE>`  
`<U>`(mail)  
`<S>`(MAIL) )`<U>`(send)  
`<S>`(To: )`<U>`(nodename::Courtney)  
`<S>`(%MAIL-E-NOSUCHUSR, no such user COURTNEY at node NODENAME)  
`<ENDINTERACTIVE>`

This example produces the following output:

The following example of VAXMAIL contains messages from both the system and a user of the system:

```
mail
MAIL> send
To: nodename::Courtney
%MAIL-E-NOSUCHUSR, no such user COURTNEY at node NODENAME
```



The following example shows how to use the <UNDERLINE> tag with the <ENDUNDERLINE> tag. This example produces the same output as the first example.

**2** <P>If you use the appropriate flags, your output may look like this:  
<P>  
<SAMPLE\_TEXT>  
<UNDERLINE>The Decline and Fall of the Roman Empire<ENDUNDERLINE>  
<ENDSAMPLE\_TEXT>

## <UPDATE\_RANGE>

---

## <UPDATE\_RANGE>

Marks the location at which a new section of updated pages begins.

---

### FORMAT

**<UPDATE\_RANGE>**(*start page* \ { *end page* } )  
*EOF*

---

### ARGUMENTS

#### ***start page***

Specifies the page number from the printed documentation of the first updated page you want to print.

#### ***end page***

Specifies the last page to be included in a set of update pages. VAX DOCUMENT automatically generates point-numbered pages if text within the bounds of the <UPDATE\_RANGE>(START PAGE) and <UPDATE\_RANGE>(END PAGE) tags will not fit within those pages.

#### ***EOF***

This keyword specifies that the update range continues to the end of a chapter or section.

---

### related tags

- <MARK>
- <REVISION>

---

### restrictions

The <UPDATE\_RANGE> tag has no effect on Bookreader output.

Valid only in the context of a <REVISION> tag, and the <REVISION> tag must specify the keyword UPDATE. If you do not specify the <REVISION> tag, no output is generated from the <UPDATE\_RANGE> and <ENDUPDATE\_RANGE> tags.

---

### required terminator

<ENDUPDATE\_RANGE>

---

## DESCRIPTION

The <UPDATE\_RANGE> tag marks the location at which a new section of updated pages begins. Files coded with the <REVISION>(UPDATE \ *update info*) tag and containing <UPDATE\_RANGE> tags process as follows:

- When the text formatter processes a file that has been marked as an update, it processes all text and commands in the file, but does not produce actual output for the DVI file, except for those pages marked within an update range.

- When the text formatter reaches the beginning of an update range, it sets the page number to the page number specified as the start of the update. Therefore, it is not important that previous versions of the text formatter file were modified for pagination during final production. All pages processed outside of an update range are not output.

The following rules apply to the placement of the <UPDATE\_RANGE> tag:

- If the text on the first update page is in the middle of a text element (for example, a paragraph, a list element, a code example), then the <UPDATE\_RANGE> tag must precede the word of text that is the first word on the page.
- If the text on the first update page represents the beginning of a new text element (for example, the <P>, <LIST>, <LE>, <CODE\_EXAMPLE> tags, and so on), you must place the <UPDATE\_RANGE> tag immediately preceding the tag for the text element.
- If an update range begins on a page that starts with a continued table, the file must specify the start of the update range on the odd-numbered page preceding the beginning of that table, or (if the table begins on an odd-numbered page) the page on which the table begins. If these pages are not to be a part of the update, you can discard them.
- If one or more pages before an update range begins contain a **floating** figure or example, you must modify the <FIGURE\_ATTRIBUTES> or <EXAMPLE\_ATTRIBUTES> tags to specify the **KEEP** argument. This prevents the text formatter from floating the figure or example to the top of the first update page.

Put a <COMMENT> tag in the file to indicate that the modification was made for the purposes of the update only. For example:

```
<COMMENT>(KEEP added to example for update only...)
```

When the file is subsequently revised, you can remove the **KEEP** arguments.

- When you process a file that contains the <REVISION>(UPDATE) tag, the table of contents and index are handled as follows: if you specify /CONTENTS and /INDEX on the command line, you receive a table of contents and an index even if you do not specify the <CONTENTS\_FILE> and <INDEX\_FILE> tags within the <UPDATE\_RANGE> and <ENDUPDATE\_RANGE> tags.

---

## EXAMPLE

In the following example, the updated material begins on page 5 and continues through page 24. When page 24 is reached, the page numbering becomes 24.1, 24.2, and so on, until the end of the update range is reached. You must place the <ENDUPDATE\_RANGE> tag in the source file at the position corresponding to the place at which an update sequence ends.

## <UPDATE\_RANGE>

<REVISION>(UPDATE\November 1990)

.  
.

<UPDATE\_RANGE>(5\24)

<P>

The first sentence on page 5 goes here.

.  
.

20 or more pages of modified text go here.

.  
.

The last sentence on page 24.n goes here.

<ENDUPDATE\_RANGE>



---

## <UPPERCASE>

Labels text that you want to appear in uppercase in the final output.

---

**FORMAT**      <UPPERCASE>(text)

---

**ARGUMENTS**    *text*  
Specifies the text you want to appear in uppercase.

---

**related tags**    • <LOWERCASE>

---

**DESCRIPTION**    The <UPPERCASE> tag labels text that you want to appear in uppercase in the final output. In your book, there may be a text element, such as a heading, that normally appears in lowercase. Use the <UPPERCASE> tag if you need to overcome the default case in one of your tags and ensure that the result in the final output appears in uppercase.

---

**EXAMPLE**        In the following example, assume that the doctype being used causes the <HEAD2> tag to output a heading that is in lowercase, no matter what the case of the text passed to it. The <UPPERCASE> tag overrides the default in this <HEAD2> tag.

```
<HEAD2>(Here is an example of <UPPERCASE>(uppercase) text.)
```

This example produces the following output:

Here is an example of UPPERCASE text.



---

**EXAMPLE**

The following example shows how to use the <USER\_I\_MESSAGE> tag to flag a section of a file that requires further work.

```
<USER_I_MESSAGE>(Section 2 is incomplete and requires  
information from Tom Smith.)
```

Suppose the file containing this message is called "Chapter\_1.SDML." Depending on how you process this file, the message appears on your screen, in the .LIST file, or in the .LOG file as follows:

```
%TAG-I-USER_IMSG, Section 2 is incomplete and requires information  
from Tom Smith.  
Line is 68 of file part2.sdml
```

## <USER\_W\_MESSAGE>

---

## <USER\_W\_MESSAGE>

Sends a warning message to the terminal, .LIS, or .LOG file during processing of a file.

---

**FORMAT**            <USER\_W\_MESSAGE> (*warning text*)

---

**ARGUMENTS**        *warning text*  
Specifies the text you want to appear on the terminal or in the .LIS or .LOG file.

---

**related tags**        • <USER\_I\_MESSAGE>

---

**restrictions**        For batch processing, you must process the file with the /LOG qualifier on the command line in order to see any <USER\_W\_MESSAGE> tag messages.

The message text is limited to 150 characters.

---

**DESCRIPTION**        The <USER\_W\_MESSAGE> tag sends a warning message to the terminal, .LIS, or .LOG file during processing of a file.

If you process the file interactively, the message will appear on your screen. If you process the file interactively and use the /LIST qualifier on the command line, the message is also listed in the .LIS file.

If you process the file in batch and specify the /LOG qualifier on the command line, the message is listed in the .LOG file.

**Note:** This type of message is counted by the tag translator as a warning message. After thirty warning messages, tag translation halts. Therefore, use the <USER\_W\_MESSAGE> tag for generating messages only if you are issuing a few messages. For a file that contains many messages, tag messages with <USER\_I\_MESSAGE> tags and process the file with the /LOG qualifier.

You can use the tag to broadcast any important information that should be noted during processing; for example, you might use it as a reminder that a part of the file is incomplete.

The tag translator displays the line number and the file name in which the tag appears. The message is output in the following format:

```
%TAG-W-USER_WMSG, at tag <USER_W_MESSAGE> on line nnn of file  
      fffffff  
      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

---

**EXAMPLE**

The following example shows how to use the <USER\_W\_MESSAGE> tag to identify a notice in a file.

<USER\_W\_MESSAGE>(Reviewers: Please note missing parameters here.)

Suppose the file containing this message is called "Reviewers\_copy.SDML." Depending on how you process this file, the message appears on your screen, in the .LIST file, or in the .LOG file as follows:

```
%TAG-W-USER_WMSG, at tag USER_W_MESSAGE on line nn of file
Reviewers_Copy.SDML.
Reviewers: Please note missing parameters here.
```

## <VALID\_BREAK>

---

## <VALID\_BREAK>

Labels a permissible page break within a monospaced example (created with the <CODE\_EXAMPLE>, <DISPLAY>, <INTERACTIVE>, or <LINE\_ART> tags).

---

### FORMAT <VALID\_BREAK>

---

**ARGUMENTS** *None.*

---

**related tags**

- <CODE\_EXAMPLE>
- <DISPLAY>—in the SOFTWARE doctype
- <INTERACTIVE>
- <LINE\_ART>

---

**restrictions**

Valid only in the context of monospaced examples created with the <CODE\_EXAMPLE>, <DISPLAY> (in the SOFTWARE doctype), <INTERACTIVE>, or <LINE\_ART> tags.

---

### DESCRIPTION

The <VALID\_BREAK> tag labels a permissible page break within a monospaced example (created with the <CODE\_EXAMPLE>, <DISPLAY>, <INTERACTIVE>, or <LINE\_ART> tags).

The text formatter attempts to keep an example together on a single page. If there is not enough room for an example on the current page, the text formatter chooses page breaks using blank lines in the example as valid places to break. If your example contains no blank lines, or if you want to specify better breaking points, you can use the <VALID\_BREAK> tag to specify the places that are acceptable page breaks.

This tag has no effect for Bookreader output.

---

### EXAMPLE

The following example shows how to use the <VALID\_BREAK> tag.

```
<INTERACTIVE>
<S> ($)
<U> (@SYS$SYSTEM:SHUTDOWN)

<S> (
    SHUTDOWN -- Perform an Orderly System Shutdown)
<ELLIPSIS>
<S> (CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, OPER1)
<S> (OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10, OPER11,)
<S> (OPER12)
<VALID_BREAK>
```

**<VALID\_BREAK>**

```
<S>(%SHUTDOWN-I-DISLOGINS, Interactive logins will now be disabled.)  
<S>(%SET-I-INTSET, login interactive limit = 0 current interactive value = 17)  
<S>(%SHUTDOWN-I-SHUTNET, The DECnet network will now be shut down.)  
<S>(%SHUTDOWN-I-STOPQUEMAN, The queue manager will now be stopped.)  
<ENDINTERACTIVE>
```

## <VALID\_TABLE\_ROW\_BREAK>

---

## <VALID\_TABLE\_ROW\_BREAK>

Marks a permissible place that a first-level table row, not a **nested table** row, may be broken across pages.

---

**FORMAT**            <VALID\_TABLE\_ROW\_BREAK>

---

**ARGUMENTS**        *None.*

---

**related tags**        • <NESTED\_TABLE\_BREAK>

---

**restrictions**        Valid only in the context of a <TABLE> tag.  
Valid only in the last column of a 2- or 3-column, first-level table; you cannot specify this tag in a nested table.

---

**DESCRIPTION**        The <VALID\_TABLE\_ROW\_BREAK> tag marks a permissible place that a first-level table row, not a **nested table** row, may be broken across pages. If you do not use the <VALID\_TABLE\_ROW\_BREAK> tag inside a long table row, the text formatter tries to keep all the text on the same page of output and might issue an error message. In extreme cases, the text formatter could run out of memory trying to process the table row and terminate processing.

This tag has no effect for Bookreader output.

---

**EXAMPLE**            The following example shows how to use the <VALID\_TABLE\_ROW\_BREAK> tag in a table with a long table row.

```
<TABLE>
<TABLE_ATTRIBUTES>(MULTIPAGE)
<TABLE_SETUP>(2\9)
<TABLE_ROW>(Item\Text that goes on for several paragraphs.
<VALID_TABLE_ROW_BREAK>
Text that is still inside the first table row.
<P>
<VALID_TABLE_ROW_BREAK>
The last paragraphs for this table row.)
<ENDTABLE>
```



---

## <VARIABLE>

Labels a program variable or number you want to typographically distinguish.

---

**FORMAT**            <VARIABLE>(variable name)

---

**ARGUMENTS**        *variable name*  
Specifies the name of the variable to be typographically distinguished.

---

**related tags**        • <KEYWORD>

---

**DESCRIPTION**      The <VARIABLE> tag labels a program variable or number you want to typographically distinguish. Use this tag to format a variable consistently throughout a document and a document set.

---

**EXAMPLE**            The following example shows how to use the <VARIABLE> tag.

```
<P>At this point in the processing,  
<VARIABLE>(VERSION_NUMBER) has the  
value of 2.
```

This example produces the following output:

At this point in the processing, *VERSION\_NUMBER* has the value of 2.

## <VBAR>

---

## <VBAR>

Specifies a vertical bar anywhere in your file. You must use this tag, however, to specify a vertical bar in an argument to a tag.

---

### FORMAT <VBAR>

---

### ARGUMENTS *None.*

---

### related tags

- The following tags label other characters that you must tag when they occur in an argument to a tag:

<AMPERSAND>  
<BACKSLASH>  
<CPAREN>  
<OPAREN>

---

**restrictions** Valid only in an argument to a tag.

---

**DESCRIPTION** The <VBAR> tag specifies a vertical bar anywhere in your file. You must use this tag, however, to specify a vertical bar in an argument to a tag. If you use a literal vertical bar within an argument to a tag, the tag translator reads the vertical bar as beginning a section of text it should treat literally. This is because the tag translator uses a vertical bar ( | ) to begin a quoted string. The vertical bar may prevent the tag translator from correctly evaluating a tag, thus causing an error in your output.

---

**EXAMPLE** The following example shows how to use the <VBAR> tag.

```
<SUBHEAD1>(Labeling the Vertical Bar (<VBAR>) Within Your  
Code\label_vbar)  
<P>To pass a vertical bar (|) in an argument to a tag through  
to your output, . . .
```

This example produces the following output:

#### **Labeling the Vertical Bar (|) Within Your Code**

To pass a vertical bar ( | ) in an argument to a tag through to your output, . . .

---

<X>

Creates a main index entry. For a printed book, the <X> tag creates a reference to the page on which this tag appears. For a book you create for Bookreader, the <X> tag creates a **hotspot** on the index entry that you **click on** to display the associated text.

---

**FORMAT**

<X>(index entry[ \ attribute])

---

**ARGUMENTS**

***index entry***

Specifies the index entry you want to appear in the index. The capitalization you use is the capitalization that appears in the index.

***attribute***

This is an optional argument. It specifies the attributes that control the sorting and formatting of the index entry. Each attribute must be a separate argument.

Attribute	Function
<XAPPEND>(string)	Appends the specified string to the page reference in the index.  You must leave a space between the left parenthesis of the <XAPPEND> tag's argument and the word you want to append, in order to have a space in the output between the page number and the appended word.  In page-range references, append the string to the <X>(...\ END) tag, not to the <X>(...\ BEGIN) tag. The <XAPPEND> tag has no effect for Bookreader output.
BEGIN	Begins a page-range reference. When you use this attribute, you must pair it with a following <X> tag that has identical text (including capitalization) and the END attribute. BEGIN and END have no effect for Bookreader output.
BOLD	Causes the page reference number to appear in boldface. This is distinct from any boldfacing that appears in the text of the index entry itself. Use the <EMPHASIS> tag if you want to bold the index entry. BOLD has no effect for Bookreader output.
BOTH	Specifies an index entry that you want to appear in both a single-document and a master index. BOTH has no effect for Bookreader output.
END	Ends a page-range reference. When you use this attribute, you must pair it with a previous <X> tag that has identical text (including capitalization) and the BEGIN attribute. BEGIN and END have no effect for Bookreader output.

## <X>

Attribute	Function
ITALIC	Causes page reference numbers to appear in italics in your index entry. If you specify both BOLD and ITALIC, the entry is output in bold italics. Use the <EMPHASIS> tag if you want to bold the index entry. ITALIC has no effect for Bookreader output.
MASTER	Specifies that an index entry appear only in a master index. A master index is an index generated by specifying /MASTER on the DOCUMENT command line. By default, the entries appear only in a single-document index. Using /MASTER allows you to specify different entries for a single book's index and for the master index of the document set. MASTER has no effect for Bookreader output.
NOMASTER	Specifies that an index entry appear only in a single-document index and not in a master index. NOMASTER has no effect for Bookreader output.
<XSORT>(string)	Uses the specified string as the sort key when placing this entry in the index, forcing the entry to sort in a place different from where it would by default.

### related tags

- <XS>
- <XSUBENTRY>
- <Y>

### restrictions

Invalid in informal code examples. Using this tag in a code example interferes with the formatting of the example.

In tables, place <X> tags directly next to the items you want to index (within the arguments to the <TABLE\_ROW> tag, so that the page references will be correct if the table breaks across pages), or after the <TABLE\_HEADS> or <TABLE\_SETUP> tags.

In tables you create for Bookreader, place <X> tags after <TABLE\_HEADS>, <TABLE\_ROW>, or <TABLE\_SETUP> tags so that the index entry points to the table **pop-up window**.

If you want an index entry to point to a small section of text, you must place the index tag within the section of text, so that the index entry refers to the correct page.

If you want an index entry to point to a large section of text denoted by a heading, you must place the index tag after the relevant <HEADn> tag, not before it, so that the index entry refers to the correct page.

### DESCRIPTION

The <X> tag creates a main index entry in either a single-document or a master index. A master index is an index generated by specifying the /MASTER\_INDEX qualifier on the DOCUMENT command line; a single-document index is an index generated by specifying /INDEX on the command line.

For a printed book, the <X> tag creates a reference to the page on which this tag appears. For a book you create for Bookreader, the <X> tag creates a hotspot on the index entry that you click on to display the associated text. The entry is composed of the main entry and optional subentries, which you specify in the first argument to the tag using the <XSUBENTRY> tag. See the <XSUBENTRY> tag description for more information.

Code your index entries within the information you want the entry to point to, which ensures that the index will point to the correct page or section of information.

You can control how the index entry is sorted and how it appears by specifying particular keywords as *attribute* arguments.

For example, the <XAPPEND> attribute is useful when appending text to the end of a page reference, as when referring to an example (5-4ex) or to a table (9-8tab). You can also use the <XAPPEND> attribute to append “ff” or “passim”, for example, to the page number. The <XAPPEND> attribute has no effect for Bookreader output.

The appended string is boldfaced or italicized if its page reference is boldfaced or italicized. You can use the BOLD and ITALIC attributes together. BOLD and ITALIC have no effect for Bookreader output.

You can create an index entry for several inclusive pages by using the BEGIN and END attribute. When using a BEGIN and END pair, you must be certain that the index entries are identical and that any other attributes you specify are the same for each entry, as well. The BEGIN and END pair have no effect for Bookreader output.

If you have an index entry for which you want to control the sorting, use the <XSORT> attribute. For example, you can sort the word “Cat” under the “-D-” entries. If you want to control the sorting of nonalphabetic characters, use the /INDEX=SORT= qualifier on the DOCUMENT command line. See Appendix A for more information on the command line.

You can specify that an index entry appear in a master index by using the MASTER attribute. No entry marked with the MASTER attribute appears in the single-document index.

You can specify an index entry that you do not want to appear in a master index by using the NOMASTER attribute. This attribute limits the entry to appear only in a single-document index.

You can specify an index entry that you want to appear in both a single-document and a master index by using the BOTH attribute.

If you do not specify an attribute, the entry appears only in the single-document index unless you override this default with the following DOCUMENT command line qualifiers:

`\INDEX=OVERRIDE_MASTER`

Creates a single-document index containing both master and single-document index entries.

<X>

`\MASTER_INDEX=OVERRIDE_MASTER` Creates a master index containing both master and single-document index entries.

An index entry that you label with the `NOMASTER` attribute is included only in the single-document index regardless of which “`OVERRIDE`” qualifier you use on the command line.

An index entry that you label with the `BOTH` attribute is included in both the single-document and the master index regardless of which “`OVERRIDE`” qualifier you use on the command line.

Table 1–5 summarizes how the <X> tag attributes and the command line indexing options for a single-document and a master index interact. An “X” specifies that an index entry with the listed tag attribute runs with the command line option to produce an index.

**Table 1–5 Interaction of <X> Tag Arguments and Command Line Indexing Options**

<X> Tag Attributes	Command Line Options			
	For a Single-document Index		For a Master Index	
	<code>/INDEX</code>	<code>/INDEX=OVERRIDE</code>	<code>/MASTER</code>	<code>/MASTER=OVERRIDE</code>
none	X	X		X
BOTH	X	X	X	X
MASTER		X	X	X
NOMASTER	X	X		

For example, if you want to specify that certain index entries appear in a master index but not in a single-document index, code the entries using the `MASTER` attribute.

If you want to generate a single-document index that includes index entries that you coded with the `MASTER` attribute, use the `/INDEX=OVERRIDE` qualifier on the command line.

If you want to generate a master index that includes index entries that you coded without the `MASTER` attribute, use the `/MASTER=OVERRIDE` qualifier on the command line.

If you want to generate a master index that includes some of your coded index entries (both with and without using the `MASTER` attribute) while excluding other entries, do the following:

- 1 Edit your `.SDML` file by adding the `NOMASTER` attribute to the entries you want excluded.
- 2 Process your book using the `/MASTER=OVERRIDE` qualifier on the command line.

If you want to generate a single-document index that includes some but not all of the entries you coded using the MASTER attribute, without having to remove the MASTER attribute from the entries you want to include, do the following:

- 1 Edit your .SDML file by changing the MASTER attribute to BOTH for those entries you want to include.
- 2 Process your book to generate a single-document index by using the /INDEX qualifier on the command line.

Consider the following when indexing for Bookreader output:

- Be sure to place your index entries within the information you want the entry to point to, which ensures that the entry in the index will point to the correct topic of information.
- You can place index tags anywhere in your table, except directly after the <TABLE> tag, or else the entries will point to the previous text instead of to the table in the **pop-up window**.
- In **formal examples** and **formal figures**, place the index tags before the <ENDEXAMPLE> or <ENDFIGURE> tag, so that **clicking on** the entry in the index causes the actual example or figure to **pop up**. Do not place index entries within a code example, or the code example may not be formatted correctly.

Refer to Appendix A for more information on using the /INDEX qualifier on the DOCUMENT command line.

---

## EXAMPLES

The following example shows how to position the <X> tag to provide an index entry for a chapter.

```
1 <CHAPTER>(Creating Routines\creating_routine_sec)
  <X>(Routines<XSUBENTRY>creation of)
  <P>
  In order to create a routine, you must first . . .
```

If the previous text appeared on page 11-2, the <X> tag would create the following output under the heading “R”:

```
Routines
  creation of, 11-2
```

The following example shows how to create an inclusive index entry using the BEGIN and END attributes.

```
2 <X>(File structure\begin)
  .
  .
  .
  <X>(File structure\end)
```

This example produces the following output under the heading “F”:

```
F
  File structure, 5-4 to 5-7
```

## <X>

The following example shows how to use the <XAPPEND> attribute. Notice that you must leave a space between the left parenthesis of the <XAPPEND> attribute's argument and the word you want to append, in order to have a space in the output between the page number and the appended word.

```
3 <X>(File structure\begin)
  .
  .
  .
<X>(File structure\end\<XAPPEND>( passim))
```

This example produces the following output under the heading "F":

File structure, 5-4 to 5-7 passim

The following example shows how to code multiple subentries for a particular main entry.

```
4 <X>(Debugger<XS>features of)
  .
  .
  .
<X>(Debugger<XS>exiting from)
  .
  .
  .
<X>(Debugger<XS>invoking)
```

This example produces the following output under the heading "D":

Debugger  
 exiting from, 2-4  
 features of, 1-7  
 invoking, 3-9

The following example shows where to put an index tag in a table, and creates an index entry that correctly specifies the page that the text "bananas" appears on, no matter where VAX DOCUMENT breaks the table.

```
5 <TABLE_ROW>(apples\oranges\pears)
<TABLE_ROW>(<X>(Tropical fruits<XS>bananas)bananas\pineapples\mangos)
<TABLE_ROW>(blackberries\blueberries\strawberries)
```

This example produces the following output under the heading "T":

Tropical fruits  
 bananas, 3-21

The following example shows how to use the <XSORT> attribute. "Tsar" is placed with the "C" entries after "Czar" rather than with the "T" entries.

```
6 <X>(Tsar<XSUBENTRY>country of\<XSORT>(Czar))
```

This example produces the following output:

Cat, 3-6  
Czar, 1-16  
Tsar  
 country of, 3-21



The following example shows how to code an index entry that you want to appear only in a master index.

7 <X>(Classical music\MASTER)

The following example shows how to code an index entry that you want to appear only in an individual index and not in a master index.

8 <X>(Jazz music\NOMASTER)

The following example shows how to code an index entry that you want to appear in both an individual and a master index.

9 <X>(Alternative rock music\BOTH)

## <XS>

---

## <XS>

Separates the main index entry from the first subentry, the first subentry from the second, and the second subentry from the third. Used within the argument to the <X> or <Y> tags. For a printed book, the <XS> tag creates a reference to the page on which the tag appears. For a book you create for Bookreader, the <XS> tag creates a **hotspot** on the index entry with no page reference. (The <XS> tag is identical to the <XSUBENTRY> tag.)

---

**FORMAT**            **<XS>** (*subentry-1*[<XS>*subentry-2*[<XS>*subentry-3*]])

---

**ARGUMENTS**        **subentry**  
Specifies up to three subentries you want to appear in the index. The capitalization you use is the capitalization that appears in the index.

---

**related tags**

- <X>
- <XSUBENTRY>
- <Y>

---

**restrictions**

Valid only within the argument to an <X> or <Y> tag.  
You can use a maximum of three subentries in the index.

---

## DESCRIPTION

The <XS> tag separates the main index entry from the first subentry, the first subentry from the second, and the second subentry from the third. For example:

```
<x>(main entry<xs>subentry-1<xs>subentry-2<xs>subentry-3)
```

For a printed book, the <XS> tag creates a reference to the page on which the tag appears. For a book you create for the Bookreader, the <XS> tag creates a hotspot on the index entry with no page reference.

An index entry for a book you create for Bookreader that contains more than one reference to another section displays all the index selections in a separate **pop-up window**. See *VAX DOCUMENT Producing Online and Printed Documentation* for a graphic example.

---

## EXAMPLES

The following example shows how to use the <XS> tag.

1 <X> (Vehicles<XS>use of)

If the previous index tags appeared on page 3-1, they would create the following index entry under the heading “V”:

```
Vehicles
  use of, 3-1
```

The following example shows how to use the <XS> tag to create a “See also” entry.

2 <X> (File structure designators)  
<Y> (File structure designators<XS>See also Header blocks)

If the previous tags appeared on page 1-3, they would create the following index entry under the heading “F”:

```
File structure designators, 1-3
  See also Header blocks
```

The following example shows how to code multiple subentries for a particular main entry.

3 <X> (Debugger<XS>features of)  
.  
.  
<X> (Debugger<XS>exiting from)  
.  
.  
<X> (Debugger<XS>invoking)

The previous tags produce the following entries under the heading “D”:

```
Debugger
  exiting from, 1-7
  features of, 2-4
  invoking, 3-9
```

The following example creates unnumbered index entries (because of the <Y> tag, which you use for unnumbered “See” and “See also” entries).

4 <Y> (CORP-AUTO report<XS>See Reports<XS>See also Cars)

Regardless of what page the previous tags appear on, they produce the following output under the heading “C”:

```
CORP-AUTO report
  See Reports
  See also Cars
```

## <XSUBENTRY>

---

## <XSUBENTRY>

Separates the main index entry from the first subentry, the first subentry from the second, and the second subentry from the third. Used within the argument to the <X> or <Y> tags. For a printed book, the <XSUBENTRY> tag creates a reference to the page on which the tag appears. For a book you create for Bookreader, the <XSUBENTRY> tag creates a **hotspot** on the index entry with no page reference. The <XSUBENTRY> tag is identical to the <XS> tag.

---

### FORMAT

**<XSUBENTRY>**(*subentry-1*[<XSUBENTRY>*subentry-2*  
[<XSUBENTRY>*subentry-3*]])

---

### ARGUMENTS

#### ***subentry***

Specifies up to three subentries you want to appear in the index. The capitalization you use is the capitalization that appears in the index.

---

### related tags

- <X>
- <XS>
- <Y>

---

### restrictions

Valid only within the argument to an <X> or <Y> tag.  
You can use a maximum of three subentries in the index.

---

### DESCRIPTION

The <XSUBENTRY> tag separates the main index entry from the first subentry, the first subentry from the second, and the second subentry from the third. For example:

```
<x>(main entry<xsubentry>subentry-1<xsubentry>subentry-2<xsubentry>subentry-3)
```

For a printed book, the <XSUBENTRY> tag creates a reference to the page on which the tag appears. For a book you create for Bookreader, the <XSUBENTRY> tag creates a hotspot on the index entry with no page reference.

An index entry for a book you create for Bookreader that contains more than one reference to another section displays all the index selections in a separate **pop-up window**. See *VAX DOCUMENT Producing Online and Printed Documentation* for a graphic example.

---

## EXAMPLES

The following example shows how to use the <XSUBENTRY> tag.

1 <X>(Vehicles<XSUBENTRY>use of)

If the previous index tags appeared on page 3-1, they would create the following index entry under the heading “V”:

```
Vehicles
  use of, 3-1
```

The following example shows how to use the <XS> tag to create a “See also” entry.

2 <X>(File structure designators)  
<Y>(File structure designators<XSUBENTRY>See also Header blocks)

If the previous tags appeared on page 1-3, they would create the following index entry under the heading “F”:

```
File structure designators, 1-3
  See also Header blocks
```

The following example shows how to code multiple subentries for a particular main entry.

3 <X>(Debugger<XSUBENTRY>features of)  
.  
.  
<X>(Debugger<XSUBENTRY>exiting from)  
.  
.  
<X>(Debugger<XSUBENTRY>invoking)

The previous tags produce the following entries under the heading “D”:

```
Debugger
  exiting from, 1-7
  features of, 2-4
  invoking, 3-9
```

The following example creates unnumbered index entries (because of the <Y> tag, which you use for unnumbered “See” and “See also” entries).

4 <Y>(CORP-AUTO report<XSUBENTRY>See Reports<XSUBENTRY>See also Cars)

Regardless of what page the previous tags appear on, they produce the following output under the heading “C”:

```
CORP-AUTO report
  See Reports
  See also Cars
```

<Y>

---

<Y>

Used with the <XSUBENTRY> tag to create a main index entry and a subentry that has no page reference for a printed book and no **hotspot** for a book you create for Bookreader. Use the <Y> and <XSUBENTRY> tags for cross-references (“See” or “See also” entries) to other index entries.

---

**FORMAT**      <Y>(index entry[ \ attribute])

---

**ARGUMENTS**    *index entry*

Specifies the main entry and subentries that appear in the index. The capitalization you use is the capitalization that appears in the index.

*attribute*

This is an optional argument. It specifies the attributes that control the sorting of the index entry. You can specify a maximum of two attribute arguments. Each attribute must be a separate argument. The possible attributes follow:

Attribute	Function
BOTH	Specifies an index entry that you want to appear in both a single-document and a master index. BOTH has no effect for Bookreader output.
MASTER	Specifies that an index entry appear only in a master index. A master index is an index generated by specifying /MASTER on the DOCUMENT command line. By default, the entries appear only in a single-document index. Using /MASTER allows you to specify different entries for a single book’s index and for the master index of the document set. MASTER has no effect for Bookreader output.
NOMASTER	Specifies that an index entry appear only in a single-document index and not in a master index. NOMASTER has no effect for Bookreader output.
<XSORT>(string)	Uses the specified string as the sort key when placing this entry in the index, forcing the entry to sort in a place different from where it would by default.

---

**related tags**

- <X>
- <XS>
- <XSUBENTRY>

---

**restrictions**

Invalid in informal code examples. Using this tag in a code example interferes with the formatting of the example.

---

## DESCRIPTION

Use the <Y> tag with the <XSUBENTRY> tag to create a main index entry and a subentry that has no page reference for a printed book and no hotspot for a book you create for Bookreader. Use the <Y> and <XSUBENTRY> tags for cross-references (“See” or “See also” entries) to other index entries. You specify the subentries by using the <XSUBENTRY> tag within the argument to the <Y> tag. See the <XSUBENTRY> tag description for more information.

The <Y> tag subentries sort alphabetically at the top of the subentry list, before the <X> tag subentries; that is, cross-references to a topic are listed before its subtopics.

For information on how to use the MASTER, NOMASTER, BOTH, and <XSORT> attributes, see the <X> tag description.

---

## EXAMPLES

The following example shows how to use the <Y> tag.

**1** <X>(File structure designators)  
<Y>(File structure designators<XSUBENTRY>See also Header blocks)

If the previous index tags appeared on page 1-3, they would create the following index entry under the heading “F”:

File structure designators, 1-3  
See also Header blocks

The following example shows how to use the <Y> tag to create an unnumbered index entry.

**2** <Y>(File structure designators<XSUBENTRY>See also Header blocks)

The previous tags create the following output under the “F” entry:

File structure designators  
See also Header blocks

The following example shows how to code multiple “See also” references for a particular main entry.

**3** <Y>(Debugger<XS>See also Debugger commands; Debugging; /DEBUG qualifier)

The previous tags produce the following output under the heading “D”:

Debugger  
See also Debugger commands; Debugging; /DEBUG qualifier

The following example shows how to use the <Y> tag to sort the entry “Cat”.

**4** <Y>(Cat<XSUBENTRY>See also Feline<XSORT>(Animals))

The previous tags produce the following output, where “Cat” is placed with the “A” entries after “Animals” rather than with the “C” entries:

Cat  
See also Feline





# A

---

## VAX DOCUMENT Command Summary

This appendix lists and describes all the VAX DOCUMENT commands. The appendix is separated into two command summaries:

- DOCUMENT[/FORMATTER]

Starts the VAX DOCUMENT formatting system.

You can find a summary of this information by typing the following command at the DCL prompt:

```
$ HELP DOCUMENT[/FORMATTER]
```

- DOCUMENT/GENERATE\_SYMBOL

Starts the VAX DOCUMENT symbol generator.

You can find a summary of this information by typing the following command at the DCL prompt:

```
$ HELP DOCUMENT/GENERATE_SYMBOL
```

The HELP file also provides a summary of commands for using the Graphics Editor. You can type the following command at the DCL prompt:

```
$ HELP DOCUMENT/GRAPHICS
```

See the *VAX DOCUMENT Graphics Editor User's Guide* for more information about the Graphics Editor.

## DOCUMENT[/FORMATTER]

---

## DOCUMENT[/FORMATTER]

Starts the VAX DOCUMENT formatting system.

---

<b>FORMAT</b>	<b>DOCUMENT[/FORMATTER]</b>	<i>input-file doctype destination</i>
	<b>Command Qualifier</b>	<b>Default</b>
	/[NO]BATCH[=qualifier-keyword]	/NOBATCH
	/CONDITION=condition-name	None.
	/[NO]CONTENTS	/NOCONTENTS
		/CONTENTS (Bookreader)
	/[NO]DEVICE_CONVERTER[=device-keyword]	/DEVICE_CONVERTER
	/[NO]DIAGNOSTICS[=file-spec]	/NODIAGNOSTICS
	/ELEMENT=file-spec	None.
	/INCLUDE=file-spec	None.
	/[NO]INDEX[=index-keyword]	/NOINDEX
	/[NO]KEEP[=filetype-keyword]	/NOKEEP
	/[NO]LIST[=file-spec]	/NOLIST
	/[NO]LOG	/LOG
	/[NO]MAP[=file-spec]	/NOMAP
	/[NO]MASTER_INDEX[=index-keyword]	/NOMASTER_INDEX
	/OUTPUT=file-spec	See text.
	/[NO]PRINT[=qualifier-keyword]	/NOPRINT
	/PROFILE=file-spec	None.
	/[NO]SYMBOLS=file-spec	/NOSYMBOLS
	/[NO]TAG_TRANSLATOR	/TAG_TRANSLATOR
	/[NO]TEXT_FORMATTER	/TEXT_FORMATTER

---

### restrictions

The /FORMATTER command is the default. It is positional; if you use it, it must immediately follow DOCUMENT.

---

### PARAMETERS *input-file*

Specifies the input file to be processed. You cannot use wildcards in the input file specification.

The default file type of the input file is SDML. If you specify qualifiers, VAX DOCUMENT determines the default file type based on the qualifiers and the destination keyword you specify. Table A-1 lists the qualifiers, the default destinations, and the default file types.

Table A-1 Default File Types

VAX DOCUMENT Qualifiers Used	Default Destination Keyword Used	Default Input File Type
None.	Any	SDML
/TAG_TRANSLATOR	Any	SDML
{ /NOTAG_TRANSLATOR } { /TEXT_FORMATTER } <sup>1</sup>	Any	TEX
{ /NOTAG_TRANSLATOR } { /NOTEXT_FORMATTER } { /DEVICE_CONVERTER }	BOOKREADER LN03 PS LINE MAIL TERMINAL	DVI_LN03 DVI_PS DVI_LINE DVI_LINE DVI_LINE
{ /NOTAG_TRANSLATOR } { /NOTEXT_FORMATTER } { /NODEVICE_CONVERTER } { /PRINT }	LN03 PS LINE MAIL TERMINAL	DVI_ BOOKREADER LN03 PS LINE TXT TERM

<sup>1</sup>Braces in this table indicate that the enclosed qualifiers occur together on the command line.

### **doctype**

The *doctype* you indicate on the command line determines both the style of your output and the SDML tags you can use. Note that some tags are valid only in specific doctypes.

A number of doctypes have a choice of *designs*. A design is a specialized form of a doctype. You specify the design keyword immediately after the doctype keyword and separate them with a period (.); for example, SOFTWARE.REFERENCE. You can abbreviate the doctype keyword; for example, you could shorten SOFTWARE.REFERENCE to S.R, as long as each keyword is unique.

See *VAX DOCUMENT Using Doctypes and Related Tags* for more information on doctypes and doctype-specific tags.

### **destination**

Specifies the output device *destination* for the document. VAX DOCUMENT supports the following output destinations; your local destinations may be different.

Destination Keyword	Formatted For	Output File Type
BOOKREADER	Bookreader output	.DECW\$BOOK
HELP	Building a Help file	.HLP
LINE	A line printer	.LINE
LN03	An LN03 laser printer	.LN03

## DOCUMENT[/FORMATTER]

Destination Keyword	Formatted For	Output File Type
PS	Any Digital-supported PostSCRIPT output device, such as the PRINTSERVER 40 or the LN03R SCRIPTPRINTER	.PS
MAIL	Sending through the VMS Mail Utility	.TXT
TERMINAL	A standard ANSI terminal, such as the VT-100	.TERM

You can abbreviate the destination keyword. For example, you could abbreviate LN03 to LN0, or even LN, as long as that destination is unique.

Note that monospaced fonts defined for all the sizes are provided for text fonts in the POSTSCRIPT destination.

You can have additional local destination keywords defined at your site. See your VAX DOCUMENT system administrator for information on local destination keywords.

### DESCRIPTION

DOCUMENT is the command you specify to start VAX DOCUMENT. It requires three parameters:

- Input File

Specifies the input file for VAX DOCUMENT. By default, this file is an SDML file containing SDML tags; however, it can also be one of the *intermediate files* generated by VAX DOCUMENT.

- Doctype

Specifies the document type for which the input file will be processed. This keyword specifies the kind of document you want to create (a letter, a software manual, a journal article, and so on).

- Destination

Specifies the final processing destination for the input file. This keyword specifies a format used by a printer, Bookreader, a terminal, or the VMS Mail Utility.

You can use qualifiers to the DOCUMENT command to create an index, master index, and table of contents, and to modify the default processing of your input file.

### QUALIFIERS

***/BATCH[=(qualifier keyword[, qualifier keyword...])]***  
***/NOBATCH***

Specifies whether VAX DOCUMENT should be run interactively or run as a batch job. The default qualifier is /NOBATCH, which specifies that VAX DOCUMENT be run interactively.

The /BATCH qualifier submits a job to SYS\$BATCH with a job name that has the same file name as the *input-file*, prefixed with the string "DOC\$." For example, the file ROUTINES.SDML would be submitted as the job DOC\$ROUTINES.

## DOCUMENT[/FORMATTER]

You can use any of the DCL SUBMIT command qualifiers with VAX DOCUMENT by using these qualifiers as keywords to the /BATCH qualifier. For example, if you want DOCUMENT to be run after 9:00 and want to be notified when it completes, you could use the following command:

```
$ DOCUMENT file-spec /BATCH=(AFTER=09:00,NOTIFY) LETTER LN03
```

When you use the /BATCH qualifier, a file is created in your current default directory that contains information about the batch job. This file has the same file name as the input file, with a default file type of LOG. When your batch job completes, this file is printed to the queue defined by the logical SYS\$PRINT, and the file is deleted.

You must define any process logical names you enter on the DOCUMENT command line in your LOGIN.COM file. Otherwise, VAX DOCUMENT is unable to translate the logical name during batch processing and issues an error message.

### ***/CONDITION=condition name***

Specifies a condition keyword for an SDML input file. This qualifier accepts a *condition name* argument that is a text string used to mark the condition being set. For more information on using the /CONDITION qualifier, see the <SET\_CONDITION> tag description and *VAX DOCUMENT Producing Online and Printed Documentation*.

The /CONDITION qualifier is valid only if tag translation is being done. If you specify /NOTAG\_TRANSLATOR with the /CONDITION qualifier, the /CONDITION qualifier is ignored, and VAX DOCUMENT issues an informational message stating that you specified conflicting qualifiers.

### ***/CONTENTS***

### ***/NOCONTENTS***

Specifies whether a table of contents file is produced. The /NOCONTENTS qualifier is the default (except when processing for Bookreader, where /CONTENTS is the default) and specifies that no table of contents is produced. When you specify the /CONTENTS qualifier, VAX DOCUMENT creates a table of contents file with a file name of *input\_filename\_CONTENTS*.

If you place the <CONTENTS\_FILE> tag in your front matter or profile SDML file, the current table of contents file is included at the corresponding point in the final printable output file (filename.LN03, filename.PS, and so on). If you do not place the <CONTENTS\_FILE> tag in your SDML file, the table of contents is not incorporated into your final output file, but is placed in the separate file, *input\_filename\_CONTENTS*, and processed separately.

For Bookreader output, a table of contents is generated automatically even if you do not specify the /CONTENTS qualifier on the command line or use the <CONTENTS\_FILE> tag in your SDML file.

If you do not specify /CONTENTS on the DOCUMENT command line when you process a file that contains a <CONTENTS\_FILE> tag, VAX DOCUMENT issues warning messages and the most recent version of the table of contents file is included. Note that this may result in an outdated table of contents being included in your document. If there is no previous table of contents file to be included, you receive a warning message and a blank page of output.

## DOCUMENT[/FORMATTER]

The /CONTENTS qualifier is valid only if text formatting is being done. If you specify /NOTEXT\_FORMATTER with the /CONTENTS qualifier, VAX DOCUMENT issues a warning message stating that you specified an illegal combination of command elements.

Note that if you process your document with the /NOTAG\_TRANSLATOR and /CONTENTS qualifiers, and the table of contents is included using the <CONTENTS\_FILE> tag, the table of contents is incorporated into your document, placed in the file *filename\_CONTENTS*, and processed separately.

**Note: The maximum length of a VMS file name is 39 characters. If you want to generate a contents file, your input file name must have no more than 30 characters, because appending \_CONTENTS to it adds 9 more characters.**

For more information on generating a table of contents, see *VAX DOCUMENT Producing Online and Printed Documentation* and the <CONTENTS\_FILE> tag description.

***/DEVICE\_CONVERTER[(device keyword [,device keyword...])]***

***/NODEVICE\_CONVERTER***

Specifies whether the device converter is run. Using /NODEVICE\_CONVERTER excludes device conversion.

The device converter reads and processes an intermediate file (with the file type *.DVI\_device*) and converts it to a file suitable for output on the destination device. The output from the device converter has the same file name as the input file you specified on the command line, and a file type based on the destination keyword you specified on the command line.

The /DEVICE\_CONVERTER qualifier optionally accepts the following keywords for special processing of your file:

Device Keyword	Description
BLANK_PAGES	Specifies that the PostScript device converter insert blank pages where needed in the PostScript output to create a two-sided document for printing on a duplex printer or for copying on a two-sided copier.
HORIZONTAL_OFFSET= <i>number of points</i>	Specifies where the text is to be positioned, relative to the left edge of the paper. The default horizontal offset for the page is one inch (72 points).  The <i>number of points</i> argument specifies the number of points the text page should be moved to the right from the left edge of the paper. This argument must be zero or a positive integer. The left edge of the paper is assumed to be zero.

Device Keyword	Description
VERTICAL_OFFSET= <i>number of points</i>	<p>Specifies where the text is to be positioned, relative to the top edge of the paper. The default vertical offset for the page is one inch (72 points).</p> <p>The <i>number of points</i> argument specifies the number of points the text page should be moved toward the page bottom from the top edge of the page. This argument must be zero or a positive integer. The top edge of the paper is assumed to be zero.</p>
STARTING_PAGE= <i>folio spec</i>	<p>Specifies the beginning page number of the first page in a range of pages to be printed. If you do not specify an ending page, the rest of the file is printed. The <i>folio spec</i> value can be any valid page number that VAX DOCUMENT produces on the page.</p>
ENDING_PAGE= <i>folio spec</i>	<p>Specifies the ending page number of the last page in a range of pages to be printed. The <i>folio spec</i> value can be any valid page number that VAX DOCUMENT produces on the page. If both the ENDING_PAGE and NUMBER_OF_PAGES keywords are used together on the same command line, VAX DOCUMENT issues an error message.</p>
NUMBER_OF_PAGES= <i>maximum pages</i>	<p>Specifies the number of pages to print when you do not specify the ENDING_PAGE keyword. If both the ENDING_PAGE and NUMBER_OF_PAGES keywords are used together on the same command line, VAX DOCUMENT issues an error message.</p> <p>The <i>maximum pages</i> value must be an integer specifying the total number of pages to print. If this number is specified as a number larger than the number of pages in the document (for example, 99999), all pages of the document are printed.</p>

Use the STARTING\_PAGE and ENDING\_PAGE keywords to specify the page numbers that are to be processed. Each of these keywords accepts a *folio spec* argument. A *folio spec* has the following syntax:

```
[[folio prefix]{separator}] {page number}
```

The following list describes the rules for each of the *folio spec* syntax elements:

- *folio prefix*

Specifies the numbers or letters that prefix the *folio spec*. A *folio prefix* can be any of the following:

- Any single letter. Specifies an appendix in your document, for example, the letter “B” in the *folio spec* B-6.
- Any number. Specifies a chapter number, for example, the number “13” in the *folio spec* 13-1.
- The keywords GLOSSARY or INDEX. Specifies a page in the glossary or index, for example, “INDEX” in the *folio spec* INDEX-6.
- The keyword PART $n$  where  $n$  is an integer of one or greater. Specifies a section begun using a <PART\_PAGE> tag, for example, “PART2” in the *folio spec* PART2-7.
- An asterisk (\*). Specifies the first section of your document.

## DOCUMENT[/FORMATTER]

- *separator*

Specifies the character that separates the *folio prefix* from the *page number*. The *separator* can be any single character that is not a space, a number, or a letter. In the *folio spec* 11-3, "-" is the *folio prefix*. You cannot use a *separator* if you do not specify a *folio prefix*.

- *page number*

Specifies the page number portion of a *folio spec*. In the *folio spec* 11-3, "3" is the *page number*. The *page number* can be a Roman number, an Arabic number, or an asterisk (\*). Roman numbers specify a page in the preface of a document, Arabic numbers specify pages outside the preface section, and the asterisk (\*) specifies the first page of the document section specified by the *folio prefix*.

The following DOCUMENT command specifies that pages 11-3 through 11-8 of file MYREPORT.DVI\_LN03 be processed by the device converter.

```
$ DOCUMENT/NOTAG_TRANSLATOR/NOTEXT_FORMATTER myreport.DVI_LN03 -  
_$_ REPORT LN03/DEVICE=(STARTING=11-3,ENDING=11-8)
```

### **/DIAGNOSTICS[=*file spec*]**

#### **/NODIAGNOSTICS**

Causes the tag translator to write VAX Language-Sensitive Editor (LSE) diagnostics records to a file. LSE uses these records during its REVIEW phase to locate and describe translation errors. The default qualifier is /NODIAGNOSTICS. See *VAX DOCUMENT Producing Online and Printed Documentation* for more information on LSE.

If you omit the file specification, the output file has the same name as the input file, with a file type of DIA.

The /DIAGNOSTICS qualifier is valid only if tag translation is being done. If you specify /NOTAG\_TRANSLATOR with the /DIAGNOSTICS qualifier, VAX DOCUMENT issues a warning message stating that you specified an illegal combination of command elements.

### **/ELEMENT=*file spec***

Names the book element that includes the input file specified on the command line. When you use the /ELEMENT qualifier to process a subelement of a book, you must also use the /PROFILE qualifier to specify the profile for the book that contains the book element. If you do not specify the file type of the element file, the default file type is SDML. For more information on processing a subelement of a book, see *VAX DOCUMENT Producing Online and Printed Documentation*.

The /ELEMENT qualifier is valid only if tag translation is being done. If you specify /NOTAG\_TRANSLATOR with the /ELEMENT qualifier, VAX DOCUMENT issues a warning message stating that you specified an illegal combination of command elements.

### **/INCLUDE=*file spec***

Specifies a VAX DOCUMENT file that you want to be included before the input file you specified on the command line. If you do not specify the file type of the file to be included, SDML is the default file type. You can use the /INCLUDE qualifier on the command line only once. For more information about using the \INCLUDE qualifier, see *VAX DOCUMENT Producing Online and Printed Documentation*.



The /INCLUDE qualifier is valid only if tag translation is being done. If you specify /NOTAG\_TRANSLATOR with the /INCLUDE qualifier, VAX DOCUMENT issues a warning message stating that you specified an illegal combination of command elements.

### ***/INDEX[=(index keyword [,index keyword...])] /NOINDEX***

Specifies whether an index file is produced. The /NOINDEX qualifier is the default and specifies that no index is produced. When you specify the /INDEX qualifier, VAX DOCUMENT creates an index file with a file name of *input\_filename\_INDEX*.

When you place the <INDEX\_FILE> tag in your profile file, the current index file is included at the corresponding point in the final printable output file (*filename.LN03*, *filename.PS*, and so on). If you do not place the <INDEX\_FILE> tag in your profile file, the index is not incorporated into your final output file, but is placed in the file *input\_filename\_INDEX*, which is printable separately.

If you do not specify /INDEX on the command line when you process a file that contains an <INDEX\_FILE> tag, VAX DOCUMENT issues a warning message and the most recent version of the index file that already existed is included in your document. This may be an outdated index. If no previous index file exists, the device converter issues a warning message.

The /INDEX qualifier is valid only if text formatting is being done. If you specify /NOTEXT\_FORMATTER with the /INDEX qualifier, VAX DOCUMENT issues a warning message stating that you specified an illegal combination of command elements.

Note that if you process your document with the /NOTAG\_TRANSLATOR and /INDEX qualifiers, and the index is included using the <INDEX\_FILE> tag, the index is incorporated into your document, placed in the file *filename\_INDEX*, and processed separately.

**Note: The maximum length of a VMS file name is 39 characters. If you want to generate an index file, your input file name must have no more than 33 characters, because appending *\_INDEX* to it adds 6 more characters.**

The following list describes the optional indexing keywords that you can use with the /INDEX qualifier. For more information on creating and generating a single-document and a master index, see the <X>, <XS>, and <Y> tag descriptions.

## DOCUMENT[/FORMATTER]

Index Keyword	Description
<pre>{ GUIDE_HEADINGS } { NOGUIDE_HEADINGS }</pre>	<p>Specifies whether alphabetic headings are created for each letter group in the index. (The entries beginning with "A" have an A at the start of the group, and so on.) The GUIDE_HEADINGS keyword is the default. The NOGUIDE_HEADINGS keyword suppresses guide headings in the index output file.</p>
<pre>SORT=( [ { LETTER }         { WORD }         [ , { NONALPHA=AFTER }             { NONALPHA=BEFORE }             { NONALPHA=IGNORE } ] ] )</pre>	<p>Specifies the sorting method used to order entries in an index.</p> <ul style="list-style-type: none"><li>• SORT=LETTER sorts the entry letter by letter and ignores spaces and hyphens. SORT=LETTER is the default.</li><li>• SORT=WORD sorts the entry letter by letter and treats spaces and hyphens as significant.</li><li>• SORT=NONALPHA positions an entry with leading nonalphabetic characters in the index based on the keyword supplied with the NONALPHA keyword.<ul style="list-style-type: none"><li>— The AFTER keyword causes an entry with leading nonalphabetic characters to be placed at the end of the index.</li><li>— The BEFORE keyword causes an entry with leading nonalphabetic characters to be placed at the beginning of the index.</li><li>— The IGNORE keyword causes an entry with leading nonalphabetic characters to be sorted by the first alphabetic characters in the entry. The default is NONALPHA=IGNORE.</li></ul></li></ul>
<pre>{ OVERRIDE_MASTER } { NOOVERRIDE_MASTER }</pre>	<p>Specifies the inclusion of index entries in both single-document indexes and master indexes; NOOVERRIDE_MASTER is the default keyword for both /INDEX and /MASTER_INDEX. OVERRIDE_MASTER and NOOVERRIDE_MASTER have no effect for Bookreader output.</p> <ul style="list-style-type: none"><li>• Use /INDEX=NOOVERRIDE_MASTER to create a single-document index that contains only the index entries that are not marked with the MASTER keyword; this is the default.</li><li>• Use /INDEX=OVERRIDE_MASTER to create a single-document index that contains both the index entries that are not marked with the MASTER keyword and the entries that are marked with the MASTER keyword.</li><li>• Use /MASTER_INDEX=NOOVERRIDE_MASTER to create a master index that contains only the index entries that are marked with the MASTER keyword; this is the default.</li><li>• Use /MASTER_INDEX=OVERRIDE_MASTER to create a master index that contains both the index entries that are not marked with the MASTER keyword and those entries that are marked with the MASTER keyword.</li></ul>

---

***/KEEP[=(filetype keyword [,filetype keyword...] )]***  
***/NOKEEP***

Specifies whether all intermediate files are kept or deleted. The default qualifier /NOKEEP indicates that the intermediate files are deleted by VAX DOCUMENT after processing.

You can use the following keywords to specify individual intermediate files that are to be kept:

---

<b>File Type Keyword</b>	<b>File Contents</b>
DVI	Specifies an intermediate output file from the text formatter. You may want to keep this file for reprocessing at a later date, or to process and print only certain pages.  The actual file type is based on the destination you specified on the DOCUMENT command line. These file types and the destination keywords that create them are listed under the description of the /TEXT_FORMATTER qualifier.
INX	Specifies an ASCII file that contains index entries in page-number order. You can use this file to create a master index.
TEX	Specifies an input file for the text formatter. Note that a TEX file processed with a certain doctype can produce errors if it is reprocessed with a different doctype.

---

## ***/LIST[=file-name]***

### ***/NOLIST***

Specifies whether a listing file is produced. The /NOLIST qualifier is the default and suppresses the generation of a listing file. If you do not specify a file specification as an argument to the /LIST qualifier, a listing file is produced with the file name of the input file specification and a default file type of LIS. If you specify a file specification as an argument, that file specification is used as the output file.

The listing file contains the following information:

- All messages generated by the tag translator
- All messages generated by the text processor
- All messages generated by the device converter
- A brief summary section. This summary includes the following information:
  - The original command line
  - The time and day that processing began
  - The total CPU time used

If you are running DOCUMENT as a batch job and want a complete log file (including output from your LOGIN.COM, symbol definitions, and so on), you should use the /LOG qualifier instead of the /LIST qualifier. Do not use /LOG, however, if you are using the symbol generator.

## ***/LOG***

### ***/NOLOG***

Specifies whether informational messages are issued during processing. The /LOG qualifier is the default and specifies that informational messages are issued. The /NOLOG qualifier suppresses informational messages.

## DOCUMENT[/FORMATTER]

The following example shows the typical informational messages displayed during tag translation:

```
$ DOCUMENT myarticle ARTICLE LN03
%DOC-I-IDENT, VAX DOCUMENT V2.0
[ T a g   T r a n s l a t i o n ]...
%TAG-I-DEFSLOADD, End of Loading of Tag Definitions
%TAG-I-ENDPASS_1, End of first pass over the input
[ T e x t   F o r m a t t i n g ]...
```

If you are running DOCUMENT as a batch job and do not need a complete log file (including output from your LOGIN.COM, symbol definitions, and so on), you can use the /LIST qualifier instead of the /LOG qualifier.

### **/MAP[=file spec]**

#### **/NOMAP**

Specifies whether all the input files processed by VAX DOCUMENT are listed in a file. This file starts with the SDML input file and includes the tag table and any SDML files specified by the /INCLUDE or /SYMBOLS qualifier or the <ELEMENT> or <INCLUDE> tags. Files that are included by other files are indented under those files in this listing. The default file type is MAP\_LIS. The default qualifier is /NOMAP.

The /MAP qualifier is valid only if tag translation is being done. If you specify /NOTAG\_TRANSLATOR with the /MAP qualifier, VAX DOCUMENT issues a warning message stating that you specified an illegal combination of command elements.

### **/MASTER\_INDEX[(index keyword [,index keyword...])]**

#### **/NOMASTER\_INDEX**

Causes indexing files to be collated into a master index file.

When you specify the /MASTER\_INDEX qualifier, VAX DOCUMENT expects a master index data file as the input file. This data file lists the index output (INX) files generated from the individual books that are being referenced in the master index. The default file type of this input file is INX\_LIST.

This qualifier accepts the same optional indexing keywords as the /INDEX qualifier. See the description of the /INDEX qualifier for a list of these keywords and their uses.

This qualifier has no effect for Bookreader output. For more information on generating a master index, see *VAX DOCUMENT Producing Online and Printed Documentation*.

### **/OUTPUT=file spec**

Specifies a new name for the output file. If you do not use the /OUTPUT qualifier, the output file is given the same file name as the input file, but with a file type based on the other qualifiers and destination chosen.

An example of a command line that uses the /OUTPUT qualifier follows:

```
$ DOC somefile LETTER LN03 /NOTEXT_FORMATTER /OUTPUT=anotherfile
```

This command line creates the file ANOTHERFILE.TEX, because processing was completed before the text formatter was run. If you omit /NOTEXT\_FORMATTER from the preceding command line, the output file is ANOTHERFILE.LN03.

The following table lists the output file types and the default destination keywords that produce them.

Default Destination Keyword	Default File Type
BOOKREADER	DECW\$BOOK
HELP	HLP
LINE	LINE
LN03	LN03
MAIL	TXT
PS	PS
TERMINAL	TERM

***/PRINT[=(qualifier keyword [,qualifier keyword...])]  
/NOPRINT***

Specifies whether the output file is printed. /NOPRINT is the default.

You can use any of the DCL PRINT command qualifiers as keywords to the /PRINT qualifier. For example, if you want to print two copies of a VAX DOCUMENT file with no flag page, you can use the PRINT command as follows:

```
$ PRINT/NOFLAG/COPIES=2 somefile.line
```

You can specify the same options using the following /PRINT qualifier on the DOCUMENT command line:

```
$ DOC/NOTAG/NOTEXT/NODEV somefile -
_$/PRINT=(NOFLAG,COPIES=2) LETTER LINE
```

You can double-space your output, but only for monospaced destinations: LINE, MAIL, and TERMINAL. For example, the following command produces double-spaced MAIL output:

```
$ DOCU somefile.mail MAIL MAIL/PRINT=SPACE
```

You can enter multiple arguments to the /PRINT qualifier keyword on the DOCUMENT command line by enclosing the arguments in three pairs of quotation marks. For example:

```
$ DOCU somefile.sdml SOFT.REF PS -
_$_\PRINT=(NOTIFY,PARAMETER="" (MESSAGE=PRINT, -
_$_DATA=POSTSCRIPT) """, QUEUE=PS_PRINT)
```

Default print queues for each of the destination keywords available at your site are established when VAX DOCUMENT is installed. These queues include all DCL PRINT command qualifiers needed for correct printing on that output device.

If you choose to specify a print queue other than SYS\$PRINT, the default, you must also specify the correct qualifier keywords needed for that queue. These keywords vary depending on the type of file you want to print (indicated by the file extension) and the type of output device. The following table shows the keywords to use for each file and output device combination.

## DOCUMENT[/FORMATTER]

Output File Type	Print Device	DOCUMENT /PRINT Qualifier-Keywords
LINE	Line Printer LN03 Laser Printer LN03 PLUS Laser Printer	/PRINT=(QUEUE=queuenam)
	LN03R SCRIPTPRINTER PRINTSERVER 40	/PRINT=(QUEUE=queuenam)
LN03	LN03 Laser Printer LN03 PLUS Laser Printer	/PRINT=(QUEUE=queuenam)
	LN03R SCRIPTPRINTER PRINTSERVER 40	/PRINT=(QUEUE=queuenam)
PS	LN03R SCRIPTPRINTER PRINTSERVER 40	/PRINT=(QUEUE=queuenam)

If you want to print your file using the DCL PRINT command, the following table shows the qualifiers you must use for each supported device.

Output File Type	Print Device	DCL PRINT Command Qualifiers
LINE	Line Printer LN03 Laser Printer LN03 PLUS Laser Printer	\$ PRINT filename.LINE _\$/QUEUE=queuenam
	LN03R SCRIPTPRINTER PRINTSERVER 40	\$ PRINT/PARAM=(DATA_TYPE=ANSI) filename.LINE _\$/QUEUE=queuenam
LN03	LN03 Laser Printer LN03 PLUS Laser Printer	\$ PRINT/NOFEED/PASSALL filename.LN03 _\$/QUEUE=queuenam
	LN03R SCRIPTPRINTER PRINTSERVER 40	\$ PRINT/PARAM=(DATA_TYPE=ANSI) filename.LN03 _\$/QUEUE=queuenam
PS	LN03R SCRIPTPRINTER PRINTSERVER 40	\$ PRINT/PARAM=(DATA_TYPE=POST) filename.PS _\$/QUEUE=queuenam

Note that if any serious errors are encountered during processing, the file is not sent to the print device.

### ***/PROFILE=file spec***

Specifies that the input file is a book element and that a cross-reference data file previously created by a VAX DOCUMENT bookbuild be used to resolve any cross references in the input file. Specify the cross-reference data file as an argument to the /PROFILE qualifier; it has a file type of XREF. For more information on using the /PROFILE qualifier, see *VAX DOCUMENT Producing Online and Printed Documentation*.

The /PROFILE qualifier is valid only if tag translation is being done. If you specify /NOTAG\_TRANSLATOR with the /PROFILE qualifier, VAX DOCUMENT issues a warning message stating that you specified an illegal combination of command elements.

***/SYMBOLS=file spec******/NOSYMBOLS***

Controls whether a file of symbol definitions is read automatically during tag translation. The default qualifier is */NOSYMBOLS*. The SDML input file accessed through the */SYMBOLS* qualifier should contain only symbol definitions created by the *<DEFINE\_SYMBOL>* and the *<DEFINE\_BOOK\_NAME>* tags. You can use the */SYMBOLS* qualifier on the command line only once. For more information on using the */SYMBOLS* qualifier, see *VAX DOCUMENT Producing Online and Printed Documentation*.

The */SYMBOLS* qualifier is valid only if tag translation is being done. If you specify */NOTAG\_TRANSLATOR* with the */SYMBOLS* qualifier, VAX DOCUMENT issues a warning message stating that you specified an illegal combination of command elements.

***/TAG\_TRANSLATOR******/NOTAG\_TRANSLATOR***

Controls whether the tag translator is run. If you do not specify this qualifier, the default qualifier is */TAG\_TRANSLATOR*.

By default, the tag translator produces an output file with the file name of the input file and a file type of *TEX*. You can then use this file as an input file to the text formatter. Using */NOTAG\_TRANSLATOR* excludes tag translation and the following related qualifiers: */CONDITION*, */ELEMENT*, */DIAGNOSTICS*, */INCLUDE*, */MAP*, */PROFILE*, and */SYMBOLS*. For more information on using the */TAG\_TRANSLATOR* qualifier, see *VAX DOCUMENT Producing Online and Printed Documentation*.

***/TEXT\_FORMATTER******/NOTEXT\_FORMATTER***

Controls whether the text formatting program is run. If you specify */NOTEXT\_FORMATTER*, VAX DOCUMENT processing stops after the tag translator completes, and text formatting and the following related qualifiers are excluded: */CONTENTS* and */INDEX*. */TEXT\_FORMATTER* is the default.

If you specify */NOTAG\_TRANSLATOR* and the input file you specify on the command line does not include a file type, the default file type is *TEX*. The output file from the text formatter can have one of several file types based on the destination entered on the command line. These default output file types and their related destination keywords are given in the following list.

<b>Destination Keyword</b>	<b>Intermediate Output File Type</b>	<b>Final Output File Type</b>
BOOKREADER	DVI_BOOKREADER	DECW\$BOOK
HELP	None.	HLP
LINE	DVI_LINE	LINE
LN03	DVI_LN03	LN03
MAIL	DVI_LINE	TXT

## VAX DOCUMENT Command Summary

Destination Keyword	Intermediate Output File Type	Final Output File Type
PS	DVI_PS	PS
TERMINAL	DVI_LINE	TERM

You can then use the output file from the text formatter as an input file to the device converter.

### EXAMPLE

When you process your input file using the DOCUMENT command, you are actually running your file through several processors. The following example shows a typical use of the VAX DOCUMENT command:

```
$ DOCUMENT myreport.SDML REPORT LN03 /CONTENTS

%DOC-I-IDENT, VAX DOCUMENT V2.0
[ T a g   T r a n s l a t i o n ]...
%TAG-I-DEFSLOADD, End of Loading of Tag Definitions
%TAG-I-ENDPASS_1, End of first pass over the input
[ T e x t   F o r m a t t i n g ]...
%TEX-I-PAGESOUT, 17 pages written.
-TEX-I-OUTFILENAME, 'DUAL:[DOCFILES]MYREPORT.DVI_LN03'
[ C o n t e n t s   G e n e r a t i o n ]...
[ T e x t   F o r m a t t i n g   C o n t e n t s ]...
%TEX-I-PAGESOUT, 1 page written.
-TEX-I-OUTFILENAME, 'DUAL:[DOCFILES]MYREPORT_CONTENTS.DVI_LN03'
[ D e v i c e   C o n v e r s i o n ]...
%DVC-I-PAGESOUT, 18 pages written to file:
  DUAL:[DOCFILES]MYREPORT.LN03
[ C o n t e n t s   D e v i c e   C o n v e r s i o n ]...
%DVC-I-PAGESOUT, 1 page written to file:
  DUAL:[DOCFILES]MYREPORT_CONTENTS.LN03
[ P r i n t i n g   F i l e ]...
Job MYREPORT (queue SYS$LN03, entry 833) started on SYS$LN03

[ P r i n t i n g   C o n t e n t s ]...
Job MYREPORT_CONTENTS (queue SYS$LN03, entry 834) started on SYS$LN03
$
```





## DOCUMENT/GENERATE\_SYMBOL

---

### QUALIFIERS

#### ***/LIST[=file name]***

#### ***/NOLIST***

Specifies whether a listing file is produced. The */NOLIST* qualifier is the default and suppresses the generation of a listing file. If you do not specify a file specification as an argument to the */LIST* qualifier, a listing file is produced with the file name of the input file specification and a default file type of LIS. If you specify a file specification as an argument, that file specification is used as the output file.

The listing file contains the following information:

- All messages generated by the tag translator
- A brief summary section. This summary includes the following information:
  - Original command line
  - Time and day that processing began
  - Total CPU time used

#### ***/PREFIX=prefix-name***

Specifies the prefix you want appended to each symbol name in your file, including any included files. If you do not specify a *prefix-name*, a default prefix of up to 25 characters of the *input-file* name is used. If the *input-file* exceeds a length of 25 characters, the remaining characters are truncated. Any alphanumeric character is valid. Invalid characters are changed to underscores only when the default prefix is used.

For example, you could use the following command:

```
$ DOCUMENT/GENERATE_SYMBOL/LIST/PREFIX=ABC MYFILE.SDML
```

# B

## Messages

---

A message has the following format:

```
% facility--severity--identification, text
```

Messages in VAX DOCUMENT can come from the following facilities:

Facility code	Facility
DOC	The DOCUMENT command line
TAG	The tag translator
TEX	The text processor
DVC	The device converter
INX	The index facility

The messages from these sources are given in the following sections.

To locate a message, use the facility code to locate the appropriate section and then look for the identification text in that section. Messages are alphabetical within sections.

For example, consider the following message:

```
%TAG-W-ISTHISTAG, at text on line 245 in file  
WRT_: [YOURNAME.BOOK] INTRO.SDML;  
Ignoring <list. Is this a tag without a closing angle bracket?
```

The facility code in this error is TAG, so you look in the section for tag translator message. The identification text is ISTHISTAG, so you locate that within the section.

## Messages

### DOCUMENT Command Messages

---

#### B.1 DOCUMENT Command Messages

ACTION\_EXCL, *qualifier* is not available for the chosen destination

**Warning:** Indicates that the default qualifier has been excluded for the desired destination and will therefore be ignored.

**User Action:** None.

AMB\_DESIGN, Ambiguous design keyword *design-keyword*

**Fatal:** An ambiguous design keyword was specified.

**User Action:** Change the specified design keyword to be a unique design keyword.

AMB\_DEST, Ambiguous destination keyword *destination-keyword*

**Fatal:** An ambiguous destination keyword was specified.

**User Action:** Change the specified destination keyword to be a unique keyword.

AMB\_PAPER\_SIZE, Ambiguous paper size value

**Error:** The specified paper size value is ambiguous.

**User Action:** Specify a unique paper value and reexecute the DOCUMENT command.

CANT\_GET\_VALUE, Cannot get value of */qualifier* qualifier

**Fatal:** A required value associated with the qualifier was not provided.

**User Action:** Specify a value for the qualifier and reexecute the DOCUMENT command.

CANT\_OPEN, Error detected opening file *file-spec*

**Error:** An error has occurred while opening the specified file.

**User Action:** Correct the error if you can or refer the problem to your system manager. After fixing the problem, reexecute the DOCUMENT command.

CANT\_USE\_SHR, Cannot use *file-spec*

**Fatal:** A shareable image file for text processing could not be found or used.

**User Action:** Consult your system manager.

DCL\_ERROR, An error was detected while parsing  
\*DCL-command*\

**Error:** An error was detected by the DCL parser.

**User Action:** Correct the error on the specified line and reexecute the DOCUMENT command.

## Messages

### DOCUMENT Command Messages

- DESIGN\_ERROR, Bad design entry in *design-file-directory* DOC\$DESIGNS data file
- Error:** An error was found in a design data file; erroneous design entries are ignored.
- User Action:** Correct the error and reexecute the DOCUMENT command.
- DESIGN\_SYNTAX, Ignoring *design-keyword* design entry
- Error:** A syntax error is detected in the design data file.
- User Action:** Correct the error and reexecute the DOCUMENT command.
- DEST\_ERROR, Bad destination entry in *destination-file-directory* DOC\$DESTINATIONS data file
- Error:** An error was found in a destination data file. Any involved entries are ignored.
- User Action:** Correct the error and reexecute the DOCUMENT command.
- DEST\_SYNTAX, Ignoring *destination-keyword* destination data entry
- Error:** An error has been detected in a destination data file.
- User Action:** Correct the errors in the destination data file and reexecute the DOCUMENT command.
- DO\_CONTENTS, [ C o n t e n t s G e n e r a t i o n ]...
- Informational:** Indicates that a contents file is being created.
- User Action:** None.
- DO\_DEVICE, [ D e v i c e C o n v e r s i o n ]...
- Informational:** Indicates that the document input is being processed into a printable form.
- User Action:** None.
- DO\_DVC\_CONTENTS, [ C o n t e n t s D e v i c e C o n v e r s i o n ]...
- Informational:** Indicates that the document contents is being processed into a printable form.
- User Action:** None.
- DO\_DVC\_INDEX, [ I n d e x D e v i c e C o n v e r s i o n ]...
- Informational:** Indicates that the document index is being processed into a printable form.
- User Action:** None.
- DO\_INDEX, [ I n d e x G e n e r a t i o n ]...
- Informational:** Indicates that an index file is being created.
- User Action:** None.

## Messages

### DOCUMENT Command Messages

DO\_MASTER\_INDEX, [MasterIndexGeneration]..

**Informational:** Indicates that a master index file is being created.

**User Action:** None.

DO\_PRINT, [PrintingFile]..

**Informational:** Indicates that your output file is being sent to the designated print queue.

**User Action:** None.

DO\_TAG, [TagTranslation]..

**Informational:** Indicates that the tag translator is processing your SDML input file.

**User Action:** None.

DO\_TEXT, [TextFormatting]..

**Informational:** Indicates that the text formatter is processing.

**User Action:** None.

DO\_TEXT\_CONTENTS, [TextFormattingContents]..

**Informational:** Indicates that the text formatter is processing the contents file.

**User Action:** None.

DO\_TEXT\_INDEX, [TextFormattingIndex]..

**Informational:** Indicates that the text formatter is processing the index file.

**User Action:** None.

ERROR\_CONVERTER, Errors found by the device converter

**Error:** The device converter has ended in error; processing cannot continue.

**User Action:** Correct the errors found by the device converter and reexecute the DOCUMENT command.

ERROR\_FORMATTER, Errors found by the text formatter

**Error:** The text formatter has ended in error; processing cannot continue.

**User Action:** Correct the errors found by the text formatter and reexecute the DOCUMENT command.

ERROR\_INDEXER, Errors found by the indexing facility

**Error:** The indexing facility has ended in error; processing cannot continue.

**User Action:** Correct the errors found by the indexing facility and reexecute the DOCUMENT command.

## Messages

### DOCUMENT Command Messages

ERROR\_IN\_DESIGN, Internal design data file error

**Fatal:** An unspecified error in the design data file was found.

**User Action:** Refer the problem to your system manager. If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ERROR\_IN\_DEST, Internal destination data file error

**Fatal:** An unspecified error in the destination data file was found.

**User Action:** Refer the problem to your system manager. If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ERROR\_TAG, Errors found by the tag translator

**Error:** The tag translator has ended in error; processing cannot continue.

**User Action:** Correct the errors found by the tag translator and reexecute the DOCUMENT command.

ERROR\_WRITE, Error writing to file *file-spec*

**Fatal:** An RMS error was reported when DOCUMENT tried to read the specified file.

**User Action:** Correct the RMS error and reexecute DOCUMENT command.

ERR\_CRE8\_LIST, Listing file *file-spec* cannot be created

**Fatal:** The specified listing file could not be created.

**User Action:** Correct the error in the listing file specification.

FILLM\_TOO\_LOW, Process open file limit too low, required value is *minimum-value*

**Fatal:** The FILLM of the current process is not large enough to run the DOCUMENT command.

**User Action:** Contact your system manager to increase the quota.

IDENT, VAX DOCUMENT *version number*

**Informational:** DOCUMENT version number identification message.

**User Action:** None.

IGNOR\_DESIGN, Ignoring optional *logical-name* design data file

**Informational:** Optional DOC\$LOCAL\_FORMATS or DOC\$STANDARD\_FORMATS design file cannot be opened. Any designs that are in that file will be unavailable.

**User Action:** Correct the problem and reexecute the DOCUMENT command or contact your system manager.

## Messages

### DOCUMENT Command Messages

INVALID\_DESIGN, *design-keyword* is not a valid design

**Fatal:** An unrecognized design parameter was specified on the DOCUMENT command line.

**User Action:** Specify a valid design parameter and reexecute the DOCUMENT command.

INVALID\_DEST, *destination-keyword* is not a valid destination

**Fatal:** An unrecognized destination keyword was specified on the DOCUMENT command line.

**User Action:** Specify a valid destination keyword and reexecute the DOCUMENT command.

INVALID\_DVI\_ARG, *qualifier-arg* is not a valid /DEVICE\_CONVERTER argument

**Fatal:** The argument value specified to /DEVICE\_CONVERTER is unrecognized.

**User Action:** Specify a valid qualifier argument and reexecute the DOCUMENT command.

INVALID\_ENTRY, *qualifier-entry* is not a valid /*qualifier* entry

**Fatal:** An invalid qualifier entry was specified on the DOCUMENT command.

**User Action:** Specify a valid qualifier and reexecute the DOCUMENT command. Refer to Appendix A in this manual for information about valid DOCUMENT command qualifiers.

INVALID\_INPUT, Cannot use *file-spec* as an input file

**Fatal:** The specified input file is invalid.

**User Action:** Specify a valid input file specification and reexecute the DOCUMENT command.

INVALID\_VALUE, Cannot use *file-spec* as a file for /*qualifier*

**Fatal:** The specified input file associated with the qualifier cannot be opened.

**User Action:** Specify a valid input file specification and reexecute the DOCUMENT command.

NEED\_VALUE, Destination entry parameter command needs /VALUE or a keyword

**Error:** Syntax error found when parsing the destination data file.

**User Action:** Refer the problem to your system manager. If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.



## Messages

### DOCUMENT Command Messages

NOMASTER\_DEST, Chosen destination does not allow /MASTER\_INDEX processing

**Fatal:** Master index processing is not allowed for the destination specified in the DOCUMENT command.

**User Action:** Either remove /MASTER\_INDEX from the DOCUMENT command or choose another destination, and reexecute the DOCUMENT command.

NO\_CONTENTS, No contents; /CONTENTS qualifier unnecessary

**Warning:** No contents information was available to generate a contents file; no contents file will be produced.

**User Action:** Do not use the /CONTENTS qualifier when processing this input file. An SDML file must contain at least one <CHAPTER>, <APPENDIX>, or <HEADn> tag to produce a table of contents file.

NO\_INDEX, No index; /INDEX qualifier unnecessary

**Warning:** No index information was available to generate an index file; no index file will be produced.

**User Action:** Do not use the /INDEX qualifier when processing this input file. An SDML file must contain at least one <X> or <Y> tag to produce an index file.

OPEN\_DESIGN, Error opening design data file in DOC\$STANDARD\_FORMATS

**Fatal:** An error occurred when trying to open the DOC\$STANDARD\_FORMATS:DOC\$DESIGNS.DAT data file.

**User Action:** Check for the existence of the DOC\$STANDARD\_FORMATS:DOC\$DESIGNS.DAT data file. If it exists, then check the file protection on that file.

OPEN\_DEST, Error opening  
DOC\$STANDARD\_FORMATS:DOC\$DESTINATIONS data  
file

**Fatal:** There was a RMS error when trying to open the destination data file.

**User Action:** Check for the existence of the DOC\$STANDARD\_FORMATS:DOC\$DESTINATIONS.DAT data file. If it exists, then check the file protection on that file.

PGFLQ\_TOO\_LOW, Process paging file quota too low, required value is  
*minimum-value*

**Fatal:** The PGFLQUOTA of the current process is not large enough to run the DOCUMENT command.

**User Action:** Contact your system manager to increase the quota.

## Messages

### DOCUMENT Command Messages

PRINT\_CONTENTS, [ P r i n t i n g C o n t e n t s ]..

**Informational:** Indicates that your contents file is being sent to the designated print queue.

**User Action:** None.

PRINT\_INDEX, [ P r i n t i n g I n d e x ]..

**Informational:** Indicates that your index file is being sent to the designated print queue.

**User Action:** None.

PROB\_CRE\_LOGIC, A problem was encountered creating logical name  
*logical-name*

**Error:** An error was returned when VAX DOCUMENT attempted to create a VMS process logical name.

**User Action:** Correct the error if you can or refer the problem to your system manager. After fixing the problem, reexecute the DOCUMENT command.

THESE\_DESIGNS, Choose one of these designs:

**Informational:** Inquires for choice of document design types, this is required input to the DOCUMENT command.

**User Action:** At the prompt, type the desired design from the choices offered.

THESE\_DESTS, Choose one of these destinations:

**Informational:** Inquires for your choice of document destinations. This is required input the DOCUMENT command.

**User Action:** At the prompt, type the desired destination from the choices offered.

TYPE\_INPUT, input file

**Informational:** Inquires for the input file.

**User Action:** At the prompt, type the name of the input file.

UNR\_DESIGN, Unrecognized design - please check your spelling

**Fatal:** An unrecognized design parameter has been specified to the DOCUMENT command.

**User Action:** Specify a valid design parameter and reexecute the DOCUMENT command.

UNR\_DEST, Unrecognized destination - please check your spelling

**Fatal:** An unrecognized destination parameter has been specified to the DOCUMENT command.

**User Action:** Specify a valid destination parameter and reexecute the DOCUMENT command.

---

## B.2 Tag Translator Messages

**ABORTFORE**, at tag or text, line *n*, file  
*file-spec*  
There have been more than *number* error messages of severity level E

**Fatal:** The tag translator did not produce an output file because it detected too many errors in the SDML source file with a severity level of Error. SDML defines the minimum number of error messages of severity E as 0, although this default value can be modified in local usage.

**User Action:** Examine the output from the preceding messages to determine the errors you must correct.

**ABORTFORW**, at tag or text, line *n*, file  
*file-spec*  
There have been more than *number* error messages of severity level W

**Fatal:** The tag translator did not produce an output file because it detected too many errors in the SDML source file with a severity level of Warning. SDML defines the minimum number of error messages of severity W as 30, although this default value can be modified in local usage.

**User Action:** Examine the output from the preceding messages to determine the errors you must correct.

**ARGBADOPR**, at tag or text, line *n*, file  
*file-spec*  
Argument 2 to tag <COUNTER> has an illegal arithmetic operator

**Warning:** The tag translator has found an illegal operator. The second argument to the <COUNTER> tag consists of pairs of operators and operands, and an optional trailing comment.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**ARGINCMPL**, at tag or text, line *n*, file  
*file-spec*  
Argument 2 to tag <COUNTER> is incomplete

**Warning:** The tag translator has not found at least one operator and one operand. The second argument to the <COUNTER> tag requires at least one operator and one operand.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Tag Translator Messages

ARGINVALD, at tag or text, line *n*, file  
*file-spec*  
Argument *number* to tag *<tagname>* is an invalid argument.

**Warning:** The argument in the specified position is not valid for the specified tag.

**User Action:** Verify the tag's argument and correct it.

ARGMENTIS, The argument is *string*

**Informational:** Follows a message that indicates that an argument is invalid, for example, when a numeric argument is required, but a character string argument is specified.

**User Action:** Use the information to correct your source file.

ARGMISSNG, at tag or text, line *n*, file  
*file-spec*  
Argument *number* to tag *<tagname>* is missing

**Warning:** The tag translator found no text for the indicated argument, but the argument is required for the indicated tag.

**User Action:** Supply a value for the indicated argument.

ARGNOTCHR, at tag or text, line *n*, file  
*file-spec*  
Argument to tag *<CHR>* is not in the range 32 to 126

**Warning:** The *<CHR>* built-in tag requires a numeric argument whose value is in the range of 32 to 126.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ARGNOTKEY, at tag or text, line *n*, file  
*file-spec*  
Argument *number* to tag *<tagname>* is not a keyword

**Warning:** The argument in the indicated position is not a valid keyword argument for the indicated tag. This message is generally followed by a message that shows the keyword that was specified.

**User Action:** Using the information in this message and the accompanying message, determine the keyword in error. Determine the valid arguments for the tag in question and correct your source file.

ARGNOTLET, at tag or text, line *n*, file  
*file-spec*  
Argument to tag *<SET\_APPENDIX\_LETTER>* is not a letter

**Warning:** The *<SET\_APPENDIX\_LETTER>* tag requires a letter (A to Z).

**User Action:** Correct the source file to supply a letter as the first character of the argument.

## Messages

### Tag Translator Messages

ARGNOTMSG, at tag or text, line *n*, file  
*file-spec*  
Argument *number* to tag *<tagname>* is not a valid message code

**Warning:** The indicated argument to the `<TAG_DIAGNOSTIC>` built-in tag is not a recognized message code.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ARGNOTNAM, at tag or text, line *n*, file  
*file-spec*  
Argument *number* to tag *<tagname>* is not a valid name

**Warning:** The indicated argument does not obey the rules for formation of a name. A name must have only letters, digits, and the underscore character. A name must not begin with two underscores. Spaces, tabs, and carriage returns are not allowed.

**User Action:** Correct the argument so that the name is valid.

ARGNOTNUM, at tag or text, line *n*, file  
*file-spec*  
Argument *number* to tag *<tagname>* is not a number

**Warning:** A non-numeric argument was specified to a tag that requires a numeric argument.

**User Action:** Verify that the argument is in the correct position in the tag's argument list. Supply a numeric argument to the tag.

ARGNOTPOS, at tag or text, line *n*, file  
*file-spec*  
Argument to tag *<tagname>* is not a positive integer

**Warning:** The `<SET_CHAPTER_NUMBER>` and `<SET_APPENDIX_NUMBER>` tags require a positive integer for the chapter or appendix number argument.

**User Action:** Correct the source file to supply a positive integer.

ARGNOTQUA, at tag or text, line *n*, file  
*file-spec*  
Argument to tag does not begin with a slash

**Error:** The argument to the `<QUALITEM>` tag must be a valid VMS qualifier, that is, it must begin with a slash.

**User Action:** Correct the source file to supply a slash character.

ARGNOTVAL, Argument *number* to tag *<tagname>* is not an integer from 1 to 6

**Warning:** Numeric argument must be an integer from 1 to 6.

**User Action:** Supply an integer from 1 to 6.

## Messages

### Tag Translator Messages

ARGNOTTYP, at tag or text, line *n*, file  
*file-spec*  
Argument *string* to tag *<tagname>* is not valid for doctype  
*string*

**Warning:** The indicated argument is not valid for this tag in the indicated doctype.

**User Action:** Verify that you specified the correct doctype keyword on the DOCUMENT command line.

ARGOUTRAN, at tag or text, line *n*, file  
*file-spec*  
Argument *number* is not in the range 0 to 9

**Warning:** The indicated argument cannot be less than 0 or greater than 9.

**User Action:** Correct the argument so that the number is in range.

ARGOVRFLW, at tag or text, line *n*, file  
*file-spec*  
More than *number* characters in an argument

**Fatal:** The argument is too large. This often indicates that an argument list was not terminated correctly. The tag translator may be including large portions of the input text as part of the argument.

**User Action:** Check the tag indicated by the location information and supply the correct terminator to the argument list.

AUXOVRFLW, at tag or text, line *n*, file  
*file-spec*  
More than *number* auxiliary files open

**Fatal:** The number of auxiliary files that can be open simultaneously is limited.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADLSTARG, at tag or text, line *n*, file  
*file-spec*  
Argument *string* is not valid for *<LIST>* type *string*

**Warning:** Verify the argument you specified to the *<LIST>* tag. Each type of list accepts different keyword arguments, for example, UPPERCASE is a valid keyword for alphabetic lists, but not for numbered lists.

**User Action:** Verify that you have specified the correct type of list. If you specify a keyword argument, be sure you spell it correctly.

## Messages

### Tag Translator Messages

BKSYMNTDF, at tag or text, line *n*, file  
*file-spec*  
Symbol *string* is undefined.  
Bookshelf will not be created.

**Error:** A symbol has been referenced but is not in the symbol table.

**User Action:** Verify that you spelled the symbol correctly and that the same symbol has been entered in the symbol table.

BNOTESOUT, at tag or text, line *n*, file  
*file-spec*  
Accumulated *string* not referenced.  
Notes will not be output

**Warning:** A document contained <BACK\_NOTE> or <BIB\_NOTE> tags, but no corresponding <BACK\_NOTES> or <BIB\_NOTES> tag was specified to output the accumulated notes.

**User Action:** Place the <BACK\_NOTES> or <BIB\_NOTES> tag in the source file at the position at which you want them output.

BOOL2DEEP, at tag or text, line *n*, file  
*file-spec*  
Boolean expression has too many nested parentheses.

**Error:** A boolean expression is nested too deep.

**User Action:** Correct the boolean expression.

BOOL\_BDCH, at tag or text, line *n*, file  
*file-spec*  
Bad character in boolean expression: *string*

**Warning:** Each item in a boolean expression must begin with one of the characters A(ND), N(OT), O(R), T(RUE), F(ALSE) or must be a left or right parenthesis.

**User Action:** Correct the boolean expression.

BOOL\_BDST, at tag or text, line *n*, file  
*file-spec*  
Bad boolean expression, evaluation incomplete

**Warning:** At the completion of the evaluation of the boolean expression, some work remained to be done.

**User Action:** Correct the boolean expression.

BOOL\_EXPN, at tag or text, line *n*, file  
*file-spec*  
Bad boolean expression, *string* precedes *string*

**Warning:** A boolean expression is improperly formed. The two terms shown in the error message cannot appear in the order shown.

**User Action:** Correct the boolean expression.

## Messages

### Tag Translator Messages

BOOL\_RTPN, at tag or text, line *n*, file  
*file-spec*  
Unmatched right parenthesis in boolean expression

**Warning:** An extra right parenthesis has been found in a boolean expression.

**User Action:** Correct the boolean expression.

BOOL\_SPEL, at tag or text, line *n*, file  
*file-spec*  
Term in boolean expression misspelled

**Warning:** One of the words AND, OR, NOT, TRUE, or FALSE is misspelled in the boolean expression.

**User Action:** Correct the boolean expression.

BOOL\_TKIN, at tag or text, line *n*, file  
*file-spec*  
Last word in boolean expression is incomplete

**Warning:** The last word in the boolean expression begins with one of the characters A, O, N, T, or F, but is not spelled out completely.

**User Action:** Correct the boolean expression.

BOXTOOBIG, at tag or text, line *n*, file  
*file-spec*  
Length *number* of boxed item exceeds *number* characters.

**Warning:** An argument to the <BOX> tag exceeds the maximum number of characters allowed. The text will be output as null.

**User Action:** Correct the <BOX> tag to specify an argument with fewer characters.

CALL\_NEST, at tag or text, line *n*, file  
*file-spec*  
Callouts inaccurately nested in a monospaced example.

**Error:** A monospaced example using callouts incorrectly nests <EXAMPLE> ... <ENDEXAMPLE> tags (or tags such as <CODE\_EXAMPLE> that produce monospaced output) with respect to the <CALLOUTS> ... <ENDCALLOUTS> tags. For example, the following is incorrect:

```
<CALLOUTS> (\PREFIX)
<CODE_EXAMPLE>
...
<ENDCALLOUTS>
<ENCODE_EXAMPLE>
```

**User Action:** Correct the SDML source file so that the begin and end tags for the example and the callouts are correctly nested. For example:

```
<CALLOUTS> (\PREFIX)
<CODE_EXAMPLE>
...
<ENCODE_EXAMPLE>
<ENDCALLOUTS>
```



## Messages

### Tag Translator Messages

CANTOPINT, Cannot open intermediate input file:  
*file\_spec*

**Fatal:** The post translator cannot open the INT\_TEX file.

**User Action:** Verify that the file specification is correct and that you have sufficient resources and access rights to open the file.

CANTOPOUT, Cannot create output file:  
*file\_spec*

**Fatal:** The post translator cannot create the TEX file for some reason.

**User Action:** Verify that the file specification is correct, and that you have sufficient resources and access rights to create the file.

CASOVRFLW, at tag or text, line *n*, file  
*file-spec*

<CASE> tags have been nested beyond the limit of *number*

**Fatal:** The <CASE> built-in tag has been nested beyond the allowed limit.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

COMNOTEND, at tag or text, line *n*, file  
*file-spec*

A <COMMENT> in a tag definition has not been terminated

**Error:** A tag's definition contains a comment that is not properly terminated.

**User Action:** Examine the tag definition to be certain that the comment text is terminated. If the comment text contains characters such as ampersand, backslash, vertical bar, or parentheses, use the <COMMENT> . . . <ENDCOMMENT> format.

CPU\_USAGE, Pass 1: *number* Pass 2: *number* Total: *number* seconds

**Informational:** The tag translator reports its CPU usage during pass 1 and pass 2 and the total for both passes.

**User Action:** None.

CPYNAMHID, at tag or text, line *n*, file  
*file-spec*

<COPY\_TAG> is referencing a tag (<tagname>) that is hidden

**Warning:** The name referenced by the <COPY\_TAG> is hidden, and therefore cannot be copied as a new tag definition.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Tag Translator Messages

CPYNAMUND, at tag or text, line *n*, file  
*file-spec*  
<COPY\_TAG> is referencing a name (*string*) that is undefined

**Warning:** The name referenced by the <COPY\_TAG> is not defined, and therefore cannot be copied as a new tag definition.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DEFSLOADD, End of Loading of Tag Definitions

**Informational:** This message is issued after the tag definitions have been loaded and before the reading of any input files. If any error messages appear before this message, the errors were detected during the loading of the tag definitions.

**User Action:** None.

DIVBYZERO, at tag or text, line *n*, file  
*file-spec*  
Argument 2 to tag <COUNTER> is attempting to divide by zero

**Warning:** Division by zero is undefined. (Processing continues. The result is as if a divisor of 1 had been used.)

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DUPHIDNAM, at tag or text, line *n*, file  
*file-spec*  
A <HIDE\_TAGS> tag is reusing the hide-name, *string*

**Warning:** The hide-name is already in use. Once tags have been hidden under a specific hide-name, that hide-name cannot again be used until the tags have been revealed with the <REVEAL\_TAGS> tag.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DUPSYMBOL, at tag or text, line *n*, file  
*file-spec*  
The symbol *string* is being defined twice.  
The earlier definition is replaced by the new definition

**Warning:** The same symbol is being defined for two different purposes. The tag translator requires that each symbol be unique. If your symbols exceed 31 characters in length, the automatic truncation to 31 characters may be causing two different symbols to look alike.

**User Action:** Verify that both symbols are defined uniquely within the first 31 characters.

## Messages

### Tag Translator Messages

EDIT\_CODE, at tag or text, line *n*, file  
*file-spec*

Argument *number* has illegal action code *char*

**Error:** The character shown is not one of the allowed action codes.

**User Action:** Correct the indicated argument to the tag.

EDIT\_LNEQ, at tag or text, line *n*, file  
*file-spec*

Lengths of delimiter, action, substitute strings are not equal

**Error:** The number of delimiter characters and the number of action codes in one of the action strings or substitute characters in the substitute string are not the same.

**User Action:** Correct the delimiter, action, or substitute arguments to the tag.

ENDNOTBEG, at tag or text, line *n*, file  
*file-spec*

<*tagname*> specified without corresponding <*tagname*>.

**Warning:** A terminating tag was specified without the tag that it terminates. The tag is ignored.

**User Action:** Verify that you correctly entered the beginning tag, or remove the extraneous ending tag from your source file.

ENDPASS\_1, End of first pass over the input

**Informational:** The tag translator reports the end of the first pass over all input files. Error messages that appear ahead of this message were issued during the first pass. Error messages that follow this message are issued during the second pass over the input files.

**User Action:** None.

EOFARGLST, End of file encountered while searching for closing parenthesis. See argument list of tag <*tagname*> on line *number* of file  
*filename*

**Fatal:** An argument list is not terminated before the end of the current input file. Either a right parenthesis is missing or an <INCLUDE> tag occurred in a tag's argument list.

**User Action:** Verify that the indicated tag's argument list is terminated and that no tag's argument list is unterminated at the end of an included file.

## Messages

### Tag Translator Messages

EOFCOLLECT, End of file encountered while searching for tag *<tagname>*.  
See *<COLLECT>* or *<PROTECT>* tag on line *number* of file  
*filename*

**Fatal:** A *<COLLECT>* or *<PROTECT>* built-in tag has encountered an end of file before encountering the indicated stop-tag.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

EOFENDCAS, End of file encountered while searching for tag *<ENDCASE>*.  
See *<CASE>* or *<CASE\_NUMERIC>* tag on line *number* of file  
*filename*

**Fatal:** A *<CASE>* (or *<CASE\_NUMERIC>*) built-in tag is not terminated before the end of file.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

EOFENDCOM, End of file encountered while searching for tag  
*<ENDCOMMENT>*.  
See *<COMMENT>* tag on line *number* of file  
*filename*

**Fatal:** Comment text, introduced by the *<COMMENT>* built-in tag, was not enclosed as an argument, and therefore requires an *<ENDCOMMENT>* tag as a terminator.

**User Action:** Verify that the comment text is either enclosed as an argument to the *<COMMENT>* built-in tag or that it is terminated with an *<ENDCOMMENT>* tag.

EOFENDLIT, End of file encountered while searching for *<ENDLITERAL>*  
or *<ENDDELAYED>*. See *<LITERAL>* or *<DELAYED>* tag on line  
*number* of file  
*filename*

**Fatal:** Literal text, introduced by the *<LITERAL>* or *<DELAYED>* tag, was not enclosed as an argument, and therefore requires an *<ENDLITERAL>* or *<ENDDELAYED>* tag as a terminator.

**User Action:** Verify that the literal text is either enclosed as an argument to the *<LITERAL>* or *<DELAYED>* tag or that it is terminated with an *<ENDLITERAL>* or *<ENDDELAYED>* tag.

EOFIGNORE, at tag or text, line *n*, file  
*file-spec*  
End of file encountered while searching for tag or label

**Fatal:** An *<IGNORE>* built-in tag has encountered the end of the input file without finding one of the tags or labels supplied in the argument list.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Tag Translator Messages

ERRCLSINP, at tag or text, line *n*, file  
*file-spec*  
Input files not closed at end of pass.

**Fatal:** Files that were included as part of the input have not been properly closed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ERRDEFLNM, at tag or text, line *n*, file  
*file-spec*  
Attempt to define logical name was unsuccessful  
Logical: *string*  
Equivalence:*string*

**Warning:** An <INCLUDES\_FILE> tag in the <PROFILE> template has failed in its attempt to define the logical name with the given equivalence string.

**User Action:** Examine the arguments to the <INCLUDES\_FILE> tag to ensure that they are a correctly formed name and a file specification. Correct the <INCLUDES\_FILE> arguments.

ERRDURGET, at tag or text, line *n*, file  
*file-spec*  
Error reading line *number* of file:  
*filename*.  
Perhaps the line is too long

**Warning:** The indicated line probably exceeds the tag translator's buffer size. Processing continues with a truncated line (which may introduce other errors if the information that was lost includes part of an argument list to a tag).

**User Action:** Edit the input file and shorten the line or break it into two lines, if possible.

ERRDURPUT, at tag or text, line *n*, file  
*file-spec*  
Error detected writing output file:  
*filename*

**Fatal:** An error was detected while attempting to write to the indicated output file.

**User Action:** Check that your process has sufficient privilege and access rights and that the output device has sufficient free space to hold the file.

ERROPEAUX, at tag or text, line *n*, file  
*file-spec*  
Cannot open auxiliary file:  
*filename*

## Messages

### Tag Translator Messages

**Warning:** An attempt to open an auxiliary file failed. The next Informational message may supply the reason for the failure. Processing continues, but attempts to read or write the auxiliary file will fail and will generate error messages.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ERROPEDMP, at tag or text, line *n*, file  
*file-spec*  
Cannot open dump file:  
*filename*

**Warning:** The <RETRIEVE> built-in tag has failed to open a file for output. The next Informational message may indicate the reason for the failure.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ERROPEINC, at tag or text, line *n*, file  
*file-spec*  
Cannot open included file:  
*filename*

**Warning:** The indicated file cannot be opened for input. The next Informational message may indicate the reason for the failure. Processing continues without the included file.

**User Action:** Verify that the correct file is being accessed and can be read. If you are using a logical name to access the file, verify that the logical name is defined. If you are relying on an <INCLUDES\_FILE> tag to define the logical name, verify that the spelling of the logical name is the same in the <INCLUDES\_FILE> tag, and that the <INCLUDES\_FILE> tag has been specified for the book element that contains the <INCLUDE> tag.

ERROPEINP, at tag or text, line *n*, file  
*file-spec*  
Cannot open input file:  
*filename*

**Fatal:** The input file specified on the command line cannot be opened. The next Informational message may indicate the reason for the failure.

**User Action:** Verify that the correct file is being accessed and can be read.

ERROPELOG, at tag or text, line *n*, file  
*file-spec*  
Cannot open log file:  
*filename*

**Warning:** The <SET> built-in tag has failed to open a file for trace output. The next Informational message may indicate the reason for the failure.

## Messages

### Tag Translator Messages

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ERROPEOUT, at tag or text, line *n*, file  
*file-spec*  
Cannot open output file:  
*filename*

**Fatal:** The tag translator cannot open the output file. The next Informational message may indicate the reason for the failure.

**User Action:** Verify that the file specification is correct and that you have sufficient resources and access rights to create the file.

ERROPESEC, at tag or text, line *n*, file  
*file-spec*  
Cannot open secondary output file:  
*filename*

**Warning:** An attempt to open a secondary output file failed. Processing continues, but output will go to the primary output file.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ERROPESIN, at tag or text, line *n*, file  
*file-spec*  
Cannot open cross reference input file:  
*filename*

**Fatal:** The tag translator failed to open the file that contains the symbols used for cross-references. The next information message indicates the reason for the failure.

**User Action:** The file that contains the symbols used in cross-referencing can be used by only one user at a time. The tag translator locks the file to ensure that users have only serial access to it. If the reason for the failure to open the file was due to the fact that another user was currently using the file, you must wait and reissue the command when the file is no longer in use.

ERROPESOT, at tag or text, line *n*, file  
*file-spec*  
Cannot open cross reference output file:  
*filename*

**Fatal:** The tag translator failed to open a file for writing the symbols used for cross references. The next Informational message may indicate the reason for the failure.

**User Action:** Verify that you have sufficient resources and access rights to create the file.

## Messages

### Tag Translator Messages

ERRPRSFSP, at tag or text, line *n*, file  
*file-spec*  
Error parsing file specification:  
*filename*

**Warning:** The file specification contains an error.

**User Action:** Correct the file specification or logical name definition.

ERRVMZONE, at tag or text, line *n*, file  
*file-spec*  
Internal error. Failure to create VM zone.

**Fatal:** The tag translator failed to create space in virtual memory for storage of data.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

EXAMINLIN, Examine line *number* of file *filename*

**Informational:** This message follows other messages of greater severity, and directs you to a possible error in the input.

**User Action:** Use the information to examine and correct your input.

EXFOOTNOT, at tag or text, line *n*, file  
*file-spec*  
Footnotes in a monospaced example exceeds maximum of 4.

**Warning:** You have specified more than four <FOOTNOTE> tags in a <CODE\_EXAMPLE> or other monospaced example. The maximum allowed is four. The footnote will not be processed.

**User Action:** Correct the source file by removing the extraneous footnotes.

EXPAPPLET, Explicit appendix letter *string* set on line *number* of file  
*filename*

**Informational:** Warns that the letter assigned to the next <APPENDIX> tag has been supplied explicitly by a <SET\_APPENDIX\_LETTER> tag. The appendixes may be lettered out of sequence.

**User Action:** None.

EXPAPPNUM, Explicit appendix number *number* set on line *number* of file  
*filename*

**Informational:** Warns that the number assigned to the next <APPENDIX> tag has been supplied explicitly by a <SET\_APPENDIX\_NUMBER> tag. The appendixes may be numbered out of sequence.

**User Action:** None.



## Messages

### Tag Translator Messages

EXPCHAPNO, Explicit chapter number *number* set on line *number* of file *filename*

**Informational:** Warns that the number assigned to the next <CHAPTER> tag has been supplied explicitly by a <SET\_CHAPTER\_NUMBER> tag. The chapters may be numbered out of sequence.

**User Action:** None.

FILEWRTNG, File *filename* written (0 length)

**Informational:** An Error or Fatal message has been issued during the tag translation of a book element. The output file for that book element is reduced to 0 length.

**User Action:** Correct the problems indicated in the Error and Fatal messages and reprocess.

FILEWRTOK, File *filename* written

**Informational:** An element of a book has been processed through the tag translation phase.

**User Action:** None.

FCMDPARMS, at tag or text, line *n*, file *file-spec*  
<tagname> specified without <tagname> in Format.  
Using <tagname> alone

**Warning:** The tags <FCMD> and <FPARMS> must be used together to produce predictable results.

**User Action:** Modify your source file so that it contains both the <FCMD> and <FPARMS> tags. Arguments to either of these tags may be null, that is, it is valid to specify <FCMD>.

FIGINDENT, at tag or text, line *n*, file *file-spec*  
The value *number* specified for a block indent exceeds the current allowed maximum of *number*.

**Warning:** A <FIGURE\_FILE> tag specified the INDENT argument and a value that exceeds the current maximum allowed. The maximum value will be used.

**User Action:** Correct the numeric argument specified with INDENT so that it does not exceed the indicated maximum.

FIGLINMAX, at tag or text, line *n*, file *file-spec*  
Monospaced example lines in <tagname> exceed maximum *number*.

**Error:** A monospaced example specified in the context of a <FIGURE> or <EXAMPLE> tags is too long to fit on the current page, and either no

## Messages

### Tag Translator Messages

<VALID\_BREAK> tags were specified to provide valid break points, or there are too many lines between <VALID\_BREAK> tags.

**User Action:** Put <VALID\_BREAK> tags at suitable places in the monospaced example to allow the pages to break automatically.

FIG\_DEPTH, at tag or text, line *n*, file  
*file-spec*  
Tag <tagname> value *number* exceeds the maximum *NUMBER*  
or is 0.

**Warning:** The indicated tag specifies a depth for a figure or space to be left for a figure that will overrun the page boundaries. Or, the argument was specified as 0. In either case, the document's default full page depth will be used.

**User Action:** Correct the tag so that you specify no more than the maximum allowed.

FIG\_WIDTH, at tag or text, line *n*, file  
*file-spec*  
Tag <tagname> value *number* exceeds the maximum *NUMBER*  
or is 0.

**Warning:** The indicated tag specifies a width for a figure or space to be left for a figure that will overrun the page boundaries. Or, the argument was specified as 0. In either case, the document's default full page width will be used.

**User Action:** Correct the tag so that you specify no more than the maximum allowed.

FILENUMNG, at tag or text, line *n*, file  
*file-spec*  
File number is out of range

**Warning:** A number supplied to the <FILE\_NAME> tag is not within the range of 1 to the number of files that have been opened.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

FILISOPEN, at tag or text, line *n*, file  
*file-spec*  
Auxiliary file:  
*filename*  
is already open

**Warning:** An <OPEN> built-in tag is specifying a file that is already open. The request is ignored and processing continues.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Tag Translator Messages

FILNOTOPN, at tag or text, line *n*, file  
*file-spec*  
Auxiliary file:  
*filename*  
is not open

**Warning:** A <READ> or <WRITE> built-in tag is addressing a file that has not been opened.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

FMTDEVICE, Tag <*tagname*> produces device-specific output.

**Informational:** A tag that produces specific formatting controls has been processed. The output may not be suitable if the file is processed for another doctype, destination, or both. For example, <FINAL\_CLEANUP>(LINE\_BREAK) tells the text processor to create a new line of output, but the line break may not be suitable on all output devices.

**User Action:** Use explicit formatting commands sparingly.

GTMAXARGS, at tag or text, line *n*, file  
*file-spec*

More than *number* arguments supplied to tag <*tagname*>

**Error:** The indicated tag does not expect more than the indicated number of arguments. You may have included a backslash character in one of the arguments, which the tag translator interpreted as an argument separator.

**User Action:** Verify that the argument list is correctly coded. If necessary, use the <BACKSLASH> tag to code a backslash character that is actually part of an argument.

HIDNOTHID, at tag or text, line *n*, file  
*file-spec*

Internal error. A hidden tag was invoked,  
but was not found in the data structure of hidden tag names

**Fatal:** The algorithm for hiding and revealing tags has failed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ICON\_TEXT, at tag or text, line *n*, file  
*file-spec*

No text supplied for <ICON>.

**Warning:** An <ICON> tag was specified, but no <ICON\_TEXT> tag specified text to accompany the graphics. No text will be output.

**User Action:** Verify that you specified <ICON\_TEXT> correctly.

## Messages

### Tag Translator Messages

IGNAMEILL, at tag or text, line *n*, file  
*file-spec*  
Argument *string* to <IGNORE> is illegal

**Warning:** The argument is not a legal name or label. The argument is dropped from the argument list, and processing continues.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

IGNAMEMIS, at tag or text, line *n*, file  
*file-spec*  
An argument to <IGNORE> tag is a null string

**Warning:** One of the arguments to an <IGNORE> tag is empty.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

IGNOREARG, Ignoring the argument

**Informational:** This message usually accompanies other error messages that indicate a faulty argument. It indicates that the faulty argument is being ignored.

**User Action:** Correct the SDML file.

IGNORESET, Ignoring <tagname> on line *number* of file  
*filename*

**Informational:** This tag is ignored when doing a book build or element build, because the numbering of book elements is done automatically.

**User Action:** Remove the tag.

IGNOTDONE, at tag or text, line *n*, file  
*file-spec*  
The <IGNORE> tag has no arguments, so it is ignored

**Warning:** Missing or illegal arguments have resulted in an <IGNORE> tag that has no legal arguments. The <IGNORE> tag is not executed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INCNOTARG, at tag or text, line *n*, file  
*file-spec*  
An <INCLUDE> tag cannot be invoked in an argument

**Error:** An <INCLUDE> tag can be invoked only in text.

**User Action:** Move the <INCLUDE> tag outside the argument list.

## Messages

### Tag Translator Messages

INCOVRFLW, at tag or text, line *n*, file  
*file-spec*

More than *number* nested levels of included files

**Fatal:** The number of included files that can be open at the same time is limited. The limit has been exceeded.

**User Action:** Consider whether the included files can be included sequentially rather than simultaneously. If file A includes file B, and file B includes file C, consider whether file A can include both B and C, one after the other.

INDENTVAL, at tag or text, line *n*, file  
*file-spec*

The *<tagname>* specifies an indent value of *number*.

This exceeds the maximum *number* allowed for this tag.

**Warning:** The INDENT argument to the specified tag specified an indent value that exceeds the maximum allowed. The maximum value will be used.

**User Action:** Correct the tag to specify an indent value less than or equal to the maximum allowed.

INT\_LOGIC, at tag or text, line *n*, file  
*file-spec*

Internal error processing tag *<NAME>*

**Error:** An SDML tag definition has an error in it.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INVINHELP, Tag *<tagname>* being ignored for HELP processing.

**Informational:** The SDML tag used is invalid in the HELP doctype. The tag and its arguments are ignored.

**User Action:** None.

INVLDEST, at tag or text, line *n*, file  
*file-spec*

Destination *string* is invalid with the *string* doctype.

**Fatal:** The destination is invalid with the doctype used.

**User Action:** Check to see that the doctype and destination you are using are compatible.

ISTHISTAG, at tag or text, line *n*, file  
*file-spec*

Ignoring *<string* terminated by *char*.

Is this a tag without a closing angle bracket?

**Warning:** An apparent tag name has been found, but without the closing angle bracket that is needed to make it a complete tag invocation.

## Messages

### Tag Translator Messages

**User Action:** If this is a typing error, simply add the closing angle bracket. If this is not a typing error (you really want the left angle bracket and the tag name to appear in the output), use the <LITERAL> tag to enclose the left angle bracket.

KEYPADROW, at tag or text, line *n*, file  
*file-spec*  
Too many keypad rows. Extra rows ignored

**Warning:** A set of tags within a <KEYPAD\_SECTION> specifies more than four <KEYPAD\_ROW> tags. The keyboard keypad has only four rows and one end row.

**User Action:** Determine which <KEYPAD\_ROW> tag is extra and remove it from your source file.

LASTAGWAS, Last occurrence of <tagname> was on line *number*

**Informational:** This message is issued when a tag is not ended before a particular context or the end-of-file. It tells you the line number of the last occurrence of the tag.

**User Action:** Use the indicated line number to determine where to correct your source file.

LISTYPWAS, List type was *string*

**Informational:** This message accompanies an error that indicates that a <LIST> tag was not terminated. It indicates the type of list, for example, NUMBERED, UNNUMBERED, and so on.

**User Action:** Use this information to correct your SDML source file.

LITOVRFW, More than *number* characters in literal. See line *number* of file  
*filespec*

**Fatal:** The size of the text encompassed by a <LITERAL> or <DELAYED> tag is limited. This error may indicate that a <LITERAL> or <DELAYED> tag was incorrectly terminated.

**User Action:** Verify that the <LITERAL> or <DELAYED> tag has been correctly coded. If the text to be encompassed by the tag exceeds the limit, you must break it up and use more than one <LITERAL> or <DELAYED> tag.

LMF\_NOLMF, at tag or text, line *n*, file  
*file-spec*  
No LMF information is specified for this document

**Warning:** LMF information is missing.

**User Action:** Check to ensure that the LMF information exists if it is required for your document. Sometimes, incorrect use of the <REFERENCE> tag within the argument to the <TITLE> tag causes this problem.

## Messages

### Tag Translator Messages

LMFINFAFT, at tag or text, line *n*, file  
*file-spec*  
Make sure *<tagname>* is specified after the LMF tags.  
No LMF information is written out.

**Warning:** The *<LMF\_INFO>* tag may be specified before the LMF tags.

**User Action:** You may want to check to ensure that the LMF tags are coded before the *<LMF\_INFO>* tag.

LMFNOALTN, at tag or text, line *n*, file  
*file-spec*  
Tag *<tagname>* may be missing

**Warning:** An *<LMF\_ALTNAME>* tag may be missing.

**User Action:** Although this tag is not required, you may want to check to ensure that, if it exists, it is not miscoded.

LMFNOINFO, at tag or text, line *n*, file  
*file-spec*  
Missing *<LMF\_INFO>* tag.

**Warning:** The *<LMF\_INFO>* tag is missing.

**User Action:** Check to ensure that the LMF information exists if it is required for your document.

LMFTAGMSG, at tag or text, line *n*, file  
*file-spec*  
Tag *<tagname>* is missing

**Error:** An LMF tag is missing.

**User Action:** Check to ensure that the *<LMF\_PRODUCER>*, *<LMF\_PRODUCT>*, *<LMF\_RELEASE\_DATE>*, and *<LMF\_VERSION\_NUMBER>* tags exist.

LMF\_TOMNY, at tag or text, line *n*, file  
*file-spec*  
LMF information is specified more than once for this document

**Error:** LMF information is specified more than once for this document.

**User Action:** You can specify LMF information only once for each document. Check for too many *<LMF>* tags.

LTMINARGS, at tag or text, line *n*, file  
*file-spec*  
Fewer than *number* arguments supplied to tag *<tagname>*

**Error:** The indicated tag requires a minimum number of arguments. Perhaps one of the arguments has an unmatched right parenthesis in it, and the tag translator has treated that character as the terminator of the argument list.

**User Action:** Review and correct the argument list. Be sure that the open parenthesis for the argument list is not preceded with a space; for example *<TAG> (ARG)* should be *<TAG>(ARG)*.

## Messages

### Tag Translator Messages

MARKUNBAL, at tag or text, line *n*, file  
*file-spec*  
<MARK> tags are unbalanced.

**Error:** A <MARK> tag and its corresponding <ENDMARK> tag cross a tag's boundary or are incorrectly nested with respect to another pair of tags. For example, if a <MARK> tag is specified before <FORMAT> and <ENDMARK> is specified before <ENDFORMAT>, the tags are considered unbalanced.

**User Action:** Find the source file location of the indicated tag and move it. In particular look for instances where a <MARK> tag is specified in a tag argument, but the <ENDMARK> tag is specified outside that tag's argument or in an argument to another tag.

MILNOCLSS, No security classification has been specified

**Informational:** The <HIGHEST\_SECURITY\_CLASS> tag has been specified to output the highest security classification. However, no <SECURITY> tags have been specified in the document to provide a classification and so no output will be produced.

**User Action:** Add a <SECURITY> tag to indicate the document's security classification or remove the <HIGHEST\_SECURITY\_CLASS> tag.

MINEXCMAX, at tag or text, line *n*, file  
*file-spec*  
Minimum argument count (*number*) exceeds maximum  
argument count (*number*).  
(Minimum argument count is being reset to zero)

**Warning:** A <DEFINE> or <REDEFINE> tag is specifying a minimum and maximum number of arguments and the minimum exceeds the maximum.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MSGIDISNG, at tag or text, line *n*, file  
*file-spec*  
The message identification *string* is invalid.  
The <HIDE\_TAGS> tag is ignored

**Warning:** The second argument to a <HIDE\_TAGS> tag is supplying an unknown message identification.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NAME2LONG, at tag or text, line *n*, file  
*file-spec*  
Condition name *string* exceeds 28 characters

**Error:** Condition names may not exceed 28 characters in length.

**User Action:** Shorten this name (and any other condition names that are too long) to 28 characters or less.



## Messages

### Tag Translator Messages

NAMETRUNC, at tag or text, line *n*, file  
*file-spec*  
Name *string* exceeds 31 characters.  
Name *string* is truncated version

**Warning:** Names may not exceed 31 characters in length. The name that was supplied is a legal name, but it is too long. It is truncated to 31 characters.

If more than one name is truncated, the shortened names may not be unique and other errors may occur depending on the type of name.

**User Action:** Shorten this name (and any other names that are too long) to 31 characters or less, so that there is no potential for other errors.

NESTEDCOM, A <COMMENT> tag on line *number* of file  
*filename*  
is nested within <COMMENT> . . . <ENDCOMMENT> that starts  
on line *number* of file  
*filename*

**Warning:** <COMMENT> tags cannot be nested.

Sometimes an error is made in typing a <COMMENT> tag in the format <COMMENT>(text). The typing error makes it appear that the tag does not have an argument list, and it is assumed that an <ENDCOMMENT> tag is present to end the comment text. During the search for the <ENDCOMMENT> tag, if another <COMMENT> tag is encountered it will appear to be nested within the first <COMMENT> tag's text.

**User Action:** Correct any typographical errors in the initial <COMMENT> tag or remove the nested <COMMENT> tag.

NOBCKNOTE, at tag or text, line *n*, file  
*file-spec*  
Accumulated back notes not referenced.  
Notes will not be output

**Warning:** A document contained <BACK\_NOTE> tags, but no corresponding <BACK\_NOTES> tag was specified to output the accumulated notes.

**User Action:** Place the <BACK\_NOTES> tag in the source file at the position at which you want them output.

NOBIBNOTE, at tag or text, line *n*, file  
*file-spec*  
Accumulated bibliographic notes not referenced.  
Notes will not be output

**Warning:** A document contained <BIB\_NOTE> tags, but no corresponding <BIB\_NOTES> tag was specified to output the accumulated notes.

**User Action:** Place the <BIB\_NOTES> tag in the source file at the position at which you want them output.

## Messages

### Tag Translator Messages

NOCONFILE, at tag or text, line *n*, file  
*file-spec*  
<CONTENTS\_FILE> tag specified without /CONTENTS qualifier.  
No new file will be generated.

**Warning:** A <CONTENTS\_FILE> tag was encountered in the input file, but the corresponding /CONTENTS qualifier was not present on the command line. No new contents file will be produced. An earlier version of a contents file, if it exists, may be included in the output.

**User Action:** If you want to include a current table of contents file, you should reissue the DOCUMENT command using the /CONTENTS qualifier.

NODEFITEM, at tag or text, line *n*, file  
*file-spec*  
Definition list specified no items before <tagname>.

**Warning:** A <DEFINITION\_LIST> tag, or a corresponding tag in the SOFTWARE doctype, was specified but no <DEFLIST\_ITEM>/<DEFLIST\_DEF> tags were specified for the list. No output will be produced.

**User Action:** Remove the definition list start/end tags or provide item text for the definition list.

NOENDPROF, at tag or text, line *n*, file  
*file-spec*  
The <PROFILE> tag was not terminated by an <ENDPROFILE> tag

**Warning:** In order for the book build process to be completed, the <PROFILE> tag must be terminated by an <ENDPROFILE> tag. The <ENDPROFILE> tag is supplied by default in this case.

**User Action:** If the <PROFILE> tag is not terminated by an <ENDPROFILE> tag, add one after the final <ELEMENT> or <INCLUDES\_FILE> tag.

NOFIGELMS, at tag or text, line *n*, file  
*file-spec*  
<tagname> has no elements.

**Warning:** A <FIGURE> or <EXAMPLE> tag was specified and terminated but there were no elements (for example, <FIGURE\_SPACE>) to place in the figure.

**User Action:** Verify that you correctly spelled and entered tags for the figure or example.

NOHIDERV, at tag or text, line *n*, file  
*file-spec*  
Ignoring request to hide the <REVEAL\_TAGS> tag

**Warning:** The <REVEAL\_TAGS> built-in tag cannot be hidden by the action of a <HIDE\_TAGS> built-in tag.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Tag Translator Messages

NOINXFILE, at tag or text, line *n*, file  
*file-spec*  
<INDEX\_FILE> tag specified without /INDEX qualifier. No new file will be generated.

**Warning:** An <INDEX\_FILE> tag was encountered in the input file, but the corresponding /INDEX qualifier was not present on the command line. No new index file will be produced. An earlier version of an index file, if it exists, may be included in the output.

**User Action:** If you want to include a current index file, you should reissue the DOCUMENT command using the /INDEX qualifier.

NONESTING, at tag or text, line *n*, file  
*file-spec*  
Invalid attempt to nest tag <tagname>. Ignored

**Warning:** A tag is referenced when its context is already active. For example, this message is issued when <CODE\_EXAMPLE> is specified when a previous <CODE\_EXAMPLE> is not terminated.

**User Action:** The indicated tag is ignored; however, you should verify your source file to determine whether the tag is a duplicate or whether, in fact, you did not terminate an earlier occurrence of the same tag.

NONEWFILE, at tag or text, line *n*, file  
*file-spec*  
<tagname> specified without *string*. No new file will be generated.

**Warning:** A <CONTENTS\_FILE> or <INDEX\_FILE> was encountered in the input file, but the corresponding /CONTENTS or /INDEX qualifier was not present on the command line. No new contents or index will be produced. An earlier version of a contents or index file, if it exists, may be included in the output.

**User Action:** If you want to include the current table of contents or index file, you should reissue the DOCUMENT command using the appropriate qualifier.

NOPROFILE, at tag or text, line *n*, file  
*file-spec*  
Tag can appear only following a <PROFILE> tag

**Fatal:** The tag is part of a profile and can be used only in that context and only within the same file as the <PROFILE> tag.

**User Action:** Check that the tag appears after a <PROFILE> tag and in the same file as the <PROFILE> tag.

## Messages

### Tag Translator Messages

NOREADOUT, at tag or text, line *n*, file  
*file-spec*  
Cannot read from file:  
*filename*  
The file is opened as output

**Warning:** A <READ> built-in tag cannot be addressed to an auxiliary file that is opened as output. (The <OPEN> tag defaults to output mode unless INPUT is specified.)

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOREFNOTE, at tag or text, line *n*, file  
*file-spec*  
Accumulated reference notes not referenced.  
Notes will not be output

**Warning:** A document contained <REF\_NOTE> tags, but no corresponding <REF\_NOTES> tag was specified to output the accumulated notes.

**User Action:** Place the <REF\_NOTES> tag in the source file at the position at which you want them output.

NOTDOCTYP, at tag or text, line *n*, file  
*file-spec*  
Tag <*tagname*> is not allowed in the *string* doctype

**Error:** The indicated tag is not meaningful in the context of the doctype specified on the command line and cannot be processed. For example, the LETTER doctype does not allow <CHAPTER> or heading-level tags.

**User Action:** Verify that you specify the correct doctype keyword on the DOCUMENT command line. If the doctype is correct, remove the tags from the SDML source file.

NOTHIDNAM, at tag or text, line *n*, file  
*file-spec*  
A <REVEAL\_TAGS> tag is referencing an unknown hide-name,  
*string*

**Warning:** The hide-name specified by a <REVEAL\_TAGS> tag is not in use.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOTINELEM, at tag or text, line *n*, file  
*file-spec*  
Tag can appear only within the context established by an element heading tag such as <CHAPTER>, <PART>, and so forth.

## Messages

### Tag Translator Messages

**Error:** When a book build is done, all text and the tags that accompany the text must appear within some element of the book. The text and tags must follow the element heading tag in order to be considered to be in the context of the element.

**User Action:** Correct the source text so that the tag that is in error follows a tag that heads an element of the book.

NOTINPROF, at tag or text, line *n*, file  
*file-spec*  
The *<tagname>* tag must appear in the same file as the  
*<PROFILE>* tag

**Error:** All the tags that make up a profile must appear in the same file.

**User Action:** Move the tag into the same file as the *<PROFILE>* tag.

NOUNDEFBI, at tag or text, line *n*, file  
*file-spec*  
Ignoring *<UNDEFINE>* or *<REDEFINE>* of the built-in definition of  
*<tagname>*

**Warning:** A tag translator built-in tag cannot be deleted by the action of a *<REDEFINE>* or *<UNDEFINE>*. The built-in tag can be “stacked” by issuing a *<DEFINE>* tag for it, however.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOWRITEIN, at tag or text, line *n*, file  
*file-spec*  
Cannot write to file:  
*filename*  
The file is opened as input

**Warning:** The *<WRITE>* built-in tag cannot write to an auxiliary file that has been explicitly opened as input.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOZONEYET, at tag or text, line *n*, file  
*file-spec*  
Tag *<tagname>* specified outside its valid zone or context.  
It must be preceded by tag *<tagname>*.

**Error:** A tag was specified outside the context that enables it, for example, a *<COPYRIGHT\_PAGE>* tag was specified when the tag *<FRONT\_MATTER>* has not been specified to enable the front matter zone.

**User Action:** Be sure that you have specified the correct tag. Add the appropriate zone-enabling tag.

## Messages

### Tag Translator Messages

NO\_OUTPUT, No output is produced for this tag

**Informational:** This message accompanies other warning messages. It indicates that the tag is being ignored and no output is produced.

**User Action:** Correct the SDML file to remove the other warning messages.

NO\_PREFIX, at tag or text, line *n*, file  
*file-spec*  
No character string prefix is available.

**Warning:** A tag in a template specified that a character-string prefix was to be used for page or formal element numbering, but no prefix was specified.

**User Action:** Specify a prefix for the section in the <SET\_TEMPLATE\_PREFIX> tag.

NUMBERARG, at tag or text, line *n*, file  
*file-spec*  
Argument *number* is not valid for <tagname>.

**Warning:** The numeric argument is not valid for the tag.

**User Action:** Consult the tag's documentation to determine the valid values.

NUMLTZERO, The argument is less than 0.

**Informational:** This message accompanies a message about an invalid numeric argument. It indicates that a negative value was specified for a tag argument which must be a positive number.

**User Action:** Consult the documentation to determine the tag's valid values.

O\_ARGNTNM, at tag or text, line *n*, file  
*file-spec*  
Argument *number* to tag <tagname> is not a valid name

**Error:** The indicated argument does not obey the rules for formation of a name. A name can have only letters, numbers, and underscores and a name cannot begin with two underscores. You can use spaces, tabs, and carriage returns.

**User Action:** Correct the argument so that the name is valid.

O\_DFINTAG, at tag or text, line *n*, file  
*file-spec*  
A <DEFINE> tag should be removed from this file.  
Consider using a <DEFINE\_SYMBOL> tag in its place.

**Fatal:** The definition of the <DEFINE> tag will not appear in the modified source file.

**User Action:** Tag definitions should be in separate files called with the /SYMBOLS qualifier on the command line.

## Messages

### Tag Translator Messages

O\_FNOTPRB, at tag or text, line *n*, file  
*file-spec*  
Possible problem with footnotes

**Warning:** The software has detected a possible problem with footnotes.

**User Action:** Check to ensure that all footnotes are coded correctly.

O\_IGNRTAG, at tag or text, line *n*, file  
*file-spec*  
Ignoring *<tagname>*.  
You have specified multiple *<tagname>* tags with the MASTER  
argument,  
ignoring all but the first tag.

**Warning:** Multiple *<SET\_ONLINE\_TOPIC>* tags with the MASTER argument  
have been specified.

**User Action:** Remove the extra *<SET\_ONLINE\_TOPIC>* tags.

O\_TAGMISN, at tag or text, line *n*, file  
*file-spec*  
Tag *<tagname>* is missing

**Warning:** A Bookreader *<ONLINE\_PART\_NUMBER>* tag is missing.

**User Action:** Check to ensure that the *<ONLINE\_PART\_NUMBER>* tag exists  
if it is required for your document.

O\_TMNYXTR, at tag or text, line *n*, file  
*file-spec*  
More than 25 characters in argument to tag *<tagname>*.

**Fatal:** There is a limit of 25 characters in the argument to the *<SYMBOL\_*  
*PREFIX>* tag.

**User Action:** Check to ensure that the argument to the *<SYMBOL\_PREFIX>*  
tag has no more than 25 characters.

OBKSYMMSG, at tag or text, line *n*, file  
*file-spec*  
Book does not contain a valid book symbol name.

**Warning:** The book is missing a valid book symbol name.

**User Action:** Check to ensure that the *<ONLINE\_INFO\_SYMBOL>* tag exists  
if it is required for your document.

OCONTOMNY, at tag or text, line *n*, file  
*file-spec*  
Too many *<tagname>* tags.

**Fatal:** Too many *<CONTENTS\_FILE>* tags have been defined for this  
document.

**User Action:** Bookreader documents are limited to one *<CONTENTS\_FILE>*  
tag. Check for too many *<CONTENTS\_FILE>* tags.

## Messages

### Tag Translator Messages

OENONSTNG, at tag or text, line *n*, file  
*file-spec*  
Invalid attempt to nest tag *<tagname>*.

**Error:** A tag is referenced when its context is already active. For example, this message is issued when an `<ONLINE_POPUP>` tag exists after a previous `<ONLINE_POPUP>` tag that you did not terminate.

**User Action:** The indicated tag is ignored; however, you should verify your source file to determine whether the tag is a duplicate or whether, in fact, you did not terminate an earlier occurrence of the same tag.

OESYMISNG, at tag or text, line *n*, file  
*file-spec*  
Missing symbol argument to a *<tagname>* tag.  
Symbol names are required when referencing headings, formal tables, examples, and figures.

**Error:** Symbols are required when referencing headings, formal tables, examples, and figures in Bookreader doctypes.

**User Action:** Supply a symbol name as an argument to the tag. You may want to use the `/GENERATE_SYMBOL` qualifier to automatically generate missing symbols.

OFILNOTCV, at tag or text, line *n*, file  
*file-spec*  
Cannot open new version of :  
*filename*  
File not converted.

**Warning:** An attempt to open an included file failed.

**User Action:** Verify that the file specification is correct, and that you have sufficient resources and access rights to create the file.

OFNONSTFM, at tag or text, line *n*, file  
*file-spec*  
Invalid attempt to place tag *<tagname>* within `<ONLINE_POPUP>/<ENDONLINE_POPUP>` tags.

**Fatal:** `<HEADn>` tags, formal examples, figures, and tables are not allowed between `<ONLINE_POPUP>/<ENDONLINE_POPUP>` tags.

**User Action:** Either remove the `<ONLINE_POPUP>/<ENDONLINE_POPUP>` tags or use an informal example, figure, or table. Also, be sure no `<HEADn>` tags are in the popup.

OFNONSTNG, at tag or text, line *n*, file  
*file-spec*  
Invalid attempt to nest tag *<tagname>*.

**Fatal:** A tag is referenced when its context is already active. For example, this message is issued when an `<ONLINE_POPUP>` tag exists after a previous `<ONLINE_POPUP>` tag that you did not terminate.



## Messages

### Tag Translator Messages

**User Action:** The indicated tag is ignored; however, you should verify your source file to determine whether the tag is a duplicate or whether, in fact, you did not terminate an earlier occurrence of the same tag.

OFONLCHNK, at tag or text, line *n*, file  
*file-spec*

Invalid attempt to place tag *<tagname>* in a formal example, figure, or table.

**Fatal:** *<ONLINE\_CHUNK>* tags are not allowed in formal examples, figures, or tables.

**User Action:** Remove the *<ONLINE\_CHUNK>* tag or use an informal example, figure, or table.

OHIVERNUM, at tag or text, line *n*, file  
*file-spec*

Version number detected on included file, *string*.  
Higher version of that file is being written.

**Warning:** The tag translator has detected a version number on an included file. A file with a higher version number is being written.

**User Action:** The source file has included a file with a specific version number and the symbol generator is writing a file with a higher version number. Either remove the version number or modify the file specification to the higher number.

OINVPREFIX, at tag or text, line *n*, file  
*file-spec*

Invalid symbol argument to */PREFIX=* qualifier on command line.

**Fatal:** Invalid characters have been detected in the symbol argument to the */PREFIX=* qualifier on the command line.

**User Action:** Check to ensure that there are no invalid characters, such as dollar signs (\$) or hyphens (-), in the symbol argument to the */PREFIX* qualifier on the command line.

OSYMNDEF, at tag or text, line *n*, file  
*file-spec*

Symbol *string* is undefined

**Error:** A symbol has been referenced but is not in the symbol table.

**User Action:** Verify that the symbol is spelled correctly and that the same symbol has been entered in the symbol table.

OTTLPGMSG, at tag or text, line *n*, file  
*file-spec*

Tag *<TITLE\_PAGE>* is missing

**Error:** A *<TITLE\_PAGE>* tag is missing. Bookreader doctypes require the *<TITLE\_PAGE>* tag.

**User Action:** Check to ensure that the *<TITLE\_PAGE>* tag exists and is coded correctly.

# Messages

## Tag Translator Messages

OTL\_LEVEL, at tag or text, line *n*, file

*file-spec*

Outline level specified, *number*, is not in range

The argument must specify 1 through 6

**Warning:** An invalid number was specified in a <LEVEL> tag in an outline. Only levels of 1 through 6 are valid.

**User Action:** Correct your source file so that no outline level has a number greater than 6.

OTL\_TITLE, at tag or text, line *n*, file

*file-spec*

<tagname> not valid within outlines.

Enter the title as an argument to the <OUTLINE> tag

**Warning:** A <TITLE> tag was specified within the context of the <OUTLINE> tags.

**User Action:** Correct your source file so that the title of the outline is supplied in arguments to the <OUTLINE> tag.

OUTLIN2BG, at tag or text, line *n*, file

*file-spec*

Output line exceeded *number* bytes, and had to be broken

**Warning:** Text in an output line exceeded the tag translator's buffer. No text was lost because the line was written as two lines, but the division into two lines may have broken a word into two words, or interfered with the correct formatting of the text. This usually happens when the tag translator inserts many formatting commands into an output line. It also happens when a tag reference on a long input line is replaced by a very long string of text.

**User Action:** Check the formatted output that corresponds to the input line specified in the location information to see if breaking the line affected the formatting.

PARSECODE, at tag or text, line *n*, file

*file-spec*

Argument *number* has illegal action code *char*

**Error:** The character shown is not one of the allowed action codes.

**User Action:** Correct the indicated argument to the tag.

PARSELNEQ, at tag or text, line *n*, file

*file-spec*

Lengths of delimiter and action arguments are not equal

**Error:** The number of delimiter characters and the number of action codes in one of the action strings are not the same.

**User Action:** Correct the delimiter or action arguments to the tag.

## Messages

### Tag Translator Messages

POSTERROR, at tag or text, line *n*, file  
*file-spec*  
Error encountered during post-translation

**Fatal:** An error has been detected during the final step in tag translation.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

PQLISNONE, at tag or text, line *n*, file  
*file-spec*  
<tagname> specifies NONE when no heading is specified

**Warning:** The keyword argument "NONE" was specified to a <PARAMDEFLIST> or <QUALDEFLIST> tag in a context in which no default heading is in effect. The use of NONE in this context is meaningless.

**User Action:** Determine if you intended to be using default headings for the <PARAMDEFLIST> or <QUALDEFLIST> tag. If not, remove the tag from your source file. Otherwise, verify whether you should have invoked one of the reference templates to set an environment for parameter/qualifier list descriptions.

PROFORDER, at tag or text, line *n*, file  
*file-spec*  
<PROFILE> tag is out of order

**Fatal:** A <PROFILE> tag has been detected after an element heading tag, such as <CHAPTER> or <APPENDIX>.

**User Action:** Place a single <PROFILE> . . . <ENDPROFILE> sequence in a file by itself in order to do a book build.

PROFQ\_ILL, at tag or text, line *n*, file  
*file-spec*  
/PROFILE qualifier illegal during a book build

**Fatal:** The /PROFILE qualifier was supplied on the command line, indicating that an element build is desired, but the SDML file contains a <PROFILE> tag, indicating that a bookbuild is desired. The /PROFILE qualifier (which indicates an element build) should not be supplied when a bookbuild is desired.

**User Action:** Invoke DOCUMENT without the /PROFILE qualifier.

QUALVALUE, at tag or text, line *n*, file  
*file-spec*  
Value *string* to /CONDITION qualifier is not a valid name

**Fatal:** The condition name supplied with the /CONDITION qualifier must contain only alphabetic and numeric characters and the underscore character.

**User Action:** Correct the spelling of the condition name.

## Messages

### Tag Translator Messages

Q\_BAD\_POS, at tag or text, line *n*, file

*file-spec*

Argument 2, *string*, is invalid for queue *name*

**Error:** A position argument supplied with a queueing built-in tag is invalid. The numeric value of the position may be out of range or an invalid keyword may have been used.

**User Action:** Correct the tag's position argument.

Q\_NONUM\_E, at tag or text, line *n*, file

*file-spec*

Entry in queue *name* is not numeric

**Error:** An <ENQUE> tag has requested that a new value be inserted in a queue in numeric sorted order. During the insertion a non-numeric entry is discovered in the queue. This can happen only if the entry was enqueued by a method other than numeric sorted order.

**User Action:** Review the <ENQUE> tags that insert entries on this queue and insure that all tags request a numeric sorted order.

Q\_TOOLONG, at tag or text, line *n*, file

*file-spec*

Concatenation of all *name* queue entries exceeds max length

**Error:** A <DEQUE>(name\ALL\RETURN) tag has been encountered. The concatenation of all the entries on the queue creates a single string that exceeds the maximum string length (65,012 bytes).

**User Action:** Adopt a different queueing method, breaking the queue into more than one queue.

REVISINFO, Document is using revision indicators

**Informational:** Indicates that the <REVISION> tag is being processed. Text that is delimited by <MARK> and <ENDMARK> tags will be accompanied by revision indicators on output.

**User Action:** If you intend for revision bars to print in your text, you do not need to take any action. However, if you intended for text indicated by <MARK> tags to not appear in the output, you should remove the <REVISION> tag from the file.

REVISTEXT, Revised pages will contain the text *string*.

**Informational:** This message verifies the text specified in the second argument to the revision tag and reminds you that only pages specified in an update range will be processed and printed.

**User Action:** None.

## Messages

### Tag Translator Messages

REVIS\_UPD, Revision will contain update pages only.

**Informational:** This message reminds you that you specified <REVISION>(UPDATE) and that your output file will only contain the text for pages specified between <UPDATE\_RANGE> and <ENDUPDATE\_RANGE> tags.

**User Action:** None.

REV\_PAGES, Update pages *number* through *number* will be output

**Informational:** This message indicates the pages that will be output for an update.

**User Action:** None.

RMVCONDITN, Removing condition *condition-name*  
on line *number* of file *filespec*

**Informational:** A <SET\_CONDITION>(REMOVE) tag is being executed.

**User Action:** None.

SAVTAGACT, at tag or text, line *n*, file  
*file-spec*

Tag <*tagname*> has a before or after action that is ignored when saving a tag table

**Warning:** You are attempting to invoke a tag that has a before action or an after action. The before action or after action cannot be executed while you are saving the tag in a tag table. The before action or after action will be executed when the saved tag table is loaded.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

SAVTAGILL, at tag or text, line *n*, file  
*file-spec*

This built-in tag cannot be invoked during the saving of a tag table

**Error:** Only certain built-in tags can be stored in a saved tag table for invocation when the saved tag table is loaded. This tag is not one that can be saved.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

SAVTAGLEV, at tag or text, line *n*, file  
*file-spec*

Tag <*tagname*> cannot be invoked within an argument list during saving of a tag table

**Error:** During the saving of a tag table, the arguments to all tags must be constant or must be made constant by quoting them.

## Messages

### Tag Translator Messages

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

SDML\_INFO, *message*

**Informational:** An SDML tag is describing action taken during tag loading.

**User Action:** None.

SECFILOPN, at tag or text, line *n*, file

*file-spec*

Cannot open a secondary output file named

*filename*

A secondary output file is already open. It is named

*filename*

**Error:** Only one secondary output file can be open at a time.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

SETCONDNTN, Setting condition *condition-name*  
on line *number* of file *filespec*

**Informational:** A <SET\_CONDITION> tag is being executed or a condition is being set in response to a /CONDITION qualifier on the command line.

**User Action:** None.

SKIPITEMS, at tag or text, line *n*, file

*file-spec*

*number* <*tagname*> outside of <*tagname*> skipped.

**Warning:** This message is usually preceded by a message indicating that an <LE> or <TABLE\_ROW> tag was encountered. It indicates the number of additional tags that were specified. The tags will not be processed, and will be written, unformatted, to the output file.

**User Action:** Determine if the tags are considered invalid because there is no preceding <TABLE> or <LIST> tag, or if the <TABLE> or <LIST> tag was specified incorrectly. Correct your source file.

STKOVRFW, at tag or text, line *n*, file

*file-spec*

More than *number* nested levels of tags. (These are tags that are invoked inside other tags' argument lists)

**Fatal:** The tag translator limits the number of tags that can be invoked simultaneously. Tag B is invoked simultaneously with tag A, if it is invoked in the argument list to tag A. Thus, <A><B><C> requires that three tags are being invoked simultaneously. This may occur unexpectedly when the argument list to one of the tags is not properly terminated.

**User Action:** Verify that the argument list of the tag that is specified in the location information has been properly terminated.

## Messages

### Tag Translator Messages

STRFREE2T, at tag or text, line *n*, file  
*file-spec*

Internal error. Attempt to free the same string twice

**Fatal:** The tag translator's memory allocation algorithm has failed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

STRFREEBA, at tag or text, line *n*, file  
*file-spec*

Internal error. Attempt to free string with bad address

**Fatal:** The tag translator's memory allocation algorithm has failed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

STRTOOBIG, at tag or text, line *n*, file  
*file-spec*

Internal error. Attempting to allocate *number*-byte string

**Fatal:** The tag translator's memory allocation algorithm has failed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

SYMFREEBA, at tag or text, line *n*, file  
*file-spec*

Internal error. Attempt to free symbol table entry with bad address

**Fatal:** The tag translator's memory allocation algorithm has failed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

SYMISSING, at tag or text, line *n*, file  
*file-spec*

Missing symbol argument.

Each book element must have a symbol argument

**Fatal:** Each element of a book must have a unique symbol so that the element can be processed independently. The symbol table cannot be saved from this run.

**User Action:** Supply a symbol as an argument to the tag.

## Messages

### Tag Translator Messages

SYMISUSED, at tag or text, line *n*, file  
*file-spec*  
The symbol *string* is already in use as a symbol of type *string*

**Error:** Each element of a book must have a unique symbol. You have supplied a symbol on this element of the book that conflicts with an earlier use of this symbol. The symbol table cannot be saved from this run.

**User Action:** Correct the use of symbols on elements of the book so that each element has a unique symbol. If you are doing a book build you must repeat the book build in order to produce a good symbol table. If you are doing an element build, just correct the symbol in this tag so that this element of the book has a unique symbol, and then repeat the element build.

SYMNOTDEF, at tag or text, line *n*, file  
*file-spec*  
Symbol *string* is undefined

**Warning:** A symbol has been referenced but is not in the symbol table.

**User Action:** Verify that the symbol is spelled correctly and that the same symbol has been entered in the symbol table by the invocation of a tag that defines the symbol.

TAGFREEBA, at tag or text, line *n*, file  
*file-spec*  
Internal error. Attempt to free tag table entry with bad address

**Fatal:** The tag translator's memory allocation algorithm has failed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

TAGINFILE, The tag was in the file *filename*

**Informational:** This message may accompany a message indicating that a tag that must be ended was not terminated. This message is output when the starting tag is not in the current input file.

**User Action:** Use the file spec given to determine which file contains the tag that was not properly terminated and terminate it.

TAGINVALD, at tag or text, line *n*, file  
*file-spec*  
Tag *<tagname>* is invalid in this context.  
The context was established by *<tagname>* on line *number* of *filename*

**Error:** A tag was specified in a context where it is not valid, for example, a header-level tag was specified within the context of a monospaced example. The message tells you what tag established the context.

**User Action:** Determine the current context at the point at which the message was issued to determine if you have forgotten a required terminator and correct your source file.



## Messages

### Tag Translator Messages

TAGINZONE, at tag or text, line *n*, file  
*file-spec*

Tag *<tagname>* is invalid in the bounds of *string*.

**Error:** A tag was specified in a context where it is not valid, for example, a header-level tag was specified within the context of a monospaced example.

**User Action:** Determine the current context at the point at which the message was issued to determine if you have forgotten a required terminator and correct your source file.

TAGNOTDEF, at tag or text, line *n*, file  
*file-spec*

Tag *<tagname>* is undefined

**Warning:** A tag has been invoked, but no definition exists for the tag. This message appears for the first invocation of an undefined tag. Subsequent invocations of the same undefined tag are counted, but the message is not repeated. A count of the total number of all invocations of undefined tags is reported at the end of the tag translator's processing.

**User Action:** Correct the spelling of the tag or supply a definition for the tag.

TAGNOTEND, at tag or text, line *n*, file  
*file-spec*

Tag *<tagname>* from line *number* not terminated

**Error:** A tag that has a required terminator was not terminated.

**User Action:** Locate the tag that was not terminated and include its terminating tag at the appropriate position in your source file.

TAGNOTHID, at tag or text, line *n*, file  
*file-spec*

Cannot reveal tag *<tagname>*. It is not hidden

**Warning:** A `<REVEAL_TAGS>` tag is requesting that a hidden tag be revealed, but the tag is not currently hidden.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

TAGNOTPRE, at tag or text, line *n*, file  
*file-spec*

Tag *<tagname>* invalid unless preceded by *string*

**Warning:** A tag is being invoked that is invalid because it is being used out of context.

**User Action:** Check the sequence of tags to ensure that the proper context is established before the tag is used. Correct the SDML source file.

## Messages

### Tag Translator Messages

TAGNOTPRX, at tag or text, line *n*, file  
*file-spec*  
Tag *<tagname>* invalid unless preceded by an example tag

**Warning:** A tag is being invoked that is invalid because it is not being used within an example.

**User Action:** Check the sequence of tags to ensure that an example tag is used before the `<VALID_BREAK>` tag. Correct the SDML source file.

TAG\_FAILS, The tag translator has detected a fatal error

**Error:** The tag translator is terminating execution with failure status. The accompanying error message supplies details of the error.

**User Action:** See the description of the accompanying error message.

TAG\_IDENT, *message*

**Informational:** This message will appear only in patched or baselevel systems. The tag translator inserts the text, which is usually some kind of internal baselevel indicator.

**User Action:** None.

TAG\_INARG, at tag or text, line *n*, file  
*file-spec*  
*<tagname>* is invalid in an argument to *<tagname>*.

**Fatal:** The indicated tag was specified in an argument to a tag that does not allow it. For example, a header level tag was specified in an argument to `<TABLE_ROW>`.

**User Action:** Verify that you correctly terminated the argument list for the tag in which this tag was specified.

TBLATTIGN, at tag or text, line *n*, file  
*file-spec*  
*<TABLE\_ATTRIBUTES>* specified following *<TABLE\_SETUP>* ignored.

**Warning:** The `<TABLE_ATTRIBUTES>` tag must be specified preceding the `<TABLE_SETUP>` tag. The specified attributes will be ignored.

**User Action:** Move the `<TABLE_ATTRIBUTES>` tag in front of the `<TABLE_SETUP>` tag.

TBLATTRIB, at tag or text, line *n*, file  
*file-spec*  
Argument *string* to tag *<tagname>* is not a valid table attribute

**Warning:** A keyword was specified in a `<TABLE_ATTRIBUTES>` tag that is not a valid keyword attribute for tables.

**User Action:** Verify that the specified value is a valid table attribute and check its spelling. Correct the SDML source file.

## Messages

### Tag Translator Messages

TBLBRKCOL, at tag or text, line *n*, file  
*file-spec*  
Table cannot be broken in column *number*

**Warning:** The <VALID\_TABLE\_ROW\_BREAK> tag is valid only in columns two through four of a table.

**User Action:** Remove the <VALID\_TABLE\_ROW\_BREAK> tag. If the text in column is too long for a page, consider restructuring the table to be less complex.

TBLCOLCNT, at tag or text, line *n*, file  
*file-spec*  
Tag <TABLE\_SETUP> argument 1 *string* is missing or is not a number in the range 2 through 9

**Error:** The <TABLE\_SETUP> tag must specify a numeric value in the range of 2 through 9 to specify the number of columns in the table. The indicated argument is invalid.

**User Action:** Correct the <TABLE\_SETUP> tag.

TBLCOLNUM, at tag or text, line *n*, file  
*file-spec*  
Table column size argument *string* is missing or is not numeric

**Error:** An argument specified as a width for table rows in setting up the table is not a numeric argument.

**User Action:** Correct the <TABLE\_SETUP> tag that specifies a non-numeric value for a table column width.

TBLCOLWID, at tag or text, line *n*, file  
*file-spec*  
Sum of table column widths *number* is too large or is 0 or less.

**Error:** The sum of the values specified for the column widths in the <TABLE\_SETUP> tag is excessive or is less than or equal to 0. The message displays the current summed value of all column widths specified.

**User Action:** Verify that the numbers you specified in <TABLE\_SETUP> are accurate. Adjust them to smaller numbers so that the table will fit. If you did not specify <TABLE\_ATTRIBUTES>(WIDE) and the doctype you are using provides an extra margin for wide tables, specify <TABLE\_ATTRIBUTES>(WIDE) in front of the <TABLE\_SETUP> tag. To request that the table appear in a smaller typesize, use <TABLE\_ATTRIBUTES>(MAXIMUM).

TBLDUPSET, at tag or text, line *n*, file  
*file-spec*  
Tag <TABLE\_SETUP> specified twice. Duplicate setup ignored.

**Warning:** A table contains two <TABLE\_SETUP> tags. The second tag is being ignored.

**User Action:** Determine which <TABLE\_SETUP> tag is correct and remove the extra one.

## Messages

### Tag Translator Messages

TBLFNOTEM, at tag or text, line *n*, file  
*file-spec*  
Table footnote number *number* exceeds maximum *number*

**Warning:** A numeric argument specified for a footnote declaration in a table exceeds the maximum. The maximum number of table footnotes is normally 12; however, if you use any non-numeric keywords in table footnotes, the maximum number allowed is reduced to 6.

**User Action:** Verify the number of footnotes in the table and the numeric arguments you use to declare and reference them.

TBLKEYDUP, at tag or text, line *n*, file  
*file-spec*  
Table already specifies <TABLE\_KEY>. Duplicate ignored.

**Warning:** Only one key can be specified in a table. The second key specified is being ignored.

**User Action:** Determine which table key is the one that you want and remove the extra one.

TBLKEYNES, at tag or text, line *n*, file  
*file-spec*  
<*tagname*> is not valid in a nested table.  
Nested table on line *number* in file *string*.

**Warning:** A <TABLE\_KEY> or <TABLE\_KEYREF> tag was specified in a nested table, that is, in a table that is specified in another table's <TABLE\_ROW>. Table keys and table key references are invalid in nested tables.

**User Action:** Place the table key and/or the reference to it in the outer table.

TBLKEYNUM, at tag or text, line *n*, file  
*file-spec*  
The <TABLE\_KEY> tag will replace table footnote 12.

**Warning:** When a key is specified in a table, you cannot specify a table footnote numbered 12. The text of the table key will replace the footnote number 12, and references to footnote 12 will be result in the table key being printed.

**User Action:** Check the arguments you specified to the <FOOTNOTE> tags in the table and do not specify a number greater than 11.

TBLKNOREF, at tag or text, line *n*, file  
*file-spec*  
Tag <TABLE\_KEYREF> tag encountered in a table that has no key.

**Warning:** A <TABLE\_KEYREF> tag is not valid because there is no key in the table to reference.

**User Action:** Verify that the <TABLE\_KEY> tag is correctly specified in the main table (it must not be specified in a nested table). If the table has no key, remove the <TABLE\_KEY> tag.

## Messages

### Tag Translator Messages

TBLNBREAK, at tag or text, line *n*, file  
*file-spec*

A nested table with more than three columns cannot be broken.

**Error:** The <NESTED\_TABLE\_BREAK> was specified in a nested table that has more than three columns.

**User Action:** Verify that the <TABLE\_SETUP> tag for the table you are trying to break was correctly specified. If the nested table has four columns, you may need to restructure your information.

TBLNONEST, at tag or text, line *n*, file  
*file-spec*  
Invalid table nesting. Ignored

**Warning:** You coded a nested table incorrectly. There are three possible problems:

- 1 You specified a <TABLE> tag with a caption argument in an argument to a <TABLE\_ROW> tag. You cannot place a formal table inside another formal table.
- 2 You specified a <TABLE> tag inside another table, but outside of a <TABLE\_ROW> tag. All text (including nested tables) that occur within a table must be coded within <TABLE\_ROW> tags.
- 3 You specified a <TABLE\_UNIT> tag inside a table that is nested in another table. <TABLE\_UNIT> tags are not valid in nested tables.

**User Action:** Based on the above situations, you should:

- 1 Remove the caption argument from the nested table tag.
- 2 Nest the table inside a <TABLE\_ROW> tag.
- 3 Remove the <TABLE\_UNIT> tags.

TBLNOROWS, at tag or text, line *n*, file  
*file-spec*

A table was specified without any table rows.

**Error:** <TABLE> and <ENDTABLE> tags were encountered in the file, but there were no <TABLE\_ROW> tags between them.

**User Action:** Verify that you specified <TABLE\_ROW> tags correctly. Or, remove the <TABLE> and <ENDTABLE> tags if you did not want to place a table there.

TBLNOTDEC, at tag or text, line *n*, file  
*file-spec*

Table footnote *number* is referenced but not declared.

**Warning:** A <FOOTREF> tag is specified in a table, but there is no corresponding <FOOTNOTE> tag at the beginning of the table. The footnote reference will not be printed.

**User Action:** Verify that you specified the <FOOTNOTE> tag correctly at the beginning of the table. Verify that the number specified in the declaration matches the number in the <FOOTREF> tag.

## Messages

### Tag Translator Messages

TBLNOTMTP, at tag or text, line *n*, file  
*file-spec*  
<tagname> invalid in table that is not multi-page

**Error:** A tag, for example <TABLE\_UNIT> was specified in a table that does not have the multipage attribute.

**User Action:** Determine whether the table is a multipage table. If so, be sure that it is not specified with the KEEP attribute.

TBLNOTROW, at tag or text, line *n*, file  
*file-spec*  
Tag <tagname> specified outside of <TABLE\_ROW>.

**Error:** A tag is specified between two <TABLE\_ROW> tags and is therefore not valid in this context.

**User Action:** Correct your source file to move the tags and related text, if any, to within the bounds of a <TABLE\_ROW> tag.

TBLNOTSET, at tag or text, line *n*, file  
*file-spec*  
Table from line *number* has no <TABLE\_SETUP> attributes.

**Error:** A <TABLE\_ROW> tag was encountered in a table without a <TABLE\_SETUP> tag.

**User Action:** Correct your source file to specify the number of columns and widths for the table.

TBLNUMCOL, Table row column count does not match setup

**Informational:** A tag, such as <TABLE\_ROW> or <TABLE\_HEADS>, specifies more or fewer arguments than indicated by the table setup. For example, a <TABLE\_SETUP> tag indicates that a table has four columns, but a <TABLE\_ROW> tag specifies only three arguments.

**User Action:** Determine whether you have not specified a column that is required, or whether you have correctly specified the <TABLE\_SETUP> tag.

TBLSPANNO, at tag or text, line *n*, file  
*file-spec*  
Argument *number* to tag <SPAN> is invalid for the current column count.

**Warning:** A <SPAN> tag's argument exceeded the number of remaining columns in the table. For example, you specified <SPAN>(3) in a table with only 2 columns.

**User Action:** Correct your source file to specify the number of columns to span. Be sure that your numeric argument, plus the current column number, do not exceed the number of columns in the table.

## Messages

### Tag Translator Messages

TEXT\_FORB, at tag or text, line *n*, file  
*file-spec*  
Text forbidden after *<tagname>* tag

**Warning:** The *<tagname>* tag cannot be followed by a text segment.

**User Action:** Correct your source file to remove the text that follows the tag or tag the text correctly.

TEXT\_NEST, at tag or text, line *n*, file  
*file-spec*  
Attempt to pop a text state that has not been pushed

**Error:** A *<SET\_TEXT\_STATE>*(POP) tag is being invoked when there is no text state saved.

**User Action:** Examine the tag definitions that contain the *<SET\_TEXT\_STATE>*(POP) tag for a missing *<SET\_TEXT\_STATE>*(PERMITTED\PUSH) tag.

TEXT\_REQD, at tag or text, line *n*, file  
*file-spec*  
Text missing after *<tagname>* tag

**Warning:** The *<tagname>* tag marks a text segment, but the text is missing.

**User Action:** Correct your source file to use the tag correctly.

TMPNONEST, at tag or text, line *n*, file  
*file-spec*  
Invalid attempt to nest reference templates.  
Last template was *string*

**Error:** A tag invoking a reference template was encountered when a reference template was already in effect; for example, a *<ROUTINE\_SECTION>* is specified before a previous *<COMMAND\_SECTION>* was terminated.

**User Action:** Correct your source file to correctly terminate the reference section that was not terminated.

TMPTAGDEF, You cannot redefine *<tagname>* using *<tagname>*.

**Informational:** A tag that defines a template tag specified a tag name that is already a defined tag. This message indicates one of the following:

- Your input file has more than one reference template section, and you use the same template tag names in both. (For example, a *<SET\_TEMPLATE\_ROUTINE>* tag may specify the same tag name in more than one occurrence of *<ROUTINE\_SECTION>*).
- You have specified a name that is the name of an SDML global tag.

**User Action:** If the message occurred because you used the same template tag names in more than one reference template section, you need take no action. If you have used a name that is the name of an SDML tag, you should select another name; future use of the SDML tag name in the same source file may produce unpredictable results.

## Messages

### Tag Translator Messages

TMPTAGEND, at tag or text, line *n*, file  
*file-spec*  
<tagname> not terminated before <tagname>.

**Warning:** A tag occurring within the context of a reference template was not terminated before the end of the file or before the beginning of the next reference section. For example, an <OVERVIEW> tag from a <COMMAND> description was not terminated before the next <COMMAND> description was encountered.

**User Action:** Correct the source file so that you correctly terminate the template tag.

TMPTAGNAM, at tag or text, line *n*, file  
*file-spec*  
*string* is an invalid template tag name.

**Error:** A template tag, for example <SET\_TEMPLATE\_COMMAND>, specifies that the tag name that already defined in SDML, or it is an invalid name, that is it contains more than 31 characters, or it contains nonalphanumeric characters.

**User Action:** Choose another name for the tag and modify your source file.

UNBALPARS, at tag or text, line *n*, file  
*file-spec*  
Unbalanced parentheses in argument *number*

**Warning:** One or more left parentheses are present without matching right parentheses. A sufficient number of right parentheses are appended to the argument to achieve a balance.

**User Action:** Correct the unbalance either by adding the proper number of right parentheses or by coding the left parentheses using the <OPAREN> tag.

UNDEFSYMS, There were *number* undefined symbol references

**Informational:** Reports the total number of references that were made to undefined symbols. Only the first reference to a specific undefined symbol is reported.

**User Action:** Either define the undefined symbol or correct the spelling of the symbols being referenced. The undefined symbols were reported in other messages.

UNDEFTAGS, There were *number* undefined tag invocations

**Informational:** Reports the total number of invocations of undefined tags. Only the first invocation of a specific undefined tag is reported.

**User Action:** Either define the undefined tags or correct the spelling of the tag being invoked. The undefined tags were reported in other messages.



## Messages

### Tag Translator Messages

UPDATINFO, Update page label *string*, *<tagname>*

**Informational:** A document that is being processed to produce update pages may have associated update information. The information indicated will be printed on the bottom of each page output.

**User Action:** Verify that the update page label is correct for your document.

USEDEFINC, Using a value that is one greater than previous value

**Informational:** Accompanies other warning messages. It indicates that a default value is being used that is one greater than a previous value.

**User Action:** Edit the SDML source file to correct the argument.

USENODLIM, Using no delimiters

**Informational:** Accompanies other warning messages. It indicates that a stacked list will be output without delimiters.

**User Action:** Edit the SDML source file to correct the argument.

USER\_IMSG, *string*  
Line is *number* of file *filename*

**Informational:** This message code is used for messages written to the terminal or output stream when a `<USER_I_MESSAGE>` tag is specified in the input file.

**User Action:** The action you take depends on the content of the user's message.

USER\_WMSG, at tag or text, line *n*, file  
*file-spec*  
*string*

**Warning:** This message code is used for messages written to the terminal or output stream when a `<USER_W_MESSAGE>` tag is specified in the input file.

**User Action:** The action you take depends on the content of the user's message.

USESPCOUT, Alignment characters will be output as spaces

**Informational:** Accompanies other warning messages. It indicates that a default value of space will be used as output for the alignment character.

**User Action:** Edit the SDML source file to correct the argument.

USETAGDEF, Using tag's default: *number*

**Informational:** Accompanies other warning messages. It generally indicates that a default value is being taken.

**User Action:** Edit the SDML source file to correct the argument.

## Messages

### Tag Translator Messages

VBARINARG, at tag or text, line *n*, file  
*file-spec*  
A vertical bar was found in argument *number* to tag  
<*tagname*>.  
Use the <VBAR> tag to code a vertical bar in an argument

**Fatal:** The tag translator treats the vertical bar character as a special operator character in arguments to tags. If you include a vertical bar character in an argument, the tag translator may overlook the right parenthesis that terminates the argument list, and may scan all of the remaining input looking for the proper terminator.

If you have mistakenly omitted the closing parenthesis of an argument list, the tag translator begins to search the remainder of your file for a closing parenthesis. If it encounters a vertical bar (perhaps in a figure or code example), it then begins to search for a matching ampersand. Thus, this error message may be issued instead of the EOFARGLST error message.

**User Action:** Correct the argument list by replacing the vertical bar with the <VBAR> tag or by properly closing the argument list.

VMOVRFLOW, at tag or text, line *n*, file  
*file-spec*  
Internal error. Failure to allocate additional space in virtual memory

**Fatal:** This error message is generally caused by one of two types of error in the SDML file. These errors, and the action that you should take to correct them, are as follows:

- An unterminated argument list. If you fail to terminate an argument list, the tag translator regards the text that follows the argument list as part of the last argument. Suppose you write

```
<EMPHASIS>(always\BOLD) press the RETURN key.
```

instead of

```
<EMPHASIS>(always\BOLD) press the RETURN key.
```

The tag translator does not realize that the right angle bracket should have been a right parenthesis. It tries to include all the following words as part of the second argument of the <EMPHASIS> tag. This means that it may store large portions of the SDML file in memory while it searches for a closing parenthesis to the argument list. If memory limits are exceeded, the VMOVRFLOW error message is issued.

- Too many tag definitions. The tag translator has no limit on the number of tag definitions. However, a limit is imposed by the memory allocation algorithm. When this limit is exceeded, it usually indicates that one or both of the following conditions are present:

## Messages

### Tag Translator Messages

- A file that contains tag definitions is being repeatedly included. This file is often a global definitions file that is being used to define tags or symbols for text substitution. You can avoid memory limitation problems by using the <CHECK\_FOR\_INCLUSION> tag at the start of such a file. This ensures that the file's definitions are actually loaded only once, even though the file is included many times.
- A tag's definition changes frequently during execution, perhaps when the tag is being used to temporarily hold a text string. If the tag's definition is established each time by the <DEFINE> tag rather than by the <REDEFINE> tag, the definitions pile up in memory. A reference to the tag always retrieves the latest definition, and if you never use the <UNDEFINE> tag to erase the latest definition, you will never see the old definitions, but they are still occupying memory space.

**User Action:** Determine if the error was caused by an unclosed argument list. The error is generally located by the information supplied in the error message.

Otherwise, examine your tag definitions to determine whether you are defining the same tag name multiple times. You can avoid memory limitation problems by defining your tags with the <REDEFINE> tag, rather than with the <DEFINE> tag. Unless you specifically intend to “stack” tag definitions, and use the <UNDEFINE> tag to “pop” definitions off the stack, you should always establish a tag definition with the <REDEFINE> tag.

XRFINCMPL, at tag or text, line *n*, file  
*file-spec*  
The file containing cross reference symbols is incomplete.  
*filename*

**Fatal:** During the reading of the file containing cross reference symbols, an error was detected that indicates that the file was not recorded correctly.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

XRFINVALD, at tag or text, line *n*, file  
*file-spec*  
The file containing cross reference symbols is invalid.  
*filename*

**Fatal:** The file was opened without error, but it does not contain cross reference symbols.

**User Action:** The wrong file is being referenced. Check the file name supplied with the /PROFILE qualifier.

## Messages

### Tag Translator Messages

XRFNORENM, at tag or text, line *n*, file  
*file-spec*

The temporary cross reference file cannot be renamed to  
*filename*

**Fatal:** The tag translator creates a temporary cross reference file with a file type TMP. When the file is complete, it renames it with a file type XREF. The rename operation has failed.

(You must have Delete privileges for the directory in which the TMP file is being renamed.)

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

---

## B.3 Text Processor Messages

ATPT, Font was read at *number* pt

**Informational:** This is an internal error.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADALPHA, Improper alphabetic constant

**Warning:** Text formatter expected to scan a constant consisting only of letters.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADCHAR, Text line contains an invalid character

**Warning:** Text formatter could not scan a character from the current input line.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADCODE, Bad code number: *value*, must be between *smaller number* and *larger number* using zero instead

**Warning:** Text formatter expected to find a code value within the range shown.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADCS, Bad control sequence: *Command*

**Warning:** Text formatter could not understand how the user meant to use the shown command in the current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADDISC, Improper discretionary list. Must contain only kerns and boxes

**Warning:** Discretionary list contained unexpected information.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

**BADDISCMATH,** Illegal discretionary break in math mode.  
The third part of a discretionary break must be empty  
in this mode. Attempting to recover...

**Warning:** Discretionary break was improperly coded.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADDUMP,** Cannot \dump inside a group

**Warning:** Command \dump occurred when there was one group or more still active.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADFONTFMT,** Fatal font format file error

**Error:** Text formatter could not load the font format file, or the font format file was fatally flawed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADFRACTION,** Ambiguous fraction ignored.

**Warning:** Text formatter could not determine the meaning of a fraction.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADGLUE,** Infinite glue. Attempting to recover

**Warning:** Text formatter was asked to typeset a box using infinite glue. Glue was made finite and the text formatter attempted to continue.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADHALIGN,** Improper \halign

**Warning:** Command \halign is invalid in current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADHRULE,** Cannot use \hrule in an \hbox.  
Use \leaders or \hrulefill instead

**Warning:** Command \hrule is invalid in current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

**BADHYPHEN,** Improper \hyphenation ignored.  
Must contain only letters and hyphens

**Warning:** Text formatter can only read letters and hyphens in a hyphenation pattern.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADHYPHMT,** Fatal hyphenation format file error

**Error:** Text formatter could not load the hyphenation format file, or the hyphenation format file was fatally flawed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADLASTBOX,** \box255 is not empty, attempting to continue

**Warning:** Text formatter expected \box255 to be empty.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADLEADERS,** \leaders not followed by proper glue

**Warning:** Glue specification was not valid in the current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADLETINPAT,** Non-letter found in \patterns

**Warning:** Text formatter can only read letters in a pattern.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADLIMIT,** Ignoring misplaced \limits or \nolimits command

**Warning:** Text formatter did not expect to see \limits or \nolimits in this context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**BADMACROFMT,** Fatal macro format file error

**Error:** Text formatter could not load the macro format file, or the macro format file was fatally flawed.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

BADMAG, Illegal magnification ratio: *bad number*, value set to *good number*

**Warning:** Text formatter could not set the magnification to the requested value.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADMATHCHAR, *font number font-number* is missing this character:  
*character*

**Warning:** Text formatter could not typeset the character shown, because that character was missing from the font shown.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADMATHDELIM, Inserted missing or misplaced math delimiter

**Warning:** Text formatter became confused while closing a math formula.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADNOALIGN, `\noalign` can only appear after `\cr`

**Warning:** Command `\noalign` is not valid in the current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADOMIT, `\omit` can only appear after `\cr` or `&`

**Warning:** Command `\omit` is not valid in the current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADOUTLOOP, Output loop repeated *number of cycles* times without doing a  
`\shipout`  
Attempting to recover. Increase `\maxdeadcycles` if necessary

**Warning:** Text formatter cycled through its output loop too many times without doing a `\shipout`.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.



## Messages

### Text Processor Messages

BADPARAMNUM, Illegal parameter number

**Warning:** Text formatter became confused while scanning parameters to a macro.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADPARAMSEQ, Parameter was numbered out of sequence

**Warning:** Text formatter became confused while scanning parameters to a macro, because a parameter was numbered out of sequence.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADPATTERNS, Bad \patterns

**Warning:** Text formatter became confused while reading a pattern.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADPREAMBLE, Cannot nest alignment preambles

**Error:** Text formatter became confused during current alignment, because it found an alignment within an alignment.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADPREFIX2, Can't use \long or \outer with *command*

**Warning:** Text formatter will ignore \long or \outer or \global when used with the shown command.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADPREFIX3, Can't use \long or \outer or \global with *command*

**Warning:** Text formatter will ignore \long or \outer when used with the shown command.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADPREVGRAF, \prevgraf = *number*. Value must be positive

**Warning:** Text formatter expected to see \prevgraf specified with a positive value.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

BADPT, Bad font size = *number* pt, replaced by 10 pt

**Warning:** Text formatter attempted to read a font at an improper font size.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report. )

BADUNBOX, Cannot unbox a *hbox* or *vbox* in *Text formatter mode* mode

**Warning:** Command unbox is invalid under current circumstances.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADUNIT, Bad unit of measure, replaced by *string*

**Warning:** Text formatter could not use the requested unit of measure. Unit was changed to shown value.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BADVALUE, *quantity = quantity*. Value must be between *smaller number* and *larger number*

**Warning:** Text formatter was expecting to read a value within the range shown.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

BUFFERSIZE, Maximum number of characters in the input buffer is *number*

**Informational:** User has exceeded total size of text formatter's input buffer.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

CANNOTRESETMAG, Cannot change magnification, using old value: *number*

**Warning:** Text formatter can only set magnification value once.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

CONFUSION, Text formatter became confused doing this: *Command*,  
job is aborting

**Error:** Text formatter became confused doing the shown activity and had to exit.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DIMENTOOBIG, Dimension too large

**Warning:** Text formatter became confused while scanning a dimension, because the dimension was too big.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DISCTOOLONG, Discretionary list is too long. Attempting to recover

**Warning:** Discretionary list was too long.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DOUBLESUB, Double subscript

**Warning:** Text formatter found a quantity with two subscripts. In these cases,  $x^1^2$  will be treated like  $x^1\{\}^2$ .

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DOUBLESUPER, Double superscript

**Warning:** Text formatter found a quantity with two superscripts. In these cases,  $x_1_2$  will be treated like  $x_1\{\}_2$ .

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DURING, Error occurred during a *command*

**Warning:** Error occurred during a specified command or control sequence.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

DURINGOR, Error occurred during a *command* or *string*

**Warning:** Error occurred during one of the two commands or control sequences shown.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

EMERGENCYSTOP, Job is aborting

**Error:** Text formatter became confused and had to exit.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ENDDURING, \end occurred during *command*

**Warning:** Command \end occurred unexpectedly.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ENDINGROUP, \end occurred inside group at level *Number*

**Warning:** Command \end occurred when there was one group or more still active.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

EOF, Unexpected end of file

**Warning:** Text formatter encountered the end of the current file unexpectedly.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

EOP, Paragraph ended before the following completed:

**Warning:** Text formatter encountered the end of a paragraph unexpectedly.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

EXTRATAB, Extra \halign or & in alignment has been changed to \cr

**Warning:** Text formatter found extra or unexpected \halign or &.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

FILENAME, *file-name*

**Informational:** Identifies current file being used as input.

**User Action:** This message always occurs in conjunction with other messages. Action depends upon associated message.

## Messages

### Text Processor Messages

**FONTDIMEN,** The following font has only *number* fontdimen parameters  
To increase the number of font parameters you must use  
\fontdimen immediately after the \font is loaded

**Warning:** Current font has too many parameters.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**FONTERROR,** Error occurred in the following font:

**Warning:** Text formatter found an error in the current font.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**FONTMEMSIZE,** Font memory size = *number* words

**Informational:** Reports the total size of text formatter's font memory.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**FONTTYPE,** Font was declared by: *string*

**Warning:** Identifies the font's type, e.g. \preloaded.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**FUNNYCS,** Missing control sequence inserted

**Warning:** Text formatter became confused while defining a control sequence.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**HPOOLSIZE,** Hyphenation string memory size = *number* bytes

**Informational:** User has exceeded total size of text formatter's hyphenation string memory.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

**IDENT,** *string*

**Informational:** Identifies the current version of the text formatter.

**User Action:** None.

# Messages

## Text Processor Messages

IGNORED, Extra *command* ignored

**Warning:** An unnecessary text formatting command was found.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

IGNOREDTEXT, All text was ignored after line *string*

**Warning:** Text formatter could not typeset any of the following lines shown.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

IGNORING, Ignored the following: *command*

**Warning:** Text formatter encountered an unexpected command and ignored it.)

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

IMPROPERCMD, Improper command found: *command*

**Warning:** command shown is not valid in the current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INCOMPLETECMD, Incomplete command found: *command*

**Warning:** Text formatter became confused while trying to scan the shown command.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INFO, *text* - on page *number*

**Informational:** A text formatting macro has generated an informational message. A problem occurred while the text formatter was processing the output. *text* provides a description of the problem. In most cases, the file will continue processing successfully and you can examine the output. The message indicates the page on which the problem occurred, and you can use this information to correct the problem

**User Action:** The following is a list of the messages you may see and the associated action that will correct the problem.

- Callout number 0 is undefined. Using 1.  
You specified a <CALLOUT> or <CO> tag as 0.

## Messages

### Text Processor Messages

- Callout number exceeds 99. Using 99.  
A <CALLOUT> or <CO> tag's value is greater than 99. POSTSCRIPT output devices cannot print numbers greater than 99.
- Error in setting note type; using default.  
A local or personal DESIGN file specified an invalid value for the \noteformattype. Refer to *VAX DOCUMENT Designing Doctypes* for value values.
- Error. Maximum number of table footnotes exceeded.  
You specified too many <FOOTNOTE> tags for a table. The largest number that is allowed is 12.
- Error pushing/popping revision bars.  
<MARK> and <ENDMARK> tags are not evenly matched or are not suppressed around heading levels in the correct sequence. If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.
- Possible problem in the output of revision bars.  
The text formatter encountered a complex situation while a revision mark (from the <MARK> tag) was active. Examine your output closely to determine if the revision mark output is correct. A revision mark may not appear or it may appear where you did not intend it to. If there is a problem, you can change your source file to decrease the range in which a revision mark is active. For example, if you have changes in two rows of a table, you might place the <MARK> and <ENDMARK> tags inside each of the two <TABLE\_ROW> tags instead of around the entire table.
- Figure is too large for a single page.  
A value specified in <FIGURE\_SPACE> or <FIGURE\_FILE> exceeds the depth of the output page. The text formatter uses only the depth of the page.  
Correct the invalid depth argument in the SDML file.
- The <FOOTREF> tag before the <FOOTNOTE> tag in the table is ignored.  
A tag <FOOTREF> tag occurs in a table for which no corresponding <FOOTNOTE> tag occurs. You must specify footnotes in tables at the beginning of the table, before any references to them occur.
- Graphics files cannot be included on this device.  
A <FIGURE\_FILE> or <ICON\_FILE> tag was encountered that specified a graphics file to be included on a device that does not support the inclusion of graphics. The tag is ignored.
- An index subentry is more than 4 levels deep; it is ignored.  
An indexing tag (<X> or <Y>) contains more than four <XSUBENTRY> tags. Only four levels of subentry are allowed.  
Correct the indexing tag by removing the excessive subentries.

## Messages

### Text Processor Messages

- Included files are not allowed to float.  
An included text formatter file in a figure is assumed to be multipage and cannot float.  
Remove the FLOAT attribute.
- Internal error while processing a table.  
The text formatter received bad input. Submit an SPR.
- Invalid table of contents entry.  
A local or personal doctype specified an invalid value for a table of contents entry. Correct the table of contents entry to use a valid code.  
If this error occurs using a standard VAX DOCUMENT design, submit an SPR.
- Monospaced example is too big to keep.  
You specified the keyword KEEP as an attribute to a <CODE\_EXAMPLE> tag or other tag that specifies a monospaced example. The number of lines in the example cannot fit on a single page of output.  
Remove the keyword KEEP from the tag and provide <VALID\_BREAK> tags to indicate acceptable page break points.
- A multipage figure is not allowed to float.  
You specified a figure with both the attributes FLOAT and MULTIPAGE. Multipage figures cannot float; the text processor outputs it at the location of its occurrence in the source file.  
Remove the FLOAT attribute.
- There is no template heading defined at this level; using level two.
- A <PAGE> tag is ignored in a floating figure.  
A <PAGE> or <FINAL\_CLEANUP>(PAGE) tag is encountered in a figure that has the default attribute FLOAT.  
Specify <FIGURE\_ATTRIBUTES>(MULTIPAGE) for all figures that are more than a page in length.
- A <PAGE> tag is ignored in a table that is not multipage.  
A <PAGE> or <FINAL\_CLEANUP>(PAGE) tag is encountered in a table that has the default attribute FLOAT.  
Specify <TABLE\_ATTRIBUTES>(MULTIPAGE) for all tables that are more than a page in length.
- Possible table paging problem.  
A table that is output on more than one page is being forced to break across pages without explicit page breaking or sufficient valid table break points.  
In most cases, the output is acceptable. The text formatter makes its best decision about breaking tables across pages when there are <TABLE\_ROW\_BREAK> tags to specify the first and last valid breaks for the table.



## Messages

### Text Processor Messages

- Revision bars cannot be output on this device.  
<REVISION>, <MARK>, and <ENDMARK> tags cannot be processed for this device. No change bars are printed.
- A <SPAN> tag at the end of a table column is ignored.  
The <SPAN> tag, which specifies that text is to span 2 or more columns in a table, must *precede* the text in the argument. For example,  
`<TABLE_ROW>(<SPAN>(3)This text sits across all three columns)`  
Move the <SPAN> tag to the beginning of the tag's argument.
- The table is too wide for the page.  
The values you specify in the <TABLE\_SETUP> tag result in a table that does not fit on the current page. Check the arguments you specified to the <TABLE\_SETUP> tag and decrease them. You are able to tell from the output how much of the table is excessive.
- A <TABLE\_HEADS> tag preceding a <TABLE\_SETUP> tag is ignored.  
You must specify table headings following the <TABLE\_SETUP> tag. Move the <TABLE\_HEADS> tag so that it precedes the <TABLE\_SETUP> tag.
- A <TABLE\_HEADS> tag within a <TABLE\_UNIT> tag is ignored.  
Do not use the <TABLE\_HEADS> tag inside <TABLE\_UNIT> tag sequences; use the <TABLE\_UNIT\_HEADS> tag instead.
- A <TABLE\_ROW> tag is encountered outside of a <TABLE> tag.  
You specify a <TABLE\_ROW> tag, but there is no <TABLE> tag. The tag is ignored. Correct the file so that a valid <TABLE> tag precedes it.
- <TABLE\_ROW\_BREAK> tags in a single-page table are ignored.  
You specify the <TABLE\_ROW\_BREAK> tag in a table that has the attribute KEEP.  
Remove the KEEP attribute from the table or remove the <TABLE\_ROW\_BREAK> tag.
- A <TABLE\_UNIT\_HEADS> tag outside of the table unit ignored.  
A <TABLE\_UNIT\_HEADS> tag is encountered before a <TABLE\_UNIT> tag to begin a table unit.  
Remove the tag or insert a <TABLE\_UNIT> tag to enclose a table unit.
- Tracing is disabled in VAX DOCUMENT.  
An instruction to accumulate tracing information occurred in a TEX file. VAX DOCUMENT does not support these instructions.
- Too many columns specified for multi-column output.  
A local or personal doctype specifies too many columns of output.  
If this error occurs using a standard doctype, submit an SPR.

## Messages

### Text Processor Messages

INITEXPAT, \patterns can only be used by Initex

**Warning:** Text formatter only recognizes \patterns when it is building format files, not during a run of VAX DOCUMENT.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INSERTED, Missing *command* inserted

**Warning:** Missing command was inserted into the input stream.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INSERTHBOX, \insert can only be added to a \vbox

**Warning:** Command \insert is not valid in the current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INSERTING, Inserting *command*

**Warning:** Text formatter attempted to resolve an unexpected situation by inserting shown command into the input stream.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INSUFEXTFON, Math formula deleted: Insufficient extension fonts

**Warning:** Math formula could not be typeset because extension fonts were inadequate or missing.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INSUFSYMFON, Math formula deleted: Insufficient symbol fonts

**Warning:** Math formula could not be typeset because symbol fonts were inadequate or missing.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INTERNALERROR, Internal error number: *error number*. Job is aborting

**Error:** Text formatter found an internal inconsistency.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

LINE, Error occurred on or around line number: *number*

**Informational:** Reports the current line of input that the text formatter is processing.

**User Action:** This message always occurs in conjunction with other messages. Action depends upon associated message.

LINETOOLONG, Line too long by *Integer remainder* points

**Informational:** The text formatter found a very lengthy line.

**User Action:** This message always occurs in conjunction with other messages. Action depends upon associated message.

LINETOOSHORT, Line too short - on page *Page number*

**Informational:** The text formatter found a line that was too short.

**User Action:** This message always occurs in conjunction with other messages. Action depends upon associated message.

MATHOVERFLOW, Arithmetic overflow

**Warning:** Result of a mathematical operation was too large.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXFONTS, Maximum number of fonts is: *number*

**Informational:** User has exceeded maximum number of fonts.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXGROUPS, Maximum number of groups is: *number*

**Informational:** User has exceeded maximum number of groups.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXHASH, Maximum number of entries in hash table is: *number*

**Informational:** User has exceeded maximum number of entries on text formatter's hash table.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

MAXHSTRINGS, Maximum number of hyphenation strings is: *number*

**Informational:** User has exceeded the maximum number of hyphenation strings.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXHYPHS, Number of hyphenation exceptions is: *number*

**Informational:** User has exceeded maximum number of hyphenations.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXINPUTS, Maximum number of simultaneous input files is: *number*

**Informational:** User has exceeded maximum number of simultaneous input files.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXNESTS, Maximum number of simultaneous semantic levels is *number*

**Informational:** User has exceeded maximum number of simultaneous semantic levels in text formatter.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXPARAMS, Maximum number of simultaneous text macro parameters is *number*

**Informational:** User has exceeded maximum number of parameters to a macro.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXSAVE, Maximum number of entries on save stack is: *number*

**Informational:** User has exceeded maximum number of entries on text formatter's save stack.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MAXSTRINGS, Maximum number of strings is: *number*

**Informational:** User has exceeded maximum number of strings.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

MEMORYSIZE, Main memory size = *number* words

**Informational:** User has exceeded the total size of the text formatter's main memory.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MESSAGE, *text* - on page *number*

**Warning:** Text formatter macro generated a warning message. The following is a list of possible messages and actions.

**User Action:** See the action associated with each of the following messages.

- Error in the window size.

An error occurred in a local or personal doctype; an invalid value was specified in a \window.

If this error occurs in a standard doctype, submit an SPR.

- Internal error in setting block indents.

Too many arguments are specified to \blockindents, or the values were not valid. This error should only occur if a local or personal DESIGN file contains an invalid value.

- Internal error in figure key definition.

Submit an SPR.

- Too many floating figures.

There are too many <FIGURE> tag sequences within a short span of text, and the default attribute "float" is in effect for all of them.

The text formatter cannot produce meaningful output when there are too many figures that are floating. When you have a large number of figures in a short span of text, you should use the <FIGURE\_ATTRIBUTES>(KEEP) tag to keep them in sequential order.

- No room for a new [item].

Internal use.

MISSINGBOOL, ' or '=' or '

**Warning:** Text formatter expected to scan one of the boolean values shown.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report. Determine which boolean value is needed and insert it in the right place.

## Messages

### Text Processor Messages

MISSINGLBRACE, Possible missing {

**Warning:** Text formatter did not see an {.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MISSINGRBRACE, Possible missing }

**Warning:** Text formatter did not see an }.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

MODE, Error occurred in *string* mode

**Informational:** Reports the current mode that the text formatter is in.

**User Action:** This message always occurs in conjunction with other messages. Action depends upon associated message.

MUERROR, Mismatched glue units. Assuming that 1mu=1pt

**Warning:** Text formatter scanned glue specifications that are incompatible.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOBOX, Expected to find \hbox or \vbox or  
\copy or \box or something like that.  
Some text may be missing from output.

**Warning:** Text formatter expected to find a box.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOEND, No \end was found

**Error:** Text formatter expected to see an \end command.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOFONTID, Could not find font identifier for the following font:

**Warning:** Font identifier for the current font is missing.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

NOINPUTPAGE, no pages in input file

**Error:** The input file named in an associated message contains no pages; therefore, there is no output to print.

**User Action:** Check SDML source file to see why no text was found.

NOINSLASTBOX, \insert255 is illegal. Changing to \insert0

**Warning:** Text formatter register \insert255 cannot be used in current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOMATHACCENT, Changed \accent to \mathaccent

**Warning:** The text formatter command \accent works differently in formulas than it does in normal text.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NONUMBER, Missing number, using zero

**Warning:** Text formatter expected to find a number.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOROOM, Exceeded memory capacity:

**Error:** Text formatter exceeded one of its memory capacities. The message immediately following indicates which quantity was exceeded.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NOUSEAFTER, After this: *command.*, the following is invalid:

**Warning:** Text formatter cannot use the command shown in the next message directly after the command shown in this message.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

NUMBERTOOBIG, Number too large, using 2147483647 instead

**Warning:** Text formatter could not use specified number because it is too big. Text formatter used 2147483647 instead, which is the largest number possible in that context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

ONEOFFOUR, Some or all of the following 4 messages apply:

**Warning:** One or more of the four messages that follow on your screen apply.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ONEOFTWO, One or both of the following messages apply:

**Warning:** One or both of the two messages that follow on your screen apply.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

ONPAGE, on page *number*

**Informational:** Identifies the current page being output.

**User Action:** This message always occurs in conjunction with other messages. Action depends upon associated message.

OUTFILENAME, *file-name*

**Informational:** Identifies the output file name.

**User Action:** None.

PAGESOUT, *number* page(s) written.

**Informational:** Reports the number of pages written to the output file.

**User Action:** None.

PAGETOOBIG, Cannot ship out huge page. More than 18 feet wide or long.

**Warning:** Text formatter became confused during typesetting of current page because page was too large.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

PAGETOOLONG, Page too long by *string.string* points

**Warning:** Text formatter could not fit the contents of the current page into boundaries of current page. Such errors are common with a line printer because it is monospaced.

**User Action:** Examine the output on the page shown to determine if there is a problem with the coding of the SDML file. You can ignore the problem if it is not serious, or you can change destinations or change the doctype, if possible.



## Messages

### Text Processor Messages

PAGETOOSHORT, Page too short - on page *Page number*

**Informational:** The text formatter had trouble composing the current page. There were not enough lines to reach the bottom. Such errors are common with a line printer because it is monospaced.

**User Action:** Examine the output on the page shown to determine if there is a problem with the coding of the SDML file. You can ignore the problem if it is not serious, or you can change destinations or change the doctype, if possible.

PATTERNEXISTS, Duplicate pattern found

**Warning:** Text formatter found a duplicate pattern.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

PATTERNMEMSIZE, Hyphenation pattern memory size = *number* words

**Informational:** User has exceeded total size of text formatter's hyphenation memory.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

POOLSIZE, String memory size = *number* bytes

**Informational:** User has exceeded total size of text formatter's string memory.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

RUNAWAY, Runaway *kind of text*

**Warning:** Text formatter became confused while scanning the shown kind of text.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

SCALED, Font was scaled: *number*

**Informational:** This is an internal error.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

SHOWCONTEXT, *text*

**Informational:** Displays the input stream that the text formatter was attempting to read when something went wrong.

**User Action:** This message always occurs in conjunction with other messages. Action depends upon associated message.

## Messages

### Text Processor Messages

SHOWTOKEN, *text*

**Informational:** Displays part of text formatter's current token list.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

TIGHTSPACING, Insufficient inter-word spacing - on page *Page number*

**Informational:** The text formatter had trouble composing a line - there was not enough white space on the line. Such errors are common with a line printer because it is monospaced.

**User Action:** Examine the output on the page shown to determine if there is a problem with the coding of the SDML file. You can ignore the problem if it is not serious, or you can change destinations or change the doctype, if possible.

TOOFEWLINES, Excess inter-line spacing - on page *Page number*

**Informational:** The text formatter had trouble composing the current page. There was too much white space between the lines. Such errors are common with a line printer because it is monospaced.

**User Action:** Examine the output on the page shown to determine if there is a problem with the coding of the SDML file. You can ignore the problem if it is not serious, or you can change destinations or change the doctype, if possible.

TOOMANYERRORS, Error limit is 30

**Error:** Text formatter found more than 30 errors.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

TOOMANYLINES, Insufficient inter-word spacing - on page *Page number*

**Informational:** The text formatter had trouble composing the current page. There was not enough white space between the lines. Such errors are common with a line printer because it is monospaced.

**User Action:** Examine the output on the page shown to determine if there is a problem with the coding of the SDML file. You can ignore the problem if it is not serious, or you can change destinations or change the doctype, if possible.

TOOMANYPARAMS, More than nine parameters were passed to a macro

**Warning:** Text formatter became confused while scanning parameters to a macro because there were too many paramters.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

## Messages

### Text Processor Messages

TRACE, *text*

**Informational:** User has turned on text formatter's tracing mechanism. Text contains information about VAX DOCUMENT's text formatter's internals.

**User Action:** None.

UNDEFINEDCS, Undefined control sequence

**Warning:** Text formatter encountered an undefined control sequence.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

UNKNOWNUNIT, Dimension unit must be: em,ex,in,pt,pc,cm,mm,dd,cc,bp,sp

**Warning:** Text formatter expected to read one of the dimensions shown.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

USINGZERO, Bad *value*, must be between *smaller number* and *larger number*, using zero instead

**Warning:** Text formatter expected to scan a value in the range shown.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

VSPLITHBOX, Attempted to \vsplit a \hbox

**Warning:** Command \vsplit is not valid in current context.

**User Action:** If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

WIDESPACING, Excess inter-word spacing - on page *Page number*

**Informational:** The text formatter had trouble composing a line; there was too much white space on the line. Such errors are common with a line printer because it is monospaced.

**User Action:** Examine the output on the page shown to determine if there is a problem with the coding of the SDML file. You can ignore the problem if it is not serious, or you can change destinations or change the doctype, if possible.

## Messages

### Device Converter Messages

---

#### B.4 Device Converter Messages

BADFILABORT, file bad—aborting run

**Error:** The device converter cannot use the font file named in an adjacent message.

**User Action:** Another file may have been given the same name. Use the DIRECTORY/FULL DCL command to see that the size and format are similar to other font files. If necessary, reinstall the font data.

BADFONT, font file bad—using blanks

**Warning:** The file that describes the font design is unusable. The program will substitute blanks for the characters in this font.

**User Action:** Use the DIRECTORY/FULL DCL command to see if the size and format of this file resembles other font files. If necessary, reinstall the font data.

BADFONTSIZE, internal font error: font  
*string* scaled size out of range.

**Error:** The device converter has detected a font size that is out of range. This may be a problem with the font file itself, or an internal device converter problem.

**User Action:** None.

BADINCDVI, bad file format or error in reading included DVI file  
*filespec*

**Error:** The device converter has encountered a read error or illegal file format in an included DVI file. See the previous message.

**User Action:** Regenerate the DVI file by running Document again.

BADINCLUDE, included file is bad

**Warning:** The program has encountered errors in processing the contents or index file requested.

**User Action:** Check that previous processing steps generated the contents or index file expected, because the included file may be an old, incompatible file, it may be another file with the same name, or it may be the correct file made unreadable because of an I/O error.

BADMETRIC, bad metric file

**Error:** The device converter has encountered errors in processing the indicated font metric file.

**User Action:** This may be another file named the same. Check the file size and format with the DIRECTORY/FULL DCL command to see if it resembles other font metric files. If necessary, reinstall the font data.

## Messages

### Device Converter Messages

BADPAPERSIZE, illegal papersize string *string*, ignored

**Warning:** An unsupported papersize string was encountered from the design file.

**User Action:** Correct the design to specify one of the accepted papersizes.

BADPOSTAMBLE, error in postamble of file  
*string*

**Error:** The device converter encountered a read or file format error in the postamble of the named file.

**User Action:** None.

BADPOSTFORMAT, illegal postamble format

**Error:** The device converter encountered a read or file format error in the postamble of the named file.

**User Action:** None.

BADPREAMBLE, error in DVI file's preamble.

**Error:** The device converter cannot accept the DVI file's preamble.

**User Action:** Regenerate the DVI file by running Document again.

BADPRIMDVI, bad file format or error reading primary DVI file:  
*filespec*

**Error:** The device converter has encountered a read error or illegal file format in the primary DVI file. See the previous message.

**User Action:** Regenerate the DVI file by running Document again.

BADPXLFILE, bad file format or error in reading PXL bitmap font file  
*filespec*

**Error:** The device converter encountered a read error or illegal file format in trying to read metrics or bitmap data for a PXL bitmap font.

**User Action:** Reinstall the font data.

BADSPECIAL, unknown DVI file special command code encountered.

**Error:** The special command code read does not match any existing defined specials.

**User Action:** None.

BOOKABORT, aborting run—book *string* not created

**Error:** The device converter received a fatal error; no DECW\$BOOK file is made.

**User Action:** Check the previous message to see what caused the fatal condition.

## Messages

### Device Converter Messages

BOOKBUILT, book was successfully built

**Informational:** The book was completed without any fatal errors.

**User Action:** None.

CANNOTCONNECT, cannot connect along a diagonal

**Warning:** While attempting to draw change bars, the program has encountered an attempt to draw a nonvertical line. Only vertical change bars are allowed.

**User Action:** Look in the macros that describe change bar positioning for errors and inconsistencies.

CANNOTCREATE, cannot create output file

**Error:** The program cannot create an output file.

**User Action:** Check that the output directory exists, that the user has permission to write in it, and that the device is available and is not full.

CANNOTOPEN, cannot open input file

**Error:** The program cannot open the input DVI file.

**User Action:** Check that the input directory exists, that the file exists or was created by earlier processing steps, that the user has permission to read the file, and that the device is available.

CANTCREATE, cannot create output file  
*filespec*

**Error:** The device converter cannot open the named file for write.

**User Action:** See next message.

CANTOPEN, cannot open file  
*filespec*

**Error:** The device converter was unable to open the named file.

**User Action:** None.

CONFIGERROR, error in configuration file

**Error:** An error has been detected in reading the file that describes PostScript fonts. An adjacent error message gives more details.

**User Action:** Check that the device is available and that the user has permission to read the file. Take the action suggested by the adjacent error message. If necessary, reinstall the font data.

DIFPAPERSIZE, papersize in design but already specified on cmd line,  
using cmd line value *string*

**Warning:** A papersize option was encountered both on the command line and in the design file. The command line value will be used.

**User Action:** Specify the desired papersize in the design file.

## Messages

### Device Converter Messages

DOCSETUP, error in generating document setup area.

**Error:** The device converter experienced a fatal error while attempting to generate the document setup area, which contains all of the font definitions needed in the document.

**User Action:** See adjacent messages.

DVIEOF, premature end-of-file in DVI file  
*filespec*

**Error:** The device converter encountered a premature end-of-file while reading the named DVI file.

**User Action:** Regenerate the DVI file by running Document again.

EXTNESTING, Extensions are being nested

**Informational:** Extensions are being nested.

**User Action:** This is not supported. Check the output carefully.

HOTSPOTERR, HOTSPOT crosses Tex page boundary:  
*string*

**Informational:** Gives the symbolic name of reference and the last chunk title that was defined.

**User Action:** Using the symbolic name and the title locate the reference being broken up by the page break and insert an <ONLINE\_CHUNK> tag before it.

EXTNESTING, Extensions are being nested

**Informational:** Extensions are being nested.

**User Action:** This is not supported. Check the output carefully.

HOTSPOTERR, HOTSPOT crosses Tex page boundary:  
*string*

**Informational:** Gives the symbolic name of reference and the last chunk title that was defined.

**User Action:** Using the symbolic name and the title locate the reference being broken up by the page break and insert an <ONLINE\_CHUNK> tag before it.

FCCANTOPEN, cannot open font configuration file  
*filespec*

**Error:** The device converter was unable to open the named file.

**User Action:** None.

FIGFILERR, Error encountered while reading:  
*filespec*

**Informational:** An error was returned when reading the figure file.

**User Action:** Check to make sure the figure is not corrupted.

## Messages

### Device Converter Messages

FIGFILOPN, cannot open figure file

**Warning:** The program cannot find the figure file named, or can find it but cannot open it.

**User Action:** Check that you have specified the right filespec, that the file exists in the path specified (or in the current default directory, if no path is specified), that you have permission to read that file, and that the file is not locked by the action of some other program running at the same time.

FIGNOTFND, cannot find figure image in figure file

**Warning:** The figure file named in an adjacent message does not contain the proper codes to identify the type of figure expected. White space appears where the figure would be placed.

**User Action:** Check the figure file.

FILEI, error in reading file  
*filespec*

**Error:** The device converter encountered an error while reading the named file.

**User Action:** None.

FILENESTING, one included file cannot include another file

**Warning:** One contents or index file may not include another.

**User Action:** Check the macros that describe contents and index files to see that they are correctly generating the filespecs. Check that the filespec generated for the contents or index file is the same as an existing file.

FILEO, error in writing file  
*filespec*

**Error:** The device converter encountered an error while writing to the named file.

**User Action:** None.

FNTCONFIG, description for font  
*string* missing or illegal in configuration file.

**Error:** The specified font is not present, or not fully described, in the file that describes PostScript fonts.

**User Action:** If necessary, reinstall the PostScript support.

FNTFILEREf, font referenced in file  
*filespec*

**Error:** The named DVI file contains a reference to the above font.

**User Action:** None.



## Messages

### Device Converter Messages

FONTCONFIG, Font description missing or illegal in configuration file

**Error:** The specified font is not present, or not fully described, in the file that describes PostScript fonts.

**User Action:** If necessary, reinstall the PostScript support.

FONTEERROR, error in font  
*string*

**Warning:** An error has occurred while processing the named font. The specific problem is identified in an adjacent message.

**User Action:** Note the name of the font. Take the action suggested by the adjacent error message.

FONTTOOBIG, too many characters in font—limit is 188

**Error:** The document uses too many characters from the named font.

**User Action:** Reinstall the font files. If the problem persists, and if you are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

FSEEK, fseek internal error, file =  
*string*

**Error:** The device converter encountered an internal error while attempting to fseek in the named file.

**User Action:** None.

IDENT, *string*

**Informational:** Reports the version of the program in use.

**User Action:** None.

IGNORING, ignoring included input file  
*filespec*

**Warning:** The program cannot process the included DVI file for the reason specified in an adjacent message.

**User Action:** Note the name of the file. Take the action suggested by the adjacent error message.

INCDVICANTOPEN, cannot open included DVI file  
*filespec*

**Warning:** The device converter was unable to open the named file.

**User Action:** None.

INCLUDED, File *filespec* was successfully included

**Informational:** The named file was successfully read and stored

**User Action:** None.

## Messages

### Device Converter Messages

INCLUDING, including input file:  
*filespec*

**Informational:** Reports that the program is processing a DVI file other than the file named on the command line. The file is typically a contents or index file.

**User Action:** None.

INITMEM, ran out of memory while initializing device converter

**Error:** This memory error occurred very early in the device conversion.

**User Action:** None.

INPUTBAD, bad input file—aborting run

**Error:** The program has encountered errors in the named input file and cannot continue.

**User Action:** Check the messages generated by earlier processing steps to see if a complete DVI file was created.

INPUTFILE, input file is:  
*filespec*

**Informational:** Reports the name of the primary DVI file, that is, the file named on the command line.

**User Action:** None.

INTERNALERROR, internal error

**Error:** A problem internal to the device converter prevents it from continuing.

**User Action:** This sometimes represents a document so large or complex that it exceeds internal buffer sizes. Sometimes you can work around this by running contents and index processing as separate jobs, rather than including them in place. You may be able to process the document in two or more pieces by using the `STARTING_PAGE` and `ENDING_PAGE` or `NUMBER_OF_PAGES` parameters to the device converter. If you receive this message and are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

INVALIDDATE, Invalid date specification. Use the dd-mm-yyyy format

**Error:** The converter has found an invalid date specification.

**User Action:** use the dd-mm-yyyy format to specify the date

INVALIDOPTN, invalid option

**Error:** This DVI special command contains an option not supported by this device converter, or the program cannot find an option at all.

**User Action:** Check to see if you have requested a figure file, contents file, index file, change bars, or other function which is not supported for this destination; check for errors in the macros that generate this special command; check if the special has all required arguments

## Messages

### Device Converter Messages

INVALIDPOINT, invalid point number *number*

**Warning:** An error has occurred in the macros used to describe change bars. Specifically, an attempt has been made to locate a point off the page, or the point is incorrectly described.

**User Action:** Look in the macros that describe change bar positioning for errors and inconsistencies.

IN\_CANTCLOSE, cannot close input file  
*filespec*

**Warning:** Error in closing the file. Processing continues.

**User Action:** None.

NOCONTENTS, no contents file found - verify that  
1 - The command line has/CONTENTS specified  
2 - The document has a CONTENTS\_FILE tag.

**Error:** No contents file was processed, which causes a fatal error in bookbuilding for the Bookreader.

**User Action:** Make sure there is a contents file and that it is included.

NOFILABORT, cannot open file—aborting run

**Error:** The device converter cannot open the named file.

**User Action:** Check to see that the device is available, that the file exists, and that the user has privileges to access that file.

NOFONT, cannot open font file—using blanks

**Warning:** The device converter cannot open the named font file.

**User Action:** Check that the device exists and is available, and that the user has permission to read the file.

NOFONTDEF, no font defined for font select  
*string*

**Error:** none.

**User Action:** None.

NOINCLUDE, cannot open included file

**Warning:** The device converter cannot open the contents or index file named in the adjacent message. Either the pathname and filename are incorrect, or the program cannot gain access because of file protection or locking by another process. Processing will continue without this file.

**User Action:** Check that the named file exists and that the user has access to it.

## Messages

### Device Converter Messages

NOINPUTPAGE, no pages in input file

**Error:** The input file named in an adjacent message contains no pages; therefore, there will be no output to print.

**User Action:** Check to see if errors reported in earlier processes caused there to be no pages generated.

NOMETRIC, cannot open metric file

**Error:** The font metric file named in an adjacent message does not exist, or the user does not have access to it because of file protection or problems with the device.

**User Action:** Check that the file exists and is available, and that the user has access to it. Check that the user is specifying a DVI input file appropriate for the destination requested.

NOSIXELIMAGE, cannot find sixel image in figure file

**Warning:** The figure file named in an adjacent message does not contain the command sequence that identifies a sixel image. White space appears where the figure is expected.

**User Action:** Check the figure file. For LN03 use, it must be a sixel file.

NOSUBSTITUTE, no adequate alternate font found

**Error:** The text formatter has requested a down-loaded font for which the device converter cannot find a reasonable substitute.

**User Action:** Check the document design for the fonts used. Check that the corresponding font files exist at the sizes desired.

NOSUCHUNIT, no such unit of measure

**Warning:** A DVI special command used to position change bars or describe graphics files contains an unknown unit of measure, or no unit.

**User Action:** Check the macros that describe change bar positioning or figure file characteristics for errors.

OFFPAGE, cannot set text outside page boundaries

**Warning:** An attempt has been made to set text outside the boundaries of the physical page specified.

**User Action:** Check the messages from the text formatter for lines too long, that you specified a destination that can print a page as large as required by your design. Also, check to see if you specified a horizontal or vertical offset that will push your design off the page.

ONPAGE, on page [*string*]

**Informational:** Reports the number of the output page on which the condition reported in the associated messages occurred.

**User Action:** Note this page number if you need to refer to the output.

## Messages

### Device Converter Messages

OUTPUTFAIL, error writing to output file

**Error:** The program encountered an error in writing to the output file. This usually means that the output disk is full, the user's disk quota is exceeded, or the output disk has become unusable.

**User Action:** Check that the output device is usable. If the disk is full or the user's disk quota is used up, delete unwanted files and run the device converter again.

OUT\_CANTCLOSE, cannot close output file  
*filespec*

**Warning:** Error in closing the file. Processing continues.

**User Action:** None.

PAGECOORDLOST, PAGE COORDINATION LOST *string*

**Error:** The converter has found a page coordination problem.

**User Action:** Make sure that two contents files are not being included and that there are no undefined symbols reported.

PAGENOTFOUND, starting page not found

**Error:** The user has specified a starting page number, but that page does not exist in the input file.

**User Action:** Check the starting page specified and the input file specified.

PAGESOUT, *number* page(s) written to file:  
*filespec*

**Informational:** This reports the number of pages written to the output file named.

**User Action:** No direct action needed; however, if the number is zero or less than expected, check other messages for errors or examine the commands, filespecs, options, and input files for problems.

PAGETOOCOMPLEX, cannot hold all font data—using blanks

**Warning:** The amount of font data required to describe all the characters on this page will not fit in the LN03 font memory, so the characters in the current font will be represented as blanks.

**User Action:** If using a design that allows you to change text\_size, choose a smaller size, or use only one kind of emphasis on the page, if possible, or move portions of the text to adjacent pages.

PASS1INCFNAMS, error on first pass through primary DVI file.

**Error:** The device converter experienced a fatal error on its first pass through the primary DVI file in which it was attempting to generate a list of included filenames.

**User Action:** See adjacent messages.

## Messages

### Device Converter Messages

PFCANTOPEN, cannot open included ps plotfile  
*filespec*

**Warning:** The device converter was unable to open the named file.

**User Action:** None.

PLOTFILEFAIL, cannot open figure file

**Warning:** The program cannot find the figure file named, or can find it but cannot open it.

**User Action:** Check that you have specified the right filespec, that the file exists in the path specified (or current default directory, if no path is specified), that you have permission to read that file, and that the file is not locked by the action of some other program running at the same time.

PRIMDVICANTOPEN, cannot open primary DVI file  
*filespec*

**Error:** The device converter was unable to open the named file.

**User Action:** None.

PROLOGERROR, error in prolog file

**Error:** The device converter cannot read its file that contains the PostScript prolog for output.

**User Action:** Check that the device is available, that the file exists, and that the user has privileges to access the file. If necessary, reinstall the PostScript support.

READFCMEM, ran out of memory while reading the font configuration file

**Error:** This memory error occurred early in the device conversion.

**User Action:** None.

SHELFBUILT, shelf was successfully built

**Informational:** The shelf was completed without any fatal errors.

**User Action:** None.

SPECIALERROR, error in \special  
*string*

**Warning:** This lists the text of a DVI special command as received by the device converter. This command is generated by macros, but may contain some user-specified information, such as a filespec or size. The specific problem associated with this special is described in an adjacent message.

**User Action:** See the adjacent message for a description of the problem. If necessary, check this message for a filespec or size that may have been incorrectly entered in a tag describing a figure file, contents file, or index file.

## Messages

### Device Converter Messages

SPECTOOLONG, \special too long—will try to process anyway

**Warning:** This DVI special command exceeds the maximum length of 1000 characters. The program will truncate the command and try to process it.

**User Action:** Check the macros that generate this special command; they may be appending too many spaces or extraneous text following the command.

TITLETOOLONG, Title has been truncated to 256 characters

**Informational:** The chunk title is limited to 256 characters

**User Action:** Use shorter titles that will fit in the window.

TOLONG, device & directory spec too long cannot exceed 48 characters

**Error:** the complete file spec is too long.

**User Action:** Move the files to the directory where the device and directory specs total length is 48 characters or less.

TOOMANYDVI, too many included input files

**Warning:** Too many included contents or index files have been requested.

**User Action:** Remove excess <CONTENTS\_FILE> and <INDEX\_FILE> tags from the input file.

TOOMANYFONTS, too many fonts—limit is 100

**Error:** The document contains too many fonts.

**User Action:** Change the document design so it does not require so many fonts, or use SDML tags that do not require so many fonts, or break the document into separate documents, or process the contents or index separately.

TOOMANYGLYPHS, too many glyphs used in all fonts combined

**Error:** The document contains too many characters in too many fonts. The LN03 cannot hold this many characters.

**User Action:** Break the device conversion of this document into two or more jobs, specifying the starting page and ending page or number of pages. End one job before the page on which this error was reported.

UNDEFSYM, *string*

**Error:** A reference to an undefined symbol was detected.

**User Action:** Check that symbol names in references are legitimate.

## Messages

### Device Converter Messages

UNRECOGPTN, unrecognized option

**Warning:** This DVI special command contains an option not supported by this device converter, or the program cannot find an option at all.

**User Action:** Check to see if you have requested a figure file, contents file, index file, change bars, or other function that is not supported for this destination. Check for errors in the macros that generate this special command.

VMBADFREE, Internal device converter error, failure to deallocate virtual memory

**Error:** The device converter has failed while trying to deallocate virtual memory.

**User Action:** Review any related informational messages, correct the problem, and reexecute the DOCUMENT command. If you are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.

VMOVRFLOW, Internal device converter error, failure to allocate additional virtual memory

**Error:** The device converter has failed while trying to allocate additional virtual memory.

**User Action:** Review any related informational messages, correct the problem, and reexecute the DOCUMENT command. This error may be caused by the user having too small of a pagefile quota (PGFLQUO). If so, increase this value and reexecute the DOCUMENT command. If you are under a service contract with Digital, call your customer service center. Otherwise, submit a Software Performance Report.



---

## B.5 Index Facility Messages

BADATTR, illegal tag argument attribute list syntax in tag  
*string*.

**Error:** A tag argument attribute list in the INX file contains illegal syntax or the INX file is corrupted.

**User Action:** Try reprocessing to get a new INX file. If the problem persists, contact Digital.

BADENTRYINCLUDE, warning: master, NOMASTER and BOTH on same  
entry for entry  
*string*  
using BOTH.

**Warning:** The indexer has encountered an entry that contains MASTER, NOMASTER and BOTH entry inclusion attributes. It will convert this entry to be a BOTH entry.

**User Action:** Edit the SDML file and remove the NOMASTER attribute from the entry.

BADINXCHAR, Warning: illegal char/s after  
*string*,  
attempting to continue.

**Warning:** The indexer has encountered one or more characters in between tags and tag arguments. The INX file may be corrupted. The indexer will ignore these characters and attempt to continue processing.

**User Action:** Try reprocessing to get a new INX file. If the problem persists, contact Digital.

BADTAG, unrecognized tag  
*string*

**Error:** The indexer has encountered an unrecognized tag in reading the INX file. The INX file may be corrupted.

**User Action:** Try reprocessing to get a new INX file. If the problem persists, contact Digital.

BEGINENTRY, warning: begin and end on same entry for entry  
*string*  
converting to a single page-number entry.

**Warning:** The indexer has encountered an entry that contains both a begin and end page-range attribute. It will convert this entry to be a single page entry.

**User Action:** Edit the SDML file and separate the begin-end page-range entries.

## Messages

### Index Facility Messages

BEGINONLY, begin page-range without end page-range for entry *string*, treating as single entry.

**Warning:** The indexer has encountered a begin page-range without a corresponding end page-range. It will treat the begin page-range as a single page number reference.

**User Action:** Edit the SDML file and add an end page-range.

CANTFREE, internal error in freeing virtual memory

**Warning:** The indexer encountered an internal error in attempting to free some virtual memory.

**User Action:** Contact Digital.

CREATED, *filespec* created

**Success:** The indicated output file has been created.

**User Action:** None.

CREATED, *filespec* created

**Success:** The indicated output file has been created.

**User Action:** None.

ENDONLY, end page-range without begin page-range for entry *string*, treating as single entry.

**Warning:** The indexer has encountered an end page-range without a corresponding begin page-range. It will treat the end page-range as a single page number reference.

**User Action:** Edit the SDML file and add a begin page-range.

FILEI, error in reading file  
*filespec*

**Error:** The indexer encountered an error while reading the specified file. This message is usually accompanied by a VAX RMS message indicating the reason for the failure.

**User Action:** Take corrective action based on the associated message.

FILEO, error in writing file  
*filespec*

**Error:** The indexer encountered an error while writing to the specified file. This message is usually accompanied by a VAX RMS message indicating the reason for the failure.

**User Action:** Take corrective action based on the associated message.

## Messages

### Index Facility Messages

FOLIOTOOLONG, warning: folio string too long, ignoring. inx tag is  
*string*

**Warning:** The indexer has attempted to generate a folio string that exceeds the length of its internal folio buffer. It will ignore the INX tag that caused this.

**User Action:** Contact Digital.

IDENT, *string*

**Informational:** Indicates the version of the INDEX utility. This message is produced by specifying the /IDENTIFICATION qualifier.

**User Action:** None.

IGNOREDUPREF, ignoring duplicate reference in tag  
*string*

**Warning:** The indexer has encountered a duplicate entry tag on the same page and will ignore it.

**User Action:** Edit the SDML file and remove the duplicate entry.

IGNOREINPGRNG, page number references encountered in between page-  
range begin and end for entry  
*string*, ignoring.

**Warning:** The indexer has encountered one or more single page number references in between a begin-end page-range sequence. It will ignore these in between references.

**User Action:** Edit the SDML file and remove references in between the begin-end sequence.

IGNOREINX\_LISTLINE, ignoring INX\_LIST file line.

**Warning:** The above INX\_LIST file line will be ignored.

**User Action:** Edit the INX\_LIST file and correct the erroneous line.

INX\_LIST, error on line *string/filespec* of *string*

**Warning:** An error was detected in the specified line of the named INX\_LIST file.

**User Action:** Edit the INX\_LIST file and correct the erroneous line.

IN\_CLOSE, Error in closing input file *filespec*

**Warning:**

**User Action:**

LANGTRUNC, Warning: tag argument too long.<\ >argument truncated to  
*string*.

**Warning:** The indexer has encountered a <LANGUAGE> tag argument that is too long, and is truncating it to the specified string.

**User Action:** Edit the SDML file and shorten the <LANGUAGE> tag argument.

## Messages

### Index Facility Messages

LANG\_IGNORED, The <LANGUAGE> tag is not at the beginning of the file and will be ignored.

**Warning:** The <LANGUAGE> tag was encountered in the INX file after the first index entry and will be ignored.

**User Action:** Reposition the <LANGUAGE> tag to the start of the INX file.

LANG\_NAME, unsupported language name in tag  
*string/string*, ignoring, using *string*

**Warning:** The language specified is not supported.

**User Action:** Modify the <LANGUAGE> tag in the INX file to specify Danish, English, Finnish, French, German, Italian, Norwegian, Portuguese, Spanish, Swedish, or some other supported language. NCS must have support for the specified language.

MASTERBOTH, warning: master and BOTH on same entry for entry  
*string*  
using BOTH.

**Warning:** The indexer has encountered an entry that contains both a MASTER and BOTH entry inclusion attribute. It will convert this entry to be a BOTH entry.

**User Action:** Edit the SDML file and remove the NOMASTER attribute from the entry.

MASTERNOMASTER, warning: master and NOMASTER on same entry for entry  
*string*  
using master.

**Warning:** The indexer has encountered an entry that contains both a MASTER and NOMASTER entry inclusion attribute. It will convert this entry to be a MASTER entry.

**User Action:** Edit the SDML file and remove the NOMASTER attribute from the entry.

NCSCONVERT, error in NCS conversion function.

**Error:** An internal NCS error has occurred in the indexer.

**User Action:** Contact Digital.

NCSGETCF, error in getting NCS conversion function id.

**Error:** An internal NCS error has occurred in the indexer.

**User Action:** Contact Digital.

NOCLOSEQUOTE, no closing quote.

**Warning:** The above INX\_LIST file line is missing a closing quote.

**User Action:** Edit the INX\_LIST file and correct the erroneous line.

## Messages

### Index Facility Messages

NOFNAMES, no filenames were found in  
*filespec*

**Error:** The above INX\_LIST file contains no filenames.

**User Action:** Edit the INX\_LIST file and insert one or more filenames.

NOITEMLIST, index command line information not found on item list

**Error:** The indexer could not find the command line information on its internal item list.

**User Action:** Contact Digital.

NOMASTERBOTH, warning: NOMASTER and BOTH on same entry for  
entry  
*string*  
using BOTH.

**Warning:** The indexer has encountered an entry that contains both a NOMASTER and BOTH entry inclusion attribute. It will convert this entry to be a BOTH entry.

**User Action:** Edit the SDML file and remove the NOMASTER attribute from the entry.

NOPAGENUMARG, no argument for page number tag, ignoring.

**Warning:** A page number tag is missing its argument.

**User Action:** Try reprocessing to get a new INX file. If the problem persists, contact Digital.

NOTAGARG, tag argument expected after tag  
*string*,  
but not found.

**Error:** A tag in the INX file is missing its argument.

**User Action:** Try reprocessing to get a new INX file. If the problem persists, contact Digital.

NULFILENAME, no file name found.

**Warning:** Each non-comment non-empty line in the INX\_LIST file must contain a file name.

**User Action:** Edit the INX\_LIST file and correct the erroneous line.

OPENIN, could not open *filespec* for input

**Error:** The input file does not exist, or you do not have access to the file.

**User Action:** Check the file specification for errors and check to be sure that the file is not read protected.

OPENOUT, could not open *filespec* for output

**Error:** The output file cannot be opened. This message is usually accompanied by a VAX RMS message indicating the reason for the failure.

**User Action:** Take corrective action based on the associated message.

## Messages

### Index Facility Messages

OUT\_CLOSE, Error in closing output file *filespec*

**Warning:** None.

**User Action:** None.

READING, reading file *filespec*

**Informational:** The indicated file is being read.

**User Action:** None.

SAMEPGBEGEND, warning: a begin-end page-range on same page for entry *string* converting to a single page-number entry.

**Warning:** The indexer has encountered a begin page-range entry and an end page-range entry that fall on the same page. It will convert the begin page-range entry to a single page entry and ignore the end page-range entry.

**User Action:** Edit the SDML file and remove the offending end page-range entry.

TAGARGTOOLONG, indexer tag argument internal buffer limit exceeded for tag *string*.

**Warning:** This is an indexer internal error. The buffer limit is exceeded for reading a tag argument.

**User Action:** Edit the SDML file and shorten the lengthy tag argument.

TAGSTRTOOLONG, indexer tag string internal buffer limit exceeded for tag *string*.

**Error:** This is an indexer internal error. The buffer limit is exceeded for reading a tag. The INX file may be corrupted.

**User Action:** Try reprocessing to get a new INX file. If the problem persists, contact Digital.

TOODEEP, maximum subindex depth exceeded

**Warning:** An index entry containing ten or more levels of subindexing was detected. The subindexing was ignored.

**User Action:** Generate index entries with fewer levels of subindexing.

TOOMANYFNAMES, too many filenames were found in *filespec*  
One or more will be lost.

**Error:** The above INX\_LIST file contains too many filenames for master index generation. The maximum number of INX filenames is 500.

**User Action:** Edit the INX\_LIST file and reduce the number of filenames to 500 or fewer, or split the file and run two separate jobs.

## Messages

### Index Facility Messages

UNKNOWNEQSTR, unknown string precedes "=" in tag argument attribute list in tag *string*, attempting to continue.

**Warning:** The tag argument attribute list in the INX file contains an unknown option string preceding the "=". Execution will attempt to continue.

**User Action:** Try reprocessing to get a new INX file. If the problem persists, contact Digital.

UNKNOWNKWD, unknown keyword in tag arg attribute list in tag *string*, <\ >attempting to continue.

**Warning:** The tag argument attribute list in the INX file contains an unknown keyword. Execution will attempt to continue.

**User Action:** Try reprocessing to get a new INX file. If the problem persists, contact Digital.

USEFILEBOOKNAME, using file name *string* to form book name.

**Warning:** The above INX\_LIST file line will use the file name as the book name.

**User Action:** Edit the INX\_LIST file and correct the erroneous line.

VMOVRFLOW, insufficient virtual memory

**Error:** The indexer does not have sufficient virtual memory to generate the index.

**User Action:** Either reduce the size of the index or increase the page file quota parameter (PGFLQUO) and reprocess your file.

WRITING, writing file *filespec*

**Informational:** The indicated file is being written to.

**User Action:** None.





---

# Index

---

## A

---

<ABSTRACT> • 1–2

description of • 1–2

related tags

<FRONT\_MATTER>

<PART\_PAGE>

<TITLE\_PAGE>

<ACCENT> • 1–3

description of • 1–3

related tags

<MCS>

<PARENCHAR>

using acute, grave, or umlaut • 1–3

Acute accents • 1–3

Aligning

numbers in a table • 1–8

numbers in text, a list, or a table • 1–5

text in a list • 1–4

text on the right-hand margin of a page • 1–218

Aligning text on the page • A–6

<ALIGN\_AFTER> • 1–4

description of • 1–4

related tags

<ALIGN\_CHAR>

<ALIGN\_NUMBER>

<LINE>(INDENT)

<LIST>(STACKED)

<ALIGN\_CHAR> • 1–5

description of • 1–5 to 1–7

related tags

<ALIGN\_AFTER>

<ALIGN\_NUMBER>

<TABLE\_ROW>

<ALIGN\_NUMBER> • 1–8

description of • 1–8 to 1–9

related tags

<ALIGN\_CHAR>

<ALIGN\_NUMBER>

<TABLE\_ROW>

AMPERSAND

in math expressions • 1–166

<AMPERSAND> • 1–10

description of • 1–10

related tags

<BACKSLASH>

<CPAREN>

<OPAREN>

<VBAR>

Appendix

changing default appendix letter • 1–224

<APPENDIX> • 1–11

description of • 1–11 to 1–12

related tags

<ENDAPPENDIX>

<SET\_APPENDIX\_LETTER>

<ATOP>

in math expressions • 1–166

Attributes

of examples • 1–71

of figures • 1–87

of tables • 1–242

---

## B

---

<BACKSLASH> • 1–13

description of • 1–13

in math expressions • 1–166

related tags

<AMPERSAND>

<CPAREN>

<OPAREN>

<VBAR>

BAR\_CHAR

in math expressions • 1–165

Batch processing of files • A–4

/BATCH qualifier • A–4

Bold type

in text • 1–60

of a keyword in text • 1–139

Bookbuilding

creating a profile • 1–207

creating elements • 1–56

generating an index • 1–133, A–9

generating a table of contents • 1–34, A–5

## Index

- <BOX> • 1–14
  - description of • 1–14
- Breaking a page • 1–197
  - within a monospaced example • 1–282
- Breaks
  - in columns, lines, or pages • 1–98

---

## C

---

- <CAL>
  - in math expressions • 1–166
- Calligraphic letters • 1–166
- <CALLOUT> • 1–15
  - description of • 1–15
  - related tags
    - <CALLOUTS>
    - <CALLOUT\_REF>
    - <CO>
    - <LIST>(CALLOUT)
- Callout numbers
  - labelling in an example • 1–15, 1–17, 1–24
  - labelling in text • 1–16
- <CALLOUTS> • 1–17
  - description of • 1–17 to 1–18
  - related tags
    - <CALLOUT>
    - <CALLOUT\_REF>
    - <CO>
    - <LIST>(CALLOUT)
- <CALLOUT\_REF> • 1–16
  - description of • 1–16
  - related tags
    - <CALLOUT>
    - <CALLOUTS>
    - <CO>
- Capital letters
  - small point size • 1–60
  - See also <UPPERCASE>
- Captions
  - in examples • 1–71
  - in figures • 1–87
  - in tables • 1–242
- <CASES>
  - in math expressions • 1–166
- Caution
  - labelling text as • 1–192
- <CDOTS>
  - in math expressions • 1–166
- Centering a heading • 1–22
- <CENTER\_LINE> • 1–19
  - description of • 1–19 to 1–20
  - related tags
    - <LINE>
    - <RIGHT\_LINE>
- Change bars • 1–215
- <CHAPTER> • 1–21
  - description of • 1–21
  - related tags
    - <REFERENCE>
    - <SET\_CHAPTER\_NUMBER>
- Chapters
  - changing the default chapter number • 1–225
- <CHHEAD> • 1–22
  - description of • 1–22
  - related tags
    - <CENTER\_LINE>
    - <HEAD>
    - <HEADn>
    - <SUBHEADn>
- <CHECK\_FOR\_INCLUSION> • 1–23
  - description of • 1–23
- <CHOOSE>
  - in math expressions • 1–166
- <CO> • 1–24
  - description of • 1–24
  - related tags
    - <CALLOUT>
    - <CALLOUTS>
    - <CALLOUT\_REF>
    - <LIST>(CALLOUT)
- <CODE\_EXAMPLE> • 1–25
  - description of • 1–25 to 1–28
  - related tags
    - <INTERACTIVE>
    - <VALID\_BREAK>
- Column breaks • 1–98
- Commands
  - coding command/parameter pairs • 1–78, 1–108
  - summary of • A–1
- <COMMENT> • 1–29
  - description of • 1–29 to 1–30
  - related tag
    - <ENDCOMMENT>
- <CONDITION> • 1–31
  - description of • 1–31 to 1–33
  - related tags
    - <ENDCONDITON>

<CONDITION>  
 related tags (cont'd)  
   <SET\_CONDITION>  
 /CONDITION qualifier • A-5  
 Conditions  
   for files • A-5  
 /CONTENTS qualifier • A-5  
 <CONTENTS\_FILE> • 1-34  
   description of • 1-34 to 1-35  
   related tags  
     <FRONT\_MATTER>  
     <PROFILE>  
 <COPYRIGHT\_DATE> • 1-36  
   description of • 1-36  
   related tags  
     <COPYRIGHT\_PAGE>  
     <FRONT\_MATTER>  
 <COPYRIGHT\_PAGE> • 1-37  
   description of • 1-37  
   related tags  
     <COPYRIGHT\_DATE>  
     <ENDCOPYRIGHT\_PAGE>  
     <FRONT\_MATTER>  
     <PRINT\_DATE>  
 <CP> • 1-38  
   description of • 1-38  
   related tag  
     <P>  
 <CPAREN> • 1-39  
   description of • 1-39  
   related tags  
     <AMPERSAND>  
     <BACKSLASH>  
     <OPAREN>  
     <VBAR>

---

## D

---

<DATE> • 1-40  
   description of • 1-40 to 1-41  
 <DDOTS>  
   in math expressions • 1-166  
 <DEFINE\_BOOK\_NAME> • 1-42  
   description of • 1-42 to 1-43  
   related tags  
     <DEFINE\_SYMBOL>  
     <REFERENCE>

<DEFINE\_SYMBOL> • 1-44  
   description of • 1-44 to 1-45  
   related tags  
     <DEFINE\_BOOK\_NAME>  
     <DELAYED>  
     <REFERENCE>  
 <DEFINITION\_LIST> • 1-46  
   description of • 1-46 to 1-47  
   related tags  
     <DEFINITION\_LIST\_HEAD>  
     <DEFLIST\_DEF>  
     <DEFLIST\_ITEM>  
     <ENDDEFINITION\_LIST>  
 <DEFINITION\_LIST\_HEAD> • 1-48  
   description of • 1-48  
   related tags  
     <DEFINITION\_LIST>  
     <DEFLIST\_DEF>  
     <DEFLIST\_ITEM>  
 <DEFLIST\_DEF> • 1-49  
   description of • 1-49  
   related tags  
     <DEFINITION\_LIST>  
     <DEFINITION\_LIST\_HEAD>  
     <DEFLIST\_ITEM>  
 <DEFLIST\_ITEM> • 1-50  
   description of • 1-50  
   related tags  
     <DEFINITION\_LIST>  
     <DEFINITION\_LIST\_HEAD>  
     <DEFLIST\_DEF>  
 Degree character • 1-186  
 <DELAYED> • 1-51  
   description of • 1-51 to 1-52  
   related tags  
     <DEFINE\_BOOK\_NAME>  
     <DEFINE\_SYMBOL>  
     <ENDDelayed>

Destination  
   See DOCUMENT command, destination  
   parameter  
 /DEVICE\_CONVERTER qualifier • A-6  
 Diagnostics  
   for LSE • A-8  
 /DIAGNOSTICS qualifier • A-8  
 Dialog  
   between system and user • 1-135

# Index

<DIVIDED\_BY>  
in math expressions • 1–166  
Dividing a document into parts • 1–200, 1–202  
Doctype  
on the command line  
See DOCUMENT command, doctype parameter  
<DOCTYPE> • 1–53  
description of • 1–53  
Document  
partial  
printing • A–7 to A–8  
DOCUMENT command  
for the symbol generator  
parameters • A–17  
parameters • A–2 to A–4  
destination • A–3  
doctype • A–3  
input-file • A–2  
qualifiers • A–4 to A–16  
/BATCH • A–4  
/CONDITION • A–5  
/CONTENTS • A–5  
/DEVICE\_CONVERTER • A–6  
/DIAGNOSTICS • A–8  
/ELEMENT • A–8  
/INCLUDE • A–8  
/INDEX • A–9  
/KEEP • A–10  
/LIST • A–11  
/LOG • A–11  
/MAP • A–12  
/MASTER\_INDEX • A–12  
/OUTPUT • A–12  
/PRINT • A–13  
/PROFILE • A–14  
/SYMBOLS • A–15  
/TAG\_TRANSLATOR • A–15  
/TEXT\_FORMATTER • A–15  
summary of • A–1 to A–16  
syntax of • A–2  
DOCUMENT symbol generator • A–17  
DOCUMENT symbol generator command  
parameters  
input-file • A–17  
qualifiers • A–17  
/LIST • A–18  
/PREFIX • A–18

Document type  
on the command line  
See DOCUMENT command, doctype parameter  
<DOTS>  
in math expressions • 1–166  
dot\_char  
in math expressions • 1–165  
<DOT\_TIMES>  
in math expressions • 1–166  
<DOUBLE\_QUOTE> • 1–54  
description of • 1–54 to 1–55  
related tags  
    <PARENDCCHAR>  
    <QUOTE>  
    <SINGLE\_QUOTE>  
Duplex printing • A–6

---

## E

---

<ELEMENT> • 1–56  
description of • 1–56 to 1–57  
related tags  
    <INCLUDES\_FILE>  
    <PROFILE>  
/ELEMENT qualifier • A–8  
<ELLIPSIS> • 1–58  
description of • 1–58 to 1–59  
related tag  
    <HELLIPSIS>  
Embedded tables  
breaking across pages • 1–189  
<EMPHASIS> • 1–60  
description of • 1–60 to 1–61  
related tags  
    <QUOTE>  
    <UNDERLINE>  
Emphasizing  
a keyword in text • 1–139  
a new term in text • 1–191  
a variable in text • 1–285  
<ENDABSTRACT> • 1–2  
<ENDALIGN\_CHAR> • 1–5  
<ENDAPPENDIX> • 1–11  
<ENDCALLOUTS> • 1–17  
<ENDCHECK\_FOR\_INCLUSION> • 1–23  
<ENDCODE\_EXAMPLE> • 1–25

<ENDCOMMENT> • 1–29  
 <ENDCONDITION> • 1–31  
 <ENDCOPYRIGHT\_PAGE> • 1–37, 1–62  
     description of • 1–62  
 <ENDDDEFINITION\_LIST> • 1–46  
 <ENDDDELAYED> • 1–51  
 <ENDEXAMPLE> • 1–66  
 <ENDFIGURE> • 1–81  
 <ENDFORMAT> • 1–108  
 <ENDFRONT\_MATTER> • 1–111  
 <ENDGLOSSARY> • 1–114  
 <ENDICON> • 1–124  
 <ENDINTERACTIVE> • 1–135  
 <ENDLINE\_ART> • 1–147  
 <ENDLIST> • 1–150  
 <ENDLITERAL> • 1–157  
 <ENDMARK> • 1–160  
 <ENDMATH> • 1–163  
 <ENDNOTE> • 1–192  
 <ENDPART\_PAGE> • 1–63, 1–202  
     description of • 1–63  
 <ENDPREFACE> • 1–203  
 <ENDPROFILE> • 1–207  
 <ENDSAMPLE\_TEXT> • 1–223  
 <ENDTABLE> • 1–240  
 <ENDTABLE\_KEY> • 1–250  
 <ENDTITLE\_PAGE> • 1–64, 1–269  
     description of • 1–64  
 <ENDUNDERLINE> • 1–272  
 <ENDUPDATE\_RANGE> • 1–274  
 <EQUALIGN>  
     in math expressions • 1–166  
 <EQUALS>  
     in math expressions • 1–166  
 Equations in math • 1–163  
 Error messages • B–1  
 <EXAMPLE> • 1–65  
     description of • 1–65 to 1–70  
     related tags  
         <CODE\_EXAMPLE>  
         <ENDEXAMPLE>  
         <ENDINTERACTIVE>  
         <EXAMPLE\_ATTRIBUTES>  
         <EXAMPLE\_FILE>  
         <EXAMPLE\_SPACE>  
         <INTERACTIVE>  
         <REFERENCE>  
         <S>  
         <SAMPLE\_TEXT>  
         <U>

<EXAMPLE>  
     related tags (cont'd)  
         <VALID\_BREAK>  
 Examples  
     formal • 1–65  
     informal • 1–25  
     using callouts in • 1–15, 1–17, 1–24  
 <EXAMPLE\_ATTRIBUTES> • 1–71  
     description of • 1–71 to 1–74  
     related tags  
         <EXAMPLE>  
         <EXAMPLE\_FILE>  
         <EXAMPLE\_SPACE>  
         <REFERENCE>  
 <EXAMPLE\_FILE> • 1–75  
     description of • 1–75  
     related tags  
         <EXAMPLE>  
         <EXAMPLE\_ATTRIBUTES>  
         <EXAMPLE\_SPACE>  
         <REFERENCE>  
 <EXAMPLE\_SPACE> • 1–76  
     description of • 1–76 to 1–77  
     related tags  
         <EXAMPLE>  
         <EXAMPLE\_ATTRIBUTES>  
         <EXAMPLE\_FILE>  
         <REFERENCE>

---

## F

---

<FCMD> • 1–78  
     description of • 1–78 to 1–80  
     related tags  
         <FORMAT>  
         <FPARM>  
         <FPARMS>  
 <FIGURE> • 1–81  
     description of • 1–81 to 1–86  
     related tags  
         <ENDFIGURE>  
         <FIGURE\_ATTRIBUTES>  
         <FIGURE\_FILE>  
         <FIGURE\_SPACE>  
         <ICON>  
         <ICON\_FILE>  
         <LINE\_ART>

# Index

<FIGURE>

related tags (cont'd)

<REFERENCE>

## Figures

captions of • 1–87

changing the default for vertical blank space •  
1–228

coding multipage figures • 1–87

coding wide figures • 1–87

controlling attributes of • 1–87

controlling formality of • 1–87

creating • 1–89

creating line art

See <LINE\_ART>

keeping a figure with its preceding text • 1–87

leaving space for • 1–93

margins in • 1–87

numbers in the caption • 1–87

page breaks in • 1–87

width of • 1–87

<FIGURE\_ATTRIBUTES> • 1–87

description of • 1–87 to 1–88

related tags

<FIGURE>

<FIGURE\_FILE>

<FIGURE\_SPACE>

<FIGURE\_FILE> • 1–89

description of • 1–91 to 1–92

related tags

<FIGURE>

<FIGURE\_ATTRIBUTES>

<ICON>

<ICON\_FILE>

<LINE\_ART>

<REFERENCE>

<SET\_FIGURE\_FILE\_SPACING\_DEFAULT>

<FIGURE\_SPACE> • 1–93

description of • 1–93 to 1–95

related tags

<FIGURE>

<FIGURE\_ATTRIBUTES>

<LINE\_ART>

<REFERENCE>

## Files

bookbuilding with elements • 1–56

intermediate

keeping during processing • A–10

listing

producing during processing • A–11

producing for the symbol generator • A–18

## Files (cont'd)

log

producing during processing • A–11

page parameters

changing • A–6

printing after processing • A–13

processing

assigning new output name • A–12

processing in batch mode • A–4

producing map files during processing • A–12

setting conditions • 1–31, 1–226, A–5

## File specifications

in tag arguments • 1–96

<FILE\_SPEC> • 1–96

description of • 1–96 to 1–97

related tags

<ELEMENT>

<EXAMPLE\_FILE>

<FIGURE\_FILE>

<ICON\_FILE>

<INCLUDE>

<INCLUDES\_FILE>

<TABLE\_FILE>

<FINAL\_CLEANUP> • 1–98

description of • 1–99 to 1–100

related tags

<COLUMN>, in ARTICLE and REPORT.TWOCOL  
doctypes

<LINE>

<PAGE>

<FOOTNOTE> • 1–101

description of • 1–101 to 1–103

related tags

<FOOTNOTE\_TEXT>

<FOOTREF>

<FOOTNOTE\_TEXT> • 1–104

description of • 1–104 to 1–105

related tags

<FOOTNOTE>

<FOOTREF>

<FOOTREF> • 1–106

description of • 1–106 to 1–107

related tags

<FOOTNOTE>

<FOOTNOTE\_TEXT>

Formal examples • 1–65

<FORMAT> • 1–108

description of • 1–108

related tags

<FORMAT>  
 related tags (cont'd)  
   <FCMD>  
   <FPARM>  
   <FPARMS>

Formatting  
 for special results • 1–98

<FPARM> • 1–109  
 description of • 1–109  
 related tags  
   <FCMD>  
   <FORMAT>  
   <FPARMS>

<FPARMS> • 1–110  
 description of • 1–110  
 related tags  
   <FCMD>  
   <FORMAT>  
   <FPARM>

<FRONT\_MATTER> • 1–111  
 description of • 1–111 to 1–112  
 related tags  
   <ABSTRACT>  
   <CONTENTS\_FILE>  
   <COPYRIGHT\_DATE>  
   <COPYRIGHT\_PAGE>  
   <DEFINE\_BOOK\_NAME>  
   <ENDFRONT\_MATTER>  
   <ORDER\_NUMBER>  
   <PREFACE>  
   <PREFACE\_SECTION>  
   <PRINT\_DATE>  
   <REVISION\_INFO>  
   <TITLE>  
   <TITLE\_PAGE>

<FUNC>  
 in math expressions • 1–166

---

## G

---

<GDEF> • 1–113  
 description of • 1–113  
 related tags  
   <GLOSSARY>  
   <GREF>  
   <TERM>

Generating symbols • A–17

<GLOSSARY> • 1–114  
 description of • 1–114 to 1–115  
 related tags  
   <ENDGLOSSARY>  
   <GDEF>  
   <GTERM>

Glossary entries  
 creating a cross-reference to a term • 1–116  
 creating entries anywhere in your document • 1–117  
 creating entries in your document • 1–113

Graphics files  
 including in SDML files • 1–81, 1–124

Grave accents • 1–3

<GREF> • 1–116  
 description of • 1–116  
 related tags  
   <GDEF>  
   <GLOSSARY>  
   <GTERM>

<GROUP>  
 in math expressions • 1–166

<GTERM> • 1–117  
 description of • 1–117  
 related tags  
   <GDEF>  
   <GLOSSARY>  
   <GREF>

---

## H

---

hat  
 in math expressions • 1–165

hat\_char  
 in math expressions • 1–165

<HEAD> • 1–118  
 description of • 1–118  
 related tags  
   <CENTER\_LINE> • 1–118  
   <CHEAD> • 1–118  
   <HEAD<sub>n</sub>> • 1–118  
   <SUBHEAD<sub>n</sub>> • 1–118

Headings  
 centered on a line • 1–22

<HEAD<sub>n</sub>> • 1–119  
 description of • 1–119 to 1–121  
 related tags  
   <CHEAD>

# Index

<HEAD<sub>n</sub>>

related tags (cont'd)

<HEAD>

<SUBHEAD<sub>n</sub>>

<HELLIPSIS> • 1-122

description of • 1-122

related tag

<ELLIPSIS>

<HYPHENATE> • 1-123

description of • 1-123

related tags

<FINAL\_CLEANUP>(LINE\_BREAK)

<KEEP>

<LINE>

Hyphenation

for specific line breaks • 1-123

---

<ICON> • 1-124

description of • 1-124 to 1-126

related tags

<ENDICON>

<FIGURE\_FILE>

<ICON\_FILE>

<ICON\_TEXT>

<ICON\_FILE> • 1-127

description of • 1-127 to 1-128

related tags

<FIGURE\_FILE>

<ICON>

<ICON\_TEXT>

<ICON\_TEXT> • 1-129

description of • 1-129

related tags

<ICON>

<ICON\_FILE>

Identification number at the bottom of a copyright page • 1-62

<INCLUDE> • 1-130

description of • 1-130

/INCLUDE qualifier • A-8

<INCLUDES\_FILE> • 1-131

description of • 1-131 to 1-132

related tags

<ELEMENT>

<INCLUDE>

<PROFILE>

Including a file in the current input file • 1-130

Indenting a line • 1-144

Index

effect on the index when using a <REVISION> tag in a file • 1-215

generating • 1-133, A-9

Indexing

creating main entries • 1-287

creating "See" and "See also" entries • 1-298

creating subentries • 1-294, 1-296

using <XAPPEND> in the <X> tag • 1-287

using <XSORT> in the <X> tag • 1-288

using <XSORT> in the <Y> tag • 1-298

/INDEX qualifier • A-9

<INDEX\_FILE> • 1-133

description of • 1-133 to 1-134

related tags

<CONTENTS\_FILE>

<PROFILE>

Informal examples • 1-25

Informational messages

sending to a terminal or a log file • 1-278

Input file

See SDML file

<INTEGRAL>

in math expressions • 1-167

<INTEGRAL\_LIMITS>

in math expressions • 1-167

<INTEGRAL\_NOLIMITS>

in math expressions • 1-167

<INTERACTIVE> • 1-135

description of • 1-135 to 1-137

related tags

<ENDINTERACTIVE>

<S>

<U>

<VALID\_BREAK>

Intermediate files

keeping during processing • A-10

Italicizing

a keyword in text • 1-139

a new term in text • 1-191

a variable in text • 1-285

Italics

in text • 1-60



---

## K

---

- <KEEP> • 1-138
    - description of • 1-138
    - related tags
      - <DEFINE\_SYMBOL>
      - <EMPHASIS>
      - <HYPHENATE>
  - /KEEP qualifier • A-10
  - <KEYWORD> • 1-139
    - description of • 1-139
    - related tags
      - <NEWTERM>
      - <SPECIAL\_CHAR>
      - <VARIABLE>
- 
- ## L
- 
- Labelling parts of your document • 1-200, 1-202
  - <LANGUAGE> • 1-140
    - description of • 1-140 to 1-142
  - <LBAR>
    - in math expressions • 1-167
  - <LBRACE>
    - in math expressions • 1-167
  - <LBRACKET>
    - in math expressions • 1-167
  - <LCEIL>
    - in math expressions • 1-167
  - <LE> • 1-143
    - description of • 1-143
    - related tag
      - <LIST>
  - <LFLOOR>
    - in math expressions • 1-167
  - <LINE> • 1-144
    - description of • 1-144 to 1-146
    - related tags
      - <CENTER\_LINE>
      - <FINAL\_CLEANUP>
      - <RIGHT\_LINE>
  - Line breaks • 1-98
  - <LINE\_ART> • 1-147
    - description of • 1-147 to 1-148

- <LINE\_ART> (cont'd)
    - related tags
      - <ENDLINE\_ART>
      - <FIGURE>
      - <VALID\_BREAK>
  - <LIST> • 1-149
    - description of • 1-149 to 1-156
    - related tags
      - <ALIGN\_AFTER>
      - <ALIGN\_CHAR>
      - <ENDLIST>
      - <LE>
  - Listing file
    - producing during processing • A-11
    - producing for the symbol generator • A-18
  - /LIST qualifier • A-11
    - for the symbol generator • A-18
  - <LITERAL> • 1-157
    - description of • 1-157 to 1-158
    - related tag
      - <ENDLITERAL>
  - Log files
    - producing during processing • A-11
  - /LOG qualifier • A-11
  - <LOWERCASE> • 1-159
    - description of • 1-159
    - related tag
      - <UPPERCASE>
  - <LPAREN>
    - in math expressions • 1-167
  - LSE
    - running diagnostics • A-8
- 

## M

---

- Map files
  - producing during processing • A-12
- /MAP qualifier • A-12
- Margins
  - in figures • 1-87
- <MARK> • 1-160
  - description of • 1-160 to 1-162
  - related tags
    - <ENDMARK>
    - <REVISION>
    - <UPDATE\_RANGE>
- Marking text
  - using change bars • 1-215

# Index

/MASTER\_INDEX qualifier • A-12

<MATH> • 1-163

description of • 1-163 to 1-177

related tags

<CODE\_EXAMPLE>

<ENDMATH>

<MATH\_CHAR>

<REFERENCE>

tags for math expressions

<ATOP>

<BACKSLASH>

<BAR\_CHAR>

<CAL>

<CASES>

<CDOTS>

<CHOOSE>

<DDOTS>

<DIVIDED\_BY>

<DOTS>

<DOT\_CHAR>

<DOT\_TIMES>

<EQUALIGN>

<EQUALS>

<FUNC>

<GROUP>

<HAT>

<HAT\_CHAR>

<INTEGRAL>

<INTEGRAL\_LIMITS>

<INTEGRAL\_NOLIMITS>

<LBAR>

<LBRACE>

<LBRACKET>

<LCEIL>

<LFLOOR>

<LPAREN>

<MATRIX>

<MINUS>

<MOD>

<OVER>

<OVERLINE>

<PI>

<PLUS>

<PMOD>

<PROD>

<PROD\_LIMITS>

<PROD\_NOLIMITS>

<RBAR>

<RBACE>

<MATH>

tags for math expressions (cont'd)

<RBRACKET>

<RCEIL>

<RFLOOR>

<RPAREN>

<SP>

<SQRT>

<SUBSCRIPT>

<SUM>

<SUM\_LIMITS>

<SUM\_NOLIMITS>

<SUPERSCRIPIT>

<TEXT>

<TILDE>

<TILDE\_CHAR>

<TIMES>

<TO>

<UNDERLINE>

<VARIABLE>

<VDOTS>

<VECTOR>

<X\_TIMES>

Math expressions

aligning expressions • 1-172

delimited expressions • 1-173

tags for math functions • 1-170

tags valid in math expressions • 1-166

using binary operations • 1-164

using mathematical functions • 1-170

using matrixes and cases • 1-174

using operators with and without limits • 1-169

using parentheses • 1-164

using variable names • 1-165

<MATH\_CHAR> • 1-178

description of • 1-178 to 1-185

related tags

<MATH>

<MCS>

<SPECIAL\_CHAR>

<MATRIX>

in math expressions • 1-167

<MCS> • 1-186

description of • 1-186 to 1-188

related tags

<MATH\_CHAR>

<SPECIAL\_CHAR>

Messages

sending informational messages to terminal or log

file • 1-278

Messages (cont'd)  
 sending warning messages to terminal or log file • 1–280  
 summary of • B–1 to B–101  
 <MINUS>  
 in math expressions • 1–167  
 <MOD>  
 in math expressions • 1–167  
 Multinational Character Set • 1–186

---

## N

---

Nested tables  
 breaking across pages • 1–189  
 <NESTED\_TABLE\_BREAK> • 1–189  
 description of • 1–189 to 1–190  
 related tags  
 <TABLE>  
 <TABLE\_FILE>  
 Nesting a tag  
 in an outer tag's argument • 1–51  
 <NEWTERM> • 1–191  
 description of • 1–191  
 related tags  
 <KEYWORD>  
 <VARIABLE>  
 <NOTE> • 1–192  
 description of • 1–192  
 related tag  
 <ENDNOTE>  
 Numbers  
 aligning in a table • 1–8  
 aligning in text, a list, or a table • 1–5  
 in figures • 1–87

---

## O

---

<OPAREN> • 1–193  
 description of • 1–193  
 related tags  
 <AMPERSAND>  
 <BACKSLASH>  
 <CPAREN>  
 <VBAR>  
 <ORDER\_NUMBER> • 1–194  
 description of • 1–194

<ORDER\_NUMBER> (cont'd)  
 related tag  
 <TITLE\_PAGE>  
 Output devices  
 See DOCUMENT command, destination parameter  
 Output file name  
 changing • A–12  
 /OUTPUT qualifier • A–12  
 <OVER>  
 in math expressions • 1–167  
 OVERLINE  
 in math expressions • 1–166

---

## P

---

<P> • 1–195  
 description of • 1–195 to 1–196  
 related tags  
 <CENTER\_LINE>  
 <CP>  
 <LINE>  
 <RIGHT\_LINE>  
 <PAGE> • 1–197  
 description of • 1–197 to 1–198  
 related tag  
 <FINAL\_CLEANUP>(PAGE\_BREAK)  
 Page breaks • 1–197  
 for final cleanup • 1–98  
 in figures • 1–87  
 within monospaced examples • 1–282  
 Page parameters  
 changing • A–6  
 Paragraphs • 1–195  
 continuing when interrupted • 1–38  
 Parameters  
 of the DOCUMENT command for the symbol generator • A–17  
 Parameters of the DOCUMENT command  
 See DOCUMENT command, parameters  
 <PARENDCHAR> • 1–199  
 description of • 1–199  
 Parentheses  
 in tag arguments • 1–39, 1–193  
 spacing for a character within • 1–199  
 using in math • 1–164  
 <PART> • 1–200  
 description of • 1–200 to 1–201

# Index

<PART> (cont'd)  
  related tag  
    <PART\_PAGE>  
Partial document  
  printing • A-7 to A-8  
<PART\_PAGE> • 1-202  
  description of • 1-202  
  related tags  
    <ABSTRACT>  
    <ENDPART\_PAGE>  
    <PART>  
    <RUNNING\_TITLE>  
    <TITLE>  
<P>  
  in math expressions • 1-167  
<PLUS>  
  in math expressions • 1-167  
<PMOD>  
  in math expressions • 1-167  
<PREFACE> • 1-203  
  description of • 1-203 to 1-204  
  related tags  
    <ENDPREFACE>  
    <FRONT\_MATTER>  
    <PREFACE\_SECTION>  
<PREFACE\_SECTION> • 1-205  
  description of • 1-205  
  related tags  
    <FRONT\_MATTER>  
    <PREFACE>  
/PREFIX qualifier • A-18  
Printing  
  blank pages • A-6  
  duplex pages • A-6  
  sections of a document • A-7 to A-8  
Printing files  
  double-spacing  
    after processing • A-13  
/PRINT qualifier • A-13  
<PRINT\_DATE> • 1-206  
  description of • 1-206  
  related tags  
    <COPYRIGHT\_PAGE>  
    <FRONT\_MATTER>  
Processing files  
  assigning a new output name • A-12  
  batch mode • A-4  
  checking files for inclusion • 1-23  
  generating a table of contents • 1-34

Processing files (cont'd)  
  index • A-9  
  indexing • 1-133  
  intermediate files  
    keeping • A-10  
  listing file  
    producing • A-11  
    producing for the symbol generator • A-18  
  log file  
    producing • A-11  
  map file  
    producing • A-12  
  page parameters  
    changing • A-6  
  printing  
    double-spacing • A-13  
    setting conditions • 1-31, 1-226  
    setting conditions for • A-5  
    table of contents • A-5  
    using elements for bookbuilding • 1-56  
<PROD>  
  in math expressions • 1-167  
<PROD\_LIMITS>  
  in math expressions • 1-168  
<PROD\_NOLIMITS>  
  in math expressions • 1-168  
Profile  
  generating an index • A-9  
  generating an index from • 1-133  
  generating a table of contents • A-5  
  generating a table of contents from • 1-34  
<PROFILE> • 1-207  
  description of • 1-207 to 1-208  
  related tags  
    <ELEMENT>  
    <ENDPROFILE>  
    <INCLUDES\_FILE>  
/PROFILE qualifier • A-14

---

## Q

---

Qualifiers  
  See DOCUMENT command, qualifiers  
  See DOCUMENT symbol generator command, qualifiers  
<QUOTE> • 1-209  
  description of • 1-209 to 1-210  
  related tags  
    <DOUBLE\_QUOTE>

<QUOTE>  
 related tags (cont'd)  
 <SINGLE\_QUOTE>

---

## R

---

<RBAR>  
 in math expressions • 1-168

<RBRACE>  
 in math expressions • 1-168

<RBACKET>  
 in math expressions • 1-168

<RCEIL>  
 in math expressions • 1-168

<REFERENCE> • 1-211  
 description of • 1-211 to 1-214

Renumbering after a part page • 1-63

Revising pages • 1-274

Revising text  
 using change bars • 1-215

<REVISION> • 1-215  
 description of • 1-215 to 1-216  
 related tags  
   <MARK>  
   <UPDATE\_RANGE>

<REVISION\_INFO> • 1-217  
 description of • 1-217  
 related tags  
   <FRONT\_MATTER>  
   <TITLE\_PAGE>

<RFLOOR>  
 in math expressions • 1-168

Right-justified text • 1-218

<RIGHT\_LINE> • 1-218  
 description of • 1-218 to 1-219  
 related tags  
   <CENTER\_LINE>  
   <LINE>

<RPAREN>  
 in math expressions • 1-168

<RULE> • 1-220  
 description of • 1-220  
 related tags  
   <SPAN>  
   <TABLE>  
   <TABLE\_HEADS>  
   <TABLE\_ROW>  
   <TABLE\_UNIT\_HEADS>

---

## S

---

<S> • 1-221  
 description of • 1-221 to 1-222  
 related tags  
   <EXAMPLE>  
   <INTERACTIVE>  
   <U>

<SAMPLE\_TEXT> • 1-223  
 description of • 1-223  
 related tags  
   <CENTER\_LINE>  
   <ENDSAMPLE\_TEXT>  
   <P>

Scientific characters • 1-178

SDML file  
 specifying to DOCUMENT command  
   See DOCUMENT command, input-file parameter  
 specifying to DOCUMENT symbol generator command  
   See DOCUMENT symbol generator command, input-file parameter

Sections of a document  
 printing • A-7 to A-8

<SET\_APPENDIX\_LETTER> • 1-224  
 description of • 1-224  
 related tags  
   <APPENDIX>  
   <SET\_APPENDIX\_NUMBER> in the MILSPEC doctype  
   <SET\_CHAPTER\_NUMBER>

<SET\_CHAPTER\_NUMBER> • 1-225  
 description of • 1-225  
 related tags  
   <CHAPTER>  
   <SET\_APPENDIX\_LETTER>

<SET\_CONDITION> • 1-226  
 description of • 1-226 to 1-227  
 related tag  
   <CONDITION>

<SET\_FIGURE\_FILE\_SPACING\_DEFAULT> • 1-228  
 description of • 1-228 to 1-229  
 related tags  
   <FIGURE>  
   <FIGURE\_FILE>

<SET\_TABLE\_ROW\_BREAK\_DEFAULT> • 1-230  
 description of • 1-230 to 1-231

# Index

<SET\_TABLE\_ROW\_BREAK\_DEFAULT> (cont'd)

related tag

<TABLE\_ROW\_BREAK>

<SINGLE\_QUOTE> • 1-232

description of • 1-232

related tags

<DOUBLE\_QUOTE>

<PARENDCHAR>

<QUOTE>

<SP>

in math expressions • 1-168

Spacing within parentheses • 1-199

<SPAN> • 1-233

description of • 1-233 to 1-235

related tags

<RULE>

<TABLE\_HEADS>

<TABLE\_UNIT\_HEADS>

Special formatting • 1-98

<SPECIAL\_CHAR> • 1-236

description of • 1-236 to 1-237

related tags

<MATH\_CHAR>

<MCS>

<SQRT>

in math expressions • 1-168

<SUBHEAD<sub>n</sub>> • 1-238

description of • 1-238

related tags

<CHEAD>

<HEAD<sub>n</sub>>

<SUBSCRIPT>

in math expressions • 1-168

<SUM>

in math expressions • 1-168

Summary of commands • A-1

<SUM\_LIMITS>

in math expressions • 1-168

<SUM\_NOLIMITS>

in math expressions • 1-168

<SUPERSCRIP<sub>T</sub>>

in math expressions • 1-169

Symbol definitions file • 1-44

Symbol definitions file for books • 1-42

Symbol generator command

parameters • A-17

Symbol names

default output • 1-211

referring to in other files • 1-44

Symbol names for books

referring to in other files • 1-42

Symbols

generating by the symbol generator • A-17

/SYMBOLS qualifier • A-15

Syntax

See DOCUMENT command, syntax of

System/user dialog • 1-135

---

## T

---

<TABLE> • 1-239

description of • 1-239 to 1-241

related tags

<ALIGN\_CHAR>

<ALIGN\_NUMBER>

<ENDTABLE>

<NESTED\_TABLE\_BREAK>

<REFERENCE>

<RULE>

<SPAN>

<TABLE\_ATTRIBUTES>

<TABLE\_FILE>

<TABLE\_HEADS>

<TABLE\_KEY>

<TABLE\_KEYREF>

<TABLE\_ROW>

<TABLE\_ROW\_BREAK>

<TABLE\_SETUP>

<TABLE\_SPACE>

<TABLE\_UNIT>

<TABLE\_UNIT\_HEADS>

<VALID\_TABLE\_ROW\_BREAK>

Table of contents

effect on the table of contents when using a

<REVISION> tag in a file • 1-215

generating • 1-34, A-5

Tables

attributes of • 1-242

breaking a table across pages • 1-256

breaking nested tables across pages • 1-189

creating table headings • 1-248

creating table rows • 1-254

grouping table rows • 1-263

headings for table units • 1-265

including a separate table file in an SDML file •  
1-246

leaving space for a table in a document • 1-261

## Tables (cont'd)

- outputting a rule following table headings • 1–220
- overriding the default page break • 1–230
- page breaks within table rows • 1–284
- setting up table columns • 1–259
- text exceeding column width • 1–233
- using a table key or legend • 1–250
- <TABLE\_ATTRIBUTES> • 1–242
  - description of • 1–242 to 1–245
  - related tags
    - <TABLE>
    - <TABLE\_SETUP>
- <TABLE\_FILE> • 1–246
  - description of • 1–246 to 1–247
  - related tag
    - <TABLE>
- <TABLE\_HEADS> • 1–248
  - description of • 1–248 to 1–249
  - related tags
    - <RULE>
    - <SPAN>
    - <TABLE>
- <TABLE\_KEY> • 1–250
  - description of • 1–250 to 1–252
  - related tags
    - <ENDTABLE\_KEY>
    - <FOOTNOTE>
    - <TABLE>
    - <TABLE\_KEYREF>
    - <TABLE\_SETUP>
- <TABLE\_KEYREF> • 1–253
  - description of • 1–253
  - related tags
    - <TABLE>
    - <TABLE\_KEY>
- <TABLE\_ROW> • 1–254
  - description of • 1–254 to 1–255
  - related tags
    - <ALIGN\_CHAR>
    - <ALIGN\_NUMBER>
    - <RULE>
    - <SPAN>
    - <TABLE>
    - <TABLE\_ROW\_BREAK>
    - <VALID\_TABLE\_ROW\_BREAK>
- <TABLE\_ROW\_BREAK> • 1–256
  - description of • 1–256 to 1–258
  - related tags
    - <NESTED\_TABLE\_BREAK>

- <TABLE\_ROW\_BREAK>
  - related tags (cont'd)
    - <TABLE>
    - <TABLE\_UNIT>
    - <VALID\_TABLE\_ROW\_BREAK>
- <TABLE\_SETUP> • 1–259
  - description of • 1–259 to 1–260
  - related tags
    - <TABLE>
    - <TABLE\_ATTRIBUTES>
- <TABLE\_SPACE> • 1–261
  - description of • 1–261 to 1–262
  - related tags
    - <TABLE>
    - <TABLE\_ROW>
- <TABLE\_UNIT> • 1–263
  - description of • 1–263 to 1–264
  - related tags
    - <TABLE>
    - <TABLE\_ROW\_BREAK>
    - <TABLE\_UNIT\_HEADS>
- <TABLE\_UNIT\_HEADS> • 1–265
  - description of • 1–265 to 1–266
  - related tags
    - <RULE>
    - <SPAN>
    - <TABLE>
    - <TABLE\_UNIT>

## Tag

- nested in an outer tag's argument • 1–51
- <TAG> • 1–267
  - description of • 1–267
  - related tag
    - <LITERAL>

## /TAG\_TRANSLATOR qualifier • A–15

## Text

- aligning in a list • 1–4
- aligning on the right-hand margin of a page • 1–218
- breaking a page of • 1–197
- centering in margin • 1–19
- coding literally • 1–157
- in double quotes • 1–54
- in lowercase • 1–159
- in quotes • 1–209
- in single quotes • 1–232
- in uppercase • 1–277
- keeping on the same line of output • 1–138
- labelling a note, caution, or warning • 1–192
- marking for exclusion • 1–29

## Index

### Text (cont'd)

- placing on a new line • 1-144
- special formatting of • 1-223
- underlining • 1-272
- using callouts in • 1-16

#### <TEXT>

- in math expressions • 1-169

/TEXT\_FORMATTER qualifier • A-15

#### tilde

- in math expressions • 1-166

#### tilde\_char

- in math expressions • 1-166

#### <TIMES>

- in math expressions • 1-169

#### <TITLE> • 1-268

- description of • 1-268

related tags

<PART\_PAGE>

<TITLE\_PAGE>

### Title page

- revision information • 1-217
- text at bottom of the page • 1-64

#### <TITLE\_PAGE> • 1-269

- description of • 1-269

related tags

<ENDTITLE\_PAGE>

<FRONT\_MATTER>

#### <TO>

- in math expressions • 1-169

### Trademark symbol • 1-236

- in footnotes • 1-101

---

## U

---

#### <U> • 1-270

- description of • 1-270 to 1-271

related tags

<EXAMPLE>

<INTERACTIVE>

<S>

### Umlaut accents • 1-3

### UNDERLINE

- in math expressions • 1-166

#### <UNDERLINE> • 1-272

- description of • 1-272 to 1-273

related tag

<ENDUNDERLINE>

#### <UPDATE\_RANGE> • 1-274

- description of • 1-274 to 1-276

related tags

<ENDUPDATE\_RANGE>

<MARK>

<REVISION>

### Updating pages • 1-274

### Updating text

- using change bars • 1-215

#### <UPPERCASE> • 1-277

- description of • 1-277

related tag

<LOWERCASE>

#### <USER\_I\_MESSAGE> • 1-278

- description of • 1-278 to 1-279

related tag

<USER\_W\_MESSAGE>

#### <USER\_W\_MESSAGE> • 1-280

- description of • 1-280 to 1-281

related tag

<USER\_I\_MESSAGE>

---

## V

---

#### <VALID\_BREAK> • 1-282

- description of • 1-282 to 1-283

related tags

<CODE\_EXAMPLE>

<DISPLAY>

<INTERACTIVE>

<LINE\_ART>

#### <VALID\_TABLE\_ROW\_BREAK> • 1-284

- description of • 1-284

related tags

<NESTED\_TABLE\_BREAK>

<TABLE>

#### <VARIABLE> • 1-285

- description of • 1-285
- in math expressions • 1-169

related tag

<KEYWORD>

### VAX DOCUMENT

See DOCUMENT command

#### <VBAR> • 1-286

- description of • 1-286

related tags

<AMPERSAND>



<VBAR>  
 related tags (cont'd)  
   <BACKSLASH>  
   <CPAREN>  
   <OPAREN>  
 <VDOTS>  
   in math expressions • 1–169  
 VECTOR  
   in math expressions • 1–166  
 <VECTOR>  
   in math expressions • 1–169  
 Vertical bar  
   using in an argument • 1–286

---

## W

---

Warning  
   labelling text as • 1–192  
 Warning messages  
   sending to a terminal or a log file • 1–280  
 Width  
   in figures • 1–87

---

## X

---

<X> • 1–287  
 description of • 1–287 to 1–293  
 related tags  
   <XS>  
   <XSUBENTRY>  
   <Y>

<XS> • 1–294  
 description of • 1–294 to 1–295  
 related tags  
   <X>  
   <XSUBENTRY>  
   <Y>  
 <XSUBENTRY> • 1–296  
 description of • 1–296 to 1–297  
 related tags  
   <X>  
   <XS>  
   <Y>  
 <X\_TIMES>  
   in math expressions • 1–169

---

## Y

---

<Y> • 1–298  
 description of • 1–298 to 1–299  
 related tags  
   <X>  
   <XS>  
   <XSUBENTRY>



## How to Order Additional Documentation

---

### Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

### Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

### Telephone and Direct Mail Orders

<b>Your Location</b>	<b>Call</b>	<b>Contact</b>
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	_____	Local Digital subsidiary or approved distributor
Internal <sup>1</sup>	_____	USASSB Order Processing - WMO/E15 or U.S. Area Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

---

<sup>1</sup>For internal orders, you must submit an Internal Software Order Form (EN-01740-07).



# Reader's Comments

VAX DOCUMENT  
Using Global Tags  
AA-JT84C-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

I would like to see more/less \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

What I like best about this manual is \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

What I like least about this manual is \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

I am using **Version** \_\_\_\_\_ of the software this manual describes.

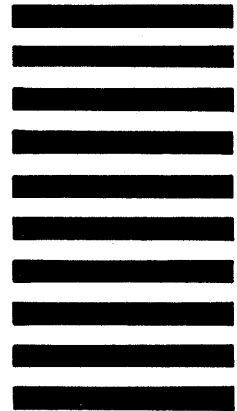
Name/Title \_\_\_\_\_ Dept. \_\_\_\_\_  
Company \_\_\_\_\_ Date \_\_\_\_\_  
Mailing Address \_\_\_\_\_  
Phone \_\_\_\_\_

--- Do Not Tear - Fold Here and Tape ---

**digital**™



No Postage  
Necessary  
if Mailed  
in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION  
Corporate User Publications—Spit Brook  
ZK01-3/J35  
110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



--- Do Not Tear - Fold Here ---

# Reader's Comments

VAX DOCUMENT  
Using Global Tags  
AA-JT84C-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

<b>I rate this manual's:</b>	<b>Excellent</b>	<b>Good</b>	<b>Fair</b>	<b>Poor</b>
Accuracy (software works as manual says)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness (enough information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clarity (easy to understand)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization (structure of subject matter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Figures (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Examples (useful)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Index (ability to find topic)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Page layout (easy to find information)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

I would like to see more/less \_\_\_\_\_

What I like best about this manual is \_\_\_\_\_

What I like least about this manual is \_\_\_\_\_

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

I am using **Version** \_\_\_\_\_ of the software this manual describes.

Name/Title \_\_\_\_\_ Dept. \_\_\_\_\_

Company \_\_\_\_\_ Date \_\_\_\_\_

Mailing Address \_\_\_\_\_

Phone \_\_\_\_\_

--- Do Not Tear - Fold Here and Tape ---

**digital**™



No Postage  
Necessary  
if Mailed  
in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION  
Corporate User Publications—Spit Brook  
ZK01-3/J35  
110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



--- Do Not Tear - Fold Here ---