

Document Composition Facility  
SCRIPT/VS Language Reference  
Release 4.0





Document Composition Facility  
SCRIPT/VS Language Reference  
Release 4.0



**Note!**

Before using this information and the product it supports, be sure to read the general information in "Notices" on page xiv.

**Eighth Edition (September 1998)**

This edition contains information from and makes obsolete the *Document Composition Facility: SCRIPT/VS Language Reference*, SH35-0070-06. Changes or editions to the text are indicated by vertical bars in the left margin.

This edition applies to Release 4.0 of the *Document Composition Facility: SCRIPT/VS Language Reference*, Program Number 5748-XX9, and applies to all subsequent releases of this product until otherwise indicated in new releases or technical newsletters.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer, your IBM Marketing Representative, or your IBM Printing Systems Company Marketing Representative.

The IBM Printing Systems Company welcomes your comments. For your convenience, a form for reader's comments is provided at the back of this publication. If the form has been removed, you may send your comments to the following address:

INFORMATION DEVELOPMENT  
THE IBM PRINTING SYSTEMS COMPANY  
DEPARTMENT 588 BUILDING 003G  
PO BOX 1900  
BOULDER CO 80301-9191

If you prefer to send your comments electronically, use one of the following methods:

- Internet: [printpubs@us.ibm.com](mailto:printpubs@us.ibm.com)
- Fax: 1-800-524-1519

**Internet**

Visit our home page at <http://www.printers.ibm.com>

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

However, the following copyright notice protects this documentation under the Copyright laws of the United States and other countries which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

© Copyright International Business Machines Corporation 1984, 1998. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Summary of Amendments</b>	xiii
Release 4.0	xiii
Notices	xiv
Trademarks	xiv
 <b>Chapter 1. Introduction to the Language Reference</b>	 3
How to Use This Reference	3
Where to Find More Information	4
How This Reference Is Organized	4
 <b>Chapter 2. Using the SCRIPT Command</b>	 7
Online Help for the SCRIPT Command	7
SCRIPT/VS in the Interactive Environment	7
Naming the Primary Input File	8
Characteristics of an Input File	9
SCRIPT/VS in the DLF Batch Environment	11
SCRIPT/VS in the TSO Batch Environment	11
Environment Restrictions and Considerations	11
MVS/TSO Data Set Naming Conventions	12
 <b>Chapter 3. SCRIPT Command Options</b>	 17
Command Option Syntax	17
Default Options	17
Command Options and TSO DDNAMES	18
Mutually Exclusive Options	18
Logical Output Devices and Destinations	19
Printer Classes	22
SCRIPT Command Option Descriptions	24
BIND: Shift the Page to the Right	24
CHARS: Specify Fonts	24
CONTINUE: Continue Processing after a Nonsevere Error Occurs	25
CTF: Prepare Output in STAIRS/VS Condensed Text Format	25
DDUT: Enable SCRIPT/VS Utility File Redefinition	25
DEST: Name a Remote Output Station or Queue	25
DEVICE: Specify a Logical Output Device	26
SCRIPT/VS Logical Line Devices	27
SCRIPT/VS Logical PostScript Devices	28
SCRIPT/VS Logical Page Devices	28
FILE: Name a Disk File for Output	30
FONTLIB: Specify a Font Library	33
FPASSES: Specify Number of Formatting Passes	35
INDEX: Enable Back of Book Index	35
LIB: Specify Symbol and Macro Libraries	35
MESSAGE: Control Message Printing	36
NOCONT: Ending Processing after an Error Occurs	37
NODDUT: Disable SCRIPT/VS Utility File Redefinition	37
NOPROF: Suppress the Profile	37
NOSEGLIB: Ignore the Segment Library	37
NOSPIE: Prevent Entering SPIE Exit Routines	38
NOSORCDD: Specifying the Primary Input File as a Data Set Name	38

NOWAIT: Prevent Prompting for Paper Adjustment	38
NUMBER: Print the File Name and Line Number	38
OPTIONS: Name a File That Contains Options	38
PAGE: Selectively Print Pages	39
PRINT: Produce Printer Output	40
PROFILE: Specify a Profile	40
PSOUT: Specify ASCII or EBCDIC PostScript Output	41
QUIET: Suppress the Formatter's Identifier Message	42
SEARCH: Specify a Library	42
SEGLIB: Specify a Segment Library	43
SEPMASTR: Specify Output Separation Masters	44
SORCDD: Process the Primary Input File as a DDNAME	44
SPELLCHK: Enable the .SV Control Word	45
STOP: Print Separate Pages at the Terminal	45
SYOFF: Disable .SY [System Command] Control Word	45
SYON: Enable .SY [System Command] Control Word	45
SYSVAR: Set System Variable Symbols	45
TERM: Display the Output at the User's Terminal	46
TLIB: Specify Spelling Verification and Hyphenation Libraries	46
TWOPASS: Prepare the Document with Two Formatting Passes	47
UNFORMAT: Print All Input Lines without Formatting	47
UPCASE: Print Lowercase Letters as Uppercase	48
@user-option: User-defined Options	48

<b>Chapter 4. SCRIPT/VS Control Words</b>	53
Control Word Syntax	53
Control Word Separator	53
Macros	54
The Control Word Modifier	54
The Input Line	55
Notational Conventions	55
Type 1 Control Words	55
Reading Syntax Diagrams	56
Space Units	58
SCRIPT/VS Control Word Descriptions	59
... [Set Label]	60
.AA [Associate APF]	62
.AM [Adjust Margin]	64
.AN [And]	66
.AP [Append]	68
.AR [Area]	70
.BC [Balance Columns]	74
.BF [Begin Font]	75
.BL [Blank Line]	78
.BM [Bottom Margin]	79
.BR [Break]	80
.BX [Box]	81
.CB [Column Begin]	91
.CC [Conditional Column Begin]	92
.CD [Column Definition]	93
.CE [Center]	95
.CG [Copy Group]	97
.CL [Column Line Length]	99
.CM [Comment]	100

.CP [Conditional Page Eject]	101
.CR [Color]	102
.CS [Conditional Section]	105
.CT [Continued Text]	107
.DA [Define Area]	108
.DC [Define Character]	112
.DD [Define Data File-Id] in the CMS Environment	120
.DD [Define Data File-id] in the MVS/TSO Environment	122
.DD [Define Data File-Id] in the ATMS Environment	125
.DD [Define Data File-Id] in the DLF Environment	127
.DF [Define Font]	130
.DH [Define Head Level]	136
.DL [Dictionary List]	140
.DM [Define Macro]	142
.DO [Define Object]	147
.DR [Define Rule]	151
.DS [Double Space Mode]	154
.DU [Dictionary Update]	155
.DV [Define Variable]	157
.EC [Execute Control]	161
.EF [End of File]	163
.EL [Else]	164
.EM [Execute Macro]	166
.ES [Extra Space]	167
.EQ [Equation]	169
.FL [Float]	170
.FN [Footnote]	172
.FO [Format Mode]	174
.FV [Format Vertically]	177
.GO [Goto]	179
.GR [Group]	181
.GS [GML Services]	183
.H0–.H6 [Head Levels 0–6]	190
.HR [Horizontal Rule]	192
.HW [Hyphenate Word]	195
.HY [Hyphenate]	196
.IC [Intercharacter Space]	201
.IE [Index Entry]	202
.IF [If]	203
.IL [Indent Line]	206
.IM [Imbed]	208
.IN [Indent]	210
.IO [Include Object]	213
.IR [Indent Right]	217
.IS [Inline Space]	219
.IT [Input Trace]	221
.IX [Index]	224
.JU [Justify Mode]	226
.KP [Keep]	227
.LB [Leading Blank]	230
.LI [Literal]	231
.LL [Line Length]	233
.LO [Lead-Out]	234
.LS [Line Spacing]	235

.LT [Leading Tab]	242
.LY [Library]	243
.MC [Multicolumn Mode]	244
.ME [Macro Exit]	245
.MG [Message]	246
.MS [Macro Substitution]	248
.NF [No Formatting]	249
.NL [Null Line]	252
.NV [Navigate]	253
.OC [Output Comment]	258
.OF [Offset]	260
.OI [Overlay Include]	262
.OR [Or]	274
.PA [Page Eject]	276
.PF [Previous Font]	278
.PI [Put Index]	279
.PL [Page Length]	282
.PM [Page Margins]	284
.PN [Page Numbering Mode]	286
.PO [PostScript]	289
.PT [Put Table of Contents]	294
.PW [Page Width]	296
.PX [Prefix]	297
.QQ [Quick Quit]	299
.QU [Quit]	300
.RC [Revision Code]	301
.RD [Read Terminal]	308
.RE [Restore Environment]	309
.RF [Running Footing]	310
.RH [Running Heading]	314
.RI [Right Adjust]	318
.RN [Reference Numbers]	320
.RV [Read Variable]	321
.SA [Save Environment]	323
.SB [Shift Baseline]	325
.SC [Single Column Mode]	327
.SD [Shading Definition]	328
.SE [Set Symbol]	337
.SI [Segment Include]	343
.SK [Skip]	351
.SM [Separation Master]	353
.SO [STAIRS/VS Output]	354
.SP [Space]	356
.SS [Single Space Mode]	358
.SU [Substitute Symbol]	359
.SV [Spelling Verification]	361
.SX [Split Text]	363
.TA [Table]	365
.TB [Tab Setting]	374
.TC [Table of Contents]	376
.TD [Table Definition]	379
.TE [Terminal Input]	393
.TH [Then]	395
.TI [Translate Input]	397



.TM [Top Margin]	399
.TP [Tab Position]	400
.TR [Translate Character]	404
.TS [Translate String]	406
.TU [Translate Uppercase]	408
.TY [Type on Terminal]	411
.UC [Underscore and Capitalize]	412
.UD [Underscore Definition]	414
.UN [Undent]	416
.UP [Uppercase]	418
.US [Underscore]	420
.UW [Unverified Word]	422
.VR [Vertical Rule]	423
.VT [Variable Text]	426
.WF [Write to File]	428
.WS [Word Space]	431
.WZ [Widow Zone]	434
 <b>Chapter 5. A Summary of SCRIPT/VS</b>	 437
The SCRIPT/VS Formatting Environment	451
 <b>Appendix A. Unsupported Control Words</b>	 461
.CO [Concatenate Mode]	462
.CW [Control Word Separator]	463
.DI [Delay Imbed]	466
.EZ [Easyscript]	468
.FM [Footing Margin]	470
.FS [Footing Space]	471
.HM [Heading Margin]	472
.HS [Heading Space]	473
.PP [Paragraph Start]	474
.PS [Page Number Symbol]	475
.RT [Running Title]	476
.SL [Set Line Space]	478
 <b>Appendix B. Related Publications and Products</b>	 479
Related Products	479
Optional Features	479
Publication Library Guide for the Document Composition Facility	480
DCF Publications	482
Related Publications	483
 <b>Glossary</b>	 485
Source Identifiers	485
References	485
 <b>Index</b>	 499



---

## Figures

1.	Example of a Type 1 Control Word . . . . .	56
2.	.PO Example Using NOBREAK . . . . .	292
3.	.PO Example Showing Proportional Scaling . . . . .	293
4.	Standard Shade Patterns for AFP Devices . . . . .	330
5.	Standard Screen Patterns for AFP Devices . . . . .	331
6.	SCRIPT/VS Terms for Parts of the Page . . . . .	438
7.	TN Translate Table for the 1403 Printer . . . . .	455



---

## Tables

1.	Abbreviations for SCRIPT Command Options	20
2.	Logical Output Device versus Output Destination	23
3.	SCRIPT/VS Logical Line Devices	27
4.	SCRIPT/VS Logical PostScript Devices	28
5.	SCRIPT/VS Logical Page Devices	28
6.	Space-Units Notation	59
7.	Characters That Delimit Words for Spelling Verification	113
8.	Summary of Initial Head-Level Characteristics	138
9.	Fields of the Active Environment	311
10.	Listings and Summaries for SCRIPT/VS	437
11.	File Identifiers of SCRIPT/VS Utility Files	438
12.	Summary of SCRIPT Options	439
13.	Control Words That Cause a Break	441
14.	Control Words That Take Effect On the Next Page	441
15.	Control Words That End a Keep, Float, Footnote, Named Area, or Table	442
16.	Control Words That Ensure the Page Is Started	442
17.	SCRIPT/VS System Symbol Names — Date and Time	443
18.	SCRIPT/VS System Symbol Names — Output Device Characteristics	443
19.	SCRIPT/VS System Symbol Names — SCRIPT Command Options	444
20.	SCRIPT/VS System Symbol Names — Page Characteristics	445
21.	SCRIPT/VS System Symbol Names — SCRIPT/VS Formatter Parameters	446
22.	SCRIPT/VS System Symbol Names — Font Information	448
23.	SCRIPT/VS System Symbol Names — Table of Contents and Index	449
24.	Attributes of a Symbol's Value	450
25.	Active Formatting Environment	451
26.	Page Environment	452
27.	Fields of the Active Environment	452
28.	Output File Formats	453
29.	Complete 3800 Printing Subsystem Model 1 and Model 3 Operating in Compatibility Mode (only in MVS) Fonts Provided with SCRIPT/VS	455
30.	Translate Tables	455
31.	Fonts Supplied with the 3800 Printing Subsystem Model 1 and Model 3 Operating in Compatibility Mode (MVS only)	456
32.	Additional Fonts Supplied with the 3800 Printing Subsystem Model 3.	457
33.	Unsupported Control Words	461
34.	Easyscript Functions	469



---

# Summary of Amendments

## Release 4.0

Updates to the Document Composition Facility (DCF), Release 4.0, includes the following:<sup>1</sup>

- **Navigation:** Allows for navigation elements to be added to your documents for use with AFP Workbench for OS/2 and Windows. See “.NV [Navigate]” on page 253 for information about the new .NV [Navigate] control word.
- **Grouping:** Allows you to group information in your document for use with AFP Workbench for OS/2 and Windows. See “.GR [Group]” on page 181 for information about the new .GR [Group] control word.
- **Defining Objects:** Allows you to define an external object to be included in your document. See “.DO [Define Object]” on page 147 for information about the new .DO [Define Object] control word.
- **Including Objects:** Allows you to include an external object in your document. See “.IO [Include Object]” on page 213 for information about the new .IO [Include Object] control word.
- **Color** Allows you to add color to your document. See “.CR [Color]” on page 102 for information about the new .CR [Color] control word.
- Other changes to this publication have been made to include corrections made to the publication or program as reported by users of the Document Composition Facility.
- **Overlay Rotation** Allows you to rotate a single overlay at 0, 90, 180, or 270 degrees relative to the physical page.

---

<sup>1</sup> Release 4.0 of the Document Composition Facility may be used with Release 3 of the Document Library Facility.

---

## Notices

### Note

This book was formatted with IBM Publishing Systems BookMaster licensed program (Program Number 5688-015) and printed on a PostScript printer. When you use the SCRIPT/VS control words, your formatted output may differ from the examples shown in this book because of differences in column line length, fonts, line spacing, device types, indention, and other formatting differences between SCRIPT/VS and the BookMaster application.

References in this publication to products or services of IBM do not suggest or imply that IBM will make them available in all countries where IBM does business or that only products or services of IBM may be used. Noninfringing equivalents may be substituted, but the user must verify that such substitutes, unless expressly designated by IBM, work correctly. No license, expressed or implied, to patents or copyrights of IBM is granted by furnishing this document. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information, which has been exchanged, should contact: IBM Corporation, Printing Systems Company Legal Department, Mail Drop 001W, Boulder, Colorado 80301 USA. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

---

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries:

IBM®	Advanced Function Printing™
PSF™	Print Services Facility™
MVS™	TSO™
VM™	VSE™
BookMaster®	ProcessMaster®
BookManager™	DisplayWrite®
VM/XA™	Virtual Machine/Extended Architecture™
ATMS-III™	IBM 3800 Printing Subsystem Models 3 and 6™
CICS™	IBM 3900 Advanced Function Printer™
AFP™	IBM 3825 Page Printer™
IBM 3827 Page Printer™	IBM 3828 Advanced Function MICR Printer™
IBM 3835 Page Printer™	IBM 3812 Page Printer™
IBM 3816 Page Printer™	IBM LaserPrinter 4028™
IBM 4224-2xx Page Printer™	IBM 4234 Page Printer™
IBM 3820 Page Printer™	IBM LaserPrinter 4019™
IBM 4250 Printer™	

The following terms are trademarks of other companies as follows:

PostScript	Adobe Systems Incorporated
Monotype	Monotype Corporation, plc.
Monotype Times New Roman	Monotype Corporation, plc.
Times Roman	Allied Linotype or its subsidiaries
New Century Schoolbook	American Type Foundry



The fonts provided by IBM are frequently designed by graphic designers in other companies and adapted for IBM printers. The IBM font listed below is a functional equivalent of a font developed by another company:

IBM Font	Functional Equivalent
Sonoran Serif	Monotype Times New Roman

Central Programming Service support and maintenance are available for the following Generalized Markup Language (GML) profiles and macros:

- The GML starter set profile: DSMPROF4
- The GML schedule profile: DSMSPROF
- The GML overhead transparency profile: DSMTPROF
- The GML memo profile: DSMMPROF
- The GML bar-code profile: DSMBPROF
- The GML SCRIPT Mathematical Formula Formatter profile: DSMFPROF
- The GML office document feature profile: DSMOPROF
- The GML office document feature macro-instruction library: DSMOGML
- The GML SCRIPT Mathematical Formula Formatter macro-instruction library: DSMFMAC
- The GML macro-instruction library: DSMGML4

Support and maintenance, however, *are not* available for these profiles and macros if they have been modified in any way. If you modify these items, it is recommended that you also maintain an **unmodified** version of these items for diagnostic purposes.

An authorized program analysis report (APAR) may not be submitted regarding the translation of literals, messages, and phrases in the starter set.



---

# Chapter 1. Introduction to the Language Reference

Chapter 1. Introduction to the Language Reference	3
How to Use This Reference	3
Where to Find More Information	4
How This Reference Is Organized	4



---

# Chapter 1. Introduction to the Language Reference

This publication is specifically designed for knowledgeable to experienced end users, document administrators, text programmers, and system programmers whose tasks might include:

- Formatting documents using SCRIPT/VS control words
- Modifying the DCF Generalized Markup Language (GML) starter set (see the *Document Composition Facility: Generalized Markup Language Starter Set Implementation Guide* for more details on modifying, designing, or adding to the GML starter set).
- Creating GML applications.

**Note:** Knowledge of operating systems is not required for general use of SCRIPT/VS, however, knowledge about your operating system might be helpful when you are evaluating the use of DCF in different operating environments.

---

## How to Use This Reference

This reference manual provides information for use with the IBM Document Composition Facility (DCF) licensed program and its component text processing program, SCRIPT/VS. This reference manual describes the SCRIPT command and the SCRIPT/VS control words.

This reference manual should be used in conjunction with the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* and the *Document Composition Facility: SCRIPT/VS User's Guide*, which describe how SCRIPT/VS is used to format documents.

The information in this publication applies equally to MVS, VSE, VM, and CICS unless specifically stated otherwise. Use of SCRIPT/VS in a CICS, CMS, or TSO environment requires the Foreground Environment Feature; use in a DLF background environment requires the Document Library Facility licensed program (Program Number 5748-XXE).

References to the 3800 Printing Subsystem apply to the 3800 Printing Subsystem Model 1, and the 3800 Printing Subsystem Models 3 and 6 (in compatibility mode), unless otherwise explicitly stated.

References to page printers apply to the following unless otherwise explicitly stated:

- AFP page printers
  - 3800 Printing Subsystem Models 3 and 6
  - 3130 Advanced Function Printer
  - 3160 Advanced Function Printer
  - 3812 PagePrinter
  - 3816 Page Printer
  - 3820 Page Printer
  - 3825 Page Printer
  - 3827 Page Printer
  - 3828 Advanced Function MICR Printer
  - 3829 Advanced Function Printer
  - 3835 Page Printer
  - 3900 Advanced Function Printer
  - LaserPrinter 4028
  - 4224 Printer
  - 4234 Printer
- 4250 Page Printer

- PostScript devices configured to accept 8-bit ASCII

The IBM 4250 Printer is an electroerosion printer that produces camera-ready copy. The Composed Document Print Facility (CDPF) licensed program is required to print SCRIPT/VS formatted output on the 4250.

**Note:** Although you can use SCRIPT/VS to format for a 4250 printer, the 4250 Printer and CDPF are no longer available from or supported by IBM.

Although IBM page printers, PostScript devices, and line printers are mentioned throughout this manual, it does not contain detailed information about IBM printers or PostScript devices. For more information, read the appropriate introduction and planning guide for the product that interests you. For example, for more information about the LaserPrinter 4019, read the *IBM Personal Publishings System User's Guide*.

---

## Where to Find More Information

This book does not contain information about any installation-defined GML applications. Users of the GML starter set application should refer to the following publications:

- *Document Composition Facility: GML Starter Set User's Guide*
- *Document Composition Facility: GML Starter Set Reference*
- *Document Composition Facility: GML Starter Set Implementation Guide*.

---

## How This Reference Is Organized

This publication contains the following chapters and appendices:

Chapter 1, "Introduction to the Language Reference," explains this publication.

Chapter 2, "Using the SCRIPT Command," explains how to specify the SCRIPT command, including naming conventions and restrictions.

Chapter 3, "SCRIPT Command Options," describes the SCRIPT command and each of its options.

Chapter 4, "SCRIPT/VS Control Words," describes each SCRIPT/VS control word in alphabetical order, giving the control word syntax, usage notes, and examples.

Chapter 5, "A Summary of SCRIPT/VS," summarizes SCRIPT/VS file names, SCRIPT command options and parameters, control words, system symbols, special characters, character sets, and 3800 Printing Subsystem fonts.

Appendix A, "Unsupported Control Words," describes control words that were obsoleted in earlier releases of DCF.

Appendix B, "Related Publications and Products," includes related products, related publications, and the publication library guide for DCF.

"Glossary," lists DCF terms and definitions.

---

## Chapter 2. Using the SCRIPT Command

<b>Chapter 2. Using the SCRIPT Command</b>	<b>7</b>
Online Help for the SCRIPT Command	7
SCRIPT/VS in the Interactive Environment	7
Naming the Primary Input File	8
CMS Naming Conventions	8
TSO Naming Conventions	8
ATMS-III Naming Conventions	9
Characteristics of an Input File	9
SCRIPT/VS in the DLF Batch Environment	11
SCRIPT/VS in the TSO Batch Environment	11
Environment Restrictions and Considerations	11
MVS/TSO Data Set Naming Conventions	12





---

## Chapter 2. Using the SCRIPT Command

You can use the SCRIPT command and its options to process and format an input file either in an interactive or in a batch environment.

---

### Online Help for the SCRIPT Command

Help information is available for the SCRIPT command in an online format in CMS and TSO.

In CMS, you get the online help for the SCRIPT command as follows:

```
help script
```

For more information on the HELP command, see *Virtual Machine/System Product CMS Command Reference*, SC19-6209.

In TSO, you get the online help for the SCRIPT command as follows:

- For a description of the function of the SCRIPT command, enter  
help script function
- For a description of the syntax of the SCRIPT command, enter  
help script syntax
- For a description of all operands, enter  
help script operands
- For a description of all the help information for the SCRIPT command, enter  
help script  
or  
help script all

For more information on the HELP command, see *TSO Extensions Version 2 Command Reference*, SC28-1881.

To invoke the help facility for the SCRIPT command from the Interactive System Productivity Facility (ISPF), you must first type in “tso.” For example:

```
tso help script
```

**Note:** Message-help information is available for the GML, SCRIPT/VS, and SMFF messages in an online format in CMS and TSO. For more information on message-help information, see *Document Composition Facility Messages*.

---

### SCRIPT/VS in the Interactive Environment

To process and format an input file in the interactive environment, simply issue the SCRIPT command along with the necessary options to control processing. SCRIPT/VS formats the input file using GML tags, macros, control words, and text that are included in the file.

The SCRIPT command can be issued as a CMS command, a TSO command, or an ATMS-III command. The format of the SCRIPT command is the same for each system, except that in TSO the options must

not be placed in parentheses, and in ATMS-III the SCRIPT command itself must be entered in lowercase. The forms of the SCRIPT command are as follows:

In CMS,

```
SCRIPT    file-id [ ( options... ]
```

In TSO,

```
SCRIPT    file-id [ options... ]
```

In ATMS-III,

```
script    file-id [ [(] options... ]  
?  
*
```

*where:*

**file-id** Is the name of the primary input file. When the input file contains imbedded or appended files, *file-id* names the primary or master file; the imbedded and appended files are named with control words in the master file. The format of the file-id depends on the environment from which SCRIPT/VS is called.

Be careful when selecting the file-id. The name *profile* is assigned as the default for the SCRIPT/VS profile and should not be used for a file name, data set name qualifier, or member name.

**?** Causes SCRIPT/VS to display a list of all the valid command options.

**\*** Is the document in ATMS-III working storage.

**options** Specify how SCRIPT/VS is to process and format the input file and where the resulting output file is to go. You can specify as many options as you think appropriate. The left parenthesis "(" preceding the option list is required in the CMS environment.

## Naming the Primary Input File

The format of the name you specify for *file-id* depends on the environment from which you call SCRIPT/VS. Except when you use SCRIPT in the TSO environment, the naming rules and conventions apply equally to the primary input file, the profile, and any imbedded or appended files.

**CMS Naming Conventions:** The *file-id* of a CMS file to be processed is given in the form:

```
filename [filetype [filemode] ]
```

If *filetype* is omitted, a file type of SCRIPT is assumed. If *filemode* is omitted, the CMS search sequence is used to locate the file on an accessed CMS disk. If you want to specify the file mode, you must also give the file type because these parameters are positional.

**TSO Naming Conventions:** In TSO, the primary input file may be specified as a partially- or fully-qualified data set name. If the *file-id* given is not fully-qualified (enclosed in single quotation marks), the current TSO data set name prefix (the userid by default) is appended to the *file-id* as the left-most qualifier, and TEXT is added (unless it already appears) as the right-most qualifier. For example:

Specified DSNAME	Actual DSNAME
A	userid.A.TEXT
A.TEXT	userid.A.TEXT
DOC(CHAP1)	userid.DOC.TEXT(CHAP1)
'DPJK1.X.Y'	DPJK1.X.Y
(CHAP2)	userid.TEXT(CHAP2)

**Note:** You can also specify that the *file-id* is a ddname with the SORCDD command option. For more information on using the SORCDD option, see “SORCDD: Process the Primary Input File as a DDNAME” on page 44.

For a more detailed explanation of the data set naming conventions for TSO and SCRIPT/VS, see “MVS/TSO Data Set Naming Conventions” on page 12.

**ATMS-III Naming Conventions:** Documents in an operator's working storage can be formatted with the command

```
script *
```

Documents that are to be formatted from permanent storage or imbedded or appended can be specified in a fully-qualified way, such as:

```
'docname:opnum;getw'
```

This results in a search for the document named *docname* with a getword of *getw* belonging to the user whose operator number is *opnum*. A qualified name always results in an explicit search without subdocument index search. A name can be qualified by the use of only the colon character (:) without any opnum. This form of qualification signifies that the document is to be explicitly located and read from the requesting user's permanent storage.

If a getword is specified, it must match the document getword even though the document belongs to the requesting user. If a getword is not specified for a document that does not belong to the requestor, it must have a getword of *any*.

Documents in an operator's permanent storage can also be formatted by transmitting a request to an appropriate SCRIPT/VS peripheral queue:

```
XFO;qname;docname:opnum;getw;options
```

where *qname* is the name of a SCRIPT/VS output queue and *options* are any valid SCRIPT/VS command options.

**Note:** ATMS-III always adds an appropriate destination option, such as PRINT or CTF, to the user's options when the peripheral queue is processed. TERM is always added when the SCRIPT command is issued from a terminal.

## Characteristics of an Input File

This section outlines general characteristics and restrictions on input files. Within this section, input files refer to: the primary input file, an input file, imbed and append files, the profile, and the SCRIPT macro and search libraries. The following is a list of the characteristics of input files in the three environments:

- In a CMS environment:
  - A file type of SCRIPT
  - Fixed-length or variable-length records, with a maximum of 256 bytes per record, including the length of resolved symbols and control word specifications, which may either be explicitly used in

the source file, generated by the GML Starter Set or generated by SCRIPT/VS (.LB [Leading Blank] and .LT [Leading Tab]).

Because of the way SCRIPT/VS processes residual text, the residual text for a GML tag must not exceed 250 characters. If the residual text exceeds 250 characters, the input line is split, no message is issued, and the portion of characters over 250 is treated as text. For more information on residual text, refer to Chapter 1 in the *Document Composition Facility: GML Starter Set Implementation Guide*.

- With or without line numbers. If the lines are numbered, the number must be in positions 1 through 8 of each record. When the input file is processed, the first line is examined for a line number. If positions 1 through 8 of the record contain 8 consecutive numeric characters, the file is assumed to be numbered, and the first 8 positions of each record in the file are not scanned for control words or text.

- In a TSO environment:

- Data set organization of PO (partitioned organization) or PS (physical sequential)

**Note:** SCRIPT/VS macro and search libraries must be partitioned direct-access storage device (DASD) data sets.

- Fixed-length or variable-length records, blocked or unblocked, with a maximum of 256 per record including the control word specification and the length of resolved symbols.
- Fixed-length or variable-length records, with a maximum of 256 bytes per record, including the length of resolved symbols and control word specifications, which may either be explicitly used in the source file, generated by the GML Starter Set, or generated by SCRIPT/VS (.LB [Leading Blank] and .LT [Leading Tab]). Because of the way SCRIPT/VS processes residual text, the residual text for a GML tag must not exceed 250 characters. If the residual text exceeds 250 characters, the input line is split, no message is issued, and the portion of characters over 250 are treated as text. For more information on residual text, refer to Chapter 1 in the *Document Composition Facility: GML Starter Set Implementation Guide*.
- With or without line numbers. If, however, line numbers are used, all records in the file must contain line numbers, and the numbers must be in the correct columns as listed below:
  - For variable-length records, the line numbers occupy the first eight columns of the input record data.
  - For fixed-length records, the line numbers occupy the last eight columns of each record.

A data set or data set member is treated as numbered if the first record contains 8 consecutive numeric characters in the line number columns. The line number fields of a numbered data set or member are not scanned for control words or text.

- In an ATMS-III environment:

- Contained in ATMS-III working or permanent storage
- Composed of variable-length records, with a maximum of 230 text characters per record
- Include uppercase and lowercase letters, numbers, and special characters
- Include ATMS-III page and unit numbers that are not included in the 230 text characters.

---

## SCRIPT/VS in the DLF Batch Environment

With DLF installed, SCRIPT/VS can be used in a batch environment under VSE and MVS. DLF can be used to run SCRIPT/VS in a batch environment under MVS, but is not required unless you use the special interface to DCF provided by DLF. (DLF provides text storage and management for the documents that SCRIPT/VS processes.) Using the DLF SCRIPT command, input files can be formatted with the SCRIPT/VS formatter in a batch environment. Input files are usually stored as documents in the library or as external data sets to DLF. The output can be directed either to a printer or to an external data set.

SCRIPT/VS files stored as documents in the library are specified by:

- A three-part document identifier, which includes:
  - Library number
  - Document name
  - Password
- Data name
- Version number.

SCRIPT/VS files stored as external data sets to DLF are specified with a ddname or data set name with the FROM operand of the DLF SCRIPT command.

See the *Document Library Facility Guide*, SH20-9165, for more information.

---

## SCRIPT/VS in the TSO Batch Environment

You can also run SCRIPT/VS in a background TSO job. See the chapter titled “Executing Foreground Commands From a Background Job” in the *TSO/E Version 2 User's Guide*, SC28-1880 for details on running TSO in a background job.

---

## Environment Restrictions and Considerations

Depending on the environment in which you are using SCRIPT/VS, certain SCRIPT command options and control word parameters are restricted.

- In the CMS environment, the following should not be used:
  - The DEST command option
  - Suboptions of the PRINT command
  - The DD, DSN, CATALOG, DATA, VERSION, PROC, and PARM parameters of the .DD [Define Data File-id] control word
  - The SCRPTFIL, SCRPTFNT, SCRPTLIB, SCRPTPRO, TEXTLIB, and SCRPTSEG TSO ddnames.
  - At the end of processing SCRIPT/VS closes all open files, whether or not those files were opened by SCRIPT/VS.<sup>2</sup> This can only be seen if SCRIPT/VS is called from within an exec or program, because CMS closes all open files when control returns to CMS. If SCRIPT/VS is being called from within an exec or program, such that this processing causes problems in reading other files from within the exec or program, read lines from your other files specifically by line number.
- In the TSO environment, the following should not be used:

- The OPTIONS, TLIB, and @user-option command options
- The DATA, VERSION, PROC, and PARM parameters of the .DD [Define Data File-id] control word.
- In the ATMS-III environment, the following should not be used:
  - The DEST, FILE, NOSPIE, NOWAIT, STOP, TLIB, and @user-option SCRIPT command options
  - The PROMPT option (implied or explicit) of the PAGE command option
  - The suboptions of the PRINT command option
  - The DD, DSN, CATALOG, DATA, VERSION, PROC, and PARM parameters of the .DD [Define Data File-id] control word
  - In VSE, the SEGLIB option should not be used.
  - The SCRPTFIL, SCRPTFNT, SCRPTLIB, SCRPTPRO, TEXTLIB, and SCRPTSEG TSO ddnames.
- In the DLF environment, the following should not be used:
  - The LIB, QUIET, NOSPIE, STOP, and TERM options of SCRIPT/VS
  - The PROMPT option (implied or explicit) of the PAGE command option
  - In VSE, the SEGLIB option.
  - The SCRPTFIL, SCRPTFNT, SCRPTLIB, SCRPTPRO, TEXTLIB, and SCRPTSEG TSO ddnames.

The DSN... parameter of the FROM suboption (which specifies the name of the data set used for input when the document library is not the source) and the DSN... parameter of the FILE option (which specifies the name of the data set used for output) are valid only in MVS.

## MVS/TSO Data Set Naming Conventions

In the TSO environment, SCRIPT/VS is a command language processor and follows most of the standard conventions for command language processors. All data sets, except temporary data sets allocated by SCRIPT/VS, must be cataloged. The ability to assign alias data set names through the MVS/VSAM catalog is not supported by the DCF product. All data sets must be cataloged under the correct name. Data set names may be specified in partially qualified or fully qualified format and must follow these conventions:

- **Partially qualified data set names:** Abbreviated specifications that are expanded by SCRIPT/VS and TSO. MVS data set names are composed of one or more 1- to 8-character qualifiers. If the data set name contains more than one qualifier, the qualifiers must be separated by periods. When a partially qualified dsname is specified for a SCRIPT/VS data set, TSO appends the current data set name prefix as the left most qualifier of the data set name. The standard TSO default data set name prefix is the TSO *userid*; however, the default may be changed, or TSO prefixing may be turned off through the PREFIX option of the TSO PROFILE command.

**Note:** SCRIPT/VS requires that each user have a unique TSO data set name prefix. If prefixing is turned off or if two or more users use the same data set name prefix, duplicate data set names may be generated by SCRIPT/VS, and allocation errors occur. A batch TSO job may require a PROFILE PREFIX(*userid*) command before the SCRIPT command.

If a unique data set name prefix cannot be used, it is the user's responsibility to ensure that unique data set names are supplied for all SCRIPT/VS output files and utility files. This can be accomplished by using fully qualified data set names on the SCRIPT/VS command line or on the .DD [Define Data

---

<sup>2</sup> See the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for more information about closing files.

File-id] control word (used with the DDUT command option), or by preallocating the necessary data sets.

For the purpose of SCRIPT/VS publications, the data set name prefix is assumed to be the standard TSO default, and the qualifier *userid* is used in examples and text.

In addition to the prefix qualifier added by TSO, SCRIPT/VS may add a suffix or right most qualifier to partially qualified data set names. The suffix qualifier depends on the function of the data set and is explained in the topics describing those command options and control words that require a data set name.

- **Fully qualified data set names:** Specifications that must be enclosed in single quotation marks and must contain the complete data set name (all qualifiers). When a fully qualified data set name is given, the name is used as supplied and is not modified by SCRIPT/VS or TSO. Fully qualified data set names must be specified for existing data sets that have names that do not follow the TSO and SCRIPT/VS conventions. Fully qualified data set names must also be specified for new or temporary data sets (such as formatted output files or utility files) if you want names that do not follow the standard conventions.

**Note:** SCRIPT/VS references all data sets internally with a 1- to 8-character file identifier (or file-id) which must be unique. For the primary input, the profile, and the lib data sets specified on the SCRIPT/VS command line, SCRIPT/VS uses the member name if the data set is a PDS and the member name is supplied on the command line. If the data set is a sequential data set or it is a PDS and no member name is supplied in the command line, SCRIPT/VS uses the right most data set name qualifier supplied on the command line. For example:

```
SCRIPT mylib(mydoc) PROF(dsmprof4) LIB('userid.r40.maclib')
```

refers to three data sets: “userid.mylib.text(mydoc),” which is a PDS containing the primary input document, “userid.dsmprof4.text,” which is the profile data set, and “userid.r40.maclib,” which is the maclib data set. The internal file-id assigned to each of the files in the order that they occur on the command line are: MYDOC, DSMPROF4, and MACLIB. Since *text* is by convention the last qualifier for SCRIPT/VS source files and the internal file-ids must be unique, some care must be taken in specifying data set names on the SCRIPT/VS command line to prevent file-id conflicts.





---

## Chapter 3. SCRIPT/VS Command Options

<b>Chapter 3. SCRIPT Command Options</b>	17
Command Option Syntax	17
Default Options	17
Command Options and TSO DDNAMES	18
Mutually Exclusive Options	18
Logical Output Devices and Destinations	19
Printer Classes	22
SCRIPT Command Option Descriptions	24
BIND: Shift the Page to the Right	24
CHARS: Specify Fonts	24
CONTINUE: Continue Processing after a Nonsevere Error Occurs	25
CTF: Prepare Output in STAIRS/VS Condensed Text Format	25
DDUT: Enable SCRIPT/VS Utility File Redefinition	25
DEST: Name a Remote Output Station or Queue	25
DEVICE: Specify a Logical Output Device	26
SCRIPT/VS Logical Devices	27
SCRIPT/VS Logical Line Devices	27
SCRIPT/VS Logical PostScript Devices	28
SCRIPT/VS Logical Page Devices	28
FILE: Name a Disk File for Output	30
CMS	30
TSO	30
FONTLIB: Specify a Font Library	33
FPASSES: Specify Number of Formatting Passes	35
INDEX: Enable Back of Book Index	35
LIB: Specify Symbol and Macro Libraries	35
MESSAGE: Control Message Printing	36
NOCONT: Ending Processing after an Error Occurs	37
NODDUT: Disable SCRIPT/VS Utility File Redefinition	37
NOPROF: Suppress the Profile	37
NOSEGLIB: Ignore the Segment Library	37
NOSPIE: Prevent Entering SPIE Exit Routines	38
NOSORCDD: Specifying the Primary Input File as a Data Set Name	38
NOWAIT: Prevent Prompting for Paper Adjustment	38
NUMBER: Print the File Name and Line Number	38
OPTIONS: Name a File That Contains Options	38
PAGE: Selectively Print Pages	39
PRINT: Produce Printer Output	40
PROFILE: Specify a Profile	40
PSOUT: Specify ASCII or EBCDIC PostScript Output	41
QUIET: Suppress the Formatter's Identifier Message	42
SEARCH: Specify a Library	42
SEGLIB: Specify a Segment Library	43
SEPMASTR: Specify Output Separation Masters	44
SORCDD: Process the Primary Input File as a DDNAME	44
SPELLCHK: Enable the .SV Control Word	45
STOP: Print Separate Pages at the Terminal	45
SYOFF: Disable .SY [System Command] Control Word	45
SYON: Enable .SY [System Command] Control Word	45
SYSVAR: Set System Variable Symbols	45

TERM: Display the Output at the User's Terminal . . . . .	46
TLIB: Specify Spelling Verification and Hyphenation Libraries . . . . .	46
TWOPASS: Prepare the Document with Two Formatting Passes . . . . .	47
UNFORMAT: Print All Input Lines without Formatting . . . . .	47
UPCASE: Print Lowercase Letters as Uppercase . . . . .	48
@user-option: User-defined Options . . . . .	48

---

## Chapter 3. SCRIPT Command Options

---

### Command Option Syntax

SCRIPT command options control how SCRIPT/VS processes and formats your input file. Some of the options have suboptions (suboptions are enclosed in parentheses). Options and suboptions are separated by blanks.

SCRIPT command option considerations for the following environments:

- In TSO, a comma is used as a separator and therefore cannot be used in a suboption as anything other than a separator. The number of characters that you can enter in one command is limited to 4096 characters.
- In CMS, the character limit is controlled by CMS command tokenizing. CMS parses each option, suboption, and parenthesis as 8-byte tokens. For example, BIND (10 10) becomes 5 tokens of 8 bytes each, which totals 40 bytes. The current maximum number of bytes available for command line tokenizing is 512 bytes. If the total number of options, suboptions, and parentheses exceeds 64 bytes, truncation occurs.
- In other environments, the command line is limited to 255 bytes. If additional characters are needed, you must use the OPTIONS parameter as described in this chapter. However, the OPTIONS parameter is invalid in TSO.

The name of each option can be shortened to its minimum abbreviation. In TSO, ambiguous truncations are not accepted, and you are prompted to reenter the option. In other systems, ambiguous truncations are accepted and interpreted as shown in Table 1 on page 20.

### Default Options

If you specify the SCRIPT command with a *file-id* or an \* and no options, the defaults are:

For CMS,

```
TERM PROFILE (PROFILE) LIB (DSMGML4) NOCONT NODDUT SYOFF
```

For TSO,

```
TERM PROFILE (PROFILE) LIB ('SCRIPT.R40.MACLIB') NOCONT NODDUT SYOFF NOSORCDD
```

For ATMS-III,

```
TERM PROFILE (PROFILE) MESSAGE (DELAY) NOCONT
```

For DLF,

```
PRINT PROFILE (PROFILE) MESSAGE (DELAY) NOCONT
```

All other options must be either explicitly specified when desired or the defaults changed at installation time. Refer to “Tailoring SCRIPT/VS for Your Installation” in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.

## Command Options and TSO DDNAMES

TSO ddnames are provided as an alternative to using the LIB, SEARCH, SEGLIB, FONTLIB, PROFILE, and FILE command options.

The following table lists all of the command options that have corresponding ddnames:

Command Option	TSO DDNAME
FILE	SCRPTFIL
FONTLIB	SCRPTFNT
LIB	SCRPTLIB
PROFILE	SCRPTPRO
SEARCH	TEXTLIB
SEGLIB	SCRPTSEG

## Mutually Exclusive Options

Some of the SCRIPT command options are mutually exclusive. When two such command options are specified, no error results, although one option can cancel the effect of another specified option. Within the following groups of options, the last one processed by SCRIPT/VS takes effect.

- SORCDD and NOSORCDD: The SORCDD command option notifies SCRIPT to process the primary input file as a ddname. The NOSORCDD option overrides the SORCDD option and **explicitly** specifies that the primary input file is a data set name.
- PROFILE and NOPROF: The PROFILE option specifies that a file is to be imbedded before the primary input file is processed. The NOPROF option specifies that no profile is needed.
- CONTINUE and NOCONT: These options determine whether processing is to continue after SCRIPT/VS detects an error condition and issues an error message. Even if CONTINUE has been specified, SCRIPT/VS stops processing if a *severe* or *terminal* error is encountered.
- SEGLIB and NOSEGLIB: These options determine whether the segment library is to be searched for a specified segment. If the CONTINUE option has been specified, SCRIPT/VS continues processing even if the specified segment does not exist.
- DDUT and NODDUT: These options determine if SCRIPT/VS utility file redefinition to a non-utility file is allowed. NODDUT disallows the redefinition, but DDUT allows it.
- SYON and SYOFF: These options determine if the .SY [System Command] control word is enabled or disabled. SYOFF disables .SY [System Command] and SYON enables it.

Because of the way TSO parses parameters before passing them to SCRIPT/VS, some options are processed in alphabetical order in TSO regardless of the order of entry. In other systems, they are processed in the order in which they are specified.

- CTF, FILE, PRINT, and TERM: These options specify the destination of the formatted output. If a logical output device is not specified also, SCRIPT/VS selects one based on the destination. If CTF is specified and the device type is not STAIRS, CTF is ignored. Table 2 on page 23 lists the default logical device for each destination.
- TWOPASS and FPASSES: These options determine the number of formatting passes SCRIPT/VS performs when processing an input file.

## Logical Output Devices and Destinations

SCRIPT/VS can format a document for a number of different output devices. See the logical device tables on page 27 through 28 for a list of the output devices available.

SCRIPT/VS always formats documents for some specific logical output device. A logical output device is a combination of a physical device type, such as the 3800 Printing Subsystem, form size, such as 8-1/2 by 11 inches, and a specific lines-per-inch specification, such as 6 or 8. The logical device type is specified with the DEVICE option of the SCRIPT command. For example, you can direct SCRIPT/VS to format a document for a 3800 Printing Subsystem Model 1, standard page size (8-1/2 by 11 inches) at 8 lines-per-inch with the following command:

```
SCRIPT TEST ( DEVICE(3800N8)
```

If the DEVICE option is not specified with the SCRIPT command, SCRIPT/VS uses a default logical device, usually 1403W6. This default device can be changed by the installation.

Option	Non-TSO Environments	TSO Environments
BIND	B	B
CHARS	C	CH
CONTINUE	CO	CO
CTF	CT	CT
DDUT	DD	DD
DEST		DES
DEVICE	D	DEV
FILE	F	FI
FONTLIB	FO	FO
FPASSES	FP	FP
INDEX	I	I
LIB	L	L
MESSAGE	M	M
NOCNT	NOC	NOC
NODDUT	NOD	NOD
NOPROF	N	NOP
NOSEGLIB	NOSE	NOSE
NOSORCDD		NOSO
NOSPIE	NOS	NOSP
NOWAIT	NOW	NOW
NUMBER	NU	NU
OPTIONS	O	
PAGE	P	PA
PRINT	PR	PRI
PROFILE	PRO	PRO
PSOUT	PS	PS
QUIET	Q	Q
SEARCH	S	SEA
SEGLIB	SEG	SEG
SEPMASTR	SEP	SEP
SORCDD		SO
SPELLCHK	SP	SP
STOP	ST	ST
SYOFF	SYOF	SYOF
SYON	SYON	SYON
SYSVAR	SYS	SYS
TERM	T	TE
TLIB	TL	TL
TWOPASS	TW	TW
UNFORMAT	U	UN
UPCASE	UP	UP

Table 1. Abbreviations for SCRIPT Command Options

The logical device that SCRIPT/VS uses in formatting the document is independent of the actual destination of the formatted output. For example, you can direct SCRIPT/VS to format for a 3800 Printing Subsystem and to put the formatted output in a disk file rather than send it to the printer. There are several options of the SCRIPT command that specify the destination of the output. These are:

- FILE (put it in a disk file)
- TERM (send it to your terminal)
- PRINT (send it to the printer)
- CTF ( put it in a disk file in Condensed Text Format for the STAIRS licensed program).

You can specify almost any combination of output destination and logical device. For example, if you specify

```
SCRIPT TEST ( FILE DEVICE(3800N8)
```

SCRIPT/VS formats a document for the 3800 Printing Subsystem at 8 lines-per-inch but saves the output in a disk file for later printing on a physical printer, if requested.

**Note:** There are two exceptions to this rule:

1. The CTF destination is valid only for the STAIRS logical device and is ignored if any other logical device is specified.
2. The PRINT option is not valid for the 4250 Page Printer or PostScript logical devices.

Additionally, certain destinations are invalid in certain environments. For more details, see the descriptions of the FILE, PRINT, TERM, and CTF command options also in this chapter.

If you specify only a logical device with the DEVICE option, SCRIPT/VS assumes an appropriate output destination. For example, if you specify 1403N8 as the logical device, SCRIPT/VS sends the output to an appropriate printer destination. If you specify a 4250 Page Printer logical device, SCRIPT/VS files the output in a disk file. In the ATMS-III environment, ATMS-III causes the output for a 4250 Page Printer to be sent to a CICS/VS extra partitioned dataset.

For more information on obtaining 4250 Page Printer output, see the *Composed Document Printing Facility: Data Stream Interface, Typographic Font Interface*.

Similarly, if you specify an explicit output destination, SCRIPT/VS assumes an appropriate logical device. If you specify neither a destination nor a logical device, SCRIPT/VS formats the document for and sends it to your terminal. Table 2 on page 23 shows the logical output device and output destination for a document when various combinations of options are specified.

## Printer Classes

SCRIPT/VS supports three classes of printer devices: line printers, page printers, and PostScript devices. A line printer is any printer that accepts one line of text at a time from the host system. SCRIPT/VS supports such line devices as the 1403 printer, 2741 typewriter terminal, and 3800 Printing Subsystem.

A page printer is any printer that accepts composed pages, which are constructed of composed text and images, among other things. SCRIPT/VS supports the following page printers:

- AFP page printers
  - 3800 Printing Subsystem Models 3 and 6
  - 3812 PagePrinter
  - 3816 Page Printer
  - 3820 Page Printer
  - 3825 Page Printer
  - 3827 Page Printer
  - 3828 Advanced Function MICR Printer
  - 3835 Page Printer
  - 3900 Advanced Function Printer
  - LaserPrinter 4028
  - 4224 Printer
  - 4234 Printer
- 4250 Page Printer
- PostScript devices configured to accept 8-bit ASCII

The IBM 4250 Printer is an electroerosion printer that produces camera-ready copy. The Composed Document Print Facility (CDPF) licensed program is required to print DCF formatted output on the 4250.

**Note:** Although you can use DCF to format for a 4250 printer, the 4250 Printer and CDPF are no longer available from or supported by IBM.

A PostScript device is any device that accepts the PostScript page description language. PostScript devices are not all-points addressable like page printers. The resolution of a PostScript device is independent of the PostScript data stream. SCRIPT/VS supports PostScript devices, such as the IBM 4019 LaserPrinter. PostScript devices must be configured to accept 8-bit ASCII.

Refer to the *Document Composition Facility: SCRIPT/VS User's Guide* and the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for information on font requirements for page printers and PostScript devices.



Options Specified	Logical Device	Physical Device	Output Destination
none [Other]	1403W6	1403	Printer
none [DLF]	TERM	2741 or 3270	Terminal
CTF	STAIRS	1403	1
FILE	1403W6	1403	File
PRINT	1403W6	1403	Printer
TERM	TERM	2741 or 3270	Terminal
DEVICE(1403xxx)	1403xxxx	1403	Printer
DEVICE(2741)	2741	2741	Terminal
DEVICE(3270)	3270	3270	Terminal
DEVICE(38PPxxxx)	38PPxxxx	3800-3	Printer <sup>6</sup>
DEVICE(3800xxxx)	3800xxxx	3800	Printer <sup>6</sup>
DEVICE(3820xxxx)	3820xxxx	3820	Printer <sup>6</sup>
DEVICE(PG1xxxx)	PG1xxxx	3820	Printer <sup>6</sup>
DEVICE(PG2xxxx)	PG2xxxx	3820	Printer <sup>6</sup>
DEVICE(PG3xxxx)	PG3xxxx	4224	Printer <sup>6</sup>
DEVICE(PG4xxxx)	PG4xxxx	4028	Printer <sup>6</sup>
DEVICE(AFPxxxx)	AFPxxxx	AFP	Printer <sup>6</sup>
DEVICE(AFP2xxxx)	AFP2xxxx	AFP	Printer <sup>6</sup>
DEVICE(PSxxxx)	PSxxxx	PostScript	File <sup>5</sup>
DEVICE(4250xx)	4250xx	4250	File <sup>2</sup>
DEVICE(STAIRS)	STAIRS	1403	1
CTF DEVICE(devtype)	devtype	device	3
FILE DEVICE(devtype)	devtype	device	File
PRINT	devtype	device	Printer
DEVICE(devtype)			
TERM	devtype	device	Terminal
DEVICE(devtype)			
CTF DEVICE(STAIRS)	STAIRS	1403	1
FILE DEVICE(STAIRS)	STAIRS	1403	File <sup>4</sup>
PRINT	STAIRS	1403	Printer <sup>4</sup>
DEVICE(STAIRS)			
TERM	STAIRS	1403	Terminal <sup>4</sup>
DEVICE(STAIRS)			

**Note:**

- 1 The destination of Condensed Text Format output depends upon the environment:
  - In CMS, TSO, and MVS/VS2: a file named DSMUTCTF
  - In VSE: a file named DSMUCTF
  - In ATMS-III: a CICS/VS partitioned data set.
- 2 In ATMS-III, the destination is a CICS/VS partitioned data set.
- 3 If the CTF and DEVICE options are both specified and *devtype* is not STAIRS, CTF is ignored.
- 4 This output is in STAIRS Proof format.
- 5 ATMS-III does not support PostScript.
- 6 In CMS, the default is File.

*Table 2. Logical Output Device versus Output Destination. It is the user's responsibility to ensure that the characteristics of the physical device to which the output is directed match the characteristics of the specified or implied logical device. Your installation's conventions for output classes and forms must be included in these considerations. Refer to the logical device tables on pages 27 through 28 for a description of all logical devices.*

---

## SCRIPT Command Option Descriptions

- I This section contains an alphabetical listing of descriptions of the SCRIPT command options.

### BIND: Shift the Page to the Right

The BIND option causes SCRIPT/VS to shift the formatted output of each page to the right side of the paper. The BIND option is specified as:

```
BIND ( obind ebind )  
    or  
BIND ( bind )
```

You can specify a binding for odd-numbered pages (*obind*) and a different binding for even-numbered pages (*ebind*). If *ebind* is not specified, the value of *bind* applies to both odd- and even-numbered pages. Binding values can be given in any valid space unit. Table 6 on page 59 lists space units recognized by SCRIPT/VS. The default binding for each logical device is shown in the logical device tables on page 27 through 28.

Left revision codes for the first column are placed in the binding space; revision codes for subsequent columns are placed in the gutter between columns. If sufficient space is not provided for revision codes, they are discarded.

You can override the values specified for binding with the .PM [Page Margins] control word.

**Note:** Be careful that you do not specify a BIND value so large that formatted output exceeds the width of the page for the logical device.

### CHARS: Specify Fonts

The CHARS option identifies the fonts to be used.

The CHARS option is specified as:

```
CHARS ( font1 [font2] ... )
```

For page devices, you specify an 8-character font identifier for the coded font. For PostScript devices, you specify a 1- to 8-character AFM (Adobe Font Metrics) filename.

If you specify the CHARS option, you must specify at least one font. You can specify a maximum of 31 fonts.

All of the fonts specified with the CHARS option are put into a symbol array, &\$CHAR(), where &\$CHAR(1) is the initial font specified, &\$CHAR(2) is the second font, and so on.

If you do not specify the CHARS option, the default font specified for the logical device is used. In either case, the first font specified or implied becomes the initial font.

When formatting for the 3800 Printing Subsystem Model 1, you can specify up to four uppercase-only fonts or two uppercase and two lowercase fonts, and only these fonts can be used within the document. The 3800 fonts provided with SCRIPT/VS are listed in Appendix C in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*. The document must then be printed on a system that supports the IBM 3800 Printing Subsystem, using the same fonts used for formatting:

- In CMS, if you have defined a virtual 3800 printer, you must issue the CP SPOOL or the CMS SETPRT command with the CHARS parameter.

- In MVS, the SYSOUT JCL statement must include the DCB=OPTCD=J parameter and the CHARS parameter.
- In VSE, the SETPRT JCS statement must include the TRC=Y parameter and the CHARS parameter.

When formatting for page printers and PostScript devices, use the FONTLIB option of the SCRIPT command to identify the host system font library containing the fonts to be used. The CHARS option can be used to establish the initial font used in your document. However, you are not limited to those fonts identified with the CHARS option; you can use any of the fonts in the font library.

## **CONTINUE: Continue Processing after a Nonsevere Error Occurs**

The CONTINUE option allows processing to continue after SCRIPT/VS detects an error condition and issues an error message. When SCRIPT/VS encounters an error that is too severe for processing to continue, it stops processing even when CONTINUE is specified. Severe and terminal errors cause SCRIPT/VS to terminate processing.

For a description of error types and SCRIPT/VS error messages, refer to the *Document Composition Facility: Messages* manual distributed with SCRIPT/VS.

## **CTF: Prepare Output in STAIRS/VS Condensed Text Format**

The CTF option specifies that the document be prepared as input for the STAIRS/VS licensed program. SCRIPT/VS output is placed in Condensed Text Format blocks and written to an appropriate destination:

- In CMS, a file named DSMUTCTF SCRIPT
- In TSO, a file named DSMUTCTF.TEXT
- In ATMS-III, a CICS partitioned data set
- In MVS, the file identified by the DSMUTCTF DD statement
- In VSE, the file identified by the DSMUCTF DLBL statement.

If the device option is not specified, the STAIRS logical device is assumed; if any other logical device is specified, the CTF option is ignored.

## **DDUT: Enable SCRIPT/VS Utility File Redefinition**

The DDUT option allows the redefinition of SCRIPT/VS utility files to non-utility file names using the .DD [Define Data File-id] control word. See the descriptions of the .DD control word for more information.

## **DEST: Name a Remote Output Station or Queue**

The DEST option, available only in TSO, is used to specify a remote output station at which the output document is to be printed. The DEST option is specified as

DEST (destination)

where *destination* is a 1- to 8-character station-id.

**Note:** The 8-character station-id is not valid in all versions of TSO. See your systems programmer for a list of all valid station-ids at your installation.

The DEST option is ignored if the output destination is not a printer.

## DEVICE: Specify a Logical Output Device

The DEVICE option allows you to identify the type of output device for which you want your document formatted. The logical device description includes the default page layout and characteristics of the physical output device. The DEVICE option is specified as

DEVICE ( devtype )

where *devtype* is the name of a logical output device that takes into account the physical characteristics of the device as well as the characteristics that can be changed by the operator, such as form size and lines per inch (for line devices).

For example, the characteristics of the logical output device 3800N6 are as follows:

- The physical device type is a 3800 Printing Subsystem Model 1.
- The page is 8.5 inches wide and 10 inches long.
- Six lines per inch are printed.
- The top and bottom margins are both 0.
- The page margin is 1 inch.

SCRIPT/VS logical device support allows a single physical device type to be defined as many different logical device types, each having different characteristics. The logical devices defined in SCRIPT/VS are summarized in the logical device tables on pages 27 through 28.

If you issue the SCRIPT command to format and display your document at the terminal and you do not specify the DEVICE option of the SCRIPT command, DEVICE(TERM) is assumed. If you invoke SCRIPT/VS in a DLF environment or use the PRINT option in a foreground environment and do not specify a device type, SCRIPT/VS assumes DEVICE(1403W6).

Formatted output for the 3800 Printing Subsystem contains table reference characters (TRCs). Therefore:

- For CMS, output may be directed to a virtual 3800 printer.
- For MVS, the DCB=OPTCD=J parameter must be specified on the SYSOUT JCL statement.
- For VSE, the TRC=Y parameter must be specified on the // SETPRT JCS statement.

The formatted output for page printers is a composed page data stream. The formatted output for PostScript devices is the PostScript page description language. PostScript devices are different from line printers or page printers in that the resolution of the printer is independent of the PostScript data stream.

If you specify DEVICE(STAIRS) or CTF, the document is formatted as it would be for device 1403W6, but instead of preparing the output for a line printer, SCRIPT/VS converts the output to STAIRS/VS Condensed Text Format (CTF) records and writes them to an appropriate file, as described under the CTF option. The format of STAIRS/VS Condensed Text Format blocks is given in the *Document Composition Facility: Text Programmer's Guide*.

If you specify DEVICE(STAIRS) and PRINT, FILE, or TERM, the document is written to the specified destination for proofreading.

In ATMS-III peripheral queue operations, you cannot override the DEVICE type defined by ATMS-III.

**Note:** ATMS-III does not support PostScript output.

**SCRIPT/VS Logical Devices:** These tables list the logical devices that can be specified with the DEVICE option of the SCRIPT command and the default page dimensions for each device. These are the defaults supplied by IBM. You can change the defaults at installation time. The page size can be changed with the .PW [Page Width] and .PL [Page Length] control words. The page margins can be changed with the .PM [Page Margins], .AM [Adjust Margins], .TM [Top Margin], and .BM [Bottom Margin] control words.

## SCRIPT/VS Logical Line Devices

Logical Device Type	Intended Output Device	Lines per Inch	Page Size		Margins			Line Length
			Width	Length	Bind	Top	Bottom	
TERM	( 1 )							
2741	2741	6	132	11i	2	.5i	.5i	6i
3270	3270		204	11i				
1403N6	1403	6	8.5i	11i	1i	.5i	.5i	6i
1403N8		8	8.5i	11i				
1403W6		6	13.5i	11i				
1403W8		8	13.5i	11i				
1403W6S		6	13.5i	8.5i				
1403W8S		8	13.5i	8.5i				
1403SW <sup>2</sup>		6	8.5i	11i				
STAIRS		6	13.5i	11i				
3800N6	3800	6	8.5i	10i	1i	0	0	6i
3800N8		8	8.5i	10i				
3800N12		12	8.5i	10i				
3800W6		6	13.5i	10i				
3800W8		8	13.5i	10i				
3800W12		12	13.5i	10i				
3800N6S		6	11i	7.5i				
3800N8S		8	11i	7.5i				
3800N12S		12	11i	7.5i				
3800W6S		6	13.5i	7.5i				
3800W8S		8	13.5i	7.5i				
3800W12S		12	13.5i	7.5i				
<b>Note:</b>								
<sup>1</sup> The physical device type corresponding to the TERM logical device can be either 2741 or 3270, depending upon the actual terminal type. Under ATMS, the physical device type is always 2741.								
<sup>2</sup> This is a 12-pitch device; all other 1403 devices are 10-pitch.								

Table 3. SCRIPT/VS Logical Line Devices. This table lists the logical line devices that can be specified with the DEVICE option of the SCRIPT command and the default page dimensions for each device.

## SCRIPT/VS Logical PostScript Devices

Logical Device Type	Intended Output Device	Page Size		Margins			Line Length
		Width	Length	Bind	Top	Bottom	
PSA	PostScript	8.5i	11i				
PSA90		11i	8.5i				
PSA180		8.5i	11i				
PSA270		11i	8.5i				
PSB		11i	17i	1i	.5i	.5i	6i
PSL		8.5i	14i				
PSA3		297mm	420mm				
PSA4		210mm	297mm				
PSB5		182mm	257mm				

Table 4. *SCRIPT/VS Logical PostScript Devices.* This table lists the logical PostScript devices that can be specified with the *DEVICE* option of the *SCRIPT* command and the default page dimensions for each device. The line spacing for PostScript logical devices is determined by the *.LS* [Line Spacing] control word and the fonts used in the document.

## SCRIPT/VS Logical Page Devices

**Note:** Mixing different fonts in a document may cause problems.

Logical Device Type	Intended Output Device	Page Size		Margins			Line Length
		Width	Length	Bind	Top	Bottom	
38PPN	3800-3	8.5i	10i				
38PPW		13.5i	10i	1i	0	.125i	6i
38PPNS		11i	7.5i				
38PPWS		13.5i	7.5i				
38PPW90		10i	13.5i				
38PPNS90		7.5i	11i	.5i	.5i	.5i	6i
38PPW270		10i	13.5i				
3820A	3820 <sup>1</sup>	8.5i	11i				
3820A90		11i	8.5i				
3820A180		8.5i	11i				
3820A270		11i	8.5i	1i	.5i	.5i	6i
3820L		8.5i	14i				
3820A4		210mm	297mm				
3820B4		257mm	364mm				
3820B5		182mm	257mm				
4250A	4250	8.5i	11i				
4250B		11i	17i				
4250L		8.5i	14i	1i	.5i	.5i	6i
4250A3		297mm	420mm				
4250A4		210mm	297mm				
PG1A	3820 <sup>1</sup>	8.5i	11i				
PG1A90		11i	8.5i				
PG1A180		8.5i	11i				
PG1A270		11i	8.5i	1i	.5i	.5i	6i
PG1L		8.5i	14i				
PG1A4		210mm	297mm				
PG1B4		257mm	364mm				
PG1B5		182mm	257mm				

Logical Device Type	Intended Output Device	Page Size		Margins			Line Length
		Width	Length	Bind	Top	Bottom	
PG2A PG2A90 PG2A180 PG2A270 PG2L PG2A4 PG2B4 PG2B5	3812 3816	8.5i 11i 8.5i 11i 8.5i 210mm 257mm 182mm	11i 8.5i 11i 8.5i 14i 297mm 364mm 257mm	1i	.5i	.5i	6i
PG3A PG3W PG3NS PG3L PG3A4 PG3B4 PG3B5	4224 <sup>2</sup>	8.5i 13.5i 11i 8.5i 210mm 257mm 182mm	11i 11i 8.5i 14i 297mm 364mm 257mm	1i	.5i	.5i	6i
PG4A PG4A90 PG4A180 PG4A270 PG4L PG4A4 PG4B5 PG4E	4028	8.5i 11i 8.5i 11i 8.5i 210mm 182mm 184.2mm	11i 8.5i 11i 8.5i 14i 297mm 257mm 266.7mm	1i	.5i	.5i	6i
AFPA AFPA90 AFPA180 AFPA270 AFPL AFPA4 AFPB4 AFPB5 AFPE	3820 or any AFP printer compatible at the data stream level	8.5i 11i 8.5i 11i 8.5i 210mm 257mm 182mm 184.2mm	11i 8.5i 11i 8.5i 14i 297mm 364mm 257mm 266.7mm	1i	.5i	.5i	6i
AFP2A AFP2A90 AFP2A180 AFP2A270 AFP2L AFP2A4 AFP2B4 AFP2B5 AFP2E	3820 <sup>3</sup> or any AFP printer compatible at the data stream level	8.5i 11i 8.5i 11i 8.5i 210mm 257mm 182mm 184.2mm	11i 8.5i 11i 8.5i 14i 297mm 364mm 257mm 266.7mm	1i	.5i	.5i	6i
<b>Note:</b> <sup>1</sup> The DCF physical device type specified is 3820. The actual device can be any AFP printer compatible with the 3820 at the data stream level. <sup>2</sup> The DCF physical device type specified is 4224. The actual device can be any printer compatible with the 4224 at the data stream level. <sup>3</sup> AFP2xxxx devices are the same as AFPxxxx devices except that, by default, they use outline fonts (FONTOLN) and advanced color support.							

Table 5. *SCRIPT/VS Logical Page Devices.* This table lists the logical page devices that can be specified with the *DEVICE* option of the *SCRIPT* command and the default page dimensions for each device. The line spacing for these is determined by the *.LS [Line Spacing]* control word and the fonts used in the document.

## FILE: Name a Disk File for Output

The FILE option directs the formatted output document to a direct-access file. If the DEVICE option is not specified also, the 1403W6 logical device is assumed. The FILE option is specified as

FILE [ (file-id) ]

where *file-id* names the direct-access file. If you do not specify a *file-id*, SCRIPT/VS sends the output document to a default *file-id* based on the environment. For example:

**CMS** In CMS, the *file-id* is of the form:

filename [filetype [filemode] ]

Default names for the output files are assigned based on the requested output device type. The following table lists the default names with their device types.

Device Type	Output File Name
4250x	<i>filename</i> LIST4250
38PPx	<i>filename</i> LIST38PP
3820x, PG1x, PG2x	<i>filename</i> LIST3820
AFPx	<i>filename</i> LISTAFP
AFP2x	<i>filename</i> LISTAFP2
PG3x	<i>filename</i> LIST4224
PG4x	<i>filename</i> LIST4028
PostScript	<i>filename</i> LISTPS
Line devices	<i>\$filename</i> SCRIPT
<b>Note:</b> The <i>filename</i> is the filename of the input file, rather than <i>\$filename</i> SCRIPT.	

These defaults can be changed at installation time by using the OUTFILE physical device keyword or it can be overridden by specifying a filetype with the FILE command option.

If a file with the name specified or implied already exists, SCRIPT/VS issues a message to allow you to let the replacement of the old file occur or to cancel the output.

**TSO:** In a TSO environment, *file-id* is a fully-qualified or partially-qualified data set name. The full name is determined by the following rules:

1. If a fully qualified dsname (placed within quotation marks) is given, the name is used as specified. If the fully qualified dsname is a partitioned data set, a member name must be specified.
2. If a partially-qualified name is provided, it is fully qualified as follows:
  - It is prefixed with the current TSO data set name prefix. This prefix is usually the TSO userid, but may be altered with the TSO PROFILE command.



Lowest Qualifier Provided	Is Fully Qualified As:	Requested Physical Device Type
'TEXT'	'LIST' 'LIST38PP' 'LIST3820' 'LIST4250' 'LIST4028' 'LISTAFP' 'LISTAFP2' 'LIST4224' 'LISTPS'	line devices 3800 Printing Subsystem Model 3 3820 Page Printer 4250 Page Printer LaserPrinter 4028 AFP device AFP2 device 4224 Printer PostScript device
'LIST'	Not changed for a line printer 'LIST38PP' 'LIST3820' 'LIST4028' 'LISTAFP' 'LISTAFP2' 'LIST4250' 'LIST4224' 'LISTPS'	line devices 3800 Printing Subsystem Model 3 3820 Page Printer LaserPrinter 4028 AFP device AFP2 device 4250 Page Printer 4224 Printer PostScript device
Neither 'LIST' nor 'TEXT'	Appended as the lowest qualifier 'LIST' 'LIST38PP' 'LIST3820' 'LIST4250' 'LIST4028' 'LISTAFP' 'LISTAFP2' 'LIST4224' 'LISTPS'	line devices 3800 Printing Subsystem Model 3 3820 Page Printer 4250 Page Printer LaserPrinter 4028 AFP device AFP2 device 4224 Printer PostScript device

If the FILE command option is used with any logical device type, the default rightmost qualifier of the generated output data set name is LISTxxxx. This default can be changed at installation time by using the OUTFILE physical device table keyword or it can be overridden by specifying a fully qualified data set name with the FILE command option.

3. If a file-id is not given, the name of the input data set is examined. If the rightmost qualifier of the input data set is not "TEXT," an error message is generated. In this case, a file-id *must* be given.

For example, for line devices:

File Specification	Input DSNAME	Generated Output DSNAME
FILE('DOC.OUT')	N/A	DOC.OUT
FILE(DOC.OUT)	N/A	userid.DOC.OUT.LIST
FILE(DOC.LIST)	N/A	userid.DOC.LIST
FILE(DOC.TEXT)	N/A	userid.DOC.LIST
FILE((CHAP2))	N/A	userid.LIST(CHAP2)
FILE	'DOC.TEXT'	DOC.LIST
FILE	'DOC.OTHER'	*** error ***

For page devices and PostScript devices:

File Specification	Input DSNAME	Generated Output DSNAME
FILE('DOC.OUT')	N/A	DOC.OUT
FILE(DOC.OUT)	N/A	userid.DOC.OUT.LISTxxxx
FILE(DOC.LIST)	N/A	userid.DOC.LISTxxxx
FILE(DOC.TEXT)	N/A	userid.DOC.LISTxxxx
FILE((CHAP2))	N/A	userid.LISTxxxx(CHAP2)
FILE	'DOC.TEXT'	DOC.LISTxxxx
FILE	'DOC.OTHER'	*** error ***

where xxxx is '38PP', '3820', '4250', '4028', '4224', or 'PS', depending on the device for which the document is being formatted.

If an output data set of the generated name does not exist, SCRIPT/VS creates an output data set with characteristics appropriate to the output device. See Table 28 on page 453 for more details on output data set characteristics for line printers, page printers, and PostScript devices.

When a new member is created in an existing partitioned data set, the existing record format and length are used.

If the output data set already exists, a check is made to ensure that the characteristics of that data set are compatible with the data to be produced. Specifically, if a printer-formatted document is directed to a data set that does not have the machine carriage control record format, or if a terminal-formatted document is directed to a data set that does, the command stops with an error message.

If a document is formatted for a printer and is sent to a direct-access file, the output document has printer controls imbedded in it appropriate for the specified or implied logical output device type. The first record of this file indicates the logical device for which it was formatted and the fonts used. For example, for this release of SCRIPT/VS, this initial record, which has a hexadecimal 03 (Control No Operation) carriage control character, has the following format:

```
SCRIPT/VS R4.0.0: DEVICE device CHARS font1 [... font4]
```

This information can be used to ensure that the document is printed on the same device for which it was formatted. The initial record is ignored when the document is printed.

In CMS, for example, you can use the CMS PRINT command to print the file. You should use the CC option of the CMS PRINT command so that the carriage controls are correctly interpreted. For details on the PRINT command, see *IBM VM/SP: CMS Command and Macro Reference*.

In the ATMS-III environment, the FILE option is not supported.

See Table 28 on page 453 for a description of output file record formats and lengths for the various printer types and operating environments.

**Using the SCRPTFIL DDNAME in TSO:** An alternative method for specifying the direct-access file for the formatted output in TSO is to preallocate the desired direct-access file with a ddname of SCRPTFIL.

If you don't specify the FILE option and SCRIPT/VS finds a ddname of SCRPTFIL allocated, SCRIPT/VS uses this ddname as the direct-access file for the formatted output.

If the FILE command option is not specified and a ddname of SCRPTFIL is not allocated, the existing DCF naming conventions apply.

If you specify the FILE command option and a ddname of SCRPTFIL, SCRIPT/VS uses the FILE option and ignores the SCRPTFIL ddname.

## FONTLIB: Specify a Font Library

The FONTLIB option specifies the font library to be used when formatting for page printers and PostScript devices. For example:

- In CMS, each font description resides in a separate CMS file, and all font description files have a common filetype. The FONTLIB option is specified as

```
FONTLIB [ ( [filetype [filemode]] ) ]
```

where *filetype* is the common file type of all the files; the defaults are

- FONT4250 for the 4250 printer
- FONT38PP for the 3800 Printing Subsystem Model 3
- FONT3820 for the 3820 Page Printer and 4224-2xx Printer
- FONT300 for the LaserPrinter 4028 and the AFP logical devices
- FONTOLN for AFP2 logical devices
- FONTPS for PostScript devices.

where *filemode* identifies the CMS minidisk on which the files reside. If “\*” is specified, the first-accessed CMS minidisk containing a file named DCFINDEX is used. If *filemode* is not specified, the default is “\*.”

If the FONTLIB option is not specified, the first-accessed CMS minidisk containing a file named DCFINDEX is used. Refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for more details on the DCFINDEX member of the font library.

- In TSO, each font description resides in a member of a partitioned data set. The FONTLIB option is specified as

```
FONTLIB ( dsname )
```

where *dsname* is the name of the font library data set; the defaults are

- SYS1.FONT4250 for the 4250 printer
- SYS1.FONT38PP for the 3800 Printing Subsystem Model 3
- SYS1.FONT3820 for the 3820 Page Printer and 4224-2xx printer
- SCRIPT.R40.FONTPS for PostScript devices.
- SYS1.FONT300 for the LaserPrinter 4028 and the AFP logical devices
- SYS1.FONTOLN for AFP2 logical devices

**Using the SCRPTFNT DDNAME in TSO:** An alternative method for specifying font libraries in TSO is to pre-allocate the desired libraries with the ddname of SCRPTFNT. When you have a ddname of SCRPTFNT allocated and you don't specify the FONTLIB command option, SCRIPT/VS searches all of the font libraries concatenated in SCRPTFNT. The libraries are searched in the order specified when you allocated SCRPTFNT.

When you have a ddname of SCRPTFNT allocated, the default font library is not added to the concatenation sequence. If you want the default font library to be searched, you must include it in the allocation of SCRPTFNT.

If you don't specify the FONTLIB command option and a ddname of SCRPTFNT is not allocated, the default font library is used.

If you specify the FONTLIB command option and a ddname of SCRPTFNT, SCRIPT/VS uses the FONTLIB option and ignores the SCRPTFNT ddname.

If you use a concatenated font library, the first DCFINDEX found in the concatenation must accurately represent all of the font members in the concatenation. For more information on creating a DCFINDEX with concatenated font libraries, refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*

- In ATMS-III, font definitions reside in a host system font library. The FONTLIB option is specified as FONTLIB ( ddname )

where *ddname* identifies a DD statement that gives the name of the host system font library; the defaults are:

- FONT4250 for the 4250 printer
- FONT38PP for the 3800 Printing Subsystem Model 3
- FONT3820 for the 3820 Page Printer, and 4224 Printer
- FONT300 for the LaserPrinter 4028 and the Generic AFP logical devices
- FONTOLN for AFP2 logical devices.

**Note:** ATMS-III does not support PostScript.

- When you run a batch MVS job with DLF, font definitions reside in a host system font library. The FONTLIB option is specified as

FONTLIB ( ddname )

where *ddname* identifies a DD statement that gives the name of the host system font library; the defaults are:

- FONT4250 for the 4250 printer
- FONT38PP for the 3800 Printing Subsystem Model 3
- FONT3820 for the 3820 Page Printer and 4224 Printer
- FONT300 for the LaserPrinter 4028 and the Generic AFP logical devices
- FONTOLN for AFP2 logical devices
- FONTPS for PostScript devices.

- When you run a batch VSE job with DLF, font definitions reside in a host system font library. The FONTLIB option is specified as

FONTLIB ( dlblname )

where *dlblname* identifies a DLBL statement that gives the name of the host system font library; the defaults are:

- FONT4250 for the 4250 printer
- FONT3820 for the 3820 Page Printer and 4224 Printer
- FONT300 for the LaserPrinter 4028 and the Generic AFP logical devices
- FONTOLN for AFP2 logical devices
- FONTPS for PostScript devices.

In all environments, it is the user's responsibility to ensure that the font library used in printing a document is the same one used during formatting.

**Note:** You must have the PostScript DCFINDEX file, the PostScript codepage files, and the PostScript AFM (Adobe Font Metrics) files in the requested FONTLIB to produce PostScript output.

## FPASSES: Specify Number of Formatting Passes

The FPASSES option specifies the number of formatting passes SCRIPT/VS performs when processing an input file<sup>3</sup>. These passes process all control words, but output occurs only after the last pass. FPASSES and TWOPASS are mutually exclusive, therefore the last one specified is the one that takes effect. If FPASSES or TWOPASS is not specified, SCRIPT/VS formats and prints everything in one pass. The format of the FPASSES option is

FPASSES(*n*)

where *n* is the number of formatting passes SCRIPT/VS performs. If FPASSES(2) is specified, the result is the same as TWOPASS. The &\$PSNO symbol indicates the number of formatting passes specified. The &\$PASS symbol indicates the number of the current formatting pass. The IBM-supplied maximum number of passes is 4, unless changed at installation time. The *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* describes how to change the maximum number of formatting passes SCRIPT/VS performs.

Use FPASSES if page number references are not resolved correctly after formatting the document with TWOPASS. See “TWOPASS: Prepare the Document with Two Formatting Passes” on page 47 for information about the TWOPASS option.

## INDEX: Enable Back of Book Index

The INDEX option enables the .PI [Put Index] control word. Information about using the .PI [Put Index] control word to create an index is described in “.PI [Put Index]” on page 279 and in the section on indexing in the *Document Composition Facility: SCRIPT/VS User's Guide*.

## LIB: Specify Symbol and Macro Libraries

The LIB option is valid in the CMS, TSO, and ATMS-III environments, and it specifies which libraries SCRIPT/VS is allowed to search for a definition of the symbols and macros not defined within the input file. See the chapter entitled “Processing Macros” in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for guidelines and restrictions about the use of macro libraries. In a DLF environment, the SEARCH option provides a similar facility. For example:

- In CMS, the LIB option is specified as

```
LIB (libname1 [ ... libname8 ] )
```

where *libname* is the file name of a CMS macro library. The file type is MACLIB. The CMS search sequence (alphabetic) is used to locate the library on any accessed disk.

- In TSO, the LIB option is specified as:

```
LIB (libname)
```

If the *libname* given is not fully qualified (placed within quotation marks), the userid is prefixed to the *libname* as the leftmost qualifier, and MACLIB is added (unless it already appears) as the rightmost qualifier. For a more detailed explanation of TSO data set qualification and naming conventions, see “MVS/TSO Data Set Naming Conventions” on page 12.

- In ATMS-III, the LIB option is specified as

---

<sup>3</sup> Use of this option significantly increases the processing time for your document.

LIB (opnum1 [ ... opnum8 ] )

where *opnum* is an operator number. It must include the user's number if the user's permanent storage is to be searched.

The library is searched when an undefined symbol or macro is encountered and a .LY ON, a .LY SYM, or a .LY MAC control word has been specified. The library is also searched (without regard to the setting of the .LY control word) when a symbol or macro is defined with the LIB parameter. For example:

```
.se symbolname LIB
```

```
.dm macroname LIB
```

You can specify up to eight library names in CMS, one name in TSO (although multiple libraries can be concatenated by preallocating a ddname of SCRPTLIB), or eight operator numbers in ATMS-III. If the symbol name or macro name is not defined (but the symbol or macro is defined as being in a library), SCRIPT/VS scans each library named in the LIB option (in the order given) until the symbol or macro is found. SCRIPT/VS then moves the symbol or macro definition into its symbol table, so that a second occurrence does not require a library search. If no library option is specified, the default library (if it exists) is searched for the symbol or macro. For example:

- In CMS, a symbol and macro library is a standard MACLIB file. Its file type is MACLIB and the default library is DSMGML4 MACLIB, unless changed at your installation.
- In TSO, a symbol and macro library is a partitioned data set. The default library, unless changed at your installation, is SCRIPT.R40.MACLIB. If a library is supplied with the LIB option, the specified library is linked to, and searched before, the default library.

An alternative method for specifying symbol and macro libraries in TSO is to preallocate the desired libraries with a ddname of SCRPTLIB. If this ddname is found, SCRIPT/VS searches all of the libraries linked to this ddname. The libraries are searched in the order specified when the ddname was allocated.

When the SCRPTLIB ddname is used, the default symbol and macro library is not added to the link sequence. If a library is to be searched, it must be included in the original allocation for SCRPTLIB.

**Note:** Even though these data sets are preallocated, SCRIPT/VS requires that they be cataloged.

If the LIB option is not specified and a ddname of SCRPTLIB is not allocated, the default library is used.

If the LIB option is specified and a SCRPTLIB ddname is also allocated, SCRIPT/VS uses the LIB option and ignores the SCRPTLIB ddname.

- In ATMS-III (unless changed at your installation), the LIB search is used only if the requested source cannot be located through the use of ATMS subdocument index build/connect facilities. Furthermore, the search is performed only against the permanent storage of the users whose operator numbers are specified in the LIB list. Also, if the LIB search is used and the located source does not belong to the requesting user, the document must have a getword of *any*.

If a LIB option is not specified, ATMS-III uses only its subdocument facilities to search for unresolved symbols and macros.

## MESSAGE: Control Message Printing

The MESSAGE option controls the amount and timing of the information SCRIPT/VS provides with error messages. If the MESSAGE option is not specified, SCRIPT/VS provides a short message that includes the message text and the line number and text of the input last read when the error was detected (when appropriate). The MESSAGE option is specified as:

```
MESSAGE ( [DELAY] [ID] [TRACE] )
```

You must specify at least one parameter with the MESSAGE option; you can specify two or all three parameters, separated by blanks. Each of the options can be abbreviated as a single letter.

DELAY requests that SCRIPT/VS not display messages while a document is being displayed or printed. SCRIPT/VS accumulates messages in a utility file (DSMUTMSG) and appends them to the end of the formatted output. DELAY is always used in ATMS-III.

ID causes SCRIPT/VS to include the error message identifier along with the error message.

TRACE causes SCRIPT/VS to list, whenever appropriate, the sequence of imbedded files, from the file that includes the next line after the error input line, backward to the primary input file. This is useful when a file is imbedded in many other files.

**Note:** CMS truncates messages that exceed 130 characters.

The MESSAGE option should always be specified before any other command option.

## **NOCONT: Ending Processing after an Error Occurs**

The NOCONT option ends processing after SCRIPT/VS encounters an error condition and issues an error message. This is the default.

If the CONTINUE option of the SCRIPT command is specified, SCRIPT/VS continues processing unless a severe or terminal error is encountered.

For a description of error types and SCRIPT/VS error messages, see the *Document Composition Facility: Messages* manual distributed with SCRIPT/VS.

## **NODDUT: Disable SCRIPT/VS Utility File Redefinition**

The NODDUT option disallows the redefinition of SCRIPT/VS utility files to non-utility file names using the .DD [Define Data File-id] control word. This is the default. See the descriptions of the .DD control word for more information.

## **NOPROF: Suppress the Profile**

The NOPROF option requests that SCRIPT/VS not imbed a profile document. For details about the profile, see the PROFILE option's description below.

## **NOSEGLIB: Ignore the Segment Library**

The NOSEGLIB option is ignored for line printers and PostScript devices.

The NOSEGLIB option overrides the SEGLIB option and specifies that no host system segment library should be accessed.

When formatting documents for a page printer, .SI [Segment Include] control words are processed just as if the segments they reference were not found in the segment library, but no error results.

## NOSPIE: Prevent Entering SPIE Exit Routines

The NOSPIE option is valid only in CMS and TSO.

The NOSPIE option requests that SCRIPT/VS not establish a program interrupt exit. The NOSPIE option is intended for use by the system programmer maintaining SCRIPT/VS.

## NOSORCDD: Specifying the Primary Input File as a Data Set Name

The NOSORCDD option overrides the SORCDD option and **explicitly** specifies that the primary input file is a data set name. The NOSORCDD option is valid only in TSO.

## NOWAIT: Prevent Prompting for Paper Adjustment

The NOWAIT option is valid only in CMS and TSO.

The NOWAIT option causes SCRIPT/VS to send output to your terminal without first prompting you to adjust the paper. NOWAIT option is the normal mode for output to all output devices except a typewriter terminal.

## NUMBER: Print the File Name and Line Number

The NUMBER option is ignored for page printers and PostScript devices.

On line devices, the NUMBER option causes SCRIPT/VS to print the file-id and line number of the last line read when a formatted output line is printed. The formatter reads ahead, and because the number printed is that of the last input line read, the number printed may or may not correspond to the input line number of the text printed in the output line. The file-id and line number are printed to the right of the formatted output line, and each can be no more than 8 characters long.

## OPTIONS: Name a File That Contains Options

The OPTIONS option allows you to specify a file that contains additional options to control processing and formatting of a document. The options in the file are in addition to options you specify with the SCRIPT command and with other options files. The OPTIONS option is specified as:

```
OPTIONS [ (file-id) ]
```

In CMS, if the *file-id* is not specified, the default *file-id* is SCRIPT OPTIONS; if only a file name is given, the default file type is OPTIONS.

In ATMS-III, the user can specify OPTIONS('docname:opnum;getw'). If *opnum* or *getw* is not specified, ATMS-III attempts to locate the source name using its subdocument index build and connect facilities. If it cannot be located as a subdocument, ATMS-III attempts to locate it as an explicit document in the permanent storage of the requesting user. If a qualified source name is used, ATMS-III uses only an explicit search for the document.

Each record in the options file can contain one or more options in the same format as they would appear on the SCRIPT command line. They must, however, be in uppercase. An option need not be completed on a single line (suboptions can appear on following lines), but each word must be completed in a single record. A left parenthesis must not precede the options in the file.



The options in the file are processed as though they replace the OPTIONS option. Consequently, the OPTIONS option in one option file can refer to another option file. Option files can be chained in this manner.

The OPTIONS option is valid only in CMS and ATMS-III.

## **PAGE: Selectively Print Pages**

The PAGE option allows you to print pages of formatted output selectively. The page number need not be an integer; you can use the .PN [Page Numbering Mode] control word to establish decimal, alphabetic, and roman numeral page numbers, and you can attach a prefix to each page number. The first 8 characters of the page number you specify with the PAGE option is the character string that SCRIPT/VS compares to the current page number symbol.

The PAGE option has several formats, and any number of page-range specifications can be included in the PAGE option. Note, however, that prompting mode *replaces* the remainder of the suboptions. Valid forms of page-range specifications are:

PROMPT

[FROM] frompage [TO] topage

[FROM] frompage FOR *n*

[FROM] page ONLY

If no parameter is given with the PAGE option, PAGE (PROMPT) is assumed (except under ATMS-III and DLF). Page prompting, implied or explicit, is not supported in ATMS-III and DLF environments and results in an error message and the end of SCRIPT/VS processing. All subsequent references to page prompting here apply only to CMS and TSO foreground environments.

The following are examples of valid, explicit, page-option specifications:

PAGE (FROM 10 TO 15)

PAGE (7 8)

PAGE (2 TO 5)

PAGE (viii ONLY 93 TO \*)

PAGE (FROM 94.1 FOR 3 99 ONLY)

An asterisk (\*) specified as *frompage* is interpreted as the current page; an asterisk specified as *topage* means the last page in the document.

If you specify or imply the PROMPT option, SCRIPT/VS prompts you to enter page-range specifications from your terminal. You can respond with any of the forms described above. SCRIPT/VS continues to prompt you for new page-range specifications until the end of the document is reached or you indicate an end to prompting mode by entering a null line. If you respond by entering only a single value, that value is used as the *frompage* value and the last page in the document is assumed as the *topage* value.

If there is a syntax error in your page-range specification, SCRIPT/VS issues an error message and begins prompting.

The page numbers must be entered in the same order as they appear in the output document. For example, you can specify

PAGE (6 1)

but it is meaningful only if there is, at some point following page 6, a .PN 1 or .PA 1 control word that resets the page counter to 1.

If there is no page with the number given or if SCRIPT/VS has passed the specified page, SCRIPT/VS reaches the end of the document without changing from not printing to printing, or vice versa.

## PRINT: Produce Printer Output

The PRINT option causes SCRIPT/VS to send the output document to a printer. If the DEVICE option is not specified, SCRIPT/VS assumes the 1403W6 logical device type.

In CMS, the number of copies and the output class are controlled by the CP SPOOL command and the CP CHANGE command.

In TSO, you can control the disposition of the printed output by specifying the following positional parameters with the print option:

PRINT (copies,class,fcbl,ucs)

*copies* is the number of copies desired, and defaults to "1." *class* is the SYSOUT class, and defaults to "A" when the UPCASE option is specified and "T" when it is not. *fcbl* is the forms control buffer name. *ucs* is the universal character set name. Your installation determines the appropriate values for *class*, *fcbl*, and *ucs*. For more information, see the *OS/VS2 MVS JCL* manual.

If SYSOUT parameters such as CHARS, FLASH, and OPTCD are desired, you should direct your formatted output to a file and submit a job to print the contents of the file, specifying the desired JCL parameters. Under JES2, this may also be necessary if the default LINECT value is not zero. (JES2 may insert extra page ejects into your document when it is printed.)

In ATMS-III, the number of copies and output class are controlled by the peripheral queue. When you define an ATMS-III peripheral queue for SCRIPT/VS, you must specify a CICS/VS destination ID that is consistent with the queue device type.

The PRINT option is ignored for the 4250 Page Printer and PostScript logical devices.

For more information on obtaining printed output, see the *Document Composition Facility: SCRIPT/VS User's Guide*.

## PROFILE: Specify a Profile

A profile is a SCRIPT input file that is imbedded before processing begins on the primary input file.

The PROFILE option names the file that SCRIPT/VS is to use as the profile for the document being formatted. A profile can contain frequently used symbol and macro definitions, GML application-processing functions, and text appropriate for many documents (for example, running headings and footings).

For more details about profiles, see the section on GML processing in the *Document Composition Facility: Text Programmer's Guide*.

The PROFILE option is specified as:

PROFILE [ (file-id) ]

where *file-id* names the profile. You can select different profiles to use when formatting the document for different applications.

If the PROFILE option is not specified or if *file-id* is not specified, SCRIPT/VS searches your files for a file named PROFILE.

In CMS, the default is:

PROFILE SCRIPT

In TSO, the default is:

'userid.PROFILE.TEXT'

**Using the SCRPTPRO DDNAME in TSO:** An alternative method for specifying the profile that you want SCRIPT/VS to use for processing in TSO is to pre-allocate the desired profile with the ddname of SCRPTPRO.

If you do not specify the PROFILE option and SCRIPT/VS finds a ddname of SCRPTPRO allocated, SCRIPT/VS uses this ddname to locate the associated profile.

If the PROFILE command option is not specified and a ddname of SCRPTPRO is not allocated, SCRIPT/VS searches the default profile.

If you specify the PROFILE command option and a ddname of SCRPTPRO, SCRIPT/VS uses the PROFILE option and ignores the SCRPTPRO ddname.

**Note:** IEC141I 013-18 ABEND occurs if the member name specified on the DSNNAME parameter could not be found.

In ATMS-III, the default is:

PROFILE

The PROFILE option also provides an epifile facility through the use of the .EF [End of File] control word. If End of File is indicated with the .EF control word within the PROFILE file, SCRIPT/VS proceeds immediately to the primary input file; the remainder of the PROFILE file is processed *after* the primary input file has been completed.

The .EF control word is described in “.EF [End of File]” on page 163.

## PSOUT: Specify ASCII or EBCDIC PostScript Output

The PSOUT option of the SCRIPT command allows you to specify whether SCRIPT/VS generates PostScript output in ASCII or EBCDIC when a PostScript device is specified with the DEVICE option. If a PostScript device is specified and PSOUT is *not* specified, an ASCII file is produced.

If PSOUT(ASCII) is specified, SCRIPT/VS produces an ASCII file. You can then download the file to an IBM PC and print the file on a locally attached PostScript device, such as the IBM 4019 LaserPrinter.

If PSOUT(EBCDIC) is specified, SCRIPT/VS produces an EBCDIC file. You can view the EBCDIC file on the host system, but you cannot download the file and print it.

You can query the &\$ASCI system symbol to determine whether ASCII or EBCDIC was specified on the SCRIPT command. &\$ASCI returns a 1 if ASCII output is produced and a 0 if EBCDIC output is produced.

This command option is ignored for all devices except PostScript devices.

## QUIET: Suppress the Formatter's Identifier Message

The QUIET option causes SCRIPT/VS to suppress (not display) the version identification message that is otherwise typed or displayed as a response to the SCRIPT command.

## SEARCH: Specify a Library

The SEARCH option causes SCRIPT/VS to search the specified library or partitioned data set for imbedded or appended files. In DLF/MVS and DLF/VSE, SCRIPT/VS also uses the library to locate symbols and macros not defined within the input file. In CMS, SCRIPT/VS uses the filetype specified with the SEARCH option as the default filetype for imbeds, appends, and .DD files. (For more details on libraries, see the LIB option.) The same guidelines and restrictions that apply to macro libraries also apply to search libraries. Refer to the chapter entitled “Processing Macros” in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for a description of these guidelines and restrictions. For example:

- In CMS, the SEARCH option is specified as:

```
SEARCH (filetype)
```

If SEARCH (=) is specified, the file type of the primary input file is used.

- In TSO, the SEARCH option is specified as:

```
SEARCH (libname)
```

The specified library name is used as described in “MVS/TSO Data Set Naming Conventions” on page 12.

In TSO, imbedded and appended files are qualified as follows:

- If the name of a partitioned data set is specified with the SEARCH option,  
    'userid.searchname.TEXT'  
    the library is searched for a member of the given name.
- If the TEXTLIB *ddname* has been allocated for the TSO session, the allocated libraries are searched for a member of the given name. The same restrictions that apply to search libraries also apply to macro libraries. Refer to the section on SCRIPT/VS macro libraries in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for more information.

**Note:** By using the TEXTLIB *ddname* instead of the SEARCH option it is possible to concatenate several partitioned datasets. The use of a TEXTLIB concatenation can cause unpredictable results if the .DD [Define Data File-id] control word is used with the LIB keyword to redefine members of the concatenation.

- If the SEARCH option is not specified, if no TEXTLIB exists, and if the primary input file is a partitioned data set, that library is searched for a member of the given name.
- If the SEARCH option is not specified, if no TEXTLIB exists, and if the primary input file is not a partitioned data set, that library 'userid.TEXT' is searched for a member of the given name.

- In ATMS-III, the SEARCH option is specified as:

```
SEARCH (opnum1 [ ... opnum8 ] )
```

Up to eight ATMS-III user operator numbers can be specified as an alternative source for imbedded documents.

If an imbedded document with an unqualified name cannot be located as a subdocument or as an explicit document in the permanent storage of the requesting user, ATMS-III searches, in the specified

order, the permanent storage of the operators specified. If the located documents are not in the operator's permanent storage, they must have a getword of "any."

## SEGLIB: Specify a Segment Library

The SEGLIB option specifies the segment library to be used when formatting for page printers. For example:

- In CMS, each segment resides in a separate CMS file, and all segment files have a common filetype. The SEGLIB option is specified as

```
SEGLIB [ ( [filetype [filemode]] ) ]
```

where *filetype* gives the common filetype of all segment description files. If *filetype* is not specified, the defaults are:

**PSEG4250** For the 4250 printer

**PSEG38PP** For the 3800 Printing Subsystem Model 3

**PSEG3820** All AFP printers except the 3800 Printing Subsystem Model 3.

**Note:** Even though the defaults given above indicate that segments reside in different segment libraries for the 3800 Printing Subsystem3. and the 3820 Page Printer, such segments may, in fact, share a common segment library. This is true for all environments that support segment libraries.

*filemode* identifies the CMS minidisk on which the segment description files reside. If *filemode* is not specified, the default is "\*."

- In TSO, each segment resides in a member of a partitioned data set. The SEGLIB option is specified as

```
SEGLIB ( dsname )
```

where *dsname* is the name of the segment library dataset. The defaults are:

**SYS1.PSEG4250** For a 4250 printer

**SYS1.PSEG38PP** For a 3800 Printing Subsystem Model 3

**SYS1.PSEG3820** All AFP printers except the 3800 Printing Subsystem Model 3.

**Using the SCRPTSEG DDNAME in TSO:** An alternative method for specifying segment libraries in TSO is to pre-allocate the desired libraries with a ddname of SCRPTSEG. When you have a ddname of SCRPTSEG allocated and you don't specify the SEGLIB command option, SCRIPT/VS searches all of the segment libraries concatenated in SCRPTSEG. The libraries are searched in the order specified when you allocated SCRPTSEG.

When you have a ddname of SCRPTSEG allocated, the default segment library is not added to the concatenation sequence. If you want the default segment library to be searched, you must include it in the allocation of SCRPTSEG.

If you don't specify the SEGLIB command option and a ddname of SCRPTSEG is not allocated, the default segment library is used.

If you specify the SEGLIB command option and a ddname of SCRPTSEG, SCRIPT/VS uses the SEGLIB option and ignores the SCRPTSEG ddname.

- In ATMS-III, segments reside in a host system segment library. The SEGLIB option is specified as

```
SEGLIB ( ddname )
```

where *ddname* identifies a DD statement that gives the name of the host system segment library. The default is NOSEGLIB for page printers.

- In batch DLF on MVS, segments reside in a host system segment library. The SEGLIB option is specified as

```
SEGLIB ( ddname )
```

where *ddname* identifies a DD statement that gives the name of the host system segment library. The defaults are:

**SEGLIB(PSEG4250)** For a 4250 printer

**SEGLIB(PSEG38PP)** For a 3800 Printing Subsystem Model 3

**SEGLIB(PSEG3820)** All AFP printers except the 3800 Printing Subsystem Model 3.

- In batch DLF on VSE, segments are not supported.

In all environments, it is the user's responsibility to ensure that the segment library used in printing a document is the same one used during formatting.

The SEGLIB option is ignored for line printers and PostScript devices.

## SEPMASR: Specify Output Separation Masters

Separation masters are used in the printing process to produce multi-part forms or multiple-color documents.

The SEPMASR option allows you to separate a page into multiple masters that can then be used in an offset printing process to produce color documents or as multi-part forms. Each output page is produced *n* times where *n* is the number of masters selected for that page. The *n* masters specified with the SEPMASR command option correspond to the numbers given on the .SM [Separation Master] control word. The SEPMASR command option is specified as follows:

```
SEPMASR(n0 ... n16)
```

where *n0* through *n16* represent the specified suboptions.

The SEPMASR command option accepts up to seventeen suboptions for selecting masters. If SEPMASR is specified with no suboptions, all of the separation masters defined with the .SM separation master control word are produced, including the default master.

To help you sort the pages for the masters, a message is issued for each logical output page. The message gives the page number, the values of the masters selected for that page, and the order in which they were produced. MESSAGE(DELAY) should be used with the SEPMASR command option so that the messages are retained after the document is formatted.

For more information about the .SM control word, see the “.SM [Separation Master]” on page 353. For more information about creating color separation masters, refer to the *Document Composition Facility: Text Programmer's Guide*.

## SORCDD: Process the Primary Input File as a DDNAME

An alternative method for specifying the primary input file is to use a pre-allocated ddname. To do this, pre-allocate the ddname and use it instead of the *file-id* on the SCRIPT command, followed by the new command option SORCDD. The SORCDD command option notifies SCRIPT to process the primary input file as a ddname.

You can preallocate a ddname and then use the SORCDD command option as shown in the following example:

- Allocate the primary input data set with the TSO ALLOCATE command:

```
ALLOC DD(ABCD) DSN('userid.DATA.TEXT(FILEIN)') SHR REUSE
```

- To use the ddname ABCD as the primary input file, you need to use the new command option SORCDD:

```
SCRIPT ABCD NOPROF CONT TWO SORCDD
```

**Note:** IEC141I 013-18 ABEND occurs if the member name specified on the DSNAME parameter could not be found.

## **SPELLCHK: Enable the .SV Control Word**

SPELLCHK is performed as each source line is read from the source file prior to any formatting. The SPELLCHK option causes SCRIPT/VS to verify spelling. Each word is verified using the spelling and hyphenation dictionaries specified with the .DL [Dictionary List] control word and the TLIB option of the SCRIPT command, unless spelling verification has been turned off with the .SV [Spelling Verification] control word. Spelling errors are listed, using the .UW [Unverified Word] control word, with other errors found during formatting.

The .UW control word is described in “.UW [Unverified Word]” on page 422.

## **STOP: Print Separate Pages at the Terminal**

The STOP option causes SCRIPT/VS to wait for you to press the return key before starting to type each page. Use this option when printing your output document on separate sheets of paper at a typewriter terminal.

When SCRIPT/VS stops after the first page, no message is issued. SCRIPT/VS unlocks your keyboard and is ready to type the first line of the next page. Position the top edge of the paper in the typewriter terminal and press RETURN. SCRIPT/VS resumes typing.

The STOP option is valid only for typewriter-like terminals in CMS and TSO.

## **SYOFF: Disable .SY [System Command] Control Word**

The SYOFF option, which is the default, disables the .SY [System Command] control word.<sup>4</sup> SYOFF is applicable only in the TSO and CMS environments.

## **SYON: Enable .SY [System Command] Control Word**

The SYON option enables the .SY [System Command] control word.<sup>4</sup> SYON is applicable only in the TSO and CMS environments.

## **SYSVAR: Set System Variable Symbols**

The SYSVAR option allows you to pass information to SCRIPT/VS as symbols defined when you issue the SCRIPT command. The SYSVAR option is specified as:

```
SYSVAR (x value ... x value)
```

Each *x value* pair causes the symbol &SYSVARx to be set to *value*. *x* is any alphanumeric character identifying the token. *Value* is any alphanumeric string of up to 8 characters; it cannot contain imbedded

---

<sup>4</sup> The .SY [System Command] control word is an obsolete control word. Refer to the *Document Composition Facility: Diagnosis Guide and Reference* for more information.

blanks or parentheses. Because both *x* and *value* are part of the SCRIPT statement, any lowercase characters you specify are converted to uppercase.

The maximum number of *x value* pairs is limited only by the length of the SCRIPT command line.

For example, your input file might include the lines

```
.in &SYSVARA  
.ll &SYSVARL
```

When you issue the SCRIPT command to format your document, you can specify values for indention and line width as:

```
SCRIPT ... SYSVAR (A 10 L 72)
```

The symbols on the input line are substituted for the values set by the SYSVAR option. The input lines shown above are processed by SCRIPT/VS as though they had been:

```
.in 10  
.ll 72
```

**Note:** The symbols created by the SYSVAR option are always uppercase.

- | **Note:** For more information about the SYSVAR option, see the *Document Composition Facility:*
- | *SCRIPT/VS Starter Set Users Guide.*

## TERM: Display the Output at the User's Terminal

The TERM option causes SCRIPT/VS to send the output document to your terminal. If the DEVICE option is not specified, SCRIPT/VS assumes the logical device TERM and displays the document on your terminal.

TERM is the default destination in CMS and TSO; ATMS-III forces TERM when SCRIPT/VS is called from a terminal.

## TLIB: Specify Spelling Verification and Hyphenation Libraries

The TLIB option specifies text libraries that contain IBM-supplied root word dictionaries, user-created root word dictionaries, and stem-processing routines for use in spelling verification and hyphenation. The TLIB option is specified as

```
TLIB ( libname1 [ ... libname8 ] )
```

where *libname* is the name of a CMS text library. The filetype is TXTLIB. The CMS search sequence is used to locate the library on any accessed disk. For example:

- The specified libraries are searched when a dictionary not included as part of the SCRIPT/VS load module is named in the .DL [Dictionary List] control word. Both the dictionary and stem-processing routines are loaded from the libraries.
- If the TLIB option is not specified, the library searched is DSMHYLIB TXTLIB.
- If the dictionaries used are included as part of the SCRIPT/VS load module when it is created, no library is needed.
- The default text library (DSMHYLIB) contains algorithmic hyphenation dictionaries for these languages: Danish, Dutch, American English, Canadian English, United Kingdom English, Finnish, Canadian French, French National, German, Icelandic, Italian, Norwegian, Portuguese, Spanish, and Swedish. It also contains spelling verification dictionaries for these languages but the following dictionaries are empty: Danish, Finnish, Icelandic, Norwegian, Portuguese, and Swedish. However, you can create



user dictionaries for spelling verification for these languages using the Dictionary Maintenance Program supplied with DCF.

**Note:** Refer to the *Document Composition Facility: Text Programmer's Guide* for more information on algorithmic hyphenation.

The TLIB option is valid only in CMS.

## TWOPASS: Prepare the Document with Two Formatting Passes

The TWOPASS option causes SCRIPT/VS to process the input file in two passes. Both passes process all control words, but output occurs only on the second pass. TWOPASS and FPASSES are mutually exclusive, therefore the last one specified is the one that takes effect. Unless you specify TWOPASS or FPASSES, SCRIPT/VS formats and outputs everything in one pass.

Two formatting passes are required when a symbolic value is needed earlier in the document than when it is set; for example, a page number in a table of contents or list of figures at the beginning of a document. The first formatting pass allows SCRIPT/VS to collect head-levels and corresponding page numbers. The second formatting pass, which produces output, places the page numbers from the first pass into the table of contents.

You can produce an accurate table of contents with a single formatting pass by having SCRIPT/VS prepare it at the end of the output document. To get a table of contents this way, put the control word that formats the table of contents at the very end of your input document. To obtain correct pagination for your table of contents, reset the page number (specifying the roman numeral iii, for example) in your input file just before the table of contents control word. After your formatted document has been printed, you can physically move the table of contents pages to the front of the document.

You can also use the TWOPASS option to detect errors in an input file. If you process a document with TWOPASS and without CONTINUE, the second pass does not begin unless the first pass has completed.

If TWOPASS is used while processing a file that uses .TE [Terminal Input], text entered as a result of .TE on the first pass is excluded from the formatted output. Text entered during the second pass, however, is formatted. You can use the TWOPASS symbol, &\$TWO (which is equal to 1 if TWOPASS was specified), and the .IF [If] control word to skip the .TE on the first pass. The &\$PASS symbol indicates the current pass number as 1 or 2. Refer to the description of the .IF [If] family of control words in the logical processing section of the *Document Composition Facility: Text Programmer's Guide*.

**Note:** SCRIPT/VS symbol and conditional-processing functions can cause an input file to appear different on the second pass from the first pass. As a result, page numbers might not be accurate in the table of contents or in other cross-references. If you specified TWOPASS and the table of contents is placed at the front of the document, it reflects the page numbers on the first pass. If it is placed at the end of the document, it reflects the page numbers on the second pass. If the FPASSES option was used and more than two formatting passes were specified, cross-references should be resolved accurately. See "FPASSES: Specify Number of Formatting Passes" on page 35 for information on the FPASSES option.

## UNFORMAT: Print All Input Lines without Formatting

The UNFORMAT option generally lists only those input lines that are processed by SCRIPT/VS. For example, input lines that are bypassed because of a .GO [Goto] control word or a .CS [Conditional Section] control word are not shown in the unformatted listing.

Running heading (.RH), running footing (.RF), footnote leader (.FN LEADER), and define macro (.DM) definitions, are listed when they are defined rather than when they are executed. If the default definitions are used, the definitions are not listed. Macro definitions within a macro library are not listed.

Some lines not in the primary input file might be printed. When SCRIPT/VS encounters the .IM [Imbed] or .AP [Append] control words, the contents of the imbedded or appended file are included following the control word. In the unformatted listing, SCRIPT/VS puts the line

```
. *====> IMBED/APPEND FILE: file-id
```

at the beginning of each imbedded or appended file. SCRIPT/VS puts the following line after the last line of an imbedded or appended file:

```
. *<=== END OF FILE: file-id
```

If the NUMBER option is used with UNFORMAT, the file-id and line number are printed on the left instead of the right.

Changes in input caused by responses to the STEP parameter of the .IT [Input Trace] control word are not shown.

The UNFORMAT option is ignored for page printers and PostScript devices.

## **UPCASE: Print Lowercase Letters as Uppercase**

The UPCASE option causes SCRIPT/VS to convert all lowercase letters to uppercase, but only for the formatted output. This option should be specified when the output is directed to a printer that cannot print lowercase letters.

## **@user-option: User-defined Options**

In CMS, additional options can be defined as needed by prefixing the new option with the character @. For example:

```
@duplex
```

User-options can also have suboptions. For example:

```
@columns (2)
```

User options are saved with their suboptions, if any, but without the @ delimiter or the parentheses surrounding the suboptions, and can be executed or extracted at any time with the .GS [GML Services] control word. Refer to the description of GML processing in the *Document Composition Facility: Text Programmer's Guide* for further details.

## Chapter 4. SCRIPT/VS Control Words

<b>Chapter 4. SCRIPT/VS Control Words</b>	53
Control Word Syntax	53
Control Word Separator	53
Macros	54
The Control Word Modifier	54
The Input Line	55
Notational Conventions	55
Type 1 Control Words	55
Reading Syntax Diagrams	56
Style Rules	56
Symbols	57
Required Parameters	57
Optional Parameters	57
Repeating Parameters	58
Fragments	58
Space Units	58
SCRIPT/VS Control Word Descriptions	59
... [Set Label]	60
.AA [Associate APF]	62
.AM [Adjust Margin]	64
.AN [And]	66
.AP [Append]	68
.AR [Area]	70
.BC [Balance Columns]	74
.BF [Begin Font]	75
.BL [Blank Line]	78
.BM [Bottom Margin]	79
.BR [Break]	80
.BX [Box]	81
.CB [Column Begin]	91
.CC [Conditional Column Begin]	92
.CD [Column Definition]	93
.CE [Center]	95
.CG [Copy Group]	97
.CL [Column Line Length]	99
.CM [Comment]	100
.CP [Conditional Page Eject]	101
.CR [Color]	102
.CS [Conditional Section]	105
.CT [Continued Text]	107
.DA [Define Area]	108
.DC [Define Character]	112
.DD [Define Data File-Id] in the CMS Environment	120
.DD [Define Data File-id] in the MVS/TSO Environment	122
.DD [Define Data File-Id] in the ATMS Environment	125
.DD [Define Data File-Id] in the DLF Environment	127
.DF [Define Font]	130
.DH [Define Head Level]	136
.DL [Dictionary List]	140
.DM [Define Macro]	142

	.DO [Define Object]	147
	.DR [Define Rule]	151
	.DS [Double Space Mode]	154
	.DU [Dictionary Update]	155
	.DV [Define Variable]	157
	.EC [Execute Control]	161
	.EF [End of File]	163
	.EL [Else]	164
	.EM [Execute Macro]	166
	.ES [Extra Space]	167
	.EQ [Equation]	169
	.FL [Float]	170
	.FN [Footnote]	172
	.FO [Format Mode]	174
	.FV [Format Vertically]	177
	.GO [Goto]	179
	.GR [Group]	181
	.GS [GML Services]	183
	.H0-.H6 [Head Levels 0–6]	190
	.HR [Horizontal Rule]	192
	.HW [Hyphenate Word]	195
	.HY [Hyphenate]	196
	.IC [Intercharacter Space]	201
	.IE [Index Entry]	202
	.IF [If]	203
	.IL [Indent Line]	206
	.IM [Imbed]	208
	.IN [Indent]	210
	.IO [Include Object]	213
	.IR [Indent Right]	217
	.IS [Inline Space]	219
	.IT [Input Trace]	221
	.IX [Index]	224
	.JU [Justify Mode]	226
	.KP [Keep]	227
	.LB [Leading Blank]	230
	.LI [Literal]	231
	.LL [Line Length]	233
	.LO [Lead-Out]	234
	.LS [Line Spacing]	235
	.LT [Leading Tab]	242
	.LY [Library]	243
	.MC [Multicolumn Mode]	244
	.ME [Macro Exit]	245
	.MG [Message]	246
	.MS [Macro Substitution]	248
	.NF [No Formatting]	249
	.NL [Null Line]	252
	.NV [Navigate]	253
	.OC [Output Comment]	258
	.OF [Offset]	260
	.OI [Overlay Include]	262
	.OR [Or]	274
	.PA [Page Eject]	276

.PF [Previous Font]	278
.PI [Put Index]	279
.PL [Page Length]	282
.PM [Page Margins]	284
.PN [Page Numbering Mode]	286
.PO [PostScript]	289
.PT [Put Table of Contents]	294
.PW [Page Width]	296
.PX [Prefix]	297
.QQ [Quick Quit]	299
.QU [Quit]	300
.RC [Revision Code]	301
.RD [Read Terminal]	308
.RE [Restore Environment]	309
.RF [Running Footing]	310
.RH [Running Heading]	314
.RI [Right Adjust]	318
.RN [Reference Numbers]	320
.RV [Read Variable]	321
.SA [Save Environment]	323
.SB [Shift Baseline]	325
.SC [Single Column Mode]	327
.SD [Shading Definition]	328
.SE [Set Symbol]	337
.SI [Segment Include]	343
.SK [Skip]	351
.SM [Separation Master]	353
.SO [STAIRS/VS Output]	354
.SP [Space]	356
.SS [Single Space Mode]	358
.SU [Substitute Symbol]	359
.SV [Spelling Verification]	361
.SX [Split Text]	363
.TA [Table]	365
.TB [Tab Setting]	374
.TC [Table of Contents]	376
.TD [Table Definition]	379
.TE [Terminal Input]	393
.TH [Then]	395
.TI [Translate Input]	397
.TM [Top Margin]	399
.TP [Tab Position]	400
.TR [Translate Character]	404
.TS [Translate String]	406
.TU [Translate Uppercase]	408
.TY [Type on Terminal]	411
.UC [Underscore and Capitalize]	412
.UD [Underscore Definition]	414
.UN [Undent]	416
.UP [Uppercase]	418
.US [Underscore]	420
.UW [Unverified Word]	422
.VR [Vertical Rule]	423
.VT [Variable Text]	426

.WF [Write to File]	428
.WS [Word Space]	431
.WZ [Widow Zone]	434

---

## Chapter 4. SCRIPT/VS Control Words

This section describes each control word in the SCRIPT/VS language. All parameters are shown with descriptions of their effect on processing. Usage notes and examples are included.

---

### Control Word Syntax

All control words have 2-character names. A control word is identified by a period (.) in the first character position of an input line, followed by the 2-character name. If the control word accepts parameters, they follow the control word name and are separated from each other by blanks:

```
.du add raccoon giraffe llama
```

The blank separating the control word name from the first parameter is optional; if you omit it, SCRIPT/VS inserts it except when specifying the OFF parameter of the .CS [Conditional Section], .DM [Define Macro], .LI [Literal], and .WF [Write to File] control words, which require the blank. Thus,

```
.cecenter this line
```

is processed as

```
.ce center this line
```

**Note:** If you omit the first blank and the control word name and first parameter together form a valid macro name, the macro is processed rather than the control word, if macro substitution is on.

### Control Word Separator

The control word separator character can be used to enter several control words on a single line:

```
.sk .5i;.fo on;.in .25i
```

SCRIPT/VS scans every control word line for the control word separator character. If one is found, the line is divided at that point, and the part of the line before the control word separator is processed as a complete control word line. The remainder, to the right of the control word separator, becomes the next input line. The period in .fo on in this example appears in the first character position, allowing .fo to be recognized as a control word.

The control word separator character can also be used to place a control word within a line of text. If text, control word separators, and GML tags are mixed on a single line, then blanks in that line may disappear. For example,

```
an under;.us on;score;.us off;d word.
```

results in: an underscored word.

SCRIPT/VS also scans every text line for the control word separator character. If one is found and if it is immediately followed by a period and a valid 2-character control word name, the line is divided at that point. The part of the line preceding the control word separator is processed as a line of text with continuation, regardless of the formatting mode, and the remainder of the line to the right of the control word separator becomes the next input line. If a control word separator character is found in a text line, but is not followed by a control word, it is treated as text.

**Note:** Macros are not recognized in text lines. The .EM [Execute Macro] control word must be used to process macros in text lines.

The character to be used as the control word separator can be changed with the `.DC CW` [Define Character] control word. The `&$CW` system symbol contains the control word separator and can be used in its place.

## Macros

SCRIPT/VS macros are invoked in the same way as control words with a period in the first character position of an input line. Macro names, however, can be up to 10 characters long. Parameters can be specified on a macro line in the same way as on a control word line, except that the blank before the first parameter is not optional. If a macro called **para** were defined with the `.DM` [Define Macro] control word, it would be invoked in the same way as a control word:

```
.para parm1 parm2
```

If a macro is defined with the same name as a control word, the macro is processed instead of the control word when macro substitution is on. This allows you to redefine the function of a control word. Macro substitution can be turned on using the `.MS` [Macro Substitution] control word.

## The Control Word Modifier

The SCRIPT/VS control word processor recognizes a single quotation mark (') after the period as a control word modifier. For example:

```
.'ce Center this line.
```

The control word modifier changes the usual operation of the control word processor in two important ways:

1. No macro search is done. Even if a macro of the given name exists, the control word is invoked, not the macro.
2. No control word separator scan is done. Any control word separators in the line are left there as ordinary text characters. Thus, a control word entered with the control word modifier must be the last control word on that line.

Because no control word separator scan is done, a control word that accepts a line of text can be entered with the control word modifier to protect any separator characters that appear in the line as part of the text:

```
.'ce centered line; one line.  
.'h3 Using the ; in text
```

However, if the line contains a symbol that, when resolved, contains a control word separator character in the first position of the new input line, it is not ignored. You must either specify a `.DC CW OFF` immediately before the line or use the control word separator symbol (`&$CW`).

The control word modifier can be used with any control word except `.DM` [Define Macro] , `.LI` [Literal] , and `.WF` [Write to File] when the ON or OFF parameter is specified.

**Note:** If you get unexpected results when using the `.SE` control word to set symbols, specifying the control word modifier may correct the problem.



## The Input Line

The maximum number of characters in an input line is 256 after all symbols have been resolved. Only 256 characters per input line are read in from the input file. If symbol substitution extends the input line past 256 characters, any data past the 256-character mark is cut off and handled as the next input line. If the input line is a control word, this causes the cut off portion of parameters to be placed as text. Neither the .CT control word nor the continuation character can be used to continue control word parameters across multiple input lines.

---

## Notational Conventions

The format of each control word is described as shown in Figure 1 on page 56.

For information on reading the syntax diagrams, see to “Reading Syntax Diagrams” on page 56.

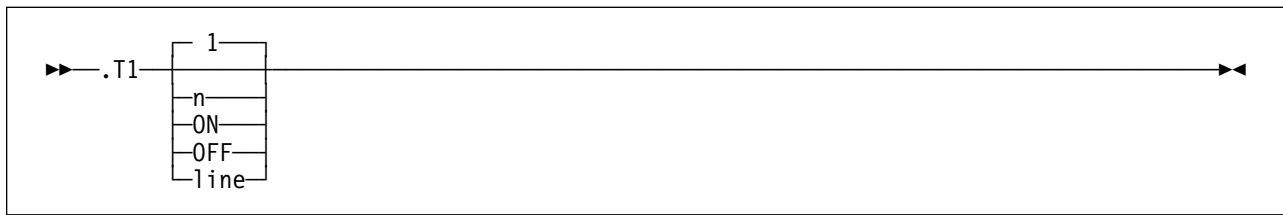
National characters refer to the alphabetic characters (uppercase and lowercase), the numerals, @, #, and \$.

The purpose of the examples in this book is to illustrate various formatting techniques using the SCRIPT/VS control words. Because of various factors, such as column line length, hyphenation dictionaries, and algorithmic hyphenators, example results may not always be identical to that shown; however, the effect of the control word is the same.

## Type 1 Control Words

There are several control words, called Type 1 control words, all of which have the same syntax and accept the same parameters. There are other control word types as well, and their syntax is explained in the individual control word descriptions.

Figure 1 is an example of a fictitious control word .t1, and is used in this discussion to represent any Type 1 control word.



- n* Is a positive integer that indicates the number of input lines to be processed by the Type 1 control word.  
The default is 1, meaning that the next input line after the control word is to be processed by this control word.
- ON** Starts an open-ended range of input lines to be processed by the Type 1 control word, until terminated with the OFF parameter.
- OFF** Stops the effect of the Type 1 control word, whether it was started with the ON parameter or with a number *n* that has not yet been exhausted.
- line* Is a single *line* that is to be processed by the Type 1 control word.  
The single input *line*
- `.t1 this is a line`
- is equivalent to the two lines
- `.t1 1`  
`this is a line`

Figure 1. Example of a Type 1 Control Word

The line given on a Type 1 control word is assumed to start with the first nonblank character after the control word. Thus, the following two forms operate identically:

```
.t1 this is a line
.t1  this is a line
```

The keywords ON and OFF and integers given in *n* are recognized only if they are the only parameters on a control word line. If there are other parameters, then all parameters are assumed to be *line*. Thus, Type 1 control words in the form:

```
.t1 On old Olympus' towering top
.t1 555 Bailey Avenue
```

are taken as control words followed by a line of text and not as requests to turn on processing or to process 555 input lines.

## Reading Syntax Diagrams

The syntax for SCRIPT/VS control words is shown using graphic notation. To read the diagrams, move from left to right and top to bottom, following the main path line.

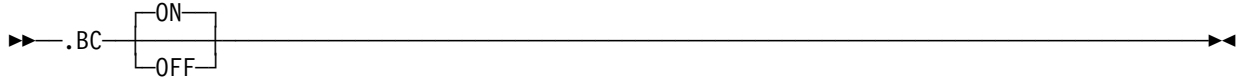
**Style Rules:** Syntax diagrams use the following style rules to show how to enter commands and parameters:

- A word in all italic, lowercase letters shows a parameter that you replace. For example:

*number*

shows that you replace number with a number like 10 or 15.

- A parameter above the line shows the default parameter. For example, ON is the default parameter in the syntax diagram for the .BC [Balance Column] control word:



**Symbols:** Syntax diagrams use symbols to help you follow the flow of information they communicate.

- Statements begin with:

▶▶—

and end with:

—▶▶

- Statements longer than one line continue to a second line with:

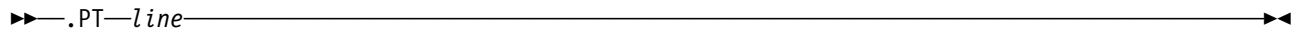
→

where they resume with:

▶—

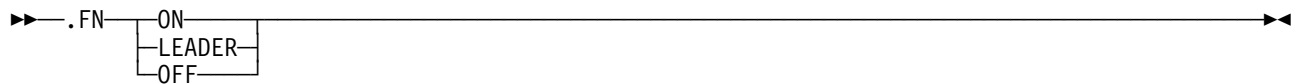
See “.DA [Define Area]” on page 108 for an example of a command statement that continues to additional lines.

**Required Parameters:** A parameter that you must include is displayed on the main path line. For example, the syntax diagram for the .PT [Put Table of Contents] control word:



shows that you must follow the control word with its required parameter.

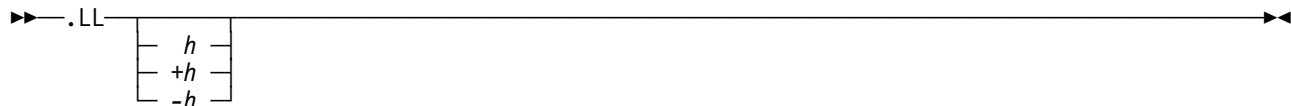
If there are two or more required parameters from which to choose, the parameters are shown with the first choice on the main path line and the other choices on branch lines under it. For example, the syntax diagram for the .FN [Footnote] control word:



shows that you must type the control word in any of the following ways:

.FN ON  
.FN LEADER  
.FN OFF

**Optional Parameters:** Parameters that you can include with a control word are displayed on branch lines below the main path line. For example, the syntax diagram for the .LL [Line Length] control word:



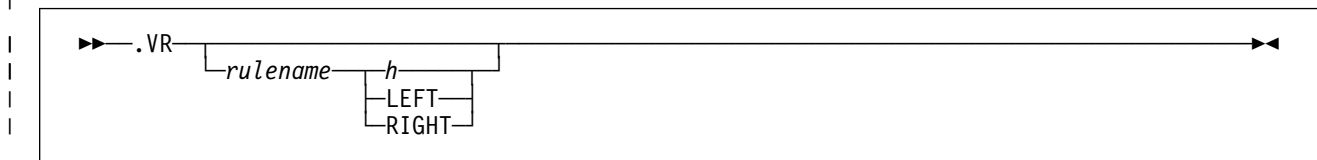
shows that you can type the control word in one of these ways:

```

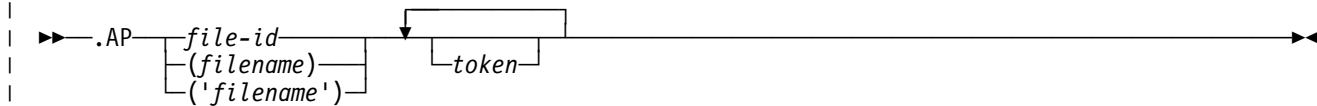
| .LL
| .LL h
| .LL +h
| .LL -h

```

Branch lines can include branch lines of their own. An example of this is the syntax diagram for the .VR [Vertical Rule] control word:

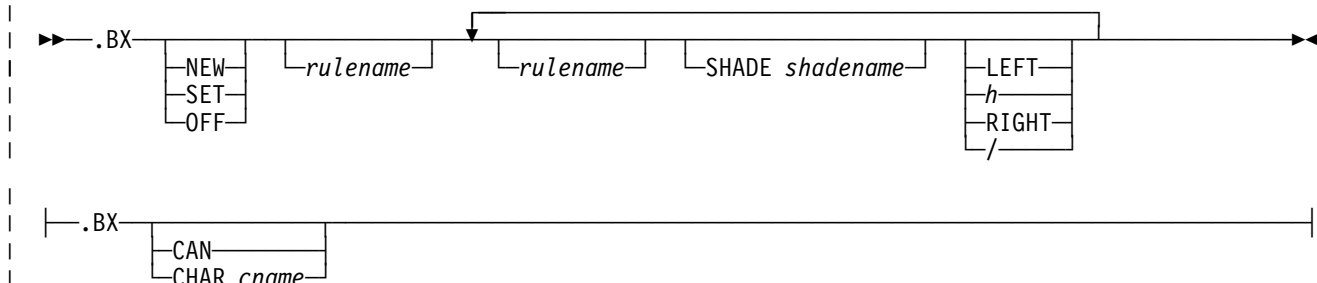


**Repeating Parameters:** An arrow on a line above a parameter means that you can repeat the parameter, or enter more than one of the listed parameters. An example of this is the syntax diagram for the .AP [Append] control word:



The arrow above *token* means you can include one or more token parameters with the .AP control word.

**Fragments:** A vertical line at the beginning of a syntax diagram indicates a different form of the control word. An example of this is the syntax diagram for the .BX [Box] control word:



## Space Units

Many SCRIPT/VS control words accept space units as parameters. Vertical and horizontal space units are designated as *v* and *h*, respectively, in the control word descriptions. Any of the notations shown in Table 6 on page 59 can be used.

Space Unit	Specified As	Examples
Centimeter	aCM	4.25cm 2,54cm 15cm
Character (Horizontal)	a	5 12.5 1,33
Cicero	nCp	c12 (12 didot points) 2c3 (2 Ciceros and 3 didot points) c1.5 (1.5 didot points)
Device Unit (Horizontal)	nDH	10dh 600dh
Device Unit (Vertical)	nDV	10dv 600dv
Em space (Horizontal)	aMH or aM	6mh 6m .33mh .33m
Em space (Vertical)	aMV	1mv .5mv
Inch	ai	3.5i 6,5i .75i
Line (Vertical)	a	2 3.5 1,75
Millimeter	aMM	12.7mm 25,4mm 100mm
Pica	nPp	p6 (6 points) 3p2 (3 picas and 2 points) p1.5 (1.5 points)
<p><b>Where:</b></p> <p><i>a</i> is a number of centimeters, characters, ems, inches, lines, or millimeters. The number can be fractional, with up to two decimal positions. Either a period (.) or comma (,) can be used to separate the integral and fractional portions of the number.</p> <p><i>n</i> is a number of whole ciceros, picas, or device units.</p> <p><i>p</i> is a number of points. (There are 12 didot points in a cicero, 12 points in a pica, and 72 pica points in an inch.)</p> <p><b>Warning:</b> All space units, except device units, are subject to rounding and truncation. Refer to <i>Document Composition Facility: SCRIPT/VS Text Programmer's Guide</i> for more information.</p>		

Table 6. Space-Units Notation

## SCRIPT/VS Control Word Descriptions

Following is a detailed description of each control word in the SCRIPT/VS language.

**Note:** The defaults used to format this book are not the same as the formatting defaults for the GML Starter Set. Your printed results of the examples may be different than the examples printed here because your column lengths, line lengths, and indents may be different than the ones we used.

The defaults of some of the control words are overridden by the profiles.

... [Set Label]

---

## ... [Set Label]

### Function

The ... [Set Label] control word marks a line in a SCRIPT/VS file or macro so that the line can be referred to in a .GO [Goto] control word.

### Syntax

►► ...*label* — *line* —►

### Parameters

*label* Identifies a line in a SCRIPT/VS file or macro. The *label* can be up to 8 characters long.

*line* Is the active part of this input line. The *first nonblank* character after the label is treated as if it were the beginning of a new input line; it may be a control word or a text line. Text associated with a label cannot begin with blanks.

**Default:** None

### Remarks

1. The ... [Set Label] control word must be the first control word on the input line, beginning in column one.
2. When the ... [Set Label] control word is encountered, SCRIPT/VS saves the location of the line so it can be found again if its label is ever used as the subject of a .GO [Goto] control word. A label can be followed by any valid SCRIPT/VS input line, or the label alone can occupy the line.
3. Use of labels with the .GO control word is restricted to the current input file or macro. SCRIPT/VS can only branch to a label within the same input file or macro. When a new file is imbedded or appended, a new set of labels is in effect while that file is being processed.
4. Every label in a particular file must be unique. If two identical labels are found in the same file, an error message is issued.

Multiple labels with the same name are allowed in macros, but when SCRIPT/VS is searching for labels in a macro, it finds only the first occurrence of the label.

5. A space is not required after the control word. To set a label with the name HERE, either "... HERE" or "...HERE" can be used.

6. The use of labels and the .GO [Goto] control word for conditional looping within an imbed file may require an .EF CLOSE [End of File] control word at the end of the imbed file in order to explicitly close it.

### Performance Considerations

The .GO function can be relatively inefficient in files. You should use it only in situations where it is the best way to achieve the required results. When going to a label that is *later* in the input file, it is most efficient when the label is not far from the .GO; when going to a label that is *earlier* in the file, it is most efficient when the label is near the beginning of the file.

Label processing in macros is much more efficient than in files. However, it is most efficient to branch to a label that is early in a macro, because labels are always searched for from the top of the macro.

Refer to “.GO [Goto]” on page 179 for a comparison of the use of labels for looping in files versus macros.

## Examples

You have created a file, REPORT1, that contains a summary of activity for January; another file, REPORT2, for February; REPORT3 for March; and so on. If you want to create a year-to-date report by imbedding all the report files for up to last month's report, you could use this SCRIPT/VS macro:

```
.dm reports on
.se *i = 1
...loop .im report&*i
.se *i = &*i + 1
.if &*i le &SYSMONTH .go loop
.dm off
.ms on
.reports
```

When the REPORTS macro is executed, the first value of the symbol “&\*i” is 1, and the name of the imbedded file “report&\*i” becomes “report1.” The next control word adds one to the value of the symbol “&\*i”; it is now 2. If the value of the counter is less than or equal to the number of the current month, the loop is processed again. This time the filename “report&\*i” becomes “report2.” The loop continues until the counter is greater than the number of the current month.

---

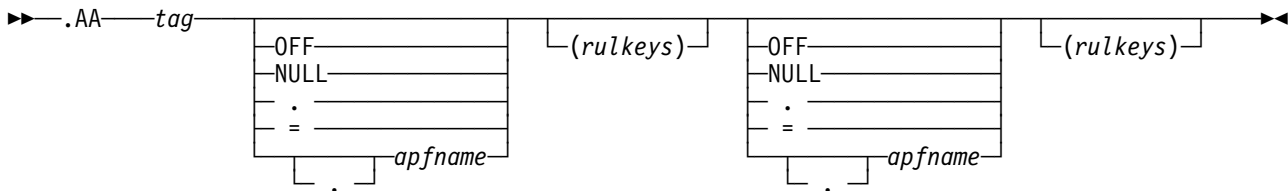
## .AA [Associate APF]

### Function

The .AA [Associate APF] control word can be used to associate a GML tag with the application processing functions (APFs) that are to be invoked to perform the tag's processing functions and to specify the rules for scanning the attributes for the tag. Two APFs can be associated with the tag—one for the processing when the tag is preceded by the GML tag delimiter and one for the processing when the tag is preceded by the GML end-tag delimiter. A GML tag name can be up to 8 characters long and can consist of letters, numbers, and the characters @, #, and \$ (except that the first character cannot be numeric).

The tag name can be entered in either uppercase or lowercase. (See the description of the .DC [Define Character] GML and .GS [GML Services] PREFIX control words.) An APF is a SCRIPT/VS executable macro or control word.

### Syntax



### Parameters

- tag* Specifies the GML tag to be associated with an APF.
- OFF** Indicates that this explicit tag-to-APF association is to be deleted. In this case, an APF can still be determined for a tag by class mapping. You can use .AA to set up an explicit association for only a start-tag or only an end-tag, and let the other APF be determined by class mapping.
- NULL** Indicates that this GML tag (or end-tag) is to result in no processing.
- .** Indicates that this GML tag (or end-tag) is to result in no processing. This option performs the same function as the NULL option.
- =** Indicates that the APF association for this tag (or end-tag) is to remain unchanged.
- apfname* Indicates that the specified macro is to be processed when the GML tag (or end-tag) is encountered.
- rulkeys* Parameters that set the rules for attribute scanning for this tag-to-APF association. The recognized rule parameters are ATT, NOATT, VAT, NOVAT, STOP, NOSTOP, MSG, and NOMSG. See “.GS [GML Services]” on page 183 for more information about the meaning of these parameters and the scanning rules they define.
- Rules are recognized only for tags in which an explicit APF association has been specified.
- The default rules for start tags are (ATT NOVAT STOP NOMSG). The default for end tags is (NOATT).



## Remarks

1. The first set of parameters apply to the tag. The second set of parameters apply to the end-tag.
2. If no scanning rules are given for this .AA, then the current rules, as most recently set by .GS RULES, are used. Once a .AA association is set up, it has scanning rules with it that remain unchanged, even if new rules are defined with .GS RULES.
3. You can change the APF association with another .AA without changing the scanning rules, or you can change the rules without changing the APF association. If you specify = for the APF name and give a new list of rules, the association remains the same, but the rules are changed. If you specify a new APF name and don't give a list of rules, the APF association is changed, but the rules remain the same.
4. An empty list of rules, that is, left and right parentheses with nothing in between, causes the current rules to be used. (The current rules are also used if this is the first .AA for this tag and you gave no rules.)

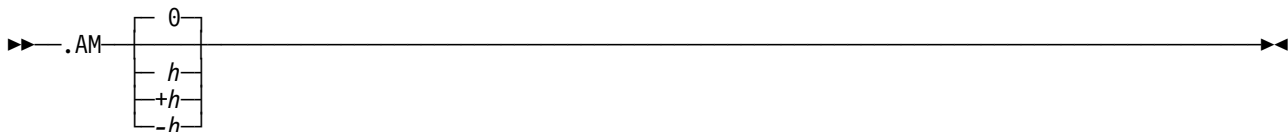
---

## .AM [Adjust Margin]

### Function

Use the .AM [Adjust Margin] control word to temporarily adjust the left margin without affecting the page margin setting. Figure 6 on page 438 shows how this control word interacts with other control words when formatting a page.

### Syntax



### Parameters

*h* Specifies the amount of horizontal space (in any valid space unit) to be used as the adjust margin value. When formatting text, SCRIPT/VS uses the adjust margin value in conjunction with the current page margin value to determine where on the page to align the text. If omitted, 0 is assumed, and the adjust margin value is reset.

If *+h* or *-h* is specified, the current adjust margin value is increased or decreased accordingly.

**Initial Setting:** 0

**Default:** Restores the initial setting.

### Notes

- .AM causes a break.
- .AM ends named areas, or tables.
- .AM takes effect on the current page.
- .AM applies to both even-numbered and odd-numbered pages.
- The adjust margin value is contained in the active environment.

### Remarks

1. Adjust margin applies only to the left margin, similar to .PM [Page Margins].
2. The new margin to be used when formatting output is computed by adding the adjust margin value to the page margin setting.
3. All column positions, tab positions, indents, rules, text, boxes, tables, and section and body areas are formatted relative to the new margin value.
4. Page areas and reference numbers are not affected by the .AM control word.
5. The adjust margin value applies only to the *starting position* of body and section areas, not to the text inside the area.
6. Ensure that the adjust margin value is not so large that no room is left on the page for text. If the line length, page margin, and adjust margin values are larger than the page width value, you may get an error message.
7. SCRIPT/VS does not change the column line length to account for the adjusted margin. Therefore, you need to change the column line length if the column line length plus the .AM value is too long.

8. Specifying a negative value ( $-h$ ) is not allowed if the resulting adjust margin value would be less than zero. For example:

```
.am 5  
.am -6
```

The second .AM causes an error, because the net adjust margin value would be  $-1$ . The control word still causes a break, and the adjust margin value is set to zero.

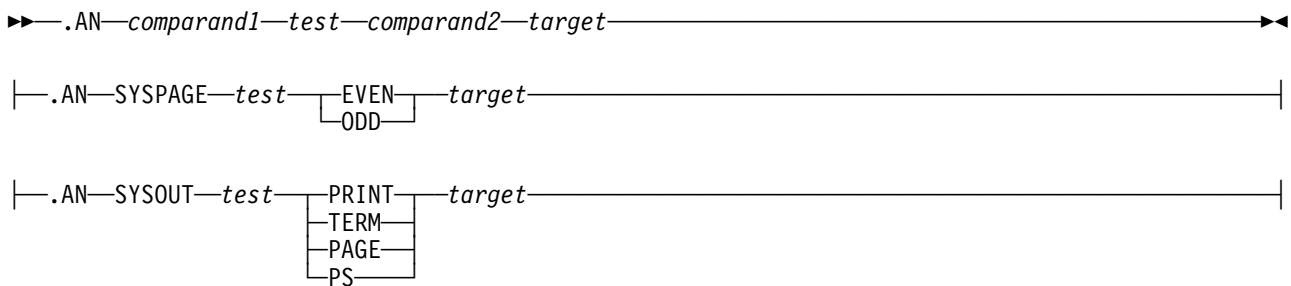
---

## .AN [And]

### Function

Use the .AN [And] control word in conjunction with the .IF [If] control word to process SCRIPT/VS input lines conditionally. The result of the test performed is logically ANDed to the result of the most recently performed .IF [If], .AN [And], or .OR [Or] control word to determine whether the target is to be processed.

### Syntax



### Parameters

**comparand1** Any string to be used as the first comparand. This comparand can be the value of a set symbol.

**comparand2** Any string to be used as the second comparand. This comparand can also be the value of a set symbol.

**test** A 1- or 2-character code that indicates the kind of comparison to make between the two comparands. The codes are:

Codes	Meaning
eq =	equal
ne $\neq$ <>	not equal
gt >	greater than
lt <	less than
ge $\geq$	greater than or equal
le $\leq$	less than or equal

**target** Any valid SCRIPT/VS input line containing a control word, a macro, or text. If this condition and the most recently performed .IF [If], .OR [Or], or .AN [And] are true, the target line is processed next, with the first nonblank character after .AN treated as the first position of the subject line. Otherwise, the target line is ignored, and processing continues with the input line that follows the .IF control line.

**SYSPAGE** Determines whether the page currently being processed is an even- or odd-numbered page.

**EVEN** Indicates that the test is for an even-numbered page.

**ODD** Indicates that the test is for an odd-numbered page.

**SYSOUT** Determines whether SCRIPT/VS output is being formatted for a printer or a terminal.

**PRINT** Indicates that the test is to determine if output is being formatted for a 1403 or a 3800.

**TERM** Indicates that the test is to determine if output is being formatted for a 2741 or a 3270.

- PAGE** Indicates that the test is to determine if output is being formatted for a page printer.
- PS** Indicates that the test is to determine if output is being formatted for a PostScript device.

In SCRIPT/VS, more variety is possible in output formatting than can be determined with the SYSOUT parameter alone. You can use the SCRIPT/VS system symbols '&\$LDEV' and '&\$PDEV' to determine the actual logical and physical devices.

## Remarks

1. For readability, an optional D can be added without an intervening blank to the .AN control word. This allows the control word to be written as .AN or .AND. However, the short form (.AN) is recommended for performance reasons.
2. The .AN [And] and .OR [Or] control words, in conjunction with .IF [If], .TH [Then], and .EL [Else], allow you to construct complex logic statements.
3. The .AN control word itself does not cause a break; the target control word might, if it is processed.
4. Each of the comparands can be up to 255 characters in length, and the shorter comparand is extended to the length of the longer with trailing blanks.
5. You must turn substitution off with the .SU [Substitute Symbol] control word if any symbols contain imbedded blanks, parentheses, or control word separators. SCRIPT/VS then does symbol substitution on the comparands at the time the .AN is processed, rather than at initial input time. This is necessary so that the test to be performed and the target of the .AN can be correctly identified and performed.

## Examples

- The following input lines are all equivalent:  

```
.if &a./&c eq &b./&d .ty Yes.
.if &a eq &b .if &c eq &d .ty Yes.
.if &a eq &b .and &c eq &d .ty Yes.
```
- You can use a .AN control word in conjunction with a .IF control word. For example, in the following case:

```
.if &u'&city = ABERDEEN
.an &u'&state = MARYLAND
.th .se water = 'Chesapeake Bay'
```

If the symbol &city has been set to the string "Aberdeen" (any mixture of uppercase and lowercase letters) and if the symbol &state has been set to the string "Maryland" (also in any mixture of uppercase and lowercase letters), then the symbol &u'&city resolves to the string "ABERDEEN" and the symbol &u'&state resolves to the string "MARYLAND." With both the IF and the AND conditions true, the symbol &water is then set to the string "Chesapeake Bay."

---

## .AP [Append]

### Function

Use the .AP [Append] control word to insert an additional SCRIPT/VS file at the end of the file just processed.

### Syntax



### Parameters

- file-id* Names the file to be appended. The name can be up to 8 characters long.
- The *file-id* can be associated with an external file or data set with the .DD [Define Data File-id] control word. If no .DD has been executed for the *file-id*, an external file or data set name is built by SCRIPT/VS from the given *file-id*, using the rules for the current environment, as described for ATMS-III in “Naming the Primary Input File” on page 8 and for other environments in “SEARCH: Specify a Library” on page 42.
- filename* The real name of the file or data set to be appended, and it must be enclosed in parentheses.
- If no .DD has been executed for the file or data set, SCRIPT/VS assigns an 8-character name to be associated with that file or data set.
- If the *filename* contains lowercase or special characters, it must be enclosed in single quotation marks ( ' ') and parentheses.
- token* Positional values passed to the file to be appended. A *token* can be up to 8 characters long.
- The first *token* (word) becomes the value of the symbol &1, the second token becomes the value of the symbol &2, and so on. The symbol &0 contains the number of tokens that were passed; up to 14 can be specified.

### Remarks

1. When the .AP control word is encountered, the current file is closed, and the appended file is processed as a continuation of the current file. Text or control words following the .AP control word in the current file are not processed.
2. The symbols &1 through &14 are reset whenever a .IM or .AP control word is processed, and the token &0 is reset to the number of nonnull tokens. If you want to leave token &1 unset but set token &2, you can use a percent sign (%) in place of token1 (or any other token you want left unset).
3. Identifiers provided by the NUMBER option of the SCRIPT command, error messages, and trace output all use the internal 8-byte file-id to describe a file.
4. If a GML tag contains an append request and the GML tag is at the end of the input file, a comment statement (.) must follow the tag in order to prevent an end-of-file condition from preventing the append from being processed.
5. The maximum number of tokens that can be passed to the append file is reduced if you use the .AP ('filename') or .AP (filename) form of the append control word. The maximum number of tokens is reduced by two, one for each parenthesis, and an additional one for each additional word in the

*filename*. SCRIPT/VS issues a warning message if there are more than 16 tokens on the control word line.

6. You cannot append any file that is already active.

## Examples

- .ap abc 10

The input file is closed. The contents of the SCRIPT/VS file ABC are processed immediately following the line of the current file that precedes the .AP request. The token 10 is passed to the appended file, so if the file ABC contains a control word of the form

```
.in &1
```

after substitution, the result is

```
.in 10
```

- You can append a file with a filetype other than that of the default. For example, you could enter  
.ap (troy exec a)  
and the file TROY EXEC A is added to the end of the file that has just processed.

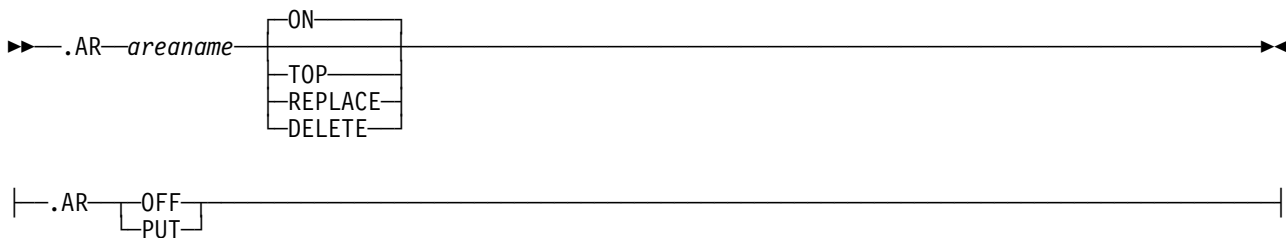
---

## .AR [Area]

### Function

The .AR [Area] control word allows you to start, add to, or end a *named* area. It also allows you to delete the contents of a *named* area, or request that section areas be placed in the output. The *named* area must have been previously defined with the .DA [Define Area] control word.

### Syntax



### Parameters

<i>areaname</i>	Identifies a <i>named</i> area that has been defined with the .DA [Define Area] control word. The identifier can contain a maximum of 16 national characters.
<b>ON</b>	Starts formatting into the designated area. If text is already in the <i>named</i> area, the new text is added to the area <i>after</i> the existing text.
<b>TOP</b>	Starts formatting into the designated area. If text is already in the <i>named</i> area, the new text is added to the area <i>before</i> the existing text.
<b>REPLACE</b>	Deletes any text that is already in the <i>named</i> area but which has not already been placed on a page, and then starts formatting into the designated area.
<b>DELETE</b>	Deletes any text that is already in the <i>named</i> area but which has not already been placed on a page. No new text is formatted into the area as a result of this parameter.
<b>OFF</b>	Ends any area that is in process. All the material in the area is now eligible to be placed on a page, but it has not yet been placed. It can still be deleted before it is placed.
<b>PUT</b>	Requests section area placement. A section break is performed, and all the normal body text up to this point is processed and placed on the page, according to the current column definition and vertical formatting status. Then, a new section is built to receive the section areas. If the material in the section areas does not all fit in one section, then as many consecutive sections and pages as required are built, each one receiving as much as it can (or is allowed to) hold, until all section areas are placed. Then, the current column definition is resumed, and subsequent body text continues just below the last section created for section areas.

**Default:** ON

### Notes

- .AR causes a break.
- .AR ends a keep, float, footnote, named area, or table.
- .AR ensures that a page is started.
- .AR PUT causes an unconditional section break.



- This control word saves and restores the current environment.

## Remarks

1. Use the .AR control word when you want to make the text that follows appear at a particular position on the page. The position must be specified with the .DA [Define Area] control word.
2. The material formatted into a *named* area is not placed on the page when the area is ended. PAGE and BODY areas are placed when the page is ended; SECTION areas are placed only when a .AR PUT control word is issued. The &AD' symbol attribute can be used to determine how much  
|     unplaced material a *named* area contains, if any. When the &AD' symbol attribute is prefixed with the  
|     &DV' symbol attribute, the depth of the unplaced text in the area is returned in vertical device units.  
|     When &AD' is not prefixed with the &DV' symbol attribute, the value returned is the actual depth of  
|     the unplaced text in the area converted to linespaces in the default font.
3. If you specify .AR PUT, then SECTION areas are placed in the BODY area of the page. .AR PUT is therefore ignored in a running heading or a running footing.
4. Area formatting has several things in common with keep, float, and footnote formatting:
  - The same control words that are disallowed in a keep are also disallowed in a named area. If one is encountered, the area is ended with a message, and then the control word is processed.
  - A named area cannot contain a keep, float, or footnote, nor can a keep, float, or footnote contain a *named* area.
  - The active environment is saved at the beginning of the area and restored at the end of it. The column line length and font are initialized to the values specified with the .DA [Define Area] control word, if any.
  - The .AR control word always ends any area that is in process before it is executed. If there is invalid syntax, nothing more is done, except issuing an appropriate message.
  - .AR OFF ends any vertical rules that may be in effect.
5. Widow-zone processing is not performed in *named* areas.
6. See the **Remarks** section following the description of “.SE [Set Symbol]” on page 337 for restrictions involving setting a symbol to ‘&’ inside a named area.
7. When putting out a section area, any vertical rules that are currently active in the body continue through the section area. If the body has multiple columns, only the vertical rules in the first column of the body continue through the section area, because section areas are handled in single-column mode.
8. Certain control words are disallowed within a named area. If one of the disallowed control words is encountered, the named area is immediately ended, as though .AR OFF had been processed. Then the disallowed control word is executed. A warning message is issued, identifying the control word that ended the named area. See Table 15 on page 442 for a list of the disallowed control words.
9. The first line in body and page areas has the extra leading specified with the EXTRA parameter of the  
|     .LS [Line Spacing] control word removed. The first line of section areas has extra leading removed if  
|     that section area is the first thing in the body of the page.

## Examples

- Suppose an area named CORNER is defined as  
     .da corner 4.5i 9i page width 1.0i  
     Text can be formatted and placed in the CORNER area by entering

## **.AR [Area]**

.hy on  
.ar corner on  
This text  
is formatted in a little block  
in the lower  
right-hand corner of the page.  
.ar off

This text  
is for-  
matted in  
a little  
block in  
the lower  
right-hand  
corner of  
the page.

- Suppose an area named RAVEN has been defined as:

```
.da raven 0 0 body
```

Text can be formatted and placed in the RAVEN area by entering:

```
.ar raven on
.ce THE RAVEN
.sk
.fo off
Once upon a midnight dreary,
  While I pondered, weak and weary,
Over many a quaint and
  Curious volume of forgotten lore--
While I nodded, nearly napping,
  Suddenly there came a tapping,
As of someone gently rapping,
  Rapping at my chamber door.
"'Tis some visitor," I muttered,
  "Tapping at my chamber door--
Only this and nothing more."
.
.
.
.ar off
```

The *named* area is placed when the page is ended, but because the poem is quite long, it most likely not fit on one page. As much as will fit is placed on the page, and the rest is saved and placed on following pages, until all the material in the area is completely placed.

- In the previous example, the title appears only on the first page. You can repeat the title by inserting it at the top of the remaining material on each subsequent page. For example, suppose that the following running heading is defined after the *named* area is filled with text:

```
.rh on
.if &AD'raven eq 0 .go empty
.ar raven top
.ce THE RAVEN (continued)
.sp
.ar off
.go end
  (skip normal heading)
...empty
  (normal running heading)
...end
.rh off
```

The running heading definition is executed on each subsequent page. The title is added at the top of the area on each page until the depth of the material remaining in the area named RAVEN is zero.

Use the `.BC` [Balance Columns] control word to cancel and restore column balancing for multiple column formatting.

## Syntax



## Parameters

**ON** Indicates that you want SCRIPT/VS to balance columns. ON is the initial setting as well as the default.

**OFF** Indicates that you do not want SCRIPT/VS to balance columns when a section break or page eject occurs.

**Initial Setting:** ON

**Note:** Column balancing mode is included in the active environment.

### Remarks

1. When column balancing is in effect, the depth of the text in each column is made as equal as possible before the material on that page is printed.
2. If a blank line that was generated by the .SK [Skip] control word ends up at the top or bottom of a column in the page body after balancing, it is discarded.
3. When column balancing is off, the depth of the text in each column is determined by explicit .CB [Column Begin] control words or by filling all columns, but no attempt is made to equalize the number of lines in all columns.
4. If a column is started explicitly by a .CB [Column Begin] or by a .CC [Conditional Column Begin] control word, text from the new column is balanced only with following columns. The new column is not balanced with preceding columns.
5. If a page eject occurs while processing multiple columns, this does not mark any column ineligible for balancing. A column eject that changes the current column from the *last* column of a page to the *first* column of the next page is the same as a page eject.
6. If you have less than 8 lines of output in multiple columns, turn off widow zone processing (.wz off). Otherwise, the columns are not balanced correctly.
7. Even if .FV JUSTIFY has been specified, the columns may not all be the same length if some columns have more vertical white space than can be redistributed within the limits set by the .LS [Line Spacing] control word.

---

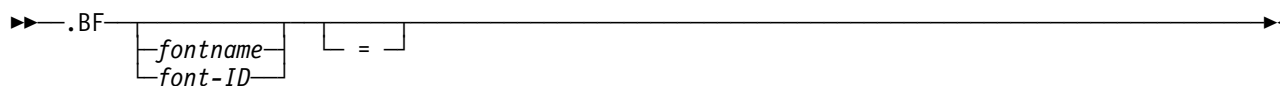
## **.BF [Begin Font]**

### **Function**

Use the .BF [Begin Font] control word to indicate the font in which subsequent text should be formatted. The current font is saved before the new font is started; the .PF [Previous Font] control word can be used to restore the saved font.

The new font to be started can be identified as a *named* font, defined with the .DF [Define Font] control word, or as an external (host system) font-ID listed with the CHARS option of the SCRIPT command. More than one font can be specified with the .BF control word, and the first valid font is used.

### **Syntax**



### **Parameters**

- |                 |  |
|-----------------|--|
| <i>fontname</i> | Name of a font, as defined with the .DF [Define Font] control word.  |
| <i>font-ID</i>  | External (host system) name of the new font, as specified in the CHARS option of the SCRIPT command.         |
| =               | Restarts the current font when none of the requested fonts are available for or used on a particular device. |

**Default:** Restarts the current font.

**Note:** The current font and the font save stack are included in the active environment.

### **Remarks**

1. The font name specified with the .BF control word can be a list of fonts previously defined with the LIST parameter of the .DF control word. For more information on using the LIST parameter, see “.DF [Define Font]” on page 130.
2. After the .BF control word has been executed, all subsequent text characters are formatted using the specified font. The specified font remains in effect until another .BF or .PF [Previous Font] control word is encountered with a different font.
3. The .BF control word saves the current font before beginning the new font; the .PF control word restores the previous font. As many as 16 fonts can be saved.
4. More than one font can be identified with the .BF control word. The first font given that has been defined with the .DF control word or was given with the CHARS option of the SCRIPT command is taken as the new font. For page printers and PostScript devices, this font must be available for use by SCRIPT/VS. An error occurs if none of the fonts given is valid. For more information, refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.
5. The system symbol array &\$CHAR contains a list of external (host system) font-IDs specified with the CHARS option of the SCRIPT command. &\$CHAR(0) contains the number of font-IDs specified, &\$CHAR(1) contains the first font-ID specified (or the default for the logical device, if the CHARS option was omitted), &\$CHAR(2) contains the second font-ID, and so on.

## .BF [Begin Font]

6. The .DF [Define Font] control word can be used to define *named* fonts that are combinations of external (host system) font-IDs and formatting functions such as underscoring and overstriking. Any *named* font can be used with .BF.
7. When a *named* font that was defined with the TYPE parameter of the .DF [Define Font] control word is started, if descriptive attributes for typeface and point size were *not* specified as part of the definition, they remain unchanged. The TYPE parameter is valid only for page printers and PostScript devices.
8. When a *named* font is started, if the CODEPAGE parameter was *not* specified as part of the definition, it remains unchanged. The CODEPAGE parameter is valid only for page printers and PostScript devices.

## Examples

- You can specify the font to be started symbolically:

```
.bf &$CHAR(2)
```

Subsequent text is set in whatever font was the second one given with the CHARS option of the SCRIPT command. If only one font was specified with the CHARS option, the value of &\$CHAR(2) is null. This is equivalent to:

```
.bf
```

The only effect is to save the current font for a corresponding .PF [Previous Font] control word.

- If you have defined a *named* font, such as

```
.df hilite us up font
```

then, when formatting for the 3800 Printing Subsystem, entering

```
.bf hilite
```

HAS THIS EFFECT.

- When formatting for a page device, you can define a *named* font by describing it. For example:

```
.df body type('monotype bodoni' 10)
```

When you enter

```
.bf body
```

all subsequent text is set in 10 point Monotype Bodoni<sup>5</sup> type, providing that Monotype Bodoni is available at your installation.

- Suppose the following *named* font is also defined:

```
.df small type(6 italic)
```

If the current font is 10 point Monotype Bodoni, and you enter

```
.bf small
```

subsequent text is set in 6 point Monotype Bodoni italic.

- You may not know exactly which fonts are available when a document is created. For example, you may prepare a document to be formatted for a 3800 Printing Subsystem without knowing what fonts are used. If you want to ensure that a piece of text is set in a bold font, you can enter:

```
.bf gb10 gb12 sb12
```

Subsequent text is formatted in the GB10 font, if it was specified with the CHARS option of the SCRIPT command. If not, GB12 is used if it was specified, and so on.

---

<sup>5</sup> Monotype Bodini is a registered trademark of the Monotype Corporation plc.

- When a document may be formatted for a variety of devices, the fonts available may vary according to the device. For example, the following font may be available only when formatting for a 1403:

```
.df os3 os rpt 3
```

The following font may be available only when formatting for a page device:

```
.df bold type(futura 10 bold)
```

- You can ensure that a piece of text is set in a bold font, regardless of the device for which your document is formatted, by entering  

```
.bf bold os3
```
- The = parameter can be used to avoid an error message in the event that none of the fonts given with .BF have been defined.

For example, suppose you have defined a font named FIGFONT for use when formatting a document for a page device and a font named HI2 for use when formatting for the 3800 Printing Subsystem Model 1. The control word

```
.bf figfont hi2
```

causes subsequent text to be formatted using FIGFONT if the output device is a page device, or HI2 if the output device is the 3800 Printing Subsystem Model 1, but causes an error message for any other device.

If you enter

```
.bf figfont hi2 =
```

no error message is issued when, for example, you are formatting for the 1403 printer; subsequent text continues to be formatted using the current font.

---

## **.BL [Blank Line]**

### **Function**

The .BL [Blank] control word is generated by SCRIPT/VS and executed whenever a blank line is processed. It results in one line of vertical white space.

### **Syntax**

►► .BL ◀◀

### **Notes**

- .BL causes a break.
- .BL ensures that the page is started.

### **Remarks**

1. Whenever SCRIPT/VS encounters a blank input line, it generates and executes a .BL control word. The .BL control word has the same effect as the .SP [Space] control word.  
  
If you want to have blank lines perform some other function, you can define a .BL macro with .DM [Define Macro]. When macro substitution has been turned on with the .MS [Macro Substitution] control word, the .BL macro is executed whenever a blank line is processed.
2. No .BL function is performed for lines processed in literal mode, caused by the .LI [Literal] control word.
3. A blank line may originate from a number of sources. Because of this, you should define a .BL macro only when a specific use in a certain part of a document requires it. Blank lines may originate from:
  - A source input file
  - A macro line that is blank
  - Terminal input (.TE)
  - A nonblank line that becomes blank as a result of symbol substitution.
4. A blank line is not the same as a null line. Null lines contain no characters and are processed by the .NL [Null Line] control word.



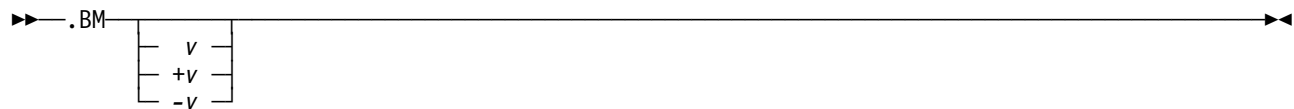
---

## **.BM [Bottom Margin]**

### **Function**

The .BM [Bottom Margin] control word specifies the amount of vertical space to be left at the bottom of subsequent output pages. Figure 6 on page 438 shows how this control word interacts with other control words when formatting a page.

### **Syntax**



### **Parameters**

*v* Specifies the amount of vertical white space to be left at the bottom of output pages.

+*v* or −*v* increases or decreases the existing bottom margin by the amount given.

**Initial Setting:** Dependent upon the logical device specified.

**Default:** Restores the initial setting.

### **Notes**

- .BM takes effect on the next page.
- .BM ends a keep, float, footnote, named area, or table.
- The size of the bottom margin is included in the page environment.

### **Remarks**

1. The running footing is placed just above the bottom margin, as shown in Figure 6 on page 438.
2. The value given should not be so large that the top margin and bottom margin together completely fill the page.
3. The size of the bottom margin is not affected by changes in line spacing.
4. The default bottom margin size for each logical device is shown in the logical device tables on pages 27 through 28.
5. Setting .BM to small values, such as .5cm (.2i) or smaller, may cause problems on page printers, because such a setting may cause characters that fall below the baseline to extend off the page.

---

## **.BR [Break]**

### **Function**

Use the .BR [Break] control word to ensure that the next input line is not concatenated with the previous line.

### **Syntax**

►► .BR ◄◄

### **Notes**

- .BR causes a break.

### **Remarks**

1. The .BR control word is necessary only when SCRIPT/VS is concatenating input lines. It causes the preceding line to be formatted as a short line, if it is shorter than the current column length.
2. Many other control words have the effect of a break. A .BR control word is not needed when one of these is present. See Table 13 on page 441 for a list of these control words.
3. A leading blank or tab on an input line has the effect of a break.

### **Examples**

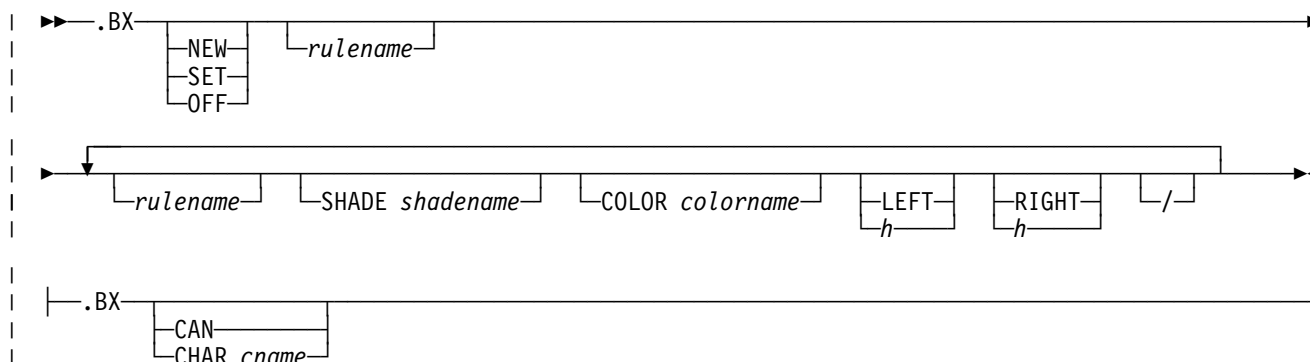
- If you enter  
Heading:  
.br  
New paragraph...  
the output appears as  
Heading:  
New paragraph...  
• If the .BR control word is not included and concatenation is on, the output appears as  
Heading: New paragraph...

## **.BX [Box]**

### **Function**

Use the .BX [Box] control word to specify where vertical rules are to be placed to produce a box and to connect the vertical rules with horizontal rules. With this control word, you can format charts or text within neatly formatted boxes.

### **Syntax**



### **Parameters**

- NEW** If a box is already in progress, a new box is started and the current box is suspended. This allows boxes to be drawn inside boxes. If no vertical rule positions are given, or if no box is currently active, the NEW function is ignored.
- SET** Causes SCRIPT/VS to suspend the current box and prepare for a new box. Unlike a NEW box, however, no horizontal rule is drawn for the top of the box.
- OFF** Causes SCRIPT/VS to finish drawing the box by ending all the vertical rules that are part of this box and by drawing one last horizontal rule. If this box was started as a NEW or SET box, the previous box is reinstated when this one is ended.
- If vertical rule positions are specified with OFF, and if no box is currently in effect, then a box bottom is drawn according to the specifications given.
- rulename** Any *named* rule that has been defined with the .DR [Define Rule] control word. The .BX control word can draw a horizontal line and start several vertical lines. On page printers, each vertical line can be drawn with a different *named* rule. The horizontal line is drawn with the first rulename parameter; if the rulename parameter is omitted for any vertical line, that line is drawn with the same *named* rule as the previous one.
- SHADE** Specifies that the inside of the box defined by the next two horizontal positions is to be shaded by the named shading definition.
- The *shadename* specifies the name of the shading definition. The name can be a maximum of 16 national characters, and it is not case sensitive. The shading definition must have been defined with the .SD [Shading Definition] control word before specifying the .BX control word and its associated parameters.
- COLOR** Specifies that the inside of the box defined by the next two horizontal positions is colored by the named color definition. The COLOR parameter applies only to printers with color capability. "colorname" can be any color previously defined with the .CR [Color] control word.

## **.BX [Box]**

<b>LEFT</b>	Indicates a vertical rule is to be placed flush against the left margin of the column, regardless of the current left indentation.																
<i>h</i>	Specifies, in any recognized space units, the horizontal position of a vertical rule for the box. The leftmost horizontal position in the group specifies the position of the left end of the horizontal rule, and the rightmost position specifies the right end of the horizontal rule. A horizontal rule is drawn across all vertical rules in a group to form the top of the box.																
<b>RIGHT</b>	Indicates a vertical rule is to be placed flush against the right margin of the column, regardless of the current right indentation.																
<b>/</b>	Indicates that the horizontal rule should not be drawn between a pair of vertical rules. That is, a slash indicates the end of a group of vertical rule positions and the beginning of another group. The horizontal position just before the slash represents the right end of a horizontal rule; the one after the slash is the left end of another. This permits you to draw several boxes side by side.																
<b>CAN</b>	Causes the box to be canceled without a box bottom. If the box being canceled is a nested box, the next higher box is reinstated.																
<b>CHAR</b>	For line devices, boxes and rules must be built with characters containing fragments of rules and rule intersections. For such devices, when the rulename parameter is not specified, SCRIPT/VS assumes an appropriate box character set based on the logical device type and current font. When rulename is specified with a .BX control word, the box character set of the rule font is used. You may override this default and specify any of the following box character sets:  <table><tr><td><b>APL</b></td><td>APL characters</td></tr><tr><td><b>GPC</b></td><td>3800 GP12 font</td></tr><tr><td><b>TNC</b></td><td>1403 TN character set</td></tr><tr><td><b>TRM</b></td><td>Terminal character set</td></tr><tr><td><b>VAN</b></td><td>Default for 2741, 3270, and 3800-3</td></tr><tr><td><b>32A</b></td><td>3270 APL characters</td></tr><tr><td><b>32T</b></td><td>3270 text characters</td></tr><tr><td><b>38C</b></td><td>SCRIPT/VS 3800 fonts</td></tr></table>	<b>APL</b>	APL characters	<b>GPC</b>	3800 GP12 font	<b>TNC</b>	1403 TN character set	<b>TRM</b>	Terminal character set	<b>VAN</b>	Default for 2741, 3270, and 3800-3	<b>32A</b>	3270 APL characters	<b>32T</b>	3270 text characters	<b>38C</b>	SCRIPT/VS 3800 fonts
<b>APL</b>	APL characters																
<b>GPC</b>	3800 GP12 font																
<b>TNC</b>	1403 TN character set																
<b>TRM</b>	Terminal character set																
<b>VAN</b>	Default for 2741, 3270, and 3800-3																
<b>32A</b>	3270 APL characters																
<b>32T</b>	3270 text characters																
<b>38C</b>	SCRIPT/VS 3800 fonts																

## **Notes**

- .BX causes a break.
- .BX ensures that the page is started.

## **Remarks**

1. The .BX control word describes an overlay structure for subsequent text that is processed by SCRIPT/VS. After the .BX control word is processed, SCRIPT/VS continues formatting output lines as usual. After each line is completely formatted, SCRIPT/VS places vertical rules at the positions indicated.
2. A .BX control word with different vertical rule specifications may be used while a box is being drawn. When this happens, a horizontal rule is drawn, all existing vertical rules are ended at the horizontal rule, and the new vertical rules are then overlaid on all subsequent text lines.

3. For the 3800 Printing Subsystem, several restrictions are imposed on the use of the .BX control word:

- Only monospace fonts may be used within a box, and all fonts used must be of the same pitch. All of the 3800 fonts distributed with SCRIPT/VS, listed in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* are monospace, with the exception of GP12, which is mixed-pitch. Boxes may be misaligned if they are drawn with proportionally spaced fonts.
- The entire box and any other nested boxes are drawn in one font.
- When a vertical rule is overlaid on a text character, the rule replaces the character, and misalignment of text may occur.
- When a vertical rule is overlaid on horizontal white space from another column, misalignment of text may occur.

4. For line devices, if a box is started while vertical rules are still going, any rulenames specified with the .BX control word are ignored. All concurrent vertical and horizontal rules must be drawn using the font and the box character set currently in use.

5. If only one rulename is given, it is used for both the horizontal and the vertical rules. If, before any horizontal positions are specified, two rulenames are given with the .BX control word, the first is used for horizontal rules and the second is used for vertical rules.

6. If rulename is not specified, the box is drawn with a default rule appropriate for the logical device:

- For line devices, the box is constructed using the box character set of the font that is current when the box is started.
- For page printers and PostScript devices, the box is drawn with a rule that is 0.3 millimeters thick. This rule is named *boxrule* and can be redefined with the .DR [Define Rule] control word. See the description of the .DR control word for more details.

7. Vertical rules that cross columns or pages extend to the bottom of the section or page, whether they are created with .BX or with the .VR [Vertical Rule] control word.

8. On page printers and PostScript devices, the depth of the lines containing horizontal rules created by .BX depends upon the linespacing currently in effect:

- If fixed linespacing has been specified with .LS NORMAL, lines containing horizontal rules have the specified depth, even if this is less than the thickness of the rule.
- If fixed linespacing is not in effect, lines containing horizontal rules have a depth equal to the thickness of the thickest rule plus the difference between the current line spacing value and the capital height in the current font. This preserves a uniform “leading” between text lines and horizontal rule lines.

9. To start a box that does not contain the extra vertical space which is explained above, use the NORMAL parameter of the .LS [Line Spacing] control word to set the linespacing to the thickness of the rule being used for the top horizontal rule on the .BX control word that is used to start the box. Restore the previous line spacing after that .BX control word.

10. A .BX control word with no parameters repeats the horizontal rule drawn by the previous .BX control word; all vertical rules continue unchanged.

11. The output from SCRIPT/VS when formatting for a PostScript, 4028 physical device, or an AFP logical device is a resolution-independent data stream. Because of the variations in printer resolutions, the same SCRIPT/VS output file might produce different results on different devices. Therefore, when you format for one of these devices, there may be one pel rounding errors at the printer that are evident in rule widths and rule intersections. If your box is shaded, this could also cause the shading to appear to not fit within the box. Moving the position of the box on the page slightly can improve the appearance.

## .BX [Box]

12. The SHADE parameter is ignored if you are formatting for a line device or a 4250 printer. For more information about the SHADE parameter, see “.SD [Shading Definition]” on page 328.
13. Ending a .BX control word with other than a horizontal specification or the RIGHT parameter results in the horizontal rules being omitted.
14. The SHADE parameter must be placed prior to the two horizontal positions defining the box which is to be shaded. Placing the SHADE parameter between the two positions, or after the two positions, results in that box not being shaded as intended.

## Examples

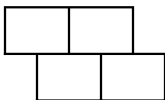
- The following input sequence could be used to center text in a box that goes from the left to the right of the current column:

```
| .bx left right
| .ce on
| Now is the winter of our discontent
| Made glorious summer by this sun of York;
| And all the clouds that lower'd upon our house
| In the deep bosom of the ocean buried.
| .ce off
| .bx off
```

```
|
|      Now is the winter of our discontent
|      Made glorious summer by this sun of York;
|      And all the clouds that lower'd upon our house
|      In the deep bosom of the ocean buried.
|
```

- When vertical rules are respecified within a box, the old vertical rules are ended and the new vertical rules begun at a horizontal rule:

```
.bx 5 10 15
.sp
.bx 8 13 18
.sp
.bx off
```



- Discontinuities in the horizontal rules drawn by .BX are indicated by a slash (/) between vertical rule positions:

```
.bx 5 9 / 15 20 / 25 31
.tb 5 15 25
&$TAB.See&$TAB.Hear&$TAB.Speak
.bx off
```

See

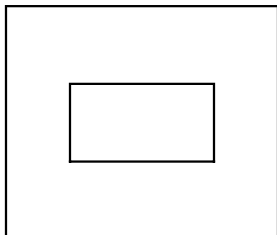
Hear

Speak

- If **.BX NEW** is specified while another box is active, the new box is nested within the outer box. For example,

```
.bx 10 30
.sp 2
.bx new 15 25
.sp 2
.bx off
.sp 2
.bx off
```

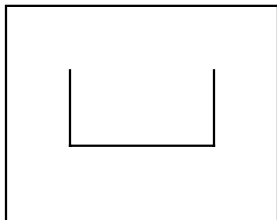
results in



- If **.BX SET** is specified while another box is active, the results are similar to those obtained when you use the **NEW** parameter, except no horizontal rule is drawn for the top of the box. For example,

```
.bx 10 30
.sp 2
.bx set 15 25
.sp 2
.bx off
.sp 2
.bx off
```

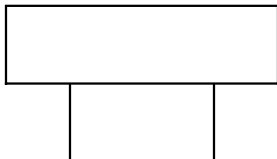
results in



- If a **.BX** control word is specified with only vertical parameters while another box is active, the current box is ended (as if **.BX OFF** had been specified) before the next box is started. For example,

```
.bx 10 30
.sp 2
.bx 15 25
.sp 2
.bx off
```

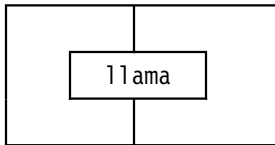
results in



## .BX [Box]

- Boxes can be “nested” by identifying the inner boxes as “NEW”:

```
.cl 30
.bx 5 15 25
.sp
.bx new 10 20
.ce llama
.bx off
.sp
.bx off
```



Notice that the inner box “covers up” one of the vertical rules from the outer box, and that this rule automatically reappears when the inner box is no longer in the way.

- The .BX [Box] control word does not, by itself, prevent a box from being split across pages. .BX is usually used within the range of another control word, such as .FL [Float] or .KP [Keep], which keeps the box and its contents together along with other text. For example:

```
.kp on
.bx 1 &$CL
.si blizzard
.bx off
Figure 2. Blizzard in Antarctica
.kp off
```

When this is formatted, the box and figure caption are kept together in the same column:

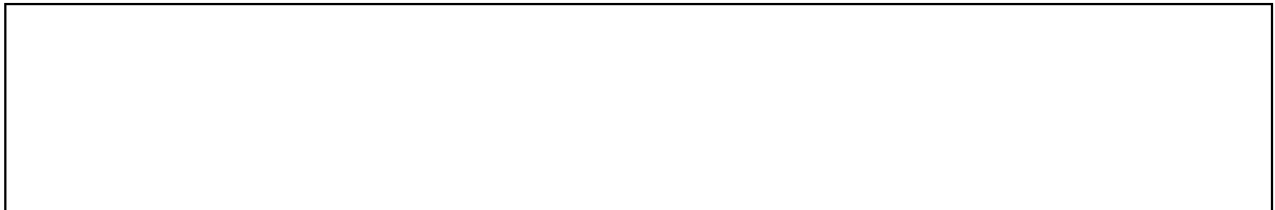


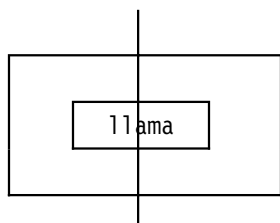
Figure 2. Blizzard in Antarctica

If the box and figure caption do not fit, a large amount of blank white space may be left at the bottom of the current column or page. If the figure is not directly related to the text around it, this white space can be eliminated by using .FL instead of .KP.



- Rules drawn with the .HR [Horizontal Rule] and the .VR [Vertical Rule] control words are independent of those produced by .BX. For example, a vertical rule can be drawn through two boxes:

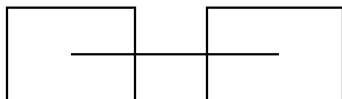
```
.cl 30
.vr 15
.sp
.bx 5 25
.sp
.bx new 10 20
.ce llama
.bx off
.sp
.bx off
.sp
.vr off
```



The vertical rule is not “covered up” by the box, because the vertical rule was independently produced with .VR.

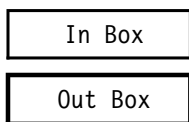
Similarly, a horizontal rule can be drawn through discontinuous boxes without disturbing them:

```
.bx 5 15 / 20 30
.sp
.hr 10 25
.sp
.bx off
```



- Boxes can be drawn with *named* rules, defined with the .DR [Define Rule] control word. For example, on page printers you can draw boxes with rules of different weights:

```
.dr thin weight .4mm
.dr thick weight .6mm
.cl 3.5i
.bx thin 1.25i 2.2i
.ce In Box
.bx off
.bx thick 1.25i 2.2i
.ce Out Box
.bx off
```

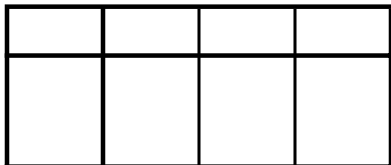


## .BX [Box]

- On page printers, a single box can be drawn with rules of different weights. For example,

```
.bx thick 3p 6p thin 9p 12p thick 15p
.sp
.bx
.sp 3
.bx off
```

results in

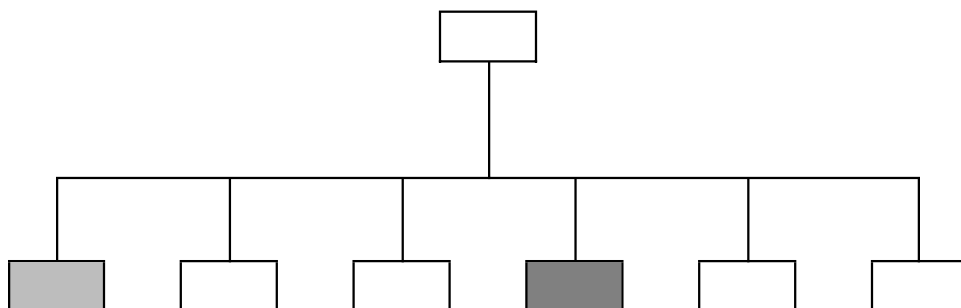


The box dimensions were specified in picas and points rather than in unqualified space units, so that the box remains the same size regardless of the initial font that determines the size of unqualified horizontal space units.

- On AFP page printers and PostScript devices, boxes can be drawn with different shading variations and with different shading patterns. For example,

```
.sd a shade light screen
.sd b shade medium
.bx 2.25i 2.75i
.sp
.bx off
.vr 2.5i
.sp 3
.hr .25i 4.75i
.vr 2.5i off
.vr .25i 1.15i 2.05i 2.95i 3.85i 4.75i ralign
.sp 2
.bx shade a 0 .5i / .9i 1.4i / 1.8i 2.3i / shade b 2.7i 3.2i / 3.6i 4.1i / 4.5i 5i
.vr off
.sp
.bx off
```

results in



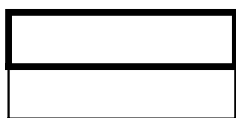
For more information on shading, see “.SD [Shading Definition]” on page 328.

- When a vertical rule is specified, an attempt is made to center the rule on the given horizontal position. If a horizontal position that is all the way to the left, such as 0mm, is given, the vertical rule cannot be centered, and the left edge of the vertical rule is placed at the specified horizontal position.

When respecifying vertical rules within a box, if the left most NEW vertical rule is centered on the same horizontal position as the left most OLD vertical rule, then the width of the horizontal rule is the same as the thickest of the OLD and NEW vertical rules. For example,

```
.dr thick weight .9mm
.bx thick 10mm 40mm
.sp
.bx 10mm 40mm
.sp
.bx off
```

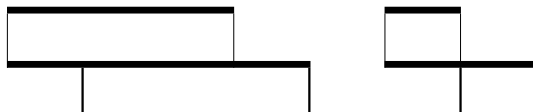
results in



- When vertical rules are respecified within a box, and the horizontal position of the left edge of the left most NEW vertical rule is to the right of the horizontal position of the left edge of the left most OLD vertical rule, the horizontal rule is that of the OLD vertical rule. For example,

```
.dr thick weight .9mm
.dr thin weight .1mm
.bx thick thin 0mm 30mm / 50mm 60mm
.sp
.bx 10mm 40mm / 60mm 70mm
.sp
.bx off
```

results in

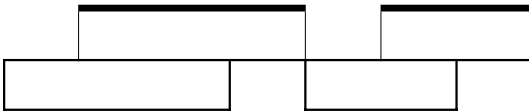


## .BX [Box]

- When vertical rules are respecified within a box, and the horizontal position of the left edge of the left most NEW vertical rule is to the left of, or equal to, the horizontal position of the left edge of the left most OLD vertical rule, the horizontal rule is that of the NEW rules. For example,

```
.dr thick weight .9mm
.dr thin weight .1mm
.bx thick thin 10mm 40mm / 50mm 70mm
.sp
.bx 0mm 30mm / 40mm 60mm
.sp
.bx off
```

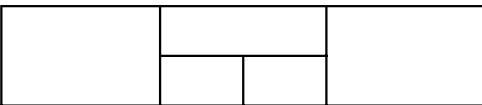
results in



- If a box is in progress and a .BX with horizontal parameters and slashes is encountered, the previous box is ended with a box bottom that is the mirror image of the previous box top. If you want to have the box bottom differ from the box top, you can change it with a .BX CAN and a .BX SET command. These commands cause the box bottom to be drawn as specified with the .BX SET command. For example, if you enter

```
.bx 6mm 27mm 49mm 70mm
.sp 1
.bx can
.bx set 6mm / 27mm 49mm / 70mm
.bx 6mm / 27mm 38mm 49mm / 70mm
.sp 1
.bx can
.bx set 6mm 27mm 38mm 49mm 70mm
.bx off
```

the result is



The CAN and SET parameters of .BX are used here to redefine the box. In this way, when the box is ended by .BX OFF or by another .BX specification, the box is ended with the horizontal rule drawn as it was specified with the .BX SET command.

---

## **.CB [Column Begin]**

### **Function**

The .CB [Column Begin] control word causes subsequent text to start a new column.

### **Syntax**

►►—.CB—————►◄

### **Notes**

- .CB causes a break.
- .CB ends a keep, float, footnote, named area, or table.
- .CB ensures that the page is started.

### **Remarks**

1. Use the .CB control word when you want to make the following text appear at the top of a new column. If the current column at the time .CB is encountered is the last column on the page, the column eject is the same as a page eject, because the next column is the first column of the next page. If you are at the top of a column, .CB still does a column eject.
2. The material following the .CB begins at the top of the new column and is moved to previous columns for the purpose of column balancing.
3. If a floating or delayed keep is waiting for the start of a new column, then the text that follows the .CB appears after the keep.
4. A column eject may be performed by certain other control words if the conditions warrant it. If this happens, the function is the same as the unconditional column eject that is caused by .CB. The other control words that can cause a column eject are:

.CC [Conditional Column Begin]  
 .H0–.H6 [Head Level 0–6]  
 .KP [Keep]

### **Examples**

The .CB control word ensures that the text following it appears at the top of a column:

```
.cb
This text falls
at the top of a column ...
```

---

## **.CC [Conditional Column Begin]**

### **Function**

The **.CC [Conditional Column Begin]** control word causes a column eject if less than a specified amount of space remains in the current column.

### **Syntax**

►► **.CC** v ►►

### **Parameters**

- v The amount of vertical space that must remain in the current column for processing to continue without a column eject. If the current column is empty and it is in the first section of the page, then no column eject is done and the previous column is subject to column balancing, if column balancing mode is in effect. If *v* is omitted and the current column contains text, a column eject is performed to the top of the next column. This is equivalent to the effect of **.BC [Balance Columns]**.

### **Notes**

- **.CC** causes a break.
- **.CC** ends a keep, float, footnote, named area, or table.
- **.CC** ensures that the page is started.

### **Remarks**

1. When the **.CC** control word is encountered, **SCRIPT/VS** checks to see if there is enough space left in the column. If there is not and the current column is not empty, a break followed by a column eject is performed. If the current column is not the last column on the page, the new column is made ineligible for column balancing with preceding columns, so that the material following the **.CC** is at the top of the new column.  
  
However, if the specified amount of space or more remains in this column or the column is empty, and if no column eject is done, subsequent column balancing may divide the text within the specified vertical space. To ensure that text is kept together, use the **.KP [Keep]** control word.
2. If a column is started explicitly by a **.CC [Conditional Column Begin]** control word, text from the new column is balanced only with following columns.

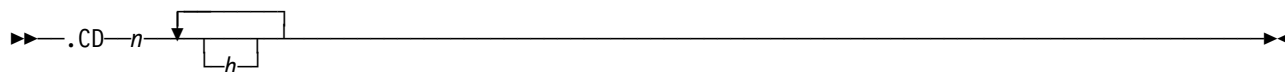
---

## .CD [Column Definition]

### Function

Use the .CD [Column Definition] control word to define how many columns are to be formatted and to position each column. Ordinary text is placed in these columns and flows from one column to the next as the columns are filled.

### Syntax



### Parameters

- n* The number of columns of output to be formatted onto each subsequent output page. The value *n* may be any number from 1 to 9.
- h* The horizontal positions of the standard columns, relative to the left edge of the page, given in horizontal space units. A position parameter of 0 indicates that the column should be flush with the left edge of the page, as determined by the BIND option of the SCRIPT command or the .PM [Page Margins] and .AM [Adjust Margin] control words. Up to nine columns can be defined.

**Initial Settings:** 1 0 46 92 0 0 0 0 0 0

### Notes

- The .CD control word causes a break and an unconditional section break.
- .CD ends a keep, float, footnote, named area, or table.
- The current column definition is included in the active environment.
- If no parameters are given, a section break occurs and columns are restarted using previous or initial settings.

### Remarks

1. The .CD control word causes a **section break** when it is processed. Therefore all the text up to that point is processed and positioned on the page using the old definition before the new definition becomes active, even if the new definition is the same as the old.
2. You can control the width of the gutter between columns by defining the column line length to a value less than the distance between column starting positions.
3. The positions of the columns do not control how wide the columns are to be; you control this by setting the column line length, using the .CL [Column Line Length] control word. If the current column line length is greater than the distance between columns, no gutter is created; the text of one column might overlay another column.

## **.CD [Column Definition]**

4. If you specify fewer positions than the number of columns and had previously specified positions on another .CD control word, those values remain in effect for any columns not respecified.

For example, if you specify

```
.cd 2 0 3i 6i
```

subsequent text is formatted into a two-column section. If you later specify

```
.cd 3
```

text is formatted into a three-column section, using the column positions given with the first .CD [Column Definition] control word.

If you specify “.CD *n*” without specifying any positions, and no previous column definition has been specified, the default values 0, 46, 92, 0, ... are used.

5. You could define columns that overlap other columns or *named* areas. The results may be undesirable, particularly on devices that do not provide overprinting.
6. If you use several different column formats in a document, you can create symbolic names (with the .SE [Set Symbol] control word) or macros (with the .DM [Define Macro] control word) to establish column definitions, column line lengths, and so on. If you use a single, one-column format and a single, multiple-column format, you can switch back and forth using the .SC [Single Column Mode] and .MC [Multicolumn Mode] control words.
7. Be careful to coordinate your .CD and .CL [Column Line Length] control words with the left margin, so that the text formatted in your columns does not exceed the width of the page.

## **Examples**

You may predefine columns without actually using them. For example, if you enter

```
.cd 2 0 10cm 20cm 30cm 40cm
```

five columns are defined, but text is formatted into only the first two of them. If you later enter

```
.cd 4
```

text is formatted into four of the five columns defined by the earlier .CD control word.



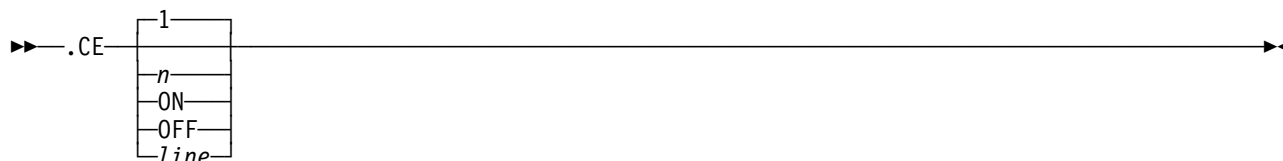
---

## **.CE [Center]**

### **Function**

Use the .CE [Center] control word to center output lines between the margins.

### **Syntax**



### **Parameters**

- n* Specifies the number of input lines to be centered. If omitted, 1 is assumed. If .CE *n* is specified when .CE ON is in effect, centering is turned off when *n* lines have been centered, or when a .CE OFF is encountered, whichever occurs first.
- ON** Specifies that subsequent text lines are to be centered.
- OFF** Terminates centering mode if it was ON, or if *n* has been specified and has not been exhausted.
- line* The line of text to be centered. The *line* is considered to start with the first nonblank character after the .CE control word.

### **Notes**

- .CE causes a break.
- This is a type 1 control word.
- The *line* form of .CE starts the page.
- The centering mode is included in the active environment.

### **Remarks**

1. The parameter ON or OFF, or the number of input lines to be centered (*n*), must be the only parameter on the control word line. A string of words that happens to start with one of these is interpreted as a single line to be centered. For example, the control word lines
 

```
.ce on top of old smokey
.ce 555 Bailey Ave.
```

 are taken to be of the .CE *line* form, not requests to turn centering on or for large numbers of lines to be centered.
2. The lines are centered between the current left margin, including any indent and offset values in effect, and the right margin. When centering is in effect, no formatting is done on the line. That is, the line is centered as it stands, and it is not filled from other input lines or justified. If a tab character appears in the line to be centered, the tab is resolved before the line is centered.
3. If the line to be centered is longer than the current column line length, the excess words are centered on a separate output line.
4. The use of ".CE *line*" while a ".CE *n*" is still in effect resets to zero the number of lines to be centered.

## **.CE [Center]**

5. The .RI [Right Adjust] control word is a variant of .CE. If either of these control words is processed, the other is canceled.
6. Contrast the .CE control word with .FO CENTER. The latter allows lines to be formatted by concatenating words until the line is nearly full, but then the filled line is centered instead of being justified, as would be the case with .FO ON.

## **Examples**

- To center one line, enter

```
.ce OFF THE RECORD
```

When this line of the file is displayed, the characters OFF THE RECORD are centered between the left and right indentation:

```
OFF THE RECORD
```

- To center several lines, enter

```
.ce on  
The Strummer Organization  
Clash Avenue, Erehwon  
61099  
.ce off
```

Each of the 3 lines between ON and OFF is separately centered:

```
The Strummer Organization  
Clash Avenue, Erehwon  
61099
```

---

## **.CG [Copy Group]**

### **Function**

Use the .CG [Copy Group] control word to specify the copy group that PSF (Print Services Facility) should use when printing the current and subsequent pages. If you only want to invoke an overlay, see “.OI [Overlay Include]” on page 262.

### **Syntax**

►► .CG *cname* ◄◄

### **Parameters**

*cname* The *cname* identifies the copy group that should be used by PSF when printing. *cname* can be up to 8 alphanumeric characters. A copy group is also known as a *medium map*.

### **Notes**

- .CG causes a break.
- .CG ends a keep, float, footnote, named area, or table.

### **Remarks**

1. For more information on copy groups, refer to the chapters listed in the following publications:
  - “Form Management Services: Making Modifications to Forms” in the *Print Management Facility User's Guide and Reference*.
  - “Form Definition and Page Definition” in one of the following publications:
    - *Print Services Facility/MVS Application Programming Guide*
    - *Print Services Facility/VSE Application Programming Guide*
    - *Print Services Facility/VM Application Programming Guide*
  - “Using Form Definition Control Commands” in the *Page Printer Formatting Aid User's Guide and Reference*.
  - “Getting a Little More Oriented” in one of the following publications:
    - *Print Services Access Facility for MVS User's Guide and Reference*
    - *Print Services Access Facility for VM User's Guide and Reference*.
2. The .CG control word is applicable only to AFP printers. For all other devices .CG causes a break, but is otherwise ignored.
3. This control word takes effect on the page on which the control word is encountered. Each .CG control word encountered on a page causes a change in the copy group. Identical consecutive copy groups on the same page are ignored. All “constant forms” copy groups are processed. Only the last “variable data” copy group is processed.
4. The user must ensure that any copy group named on the .CG control word exists in the FORMDEF used by PSF when the document is printed because SCRIPT/VS doesn't check this.
5. The first copy group in the FORMDEF is the default.
6. If you are using separation masters with a .CG control word, the copy group is invoked on all masters of the page. See “.SM [Separation Master]” on page 353 for information on the .SM control word.

## **.CG [Copy Group]**

7. If only one copy group is specified on the page and it is the same as the last copy group used on the previous page, the copy group is not changed. A copy group on the first page of a document containing a .CG control word is always processed, even if the *cgname* is for the default copy group.
8. When printing on a 3820 or 3812 Page Printer, a copy group change causes an eject to a front-facing page. When duplexing at print time, a copy group change on a back-facing page causes a blank, unnumbered page. The formatted output appears on the next front-facing page. To number the blank back-facing page, insert a page eject before the .CG control word. The copy group change occurs on the next front-facing page, and the printed output is as expected.

For example,

```
. * When front-facing pages are odd-numbered  
.if SYSPAGE = EVEN .pa  
.cg xxxx
```

or

```
. * When front-facing pages are even-numbered  
.if SYSPAGE = ODD .pa  
.cg xxxx
```

where xxxx is the name of the copy group.

---

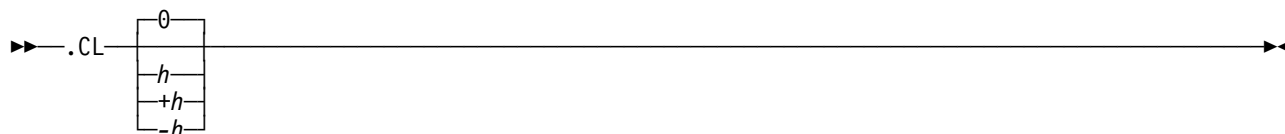
## **.CL [Column Line Length]**

### **Function**

The .CL [Column Line Length] control word sets the line length of each column of SCRIPT/VS output.

Figure 6 on page 438 shows the relationship of the .CL [Column Line Length] to the layout of a SCRIPT/VS output page.

### **Syntax**



### **Parameters**

*h* Specifies the line length of each column of formatted output. It should not be larger than the page width. It can be expressed in any valid horizontal space units.

If  $+h$  or  $-h$  is specified, the current column line length is increased or decreased accordingly.

### **Notes**

- .CL causes a break.
- The column line length is included in the active environment.

### **Remarks**

1. The .CL control word should be used in conjunction with the .CD [Column Definition] control word to define the line length of each column from the displacements given. If the column line length is greater than the space left between columns, the columns may overlay each other. An intercolumn gutter may be nested by making the column line length shorter than the distance between columns.
2. The initial width of running headings and footings is set by the .LL [Line Length] control word, not the column line length.
3. If the column line length has never been set explicitly, it has the same value as the line length. If you set the column line length to zero (.CL 0), the column line length is reset to the value of the current line length. Note that changing the column line length by means of the .LL control word means that the column line length change takes effect immediately if it is valid given the current page margins and page width settings.
4. Be careful to coordinate your .CL and .CD [Column Definition] control words with the page binding, so that text formatted to column line length does not exceed the width of the page.

---

## .CM [Comment]

### Function

Use the .CM [Comment] control word to place comments within a SCRIPT file.

### Syntax

►► .CM └───┬───►  
          |  
          └───┬───►  
              |  
              └───►  
              *comments*

### Parameters

*comments* Can be anything; this input line is not used in formatting the output. However, because this is a control word, the input line is scanned for control word separators and symbol substitution is performed.

### Remarks

1. The .CM control word allows comments to be stored in the SCRIPT file for future reference. These comments can be seen when you edit the file, or when you print it using the UNFORMAT option of the SCRIPT command.

The comments can also be used to store unique identifications that can be useful when attempting to locate a specific region of the file during editing.

2. If you want an entire line to be ignored and not scanned for control word separators, you can use another form of comment. Any line that begins with .\*. \*. is not considered to be a control word, but .CM is.

### Examples

- If you wanted to remind yourself to update the date in a text but you did not want the reminder printed, you could enter

.cm Remember to change the date.

and this line is seen only when you examine an unformatted listing of the SCRIPT file.

- You can use the .CM control word on a line that contains a control word separator. For example, if you enter

.cm Just a note.;Resume text.

then your note is not printed but the text following the control word separator is:

Resume text.

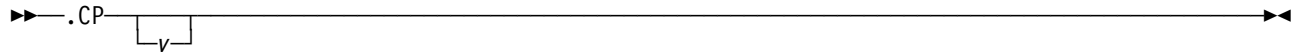
---

## **.CP [Conditional Page Eject]**

### **Function**

The .CP [Conditional Page Eject] control word causes a page eject to occur if less than the specified amount of space remains in the current column.

### **Syntax**

➤➤ .CP 

### **Parameters**

- v* The amount of vertical space that must remain in the current column for additional lines to be processed without a page eject. If *v* is omitted, a break and a page eject is done, if necessary, to get to the top of a page. A break and a page eject is not done if the current page is empty.

### **Notes**

- The .CP control word without a *v* parameter causes a break and an unconditional section break.
- .CP ends a keep, float, footnote, named area, or table.
- .CP ensures that the page is started.

### **Remarks**

1. You can use the .CP control word to guarantee that enough space (up to the maximum column depth) exists in one column to accommodate blank space left by .SP [Space] for a figure to be inserted later.
2. If you request a .CP and there is text yet to be placed in your document, as a result of a pending float, for example, it may be necessary for SCRIPT/VS to perform more than one page eject in order to satisfy the .CP request.
3. To keep formatted text together, use the .KP [Keep] control word.
4. You should use this control word only if you are in single-column mode, or if you are at the beginning of a section (for example, right after a .CD control word).

### **Examples**

If you enter

.cp 2i

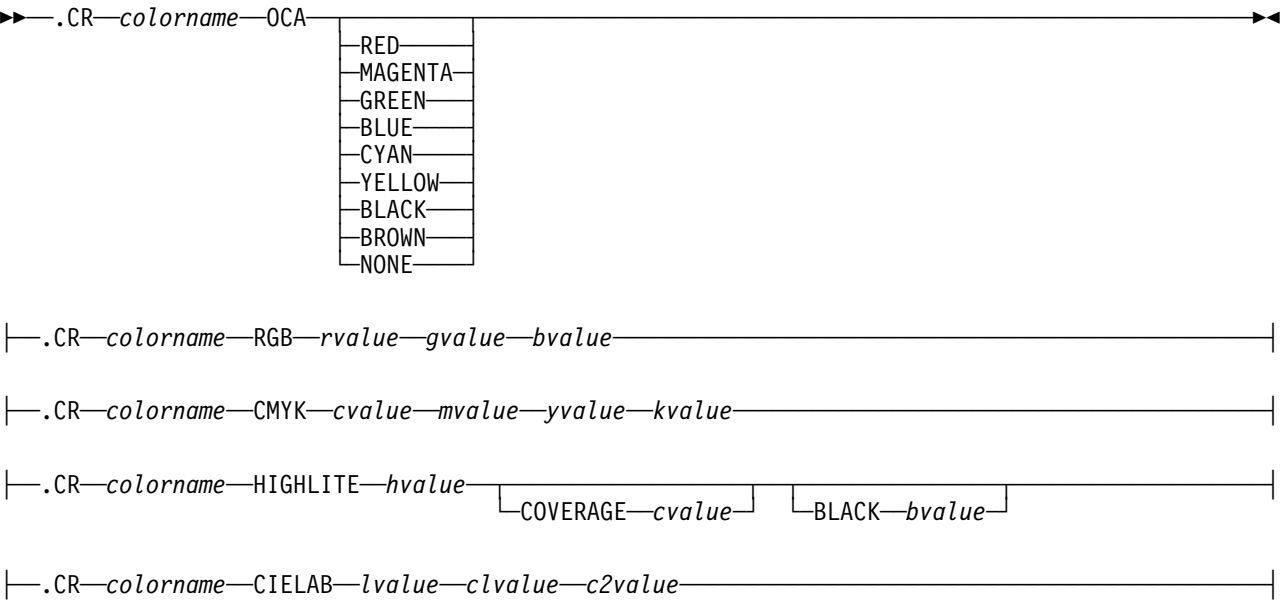
then, if less than two inches of space remain on the current column, a page eject is issued before processing continues. If two inches or more remain, processing continues on the current column.

.CR [Color]

Function

Use the .CR [Color] control word to define a “named color” for use on the .BX [Box], .DA [Define Area], .DF [Define Font], .IO [Include Object], and .TD [Table Definition] control words.

Syntax



Parameters

- colorname** The name of the color to use on subsequent .BX [Box], .DA [Define Area], .DF [Define Font], .IO [Include Object], .DR [ ], and .TD CELL [Table Definition] control words. *colorname* can consist of up to 16 National characters.
- OCA** Defines a standard OCA color model. Any of the colornames shown can be specified. NONE specifies that no color is associated with the object (the output is not visible). This is the same as “color of medium.”
- RGB** Defines the red, green, and blue color model. *rvalue* is an integer that specifies the red value and is specified as a percentage from 0 to 100. *gvalue* is an integer that specifies the green value and is specified as a percentage from 0 to 100. *bvalue* is an integer that specifies the blue value and is specified as a percentage from 0 to 100.
- CMYK** Defines the cyan, magenta, yellow, and black color model. *cvalue* is an integer that specifies the cyan value and is specified as a percentage from 0 to 100. *mvalue* is an integer that specifies the magenta value and is specified as a percentage from 0 to 100. *yvalue* is an integer that specifies the yellow value and is specified as a percentage from 0 to 100. *bvalue* is an integer that specifies the black value and is specified as a percentage from 0 to 100.
- HIGHLIGHT** Defines the highlight color model. This color space defines a request for a highlight color. Highlight colors are specific to a particular device. *hvalue* is an integer from 0 to 65535. An *hvalue* of zero indicates the presentation device default color, which is black for most devices. An *hvalue* over 65535 causes unpredictable results. The HIGHLIGHT color model is specific to the IBM InfoPrint Hi-Lite Color Post Processor.



<b>COVERAGE</b>	Indicates the percentage of the highlight color to use. <i>cvalue</i> is a percentage from 0 to 100. The COVERAGE value added to the BLACK value cannot exceed 100. If the COVERAGE value plus the BLACK value is less than 100, the remaining coverage is achieved with the color of the medium. If COVERAGE is not specified, 100 minus the BLACK value is used.
<b>BLACK</b>	Indicates the percentage of black to add to the highlight color. <i>bvalue</i> is a percentage from 0 to 100. The BLACK value added to the COVERAGE value cannot exceed 100. If BLACK is not specified, 0 is used.
<b>CIELAB</b>	Defines the CIELAB color model. <i>lvalue</i> specifies the lumincent and is specified as a decimal number from 0 to 100, with up to two decimal positions. <i>clvalue</i> and <i>c2value</i> specify the chrominance differences and are specified as integers from -127 to 127.

## Remarks

1. OCA color is supported by 4224 physical devices. All color schemes are supported by AFP2 physical devices. Color is ignored for any other physical device.
2. The degree of color support is determined by the presentation system that renders the SCRIPT/VS output file.
3. *colorname* cannot be OLDCOLOR or any of its abbreviations, because OLDCOLOR is a keyword of the .DF [Define Font] control word.
4. Shading can be accomplished by using the HIGHLIGHT keyword. Specify an *hvalue* of 0 to indicate presentation device default color, and specify the COVERAGE keyword to indicate the percentage of highlight color to apply.
5. Only the OCA color scheme is allowed when specifying a color for an included object (.IO [Include Object] control word) which is IOCA, BCOCA, GOCA, or a page segment.
6. Default colornames corresponding to the OCA color values are already defined. You can use any of the OCA color values as colornames without defining that colorname with a .CR control word.
7. The HIGHLIGHT color scheme is device specific. Results will vary from presentation device to presentation device.
8. For the HIGHLIGHT color scheme, the COVERAGE value plus the BLACK value cannot add up to more than 100. An error results if this value is higher than 100. If the COVERAGE and BLACK values add up to less than 100, the remainder of the highlighting is achieved with color of medium.
9. When the background color in areas or table cells overlap, overprinting occurs. The last color placed wipes out everything under it. The order in which things are placed on the page might not be the order in which they are found in the input file.

## Examples

- The following example shows how to define and use a font with the OCA color magenta:

```
.df magenta color magenta
.bf magenta
This text is printed in the color magenta when formatting for a device
which supports color.
.pf
This text is printed in the default font and color.
```

A .CR control word was not required for the above example because all of the OCA colors are predefined in SCRIPT/VS.

## .CR [Color]

- To define a RGB color for your text, use the following example:

```
.cr purple rgb 30 0 60
.df purple color purple
.bf purple
```

This text is printed in a color made up of 30 percent red and 60 percent blue when formatted for and printed on a device that supports color.

```
.pf
```

This text is printed in the default font and color.

- The following example defines a box that has a background color defined in the CMYK color model. The text in the box is printed in a font that uses the OCA color model.

```
.cr ygreen cmyk 10 10 50 0
.df blue color blue
.da area 0 0 width 4i section color ygreen font blue
.ar area on
```

This area has a background color of yellow green created with the CMYK format.

This text is printed in blue.

```
.ar off
```

```
.ar put
```

- The following example uses two colors defined in the CMYK color model to color the rules of a box magenta and the background of the box cyan:

```
.cr magenta cmyk 0 70 0 0
.cr cyan cmyk 60 0 0 0
.dr magenta color magenta
.bx magenta color cyan left right
```

```
.sp
```

```
.in .li
```

This is stuff in the box, the rules are magenta and the backgroun is cyan.

```
.bx off.
```

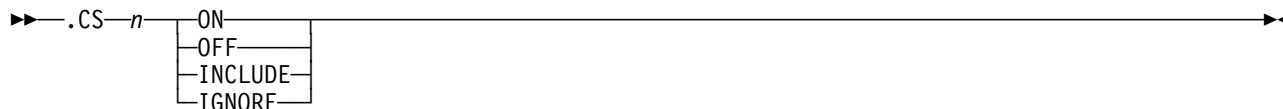
---

## **.CS [Conditional Section]**

### **Function**

The **.CS [Conditional Section]** control word allows you to designate sections of the input file that are to be conditionally included or ignored in the document.

### **Syntax**



### **Parameters**

- n*** Specifies the conditional section code number from 1 to 9.
- ON** Marks the beginning of conditional section *n*.
- OFF** Marks the end of conditional section *n*.  
If you specify **.CS *n* OFF**, there must be a blank between the control word and the first parameter.
- INCLUDE** Indicates that SCRIPT/VS is to process all the input lines between the ON and the OFF control words for conditional section *n*.
- IGNORE** Indicates that SCRIPT/VS is to bypass every line for conditional section *n* that falls between **.CS *n* ON** and **.CS *n* OFF**.

**Note:** Conditional section definitions are included in the active environment.

### **Remarks**

1. The **.CS** control word allows you to designate specific sections of your input file that can be ignored or included conditionally. You can have as many as nine separate section codes and can specify which section numbers are to be included and which are to be ignored. Each section code can be used for many sections. The **ON** and **OFF** operands identify the beginning and end of a conditional section; the **INCLUDE** and **IGNORE** operands indicate whether SCRIPT/VS should process the input lines within the conditional sections.
2. You can use conditional section codes to separate sections of a document that apply to different versions, and you can specify which version is to be formatted. You can also use this technique to identify sections of a manual that you may sometimes want to exclude.
3. Because the **.CS** control word does not cause a break, you can turn conditional sections on and off within a paragraph or even within a sentence without disrupting normal output formatting.
4. By default, all conditional section codes are assumed to be set to **INCLUDE** unless explicitly set to **IGNORE**.
5. A conditional section can contain SCRIPT/VS control words as well as text. If the section is ignored, all control words contained in that section are ignored, except  
     **.cs *n* off**  
     that marks the end of the section.

## **.CS [Conditional Section]**

6. Conditional section definitions can be nested (that is, a conditional section can contain another conditional section). A nested section is included only if all outer nestings specify INCLUDE. Otherwise, the inner nesting is never processed, because it is part of an outer section that has been ignored. If a conditional section is nested within another one, the entire section should be enclosed by the outer section.
7. The .CS control word can be used in conjunction with the .RC [Revision Code] control word to mark the conditional sections. The .TE [Terminal Input] control word can be used in interactive environments to specify which sections are to be included during processing of the input file.

## **Examples**

This example shows the use of the .CS control word to define two conditional sections and to designate that only one is to be used:

```
.cs 1 ignore
.cs 2 include
.
.
In this version of the system
there can be only
.cs 1 on
256
.cs 1 off
.cs 2 on
1000
.cs 2 off
entries in a MACLIB file.
```

Because only conditional section code 2 is to be included, the generated output line is  
In this version of the system there can be only 1000 entries in a MACLIB file.

---

## **.CT [Continued Text]**

### **Function**

The `.CT [Continued Text]` control word causes the line given to be treated as a continuation of the previous text line. If no line is given, the control word means “continue nothing,” which cancels continuation that may be in effect from a continuation character on the previous text line.

### **Syntax**

►► `.CT` *line* ◄◄

### **Parameters**

*line* The *line* to be considered a continuation of the previous text input line, even if the previous line did not end with a continuation character. The formatter normally considers that no word can span input lines. A text word can span input lines if the first line ends with a continuation character defined with the `CONT` parameter of the `.DC [Define Character]` control word or if the second line is the parameter of the `.CT` control word.

A more complete discussion of continuation can be found under the `CONT` option of the `.DC [Define Character]` control word.

### **Notes**

- `.CT` ensures that the page is started.

### **Remarks**

1. If *line* is omitted, continuation is ended even if the previous text line ended with a continuation character.
2. If a `.CT [Continued Text]` control word follows a control word that causes a break, then the text associated with the `.CT` control word becomes the first text of a new output line.
3. If the `.CT [Continued Text]` control word immediately follows a `.RE [Restore Environment]` control word, continuation may or may not take effect, because the `.RE` control word, under certain circumstances, causes a break. See “`.RE [Restore Environment]`” on page 309 for more details.
4. The `.CT` control word cannot be used to continue control word parameters across multiple input lines.

---

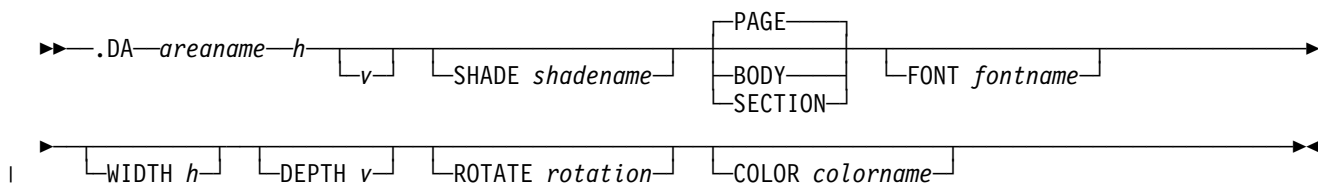
## .DA [Define Area]

### Function

Use the .DA [Define Area] control word to define a *named* area. A *named* area can receive formatted text lines in the same way a normal column, keep, float or footnote can. But the text in a *named* area is placed at a predefined location on the page and can overlay other text.

Once a named area is defined, you can use the .AR [Area] control word to identify text to be formatted into the area.

### Syntax



### Parameters

<i>areaname</i>	Name of the area to be defined. After the areaname parameter has been defined, it may be used with the .AR [Area] control word. The areaname parameter cannot be OFF or PUT. It can contain a maximum of 16 national characters.
<i>h</i>	Horizontal displacement to the upper-left corner of the area from the left edge of the region where this area is placed. A value of zero indicates that the area should be flush against the left edge of the region. The regions are designated with the PAGE, BODY, and SECTION parameters.
<i>v</i>	Vertical displacement to the upper-left corner of the area from the top of the region where this area is placed. A value of zero indicates that the area should be flush against the top edge of the region. <i>v</i> is ignored for SECTION areas.
<b>SHADE</b>	Specifies that the defined area is to be shaded with the named shading definition.  The <i>shadename</i> specifies the name of the shading definition. The name can be a maximum of 16 national characters, and it is not case sensitive. The shading definition must have been defined with the .SD [Shading Definition] control word before specifying the .DA control word and its associated parameters.
<b>PAGE</b>	Specifies that the area is to be positioned relative to the page. The horizontal and vertical displacement of the area are measured from the upper-left corner of the page. Page areas can be placed anywhere on the page and can overlay running headings and footings as well as body text. Page areas are automatically placed every time a page is ended.
<b>BODY</b>	Specifies that this area is to be positioned relative to the body of the page. The horizontal and vertical displacement of the area are measured from the upper-left corner of the body of the page. That is, the origin is just below the running heading and at the current left margin for the page. Body areas can be placed anywhere within the body and can overlay body text. Body areas are automatically placed every time a page is ended.

<b>SECTION</b>	Specifies that this area is to be placed in a body section. The horizontal displacement is relative to the current left margin, and any vertical displacement is ignored. Section areas are not automatically placed; you must request that they be placed with the PUT parameter of the .AR [Area] control word. If you don't request placement of section areas, they do not appear in the document.
<b>FONT</b> <i>fontname</i>	Initial font to be used in formatting text in the area.  If no initial font is specified for a <i>named</i> area, then the current font remains unchanged when the <i>named</i> area is started with the .AR [Area] control word.
<b>WIDTH</b> <i>h</i>	Initial column line length of the area. If WIDTH is not specified, the current column line length remains unchanged when the <i>named</i> area is started with the .AR [Area] control word.
<b>DEPTH</b> <i>v</i>	Maximum depth of the area given in vertical space units. No more than the specified depth is placed on one page or section. If no depth is given or the value specified exceeds the maximum available, then the maximum available in the region is used.  For PAGE areas, the maximum depth is the distance from the position specified as <i>v</i> to the bottom of the page. For BODY areas, the maximum depth is the distance from the position specified as <i>v</i> to the bottom edge of the body. For SECTION areas, the maximum depth is the distance from the top of the current section when the area is placed to the bottom edge of the body. As many sections as needed are put on the page to place the entire area.  If more material remains in an area after the specified or maximum available depth has been placed, the remainder is saved for the next section or page.
<b>ROTATE</b>	Gives the angle of rotation of the area. The rotation parameter is given in degrees and must be one of these multiples of 90°:  -270 -180 -90 0 90 180 270  Rotation is done in a clockwise direction about the upper-left corner of the area. This parameter is ignored for all devices that do not support rotation. The 180° rotation is not valid for the 3800 Printing Subsystem Model 3. The ROTATE parameter of the .DA control word is ignored when specified for section areas.
<b>COLOR</b>	Specifies the color to be used as the background color for this area. The COLOR parameter applies only to printers with color capability. "colorname" can be any color previously defined with the .CR [Color] control word.

## Notes

- .DA ends a keep, float, footnote, named area, or table.

## Remarks

1. The font name specified with the .DA control word can be a list of fonts previously defined with the LIST parameter of the .DF control word. For more information on using the LIST parameter, see the section on the .DF [Define Font] control word.
2. Areas may be defined that overlap other areas or columns. The results may be undesirable, particularly on devices that do not provide overprinting.
3. You can shade areas using the SHADE parameter on the .SD [Shade] control word. The SHADE parameter is ignored if you are formatting for a line device or for a 4250 printer. For more information about the SHADE parameter, see ".SD [Shading Definition]" on page 328.

## .DA [Define Area]

4. Section areas may overlap each other and may be overlapped by page and body areas, but they occupy complete sections. This prevents section areas from overlapping normal body text (text outside any area). Any vertical rules in the normal body that are currently active in the body continue through the section area. If the body has multiple columns, only the vertical rules in the first column of the body continue through the section area, because section areas are handled in single-column mode.
5. For the 1403, SCRIPT/VS assumes that the channel-one punch of the carriage tape is located one-half inch below the top edge of the page. The default top margin for 1403 devices is one-half inch, and SCRIPT/VS issues a Skip Immediate To Channel One carriage control to advance to each new page.  
  
If you set the top margin to less than one-half inch or try to place an area less than one-half inch from the top of the page, a blank page precedes the first page of output, and SCRIPT/VS spaces from each page to the next. If you also change the page length, the top of the logical page may not coincide with the top of the physical page.
6. Be careful to coordinate your .DA control words with the page binding so that the text formatted into your areas does not exceed the page width.
7. Specifying  $-90^\circ$  rotation is the equivalent of specifying  $270^\circ$  rotation.
8. The .AM [Adjust Margin] control word affects section and body areas, but not page areas. The .AM control word also applies to the starting position of section and body areas, but not to text within the page area.
9. Certain control words are not allowed within a named area. If one of the disallowed control words is encountered, the named area is immediately ended, and the disallowed control word is executed. A warning message is issued, identifying the control word that ended the named area. See Table 15 on page 442 for a list of the disallowed control words.

## Examples

- A pair of **SECTION** areas may be defined to facilitate formatting parallel translations of text in different languages. For example:

```
.da french 0 section width 8cm
.da english 9cm section width 8cm
.sp
.ar french on
.
. (french translation of text)
.
.ar off
.ar english on
.
. (english translation of text)
.
.ar off
.ar put
```

As a result of the .AR PUT control word, the text formatted into the *named* areas is placed side-by-side on the page. If the material in one or both areas exceeds the depth of the page, subsequent pages contain the remainder.

- If your device allows rotation, a rotated *named* area may be defined by entering

```
.da thumb 6i 10i page width 1.0i rotate -90
```
- With the following definition, you can create shaded thumb-tabs that could be used for quick reference to chapter headings:



```
.sd stthumb shade light
.da thumb 50p2 47p width 1.0i rotate 90 shade stthumb
.ar thumb on
.bx left right
.ce Chapter 1
.bx off
.ar off
```

which results in

By using a definition similar to the examples shown here, you can shade the thumb-tabs for an entire publication.

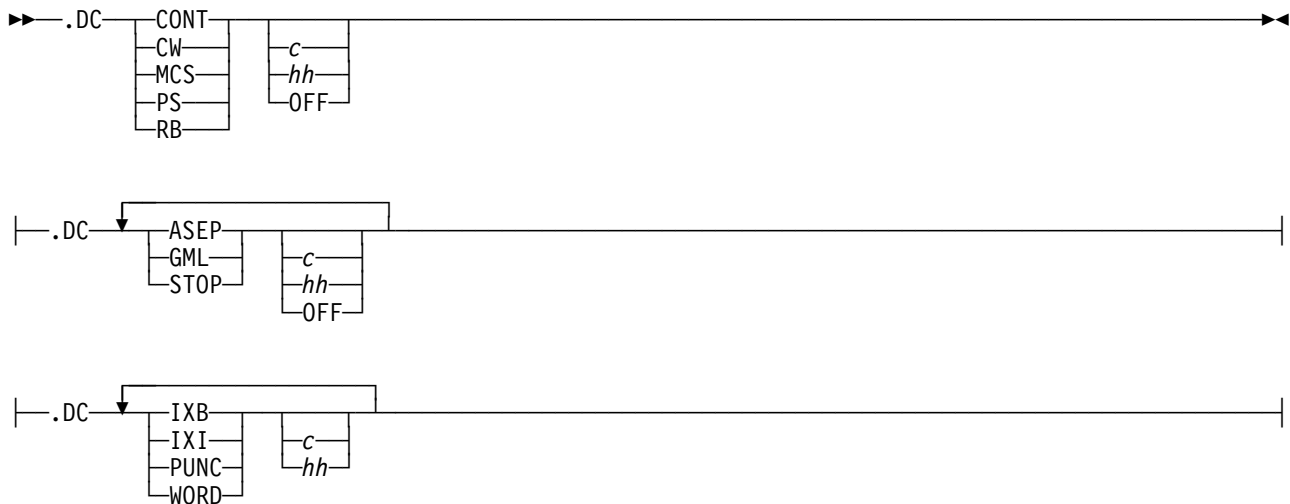
---

## .DC [Define Character]

### Function

Use the .DC [Define Character] control word to define various special characters that the formatter recognizes as having a special significance.

### Syntax



### Parameters

**c** Specifies the character (or characters) to be recognized. *c* may be any single character.

**hh** Specifies the character (or characters) to be recognized. *hh* is expressed as two hexadecimal digits.

**OFF** Causes the character (or characters) to be undefined.

**ASEP** Allows the definition of up to *four* characters that are to be used to separate array elements when an array is substituted in a document using the &name(\*) form. All characters to be used to separate array elements must be specified, including blank characters (X'40').

The initial array-separator characters are comma (,) and blank. The characters given with ASEP replace the previous array-separator characters.

**CONT** Defines a continuation character for text lines. SCRIPT/VS normally adds a word space between words on different input lines. The continuation character allows words to be built from fragments entered on several input lines. When the last character of an input text line is a continuation character, the normal interword space is not added when this line is concatenated to the next, but existing blank characters preceding the line continuation character are retained.

The line continuation character is recognized at the end of a line, whether the line contains text or control words, or a mixture of both. The continuation character may not be used to extend a control word line, but it may extend the *text data* that is associated with that control word.

Initially there is no continuation character. The character given with CONT replaces the previous continuation character.

**CW** Specifies the character to be used to separate control words on a single line.<sup>6</sup>

The initial control word separator is a semicolon (;). The character given with **CW** replaces the previous control word separator.

**GML** Defines delimiters for Generalized Markup Language tags and end-tags. GML tags and end-tags may be separately defined by the .AA [Associate APF] and .GS [GML Services] PREFIX control words. If only one character is specified, it is taken to be the GML tag delimiter, and GML end-tags are not recognized.<sup>6</sup> If two characters are specified, the second is taken to be the GML end-tag delimiter. If three characters are specified, the second and third characters are taken to be the GML end-tag delimiter.

The GML delimiter may not be a blank, a period (.), or an ampersand (&), or any character that is valid in a tag name.

The initial GML tag and end-tag delimiters are a colon (:) and a double-colon (::), respectively. Using .DC GML without giving a delimiter restores the initial values. The characters given with GML replace the previous GML delimiters.

<u>Code</u>	<u>Character</u>	<u>Code</u>	<u>Character</u>	<u>Code</u>	<u>Character</u>
05	Tab	4F		6F	?
11	Special Blank <sup>1</sup>	5A	!	7A	:
12	Special Blank <sup>1</sup>	5B	\$	7E	=
13	Special Blank <sup>1</sup>	5C	*	7F	"
16	Backspace	5D	)	8B	{
40	Blank	5E	;	9B	}
41	Required Blank <sup>2</sup>	5F		A0	-
4A	¢	61	/	AD	[
4B	. (Period)	6B	,	AF	•
4C	<	6C	%	BD	]
4D	(	6D	—		
4E	+	6E	>		

**Note:**

<sup>1</sup> Special blanks are used for justification in documents formatted for the 3800 Printing Subsystem.

<sup>2</sup> The required blank is a blank that cannot have space added to it during justification. The code point assignment of the required blank can be changed with the .DC [Define Character] control word.

Table 7. Characters That Delimit Words for Spelling Verification. The .DC [Define Character] control word can be used to add additional character word delimiters.

**IXB** Defines characters that are to be treated as blanks when sorting index entries. Characters designated with .DC IXB do not participate in the sort, but they still occupy a character position. This is true even if the blank itself is designated as an IXI (index ignore) character.

Initially, only hexadecimal 40 is treated as a blank when sorting index entries.

The characters given with IXB are added to the current list of characters already requested.

**IXI** Defines characters that are to be ignored when sorting index entries. Characters to be ignored are completely removed from index entries for purposes of sorting.

Initially, no characters are ignored when sorting index entries.

The characters given with IXI are added to the current list of characters already requested.

<sup>6</sup> If the control word separator character, the GML delimiter character, or the markup content separator is set to hexadecimal X'00' or X'40', they are undefined, just as if .DC CW OFF, .DC GML OFF, or .DC MCS OFF, respectively, had been specified.

## **.DC [Define Character]**

- MCS** Defines a markup/content separator character to be recognized, in addition to the blank, as the delimiter between GML tag attributes and the text that follows them.
- The initial markup/content separator character is the period (.). The character given with MCS replaces the previous markup/content separator.
- PS** Specifies the character to be used as the page number symbol. Every page number symbol in running headings and footings is replaced with the current page number every time the running heading or footing is formatted for a new page.
- The initial page number symbol is an ampersand (&). The character given with PS replaces the previous page number symbol.
- PUNC** Specifies characters that are to be recognized as punctuation for spelling verification. When punctuation characters occur within a word, they are retained when the word is checked against the dictionary; if they occur at the end of a word, they are removed before checking takes place.
- The initial punctuation characters are the hyphen (-) and single quote ( ' '). The characters given with PUNC are added to the current list of characters already requested.
- RB** Defines the character to be used as a required blank. Required blanks are not recognized as interword spaces for justification.
- The initial required blank is hexadecimal X'41'. The character given with RB replaces the previous required blank character. Do not attempt to change the required blank to a hexadecimal X'00', X'05', X'16', or X'40'. The results are unpredictable.
- STOP** Specifies the characters to be recognized as full-stop or “end of sentence” characters. When full-stop characters occur at the end of an input line, or precede a double quotation mark or right parenthesis character at the end of a line, SCRIPT/VS inserts an extra blank before concatenating the following input line. If the same character is defined as a continuation character and a stop character, it is interpreted as a continuation character if it occurs at the end of the line. The width of these extra blanks is determined by the .ES [Extra Space] control word.
- The initial full-stop characters are the period (.), question mark (?), exclamation point (!), and colon (:). The characters given with STOP replace the previous full-stop characters.
- WORD** Specifies characters that are to be interpreted as word delimiters during spelling verification. The spelling of each string marked with word delimiter characters is separately verified.
- The initial word delimiter characters are listed in Table 7 on page 113.
- The characters given with WORD are added to the current list of characters already requested.

**Default:** Restores the initial setting for the specified parameter.

**Note:** The continuation character (.DC CONT), control word separator (.DC CW), GML tag delimiter (.DC GML), markup content separator (.DC MCS), and page number symbol (.DC PS) are included in the active environment.

## **Remarks**

1. If the formatter control or text that follows the line continuation character causes a break, continuation is canceled for that line. A null line also cancels continuation for the previous line.
2. Care is required when using the CW parameter if the control word separator might already be set to the given character. For example, if the control word separator is already the question mark (?) and you enter  

```
.dc cw ?
```

  
this is processed as

```
.dc cw
.nl
```

because the question mark is interpreted as a control word separator. The result is to restore the initial control word separator, the semi-colon (;).

You can safely change the control word separator when it may already be set to the desired character in several ways:

- Turn off the control word separator before changing it:

```
.dc cw off
.dc cw ;
```

- Specify the new control word separator character in hexadecimal:

```
.dc cw 5e
```

- Use the control word modifier to suppress control word separator processing on the input line:

```
.'dc cw ;
```

3. Do not confuse the page number symbol with the ampersand used with the .SE [Set Symbol] control word. If you enter

```
.se currpage = &
```

the symbol &currpage is set to the current page number, regardless of the current page number symbol.

4. The blank is always recognized as a markup/content separator for GML tags, in addition to the character given with the MCS parameter.

5. The following hexadecimal code points have fixed meanings when appearing in text:

```
05  Tab
16  Backspace
40  Blank
```

These code points may not be used as the required blank character.

6. A character cannot be both a word delimiter and a punctuation character. If the same character is specified with both the PUNC and WORD parameters, the last specification is used.
7. Each language, specified with the .DL [Dictionary List] control word, has a unique word delimiter and punctuation table. Characters specified with PUNC and WORD are added to the table for the current language.
8. The current settings of several .DC parameters are available in SCRIPT/VS system symbols:

```
&$CONT  Continuation character
&$CW    Control word separator
&$EGML  GML end-tag delimiter
&$GML   GML tag delimiter
&$PS    Page number symbol
&$RB    Required blank.
```

9. The initial values for each .DC parameter are:

```
ASEP    ,40
CONT    (none)
CW      ;
GML     : : :
IXB     (none)
IXI     (none)
MCS     .
```

## .DC [Define Character]

<b>PS</b>	<b>&amp;</b>
<b>PUNC</b>	<b>- ' ,</b>
<b>RB</b>	<b>41</b>
<b>STOP</b>	<b>. ! ? :</b>
<b>WORD</b>	(See Table 7 on page 113.)

## Examples

- ASEP (Array separators)

Suppose you have defined a symbol array containing the names of geographical locations:

```
.se places() = 'Korea
.se places() = 'Mainz
.se places() = 'Venice
```

When the symbol &places(\*) is found, it is replaced with the entire symbol array, and the elements of the array are separated by the array separator characters. For example, the initial setting of the array separator characters is a comma (,) and a blank, so if you enter

```
At: &places(*)
```

the result of symbol substitution is

```
At: Korea, Mainz, Venice
```

You can specify up to four characters with the ASEP parameter. You can insert more than four characters, including control words, between array elements by setting the array separator characters to a symbol. For example:

```
.se x = ' and '
.dc asep &x .
At: &places(*)
```

The intermediate result of symbol substitution is

```
At: Korea&x.Mainz&x.Venice
```

The final result of symbol substitution is

```
At: Korea and Mainz and Venice
```

You can set the array separator to cause a break by specifying

```
.'se x = ' ;.br;'
.dc asep & x .
```

- CONT (Continuation character)

The continuation character is recognized only when it appears at the end of an input line. When it appears in the middle of a line, it is treated as an ordinary text character. For example,

```
.dc cont +&$RB
.us on
Underscored + cap+&$RB
.up ital+&$RB
ized.
.us off
```

results in

```
Underscored + capITALized.
```

The continuation character can be used to create one “logical” input line from a number of input lines. For example,

```
.dc cont -
.ce 1
this is a sin-&$RB
.up gle line
.dc cont +
```

results in the line

```
      this is a sinGLE LINE
```

- **CW (Control word separator)**

The control word separator character can be used to enter several control words on a single input line. For example, the input lines

```
.dc cw /
.sk&slaxh..of 5/(iv) Next, ...
```

are equivalent to

```
.sk
.of 5
(iv) Next, ...
```

- **GML (GML delimiters)**

The GML delimiters indicate the beginning of GML tags and end-tags; the end of a GML tag is indicated by a markup/content separator, set with the MCS parameter. For example, the initial GML tag delimiter is the colon (:); the initial GML end-tag delimiter is a double colon (::). A figure can be marked up with the FIGURE tag by entering

```
:figure
.
.
.
::figure
```

If you change the GML delimiters by entering

```
.dc gml ! ! e
```

then a figure can be marked up by entering

```
!figure
.
.
.
!efigure
```

- **IXB (Index blank)**

By default, “Mother Superior” is placed before “mother-in-law” in an index. If you specify

```
.dc ixb -
```

the latter entry is sorted as if the entry were “mother in law” and placed before “Mother Superior.”

- **IXI (Index ignore)**

By default, “William Steinburg” is placed before “Williamsburg” in an index. If you specify

```
.dc ixi 40
```

then the former entry is sorted as if the entry were “Williamsteinburg” and is placed after “Williamsburg.”

- **MCS (Markup/content separator)**

## .DC [Define Character]

The markup/content separator indicates the end of a GML tag and its attributes. For example, the default GML delimiter is a colon (:) and the default markup/content separator is a period (.). Therefore a figure reference can be marked up as

See :figref refid=plugh..

If you change the GML delimiter and markup/content separator by entering

```
.dc gml <
.dc mcs >
```

then a figure reference can be marked up as

See <figref refid=plugh>.

- PUNC (Punctuation characters)

Punctuation characters are considered to be part of a word for purposes of spelling verification, unless they appear at the end of a word.

For example, the slash (/) is initially a word delimiter character, so when the term “SCRIPT/VS” is processed by spelling verification, “SCRIPT” and “VS” are checked separately as two different words. If you enter

```
.dc punc /
```

the slash is now treated as a punctuation character, and the term “SCRIPT/VS” is checked as a single word. If you enter the term “/rubbish/,” the slashes are removed because they occur at the ends of the word, and the word “rubbish” is verified.

- PS (Page number symbol)

The page number symbol is replaced with the current page number wherever it appears in a running heading or footing. For example, the initial page number symbol is an ampersand (&), and you can use it in a running footing definition:

```
.rf on
.sp 2
.ce --- & ---
.rf off
```

This footing appears at the bottom of each subsequent page:

--- 118 ---

If your running heading or footing contains a text ampersand, you should change the page number symbol to an unused character:

```
.dc ps ?
.rf on
.sp
.bx left right
.in +2
.ir +2
.sx /Rock & Roll//Page ?/
.bx off
.rf off
```

This footing appears at the bottom of subsequent pages:

Rock & Roll	Page 118
-------------	----------

- RB (Required blank)



Ordinary blanks (hexadecimal X'40') are treated as word spaces by SCRIPT/VS. When concatenation of input lines is in effect, SCRIPT/VS may break an input line into multiple output lines at any word space. When justification of output lines is in effect, SCRIPT/VS may expand or compress word spaces to justify text.

A required blank should be used whenever a space between words is needed but a word space is not wanted. Required blanks are treated like other nonblank characters for purposes of concatenation and justification, but they are replaced with ordinary blanks when the text is printed.

For example, to ensure that a personal title is not placed on a separate line from the surname that follows, use a required blank:

```
.dc rb -
and Dr.-Von-Neumann was ...
.dc rb off
```

On input, the hyphen characters (-) are replaced with required blanks so that the name is not broken between lines.

To ensure that manually aligned text is not disrupted by justification, use a required blank:

```
.dc rb -
.of 5
(iv)-Next, insert it into ...
.dc rb off
```

The hyphen (-) is replaced with a required blank, and the width of the required blank is not increased when the line is justified.

- **STOP (Full-stop characters)**

When concatenation of input lines is in effect, SCRIPT/VS inserts a word space between the last word of each input line and the first word of the next input line. If the input line ends in a full stop, SCRIPT/VS adds both a word space and an extra space between the last word of an input line and the first word of the next input line.

A line ends in a full stop when the last character on the line (not counting a double quotation mark or right parenthesis) is a full-stop character. When you specify full-stop characters with .DC, the characters given replace any previous full-stop characters. For example, if you enter

```
.dc stop : .
```

only the colon and period result in full stops.

- **WORD (Word delimiters)**

Word delimiters are used to separate individual words for spelling verification.

For example, the hyphen (-) is initially a punctuation character, so the term “in-laws” is processed by spelling verification as a single word. If you enter

```
.dc word -
```

the hyphen is treated as a word delimiter, and the term “in-laws” is checked as two separate words.

The backslash (\) is initially neither a word delimiter nor a punctuation character, so the term “APL&X'e0.360” is verified as a single word. The backslash is not keyable on some terminals, but it can be identified as a word delimiter by entering its hexadecimal code on any terminal:

```
.dc word e0
```

The term “APL\360” is now processed as two separate words.

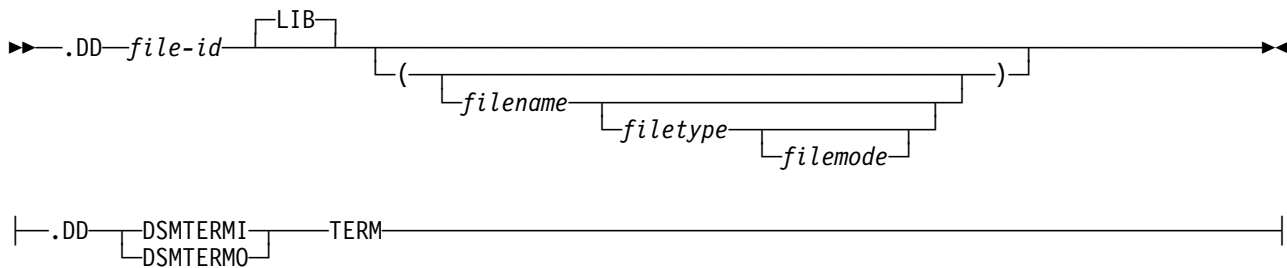
---

## .DD [Define Data File-Id] in the CMS Environment

### Function

The following are the parameters of the .DD [Define Data File-id] control word that are used in the CMS environment.

### Syntax



### Parameters

<i>file-id</i>	A 1- to 8-character SCRIPT/VS name for the file being defined. If the file-id corresponds to a file previously read and terminated by a .EF [End of File] control word, that original file is closed before the redefinition is made. If the file-id corresponds to a file that is currently active, an error occurs.
<b>LIB</b>	Is meaningful only in the in MVS/TSO environment. For compatibility purposes, this parameter can be specified in the CMS environment, but it is ignored.
<i>filename</i>	Specifies the CMS file name of the file to be assigned the specified file-id.
<i>filetype</i>	Specifies the CMS file type of the file to be assigned the specified file-id. If omitted, the file type given in the SEARCH command option is used. If the SEARCH command option is not used, a file type of "SCRIPT" is assigned.
<i>filemode</i>	Specifies the CMS file mode of the file to be assigned the specified file-id. If omitted, "*" is used.
<b>DSMTERMI</b>	Specifies the name of the terminal input file.
<b>DSMTERMO</b>	Specifies the name of the terminal output file.
<b>TERM</b>	Specifies that the terminal file (DSMTERMI or DSMTERMO) is to be redefined to the terminal. Valid only for file-ids DSMTERMI and DSMTERMO.

### Remarks

1. Parentheses are not required around *filename filetype filemode*. For compatibility with other environments, parentheses may be used.
2. SCRIPT/VS has a number of internally defined file-ids that can be redefined by the .DD control word. These file-ids are:
  - DSMTERMI — terminal input file
  - DSMTERMO — terminal output file
  - DSMUTCTF — STAIRS/VS CTF output file
  - DSMUTMSG — message file
  - DSMUTTOC — table of contents file
  - DSMUTWTF — .WF file

For example, if the file-id DSMTERMI is associated with a disk file, whenever a .RV, .RD, or .TE control word is processed, the data is read from the specified file.

Whenever any of these file-ids is the subject of a .DD control word, the file associated with the existing definition of that file-id is closed.

3. If the default command option, NODDUT, is in effect, SCRIPT/VS utility files may be redefined with the .DD control word, if the new file name is a SCRIPT/VS utility filename or if the prefix of the file type is DSM. Use of the DDUT command option allows unrestricted redefinition. See “DDUT: Enable SCRIPT/VS Utility File Redefinition” on page 25 and “NODDUT: Disable SCRIPT/VS Utility File Redefinition” on page 37 for descriptions of these command options.
4. The default definitions for the SCRIPT/VS internally defined file-ids are:

- DSMTERMI — terminal
- DSMTERMO — terminal
- DSMUTCTF — DSMUTCTF SCRIPT A
- DSMUTMSG — DSMUTMSG SCRIPT A
- DSMUTTOC — DSMUTTOC SCRIPT A
- DSMUTWTF — DSMUTWTF SCRIPT A

## Examples

- If two SCRIPT files have a file name of “names” and a filetype of “script,” one on your A-disk and the other on your C-disk, the control word .IM NAMES ordinarily imbeds the file on the A-disk, following CMS search order. The file on the C-disk would be imbedded if the following .DD were in effect:

```
.dd names names script c
```

- The following control words can be used to imbed a file that does not have a file type of “SCRIPT”:

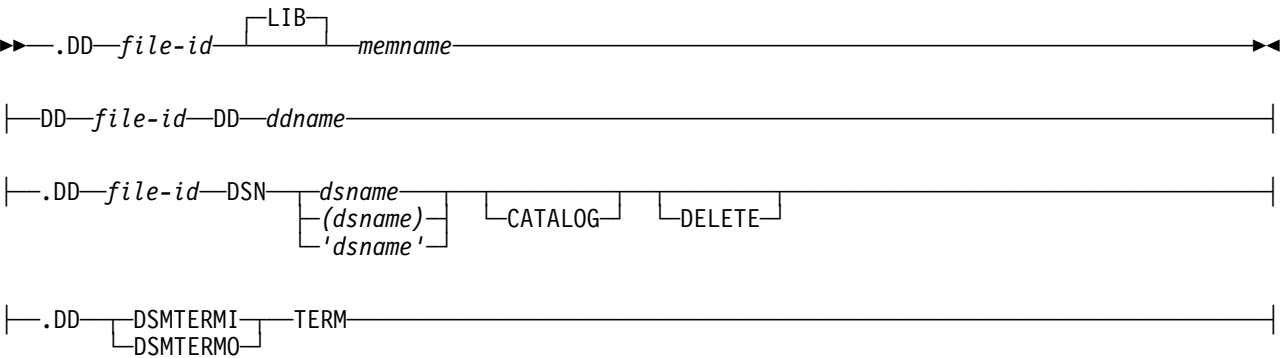
```
.dd setup setup exec a  
.im setup
```

# .DD [Define Data File-id] in the MVS/TSO Environment

## Function

The following are the parameters of the .DD [Define Data File-id] control word that are used in the MVS/TSO environment.

## Syntax



## Parameters

<i>file-id</i>	A 1- to 8-character SCRIPT/VS name for the data set being defined. If the file-id corresponds to a data set previously read and terminated by a .EF [End of File] control word, that original data set is closed before the redefinition is made. If the file-id corresponds to a data set that is currently active, an error occurs.
<b>LIB</b>	Indicates that the data set to be referred to exists in the library that was specified on the SEARCH command option. See “SEARCH: Specify a Library” on page 42 for an explanation of the SEARCH command option. LIB is the default. It is only necessary to use the .DD <i>file-id</i> LIB <i>memname</i> when the <i>file-id</i> and <i>memname</i> are different.
<i>memname</i>	Specifies the real member name of a member in the SEARCH library that is to be assigned to the new name given as the <i>file-id</i> . <i>memname</i> must reside in the first PDS in the TEXTLIB concatenation when the SEARCH command option is not specified and processing utility files.
<b>DD</b>	Specifies that the MVS data set to be referred to is specified via a DD name.
<i>ddname</i>	Specifies the DD name of the MVS data set to be given the specified file-id. <i>ddname</i> must be defined to MVS either by a user-supplied JCL DD card or by the TSO ALLOCATE command.
<b>DSN</b>	Specifies that the data set to be referred to is specified as a partially qualified or a fully qualified MVS data set name.
<i>dsname</i>	Specifies the partially qualified or the fully qualified MVS data set name to be assigned to the specified file-id. If <i>dsname</i> is a partially qualified data set name, it is fully qualified according to the TSO data set naming conventions described in “MVS/TSO Data Set Naming Conventions” on page 12. If <i>dsname</i> is a fully qualified data set name, it must be entered in uppercase and enclosed in single quotation marks. If the data set is a partitioned data set, <i>dsname</i> must include the member name.
<b>CATALOG</b>	Specifies that the data set to which the file-id refers is to be cataloged when it is closed. This parameter is valid only for SCRIPT/VS utility data sets when used with the DSN parameter to create a new data set.

<b>DELETE</b>	Specifies that the utility file should be deleted rather than catalogued when the job ends. DELETE is only valid when the <i>file-id</i> is DSMUTWTF. It is ignored for any other <i>file-id</i> .
<b>DSMTERMI</b>	Specifies the name of the terminal input data set.
<b>DSMTERMO</b>	Specifies the name of the terminal output data set.
<b>TERM</b>	Specifies that the terminal file (DSMTERMI or DSMTERMO) is to be redefined to the terminal. Valid only for file-ids DSMTERMI and DSMTERMO.

### Remarks

1. If MVS standards for data set names are violated in the .DD control word specification, the error can cause an MVS dynamic allocation or OPEN failure.
2. SCRIPT/VS has a number of internally defined file-ids that can be redefined by the .DD control word. These file-ids are:

- DSMTERMI — terminal input file
- DSMTERMO — terminal output file
- DSMUTCTF — STAIRS/VS CTF output file
- DSMUTMSG — message file
- DSMUTTOC — table of contents file
- DSMUTWTF — .WF file

For example, if the file-id DSMTERMI is associated with a data set, then whenever a .RV, .RD, or .TE control word is processed, the data is read from the specified data set. When DSMTERMI has been redefined to be a dataset, you must check for end-of-file after each .RV [Read Variable] control word by using the &E' attribute to check for the existence of the symbol.

Whenever any of these file-ids is the subject of a .DD control word, the data set associated with the existing definition of that file-id is closed and deallocated. Because the existing data set is deallocated when the file-id is redefined and utility data sets are normally allocated DISP=NEW, all existing data in the old data set is lost. If a utility data set is to be redefined several times and the data is to be retained across redefinitions, the utility data set must be preallocated with DISP=MOD.

3. If the default command option, NODDUT, is in effect, SCRIPT/VS utility data sets can be redefined with the .DD control word if the DSN parameter of the .DD control word is used to define the new data set name. Even if the DSN parameter is used, redefinition of utility data sets is allowed only if the qualifier preceding the low-level qualifier (TEXT) has a prefix of DSM. Use of the DDUT command option allows unrestricted redefinition. See “DDUT: Enable SCRIPT/VS Utility File Redefinition” on page 25 and “NODDUT: Disable SCRIPT/VS Utility File Redefinition” on page 37 for descriptions of these command options.
4. If the CATALOG parameter is used on the .DD control word to specify that the data set should be cataloged, the data set is not catalogued until the end of the job.
5. DELETE is normally used in conjunction with the CATALOG parameter. An application using the CATALOG parameter when renaming the DSMUTWTF dataset several times within a document can use the DELETE parameter to cause the dataset to be deleted at the end of the job.  
  
If DELETE is issued more than once for the same dataset, an MVS error message is issued indicating that the dataset is not allocated. The error message does not affect the SCRIPT/VS return code.
6. If a member is imbedded with a different file-id and written later with DSMUTWTF, the PDS may remain allocated until the TSO session ends. This can block the use of the PDS by other users. Use DSMUTWTF and .WF IMBED to imbed any member that is written by .WF [Write to File] any time in the document.
7. The default definitions for the SCRIPT/VS internally defined file-ids are:

- DSMTERMI — terminal

DSMTERMO — terminal  
 DSMUTCTF — 'userid.DSMUTCTF.TEXT'  
 DSMUTMSG — 'userid.DSMUTMSG.TEXT'  
 DSMUTTOC — 'userid.DSMUTTOC.TEXT'  
 DSMUTWTF — 'userid.DSMUTWTF.TEXT'

- | 8. You should not use the .DD control word to individually define members of PDSs already known to the  
 | SEARCH option or the TEXTLIB *ddname* unless it is necessary to rename the members. Defining  
 | each member under its own name with .DD causes unnecessary overhead.
- | 9. The LIB parameter of the .DD control word was designed to work with the SEARCH command option  
 | which allows only one partitioned dataset to be searched for imbedded members. The LIB parameter  
 | of the .DD control word may cause unpredictable errors if it is used with a concatenation of partitioned  
 | data sets allocated to the TEXTLIB *ddname*. Members of a concatenation should be imbedded  
 | directly with the member names. If you are forced to redefine a member of a TEXTLIB concatenation,  
 | use the DD or DSN parameter of the .DD control word to identify the specific partitioned dataset  
 | member to be redefined.

## Examples

- To give a file-id of "alpha" to member "mem3" of a partitioned data set named *userid.DOC.TEXT* enter the .DD control word.

```
.dd alpha dsn doc(mem3)
```

See "MVS/TSO Data Set Naming Conventions" on page 12 to understand how the partially qualified data set name *doc(mem3)* was expanded into the fully qualified data set name of *userid.DOC.TEXT(MEM3)*.

- To fully qualify a data set name with a member, enter

```
.dd beta dsn 'P3741.DATA.TEXT(FILIN)'
```

When a fully qualified data set name is given, it must be given in exactly the same way it is known to the system.

- | If 'P3741.DATA.TEXT' was named in the SEARCH option, or if 'P3741.DATA.TEXT' was allocated  
 | to *ddname* TEXTLIB, then the same effect could be achieved with the following statement:

```
| .im filein
```

- | If *filein* is a member in the SEARCH library, and does not need to be renamed, then it can be  
 | imbedded directly with the .IM [Imbed] control word and no .DD is needed.

- To format data from a sequential data set named *APPLDEV.TEST.TEXT* do the following:

- Create an input file named *userid.IMTEST.TEXT* that contains the following:

```
.dd testfile dd testin
.* this data file imbeds a preallocated data set with
.* the ddname of 'testin'.
.im testfile
```

- Allocate the data set with TSO ALLOCATE command:

```
alloc dd(testin) dsn('APPLDEV.TEST.TEXT')
```

- Issue the SCRIPT command:

```
script imtest
```

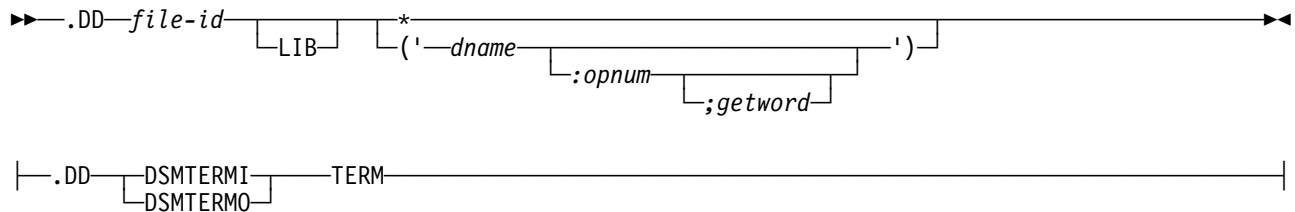
The imbedded file *testfile* is included from the sequential data set, *APPLDEV.TEST.TEXT*.

## .DD [Define Data File-Id] in the ATMS Environment

### Function

The following are the parameters of the .DD [Define Data File-id] control word that are used in the ATMS environment.

### Syntax



### Parameters

<i>file-id</i>	A 1- to 8-character SCRIPT/VS name for the document being defined. If the file-id corresponds to a document previously read and terminated by a .EF [End of File] control word, that original document is closed before the redefinition is made. If the file-id corresponds to a document that is currently active, an error occurs.
<b>LIB</b>	Indicates the document to be referred to exists in permanent storage.
<b>*</b>	Indicates the document to be referred to is the current document in the operator's working storage or the document that was specified on the TRANSMIT command.
<i>dname</i>	Specifies the name of the document in permanent storage. If the combination of document name, operator number, and access word contains lowercase letters or special characters, it must be enclosed in single quotes.
<i>opnum</i>	Specifies the operator number where the document exists. If the combination of document name, operator number, and access word contains lowercase letters or special characters, it must be enclosed in single quotes.
<i>getword</i>	Specifies the access word for the document. If an access word is specified, the control word modifier must be used on the .DD control word to prevent the semicolon before the access word from being interpreted as a control word separator. If the combination of document name, operator number, and access word contains lowercase letters or special characters, it must be enclosed in single quotes.
<b>DSMTERMI</b>	Specifies the name of the terminal input file.
<b>DSMTERMO</b>	Specifies the name of the terminal output file.
<b>TERM</b>	Specifies that the terminal file (DSMTERMI or DSMTERMO) is to be redefined to the terminal. Valid only for file-ids DSMTERMI and DSMTERMO.

### Remarks

1. SCRIPT/VS has a number of file-ids that are used by the system. These can be the subject of the .DD control word. These file-ids are:

- DSMTERMI — terminal input file
- DSMTERMO — terminal output file
- DSMUTCTF — STAIRVS/VS CTF output file
- DSMUTMSG — message file

DSMUTTOC — table of contents file  
DSMUTWTF — .WF file

2. The utility files are managed as CICS temporary storage files, and they are not accessible by the user.
3. If the document name for the primary input file is enclosed in parenthesis, an error occurs.

## Examples

- To specify an internal name for a document stored with a name of “shotnaf” in the permanent storage of operator 3, the following .DD control word would be used:  
`.dd mydoc ('shotnaf:3')`
- To specify an internal name for a document stored with a name of *hoffy* in the permanent storage of operator 0 with an access word of *zorch*, the following .DD control word would be used:  
`.'dd moredoc ('hoffy:3;zorch')`

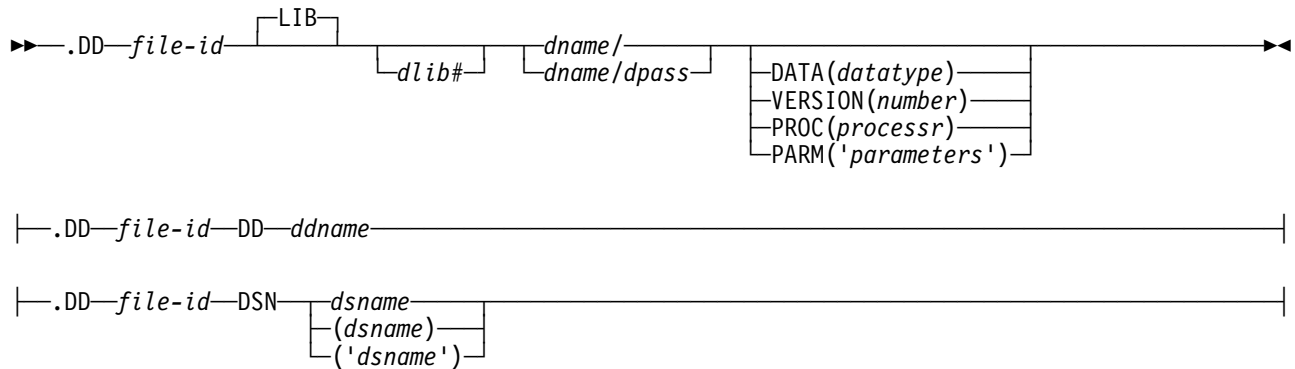


## .DD [Define Data File-Id] in the DLF Environment

### Function

The following are the parameters of the .Define Data File-id] control word that are used in the DLF environment.

### Syntax



### Parameters

<i>file-id</i>	A 1- to 8-character SCRIPT/VS name for the document being defined. If the file-id corresponds to a document previously read and terminated by a .EF [End of File] control word, that original document is closed before the redefinition is made. If the file-id corresponds to a document that is currently active, an error occurs.
<b>LIB</b>	Indicates the document to be referred to exists in the DLF library. LIB is the default. The LIB parameter can be followed optionally be a library number and password.
<i>dlib#</i>	Specifies the library number of the library to be searched. If not specified, the current user's library is searched. Refer to the <i>Document Library Facility Guide</i> for more information on library numbers and their usage with DLF.
<i>dname</i>	Specifies the document name that is to be given the specified file-id. If the combination of document name and document password contains lowercase letters or special characters, it must be enclosed in single quotes. Refer to the <i>Document Library Facility Guide</i> for more information on document names and their usage with DLF.
<i>dpass</i>	Specifies the password of the document. This parameter should be used only if a password is associated with the document dname. If the combination of document name and document password contains lowercase letters or special characters, it must be enclosed in single quotes. Refer to the <i>Document Library Facility Guide</i> for more information on document passwords and their usage with DLF.
<b>DATA</b>	Specifies the datatype of the document. Refer to the <i>Document Library Facility Guide</i> for details.
<b>VERSION</b>	Specifies the version number of the document. Refer to the <i>Document Library Facility Guide</i> for details.
<b>PROC</b>	Specifies a processor for the document. Refer to the <i>Document Library Facility Guide</i> for details.
<b>PARM</b>	Specifies parameters for the processor named by the PROC parameter. Refer to the <i>Document Library Facility Guide</i> for details.

## .DD [Define Data File-Id] in the DLF Environment

<b>DD</b>	Specifies that the data set to be referred to is specified via a DD name.
<i>ddname</i>	Specifies the DD name of the data set to be given the specified file-id. The <i>ddname</i> should be used on a user-supplied JCL DD card.
<b>DSN</b>	Specifies that the data set to be referred to is specified as a partially qualified or fully qualified data set name. This parameter is allowed only in the MVS environment.
<i>dsname</i>	Specifies the partially qualified or fully qualified MVS data set name to be assigned to the specified file-id. If <i>dsname</i> is a partially qualified data set name, it is fully qualified according to the TSO data set naming conventions described in "MVS/TSO Data Set Naming Conventions" on page 12. If <i>dsname</i> is a fully qualified data set name, it must be entered in uppercase and enclosed in single quotation marks. If the data set is a partitioned data set, <i>dsname</i> must include the member name.

### Remarks

1. If MVS or VSE data set naming standards are violated in the .DD control word specification, the error might cause an MVS or VSE dynamic allocation or OPEN failure.
2. SCRIPT/VS has a number of file-ids that can be redefined by the .DD control word. These file-ids are:

DSMTERMI — terminal input file  
DSMTERMO — terminal output file  
DSMUTCTF — STAIRS/VS CTF output file  
DSMUTMSG — message file  
DSMUTTOC — table of contents file  
DSMUTWTF — .WF file

For example, if the file-id DSMTERMI is associated with a document, then whenever a .RV, .RD, or .TE control word is processed, the data is read from the specified document.

The internally defined file-ids may be redefined only with the DD form of .DD [Define Data File-id].

3. The default MVS/DLF definitions for the SCRIPT/VS internally defined file-ids are:

MVS:

DSMTERMI — ddname = DSMTERMI  
DSMTERMO — ddname = DSMLIST  
DSMUTMSG — ddname = DSMUTMSG  
DSMUTWTF — ddname = DSMUTWTF  
DSMUTCTF — ddname = DSMUTCTF  
DSMUTTOC — ddname = DSMUTTOC

VSE:

DSMTERMI — dlblname = DSMITRM  
DSMTERMO — dlblname = DSMOTRM  
DSMUTMSG — dlblname = DSMUMSG  
DSMUTWTF — dlblname = DSMUWTF  
DSMUTCTF — dlblname = DSMUCTF  
DSMUTTOC — dlblname = DSMUTOC

## Examples

- To give a file-id of “file1” to a Document Library Facility file named “title,” with a password of “P2301” with existing library *13425*, the .DD control word would be

```
.dd file1 lib 13425 title/p2301
```

In this example, the LIB parameter could have been omitted because it is the default.

---

## .DF [Define Font]

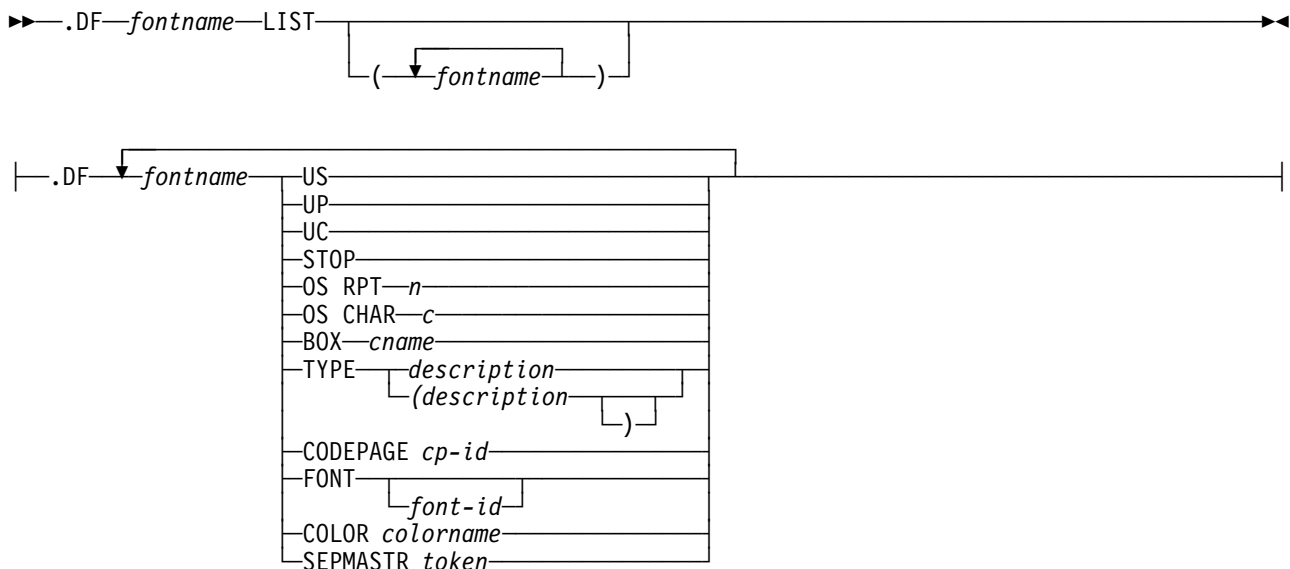
### Function

Use the .DF [Define Font] control word to specify a list of fonts or to define a *named* font for use with the .BF [Begin Font], .DA [Define Area], .DH [Define Head Level], .DR [Define Rule], .DV [Define Variable], and .RC [Revision Code] control words. This allows such internal formatting functions as underscoring and capitalization to be managed with the .BF [Begin Font] and .PF [Previous Font] control words. This also allows you to alter the characteristics of the font-ids given with the CHARS option of the SCRIPT command and provides a means of identifying fonts descriptively for page printers.

The .DF control word also provides the ability to selectively overstrike text (on impact printers) and temporarily stop typing (on typewriter terminals) to change elements.

Refer to the *Document Composition Facility: SCRIPT/VS User's Guide* for more information on defining fonts.

### Syntax



### Parameters

- fontname** Specifies the identifier of the font or list of fonts being defined by the .DF control word. After *fontname* has been defined, it may be used with the .BF, .DA, .DH, .DR, .DV, and .RC control words. It can contain a maximum of 8 nonblank characters.
- LIST** Associates a list of fonts with a *fontname* for use with the .BF, .DA, .DH, .DR, .DV, and .RC control words. If you specify more than one *fontname*, the list of *fontnames* must be enclosed in parentheses. If you specify only one *fontname* with LIST, the parentheses are not required.
- US** Specifies that this font causes text to be underscored.
- UP** Specifies that this font causes text to be capitalized.
- UC** Specifies that this font causes both capitalization and underscoring.

<b>STOP</b>	Specifies that when this font begins, the interactive typewriter terminal stops and waits for the user to signal ATTENTION. STOP may be used for changing fonts at a typewriter terminal where each “font” is actually a different typing element. When the typewriter stops, you can change the element. STOP has meaning only in the CMS environment; it is ignored in other environments or when the output destination is not a typewriter terminal.
<b>OS</b>	Specifies that this font is to be formed by overstriking. OS is ignored for devices other than 2741 and 1403. Overstriking with the underscore character ( <code>_</code> ) is honored for the 3270 and 3800 Model 1 devices.
<b>RPT</b>	Specifies that each nonblank character is to be overstruck with itself <i>n</i> times. A maximum of five overstrikes is allowed.
<b>CHAR</b>	Specifies that each nonblank character is to be overstruck with the single character <i>c</i> .
<b>BOX</b>	Specifies a box character set to be used on line devices for constructing box corners and intersections with the .BX [Box] control word. The valid box character set names are listed under “.BX [Box]” on page 81.
<b>FONT</b> <i>font-id</i>	<p>Specifies the external (host system) font to be used. If formatting for the 3800 Model 1, <i>font-id</i> must be one of those listed with the CHARS option of the SCRIPT command.</p> <p>If FONT is not specified, the <i>font-id</i> of the current font is used when <i>fontname</i> is used with the .BF control word.</p> <p>The FONT parameter is ignored for PostScript devices.</p>
<b>TYPE</b>	<p>Descriptively defines fonts for page printers and PostScript devices and is ignored for line devices. If more than one description parameter is included, it must be enclosed in parentheses. The specified typeface name is converted to uppercase.</p> <p><b>Note:</b> Not all typefaces, point sizes, weights, widths, and attributes may be available at your installation. Refer to the font library listing created by the Font Library Index Program (FLIP) to determine which typefaces, point sizes, weights, widths, and attributes are available at your installation.</p> <p>TYPE describes the <i>named</i> font being defined in terms of the following characteristics:</p> <ul style="list-style-type: none"> <li>• <i>typeface</i> gives the descriptive name of a font family. For example, for the Century Schoolbook font the descriptive name is ‘Century Schoolbook’. If <i>typeface</i> is not specified, it remains unchanged when <i>fontname</i> is used.</li> </ul> <p>If <i>typeface</i> contains blanks or parentheses, it must be enclosed in single quotation marks ( <code>' '</code> ).</p> <ul style="list-style-type: none"> <li>• <i>point size</i> is the vertical size in points of the font given as an absolute number, <i>v</i>, or a relative number, <i>+v</i> or <i>-v</i>. If the point size is not specified or if a point size of zero is given, the point size remains unchanged when the font is started. If the point size is given as an absolute number or is inherited from the previous font, the point size must be available in the font library. If the point size is given as a relative number, <i>+v</i> (or <i>-v</i>), the number <i>v</i> is added to (or subtracted from) the current point size in effect when the font is started. If the resulting point size is not available, the point size closest to the resulting point size is used. If two point sizes are equally close to the resulting point size, the larger point size is used. For example, if “-8” is specified when the current point size is 30 and the available point sizes are 20, 24, and 30, point size 24 is used. The point size specification of +0 (or -0) indicates that the current point size should be used if available. If the current point size is not available, the next larger (or smaller) available point size is used.</li> </ul>

A four-digit relative number is allowed. If a value greater than four digits is given on a relative point size specification (+*v* or -*v*), it is treated as a typeface name specification.

The point size for a PostScript font can be any integer value between 1 and 3276.

- *weight* is one of the following weight-class identifiers:

ULTRALIGHT  
EXTRALIGHT  
LIGHT  
SEMILIGHT  
MEDIUM  
SEMIBOLD  
BOLD  
EXTRABOLD  
ULTRABOLD  
OLDWEIGHT

If *weight* is not specified, MEDIUM is used.

**OLDWEIGHT** is used to inherit the weight of the font in effect when this font is started.

- *width* is one of the following width-class identifiers:

ULTRACONDENSED  
EXTRACONDENSED  
CONDENSED  
SEMICONDENSED  
NORMAL  
SEMIEXPANDED  
EXPANDED  
EXTRAEXPANDED  
ULTRAEXPANDED  
OLDWIDTH

If *width* is not specified, NORMAL is used.

**OLDWIDTH** is used to inherit the width of the font in effect when this font is started.

- *rotation* is the angle of rotation of the character images, given in degrees relative to the baseline, and must be one of the following values:

0R  
90R or -270R  
180R or -180R  
270R or -90R

If *rotation* is not specified, it remains unchanged. The rotation parameter is valid only for page printers; it is not valid for PostScript devices.

- *attr* is any of the following attribute identifiers:

ITALIC  
UNDERSCORE  
OUTLINE  
OVERSTRUCK  
OLDATTR

**OLDATTR** is used to inherit the attributes of the font in effect when this font is started, in addition to any attributes specified for the new font.

<b>CODEPAGE</b>	<p>Specifies the external code page to be used with this font. <i>cp-ID</i> must be a code page residing in the font library identified in the FONTLIB option of the SCRIPT command.</p> <p>If CODEPAGE is not specified, it remains unchanged when the font is used.</p> <p>CODEPAGE is ignored for line devices.</p>
<b>COLOR</b>	<p>Specifies the color to be used with this font. The COLOR parameter applies only to output devices with color capability. <i>colorname</i> can be any color defined previously with the .CR [Color] control word.</p> <p>A <i>colorname</i> of <b>OLDCOLOR</b> is used to inherit the color of the font in effect when this font is started.</p> <p>If COLOR is not specified, the default color for the output device is used. If the output device does not support the specified color, the COLOR parameter is ignored.</p>
<b>SEPMASR</b>	<p>Specifies that the font is being associated with output separation masters.</p> <p>The <i>token</i> is a 1- to 8-character name that identifies which separation masters this font should be associated with. The token name corresponds to the items selected for separation masters with the .SM control word.</p> <p>The special token value of ALL is used to indicate that this font should be contained in all separation masters, including the default.</p> <p>The special token value of OLDOLOR is used to inherit the separation master value of the font in effect when this font is started.</p> <p>For more information on the SEPMASR parameter, see “.SM [Separation Master]” on page 353.</p>

**Note:** Font definitions are included in the active environment.

## Remarks

- When using the LIST parameter, the following apply:
  - LIST must be the first parameter after the fontname.
  - No other attributes can be specified on the .DF control word.
  - The number of fonts in a list is not restricted.
  - The names in a list need not be defined before the list is defined.
  - A fontname in a list cannot be the name of another list. If you specify the name of another list, it is considered an undefined fontname and is ignored.
  - The single fontname on a .BF, .DA, .DH, .DR, .DV, or .RC control word can be the name of a list. Also, any fontname in a list on a .BF control word *can* be the name of a list.
  - An attempt to start a list-form font does not fail unless the entire list is exhausted without successfully starting a font.
- As many parameters as necessary may be specified with .DF. If both the FONT and TYPE parameters are given, the last one specified is used. The CODEPAGE parameter cannot be used in conjunction with the FONT parameter.
- A null font definition might result either because no parameters were specified or because none of the parameters specified applies to the current output device. When a null font is used with the .BF [Begin Font] control word, the current font remains unchanged.
- If the *named* font being defined already exists, the previous definition is completely replaced with the new one.
- When a *named* font is defined descriptively with the TYPE parameter, and typeface or point size are not specified, they remain unchanged when that font is used.

## .DF [Define Font]

6. When you use the STOP parameter under CMS, issue the command

cp term attn off

to suppress CPs normal attention acknowledgment.

7. The SEPMASTR keyword is ignored for line devices.
8. The SEPMASTR keyword is also ignored if the SEPMASTR command option id not specified.
9. A code page defines the relationship between the hexadecimal code point values in your SCRIPT/VS source file and the actual character produced when the file is printed. You should use the code page that most closely matches the keyboard that you are using. The code pages for use with special character fonts, such as Pi and Light Italic and for PostScript Symbol and Dingbat fonts are a different kind of code page. These fonts have a special set of characters and thus need special code pages to use in relating those characters to specific hexadecimal code points.
10. If a code page is specified, or inherited, that does not match the font being used, an error message is issued. The code page does not match the font when the font specified does not contain all of the characters in the code page. This situation can result in errors at print time.
11. When specifying the CODEPAGE parameter, only use code pages compatible with the character sets appropriate for the logical device specified with the DEVICE command option.
12. Refer to *About Type: IBM's Code Pages for Digitized Type*, S544-3802 for more information on the relationship between code pages and font character sets.
13. AFP outline fonts are relative metric fonts. They can be used when formatting for the DEV(AFPx) logical device types, though AFP outline fonts are not the default for those logical device types. You can mix outline and bounded box raster technology fonts in the formatting run. Be sure your DCFINDEX appropriately reflects your font libraries. The DCF Font Library Index Program does not flag duplicate fonts across font technologies.

When using outline fonts, any coded fonts you use (including the default font) must contain a vertical font size greater than zero or you will get an error message and cannot use the font. Coded fonts with horizontal scale factors equal to values other than zero or the vertical font size are not supported. Double-byte outline fonts are not supported.

When using AFP outline fonts, SCRIPT/VS always writes the second byte of the character set name on the MFC/2 structured field with a "Z" to indicate outline technology.
14. The SEPMASTR parameter is ignored for line devices.
15. The SEPMASTR parameter is also ignored if the SEPMASTR command option is not specified.

**Note:** Examples using the SEPMASTR parameter can be found in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.



## Examples

- To define font definitions for highlighted phrases with the LIST parameter, specify

```
.df font1 list (font1a font1b =)
.df font1a type (italic)
.df font1b us
.df font2b up
.df font2 list (font2a font2b =)
.df font3 list (font3a font3b =)
.fo off
.bf font1
This text is italic.
.pf
.bf font2
This text is in uppercase.
.bf font3
This text is in uppercase.
.fo on
```

The “=” ensures that the current font is used if no other font can be started and that no error message is issued. As in this example font2a, font3a, and font3b do not exist.

Here are the results:

```
This text is italic.
THIS TEXT IS IN UPPERCASE.
THIS TEXT IS IN UPPERCASE.
```

- To define a new font for 1403 output that causes capitalization and overstriking, specify
- To define a new font for the 3800 Printing Subsystem Model 1 that causes underscoring in a bold font, specify

```
.df bold up os rpt 3
```

```
.df hilite us font GB12
```

- A font for formatting footnotes on a page printer can be defined as

```
.df ftnote type(futura 6 light)
```

- A font for emphasizing key phrases can be defined as

```
.df emph type(bold italic)
```

Because neither a typeface nor a point size are included in the definition, they are not changed when the EMPH font is used. For example, if the current font is 8 point Monotype Bodoni, and you enter

```
.bf emph
```

subsequent text is set in 8 point Monotype Bodoni bold italic. But if, using the definition of the FTNOTE font used above, you enter

```
.bf ftnote
...
.bf emph
```

subsequent text is set in 6 point Futura bold italic.

- A font for printing text in blue on a device that supports color may be defined as:

```
.df bluetext color blue
```

- Refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for information on using the .DF control word with the SEPMASTR parameter.

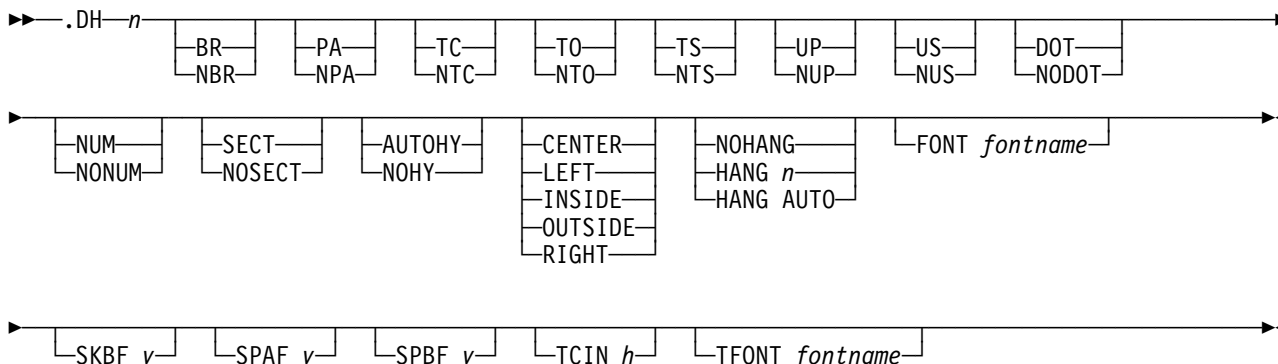
---

## .DH [Define Head Level]

### Function

Use the .DH [Define Head Level] control word to override the default characteristics of the head levels that are generated with the .H0 [Head Level 0] control words.

### Syntax



### Parameters

*n* The number of the head level to be defined. *n* may be a number from 0 to 6.

*options* Parameters that indicate how to change the definition of head level *n*. If no options are given, the default characteristics of the head level are restored.

The options recognized are:

**AUTOHY** Headings are to be automatically eligible for hyphenation, depending on whether hyphenation is on or off. Note that the actual table of contents entries are always eligible for hyphenation. This option applies only to headings in the body of the document and is the default for all head levels.

**NOHY** Headings in the body are not to be hyphenated if the heading exceeds the column line length. This does not turn off hyphenation of entries in the table of contents.

**BR** Causes a break after the head.

**NBR** No break.

**CENTER** Indicates that the text of the heading is to be centered (as in .FO CENTER).

**DOT** In the table of contents, the text of the heading is separated from the page number by “leader-dots,” and the page number is right aligned. (This is the default for all head levels that create a table of contents entry.)

**NODOT** Table of contents entries for this head level do not have “leader-dots” separating the text of the heading from the page number. In this case, the page number is not right-aligned, but it is separated from the text by two figure spaces.

<b>FONT</b> <i>fontname</i>	The name of the font to be used for the heading, or the word OFF. If OFF is specified, the previous specification of FONT is set off and the heading is set in the current font. If <i>fontname</i> is omitted and FONT is the last option specified, no syntax error results and the heading is also set in the current font. This is useful when using the &\$CHAR system symbol array for <i>fontname</i> , because the symbol array elements have null values when no corresponding fonts exist. (The default for all head levels is the current font.)
<b>HANG</b> <i>n</i>	Headings, if they occupy more than one output line, are to be formatted in the hang indent style. The amount of the indent is <i>n</i> . If an offset is in effect, HANG is ignored.
<b>HANG AUTO</b>	The indentation is automatically calculated to be equal to the heading numbers and space. This cannot be used for level zero headings because they are never numbered.
<b>NOHANG</b>	The second and subsequent output line of a heading is flush left with the current margin. This is the default.
<b>LEFT</b>	Left-justify the text of the heading (as in .FO LEFT).
<b>INSIDE</b>	Align the text of the heading against the inside margin of the column—toward the presumed binding edge of the duplexed page. This is equivalent to .FO LEFT for odd pages and .FO RIGHT for even pages.
<b>NUM</b>	Prefix this heading with a decimal number that reflects the level of the heading. The first level-one heading is numbered 1.0, the first level-two heading is numbered 1.1, the first level-three heading is numbered 1.1.1, and so on. Level zero headings are never numbered.  You can set the counters used for head numbering with the HCTR option of the .GS [GML Services] control word, but this is not necessary if you want the numbering to start with 1.0 and increase sequentially.
<b>NONUM</b>	Do not prefix this heading with a decimal number. (This is the default for all head levels.)
<b>OUTSIDE</b>	Align the text of the heading against the outside margin of the column—away from the presumed binding edge of the duplexed page. This is equivalent to .FO RIGHT for odd pages and .FO LEFT for even pages.
<b>PA</b>	Do a page eject before the head level if necessary (if not already at the top of a page).
<b>NPA</b>	No page eject.
<b>RIGHT</b>	Right-align the text of the heading (as in .FO RIGHT).
<b>SECT</b>	Cause a section break before and after the head level. SECT means that the head level itself is formatted in a single column the full width of the page, regardless of the current column definition.
<b>NOSECT</b>	No section break around heading.
<b>SKBF</b> <i>v</i>	<i>v</i> is the amount of conditional space to be skipped before the head level. <i>v</i> can be any valid vertical space unit.

## .DH [Define Head Level]

<b>SPAF</b> <i>v</i>	<i>v</i> is the amount of unconditional space to follow the head level. <i>v</i> can be any valid vertical space unit.
<b>SPBF</b> <i>v</i>	<i>v</i> is the amount of unconditional space to precede the head level. <i>v</i> can be any valid vertical space unit.
<b>TC</b>	Table of contents entry wanted.
<b>NTC</b>	No table of contents entry.
<b>TCIN</b> <i>h</i>	<i>h</i> is the amount the table of contents entry associated with the head level is to be indented.
<b>TFONT</b> <i>fontname</i>	The name of the font to be used in the table of contents for this head level entry, or the word OFF. It works similarly to FONT.
<b>TO</b>	Table of contents entry only; nothing is formatted in the body of the page.
<b>NTO</b>	No TO; the heading is formatted in the text.
<b>TS</b>	Space one line before table of contents entry.
<b>NTS</b>	No TS; the table of contents entry is not preceded by a space.
<b>UP</b>	Capitalize the head level.
<b>NUP</b>	No UP; the head level is not to be capitalized.
<b>US</b>	Underscore the head level.
<b>NUS</b>	No US; the head level is not to be underscored.

**Default:** Restores the initial setting.

**Note:** Heading definitions may be overridden by the profile you are using.

Characteristic	.H0	.H1	.H2	.H3	.H4	.H5	.H6
New page for heading		Yes					
Section breaks around heading		Yes					
Heading alignment		Outside	Left	Left	Left		
Space before heading	0	0	0	0	0	0	0
Skip before heading	0	0	3	3	3	1	1
Space after heading	0	5	2	2	2	0	0
Heading underscored		Yes	Yes		Yes	Yes	Yes
Heading capitalized		Yes	Yes	Yes		Yes	
Break before heading		Yes	Yes	Yes	Yes		
Table of Contents entry	Yes	Yes	Yes	Yes			
Table of Contents only	Yes						
Skip before T.O.C. entry		Yes					
T.O.C. indentation	0	0	0	2	4	6	8
Automatic hyphenation	0	Yes	Yes	Yes	Yes	Yes	Yes
Hanging indent	0	no	no	no	no	no	no

*Table 8. Summary of Initial Head-Level Characteristics. This table lists the initial characteristics of the .KH [Head Level n] control words. The .DH [Define Head Level] control word allows you to redefine any of these characteristics to suit your needs.*

## Remarks

1. The font name specified with the .DH control word can be a list of fonts previously defined with the LIST parameter of the .DF control word. For more information on using the LIST parameter, see “.DF [Define Font]” on page 130.
2. If you want to change more head-level variables than can easily fit on one input line, use more than one .DH control word. Each time .DH is processed, only those variables specified are changed.
3. If a head level control word is processed that causes an entry in the table of contents, the table of contents entry is saved with the specifications that are in effect at the time that head level is processed. If you change the definition of that head level later, the new definition affects only later occurrences of that head level control word.
4. For a list of the initial characteristics associated with the heading levels 0 through 6, see Table 8 on page 138.
5. The default for a .H0 is only a table of contents entry. If you want this heading to appear in your document, you must specify all the appropriate parameters.
6. The alignment parameters (INSIDE, OUTSIDE, LEFT, RIGHT, and CENTER) are used only if you have also specified the BR or SECT parameters.
7. If the heading level is defined to start a page or to cause section breaks around it, there is no keep around the heading level and the first two lines of following text.
8. The OJ and NOJ options are provided for compatibility with SCRIPT/VS Release 2. They are equivalent to OUTSIDE and LEFT, respectively.
9. If FONT *fontname* is specified and underscoring is part of the definition of the font (as specified with the US parameter of the .DF [Define Font] control word), the heading is underscored whether or not the US or NUS parameter of .DH was specified for that heading. If you use the UP or UC parameters of .DF to make capitalization or capitalization and underscoring a part of the definition of your font, these functions would also take precedence, whether or not you had specified the US or NUS, or the UP or NUP parameters of .DH.
10. On headings that cause section breaks, the SKBF parameter produces a page skip and therefore is never combined with a SPAF parameter from a previous heading.

## Examples

- To center all level 1 headings and set them in the second font specified with the CHARS option of the SCRIPT command, enter

```
.dh 1 center font &$CHAR(2)
```

If only one font is specified with the CHARS option, the value of the symbol &\$CHAR(2) is null, and the FONT parameter is ignored.

- To make level 2 headings result in exactly the same formatting as level 1 headings, enter

```
.dh 2 pa sect outside spaf 5 ts
```

- If you do not want level 5 headings to be underscored or capitalized, but you do want them to create table of contents entries, enter

```
.dh 5 nus nup tc
```

To restore the default characteristics, enter

```
.dh 5
```

---

## .DL [Dictionary List]

### Function

Use the .DL [Dictionary List] control word to specify the dictionaries to be used for spelling verification, hyphenation, and the language used for the index sort sequence.

### Syntax

►► .DL name ◄◄

| .DL—INDEX—name |

### Parameters

**name**      Name of a language. The name can be up to four characters long.

**INDEX**    Specifies the language to use for sorting the index entries in the document. Only one language is allowed for a document.

### Remarks

1. The addenda dictionary used is the first dictionary named. The base dictionary for the current language follows the addenda dictionary in the search sequence, followed by any user-created dictionaries.
2. The names of the IBM-supplied languages that are available for spelling verification, algorithmic hyphenation, and index sort sequences are:

<b>DAN</b>	Danish
<b>DUTH</b>	Dutch
<b>EAM</b>	English (American)
<b>ECAN</b>	English (Canadian)
<b>EUK</b>	English (United Kingdom)
<b>FCAN</b>	French (Canadian)
<b>FIN</b>	Finnish
<b>FNAT</b>	French
<b>GERM</b>	German
<b>ICE</b>	Icelandic
<b>ITAL</b>	Italian
<b>NOR</b>	Norwegian
<b>POR</b>	Portuguese
<b>SPAN</b>	Spanish
<b>SWE</b>	Swedish

**Note:** User dictionaries must have different names.

IBM-supplied dictionaries exist for all of the supported languages but the following dictionaries are empty: Danish, Finnish, Icelandic, Norwegian, Portuguese, and Swedish. However, you can create user dictionaries for spelling verification for these languages using the Dictionary Maintenance Program supplied with DCF. Refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for more information.

3. If the .DL [Dictionary List] control word is not used, the default language specified by your installation is used for spelling verification, hyphenation, and index sort sequence.

4. If the first dictionary named is one of the IBM-supplied base dictionaries, any user-created dictionaries currently in use is no longer used, and the stem processing routines for these languages is loaded.

The following user dictionary names should not be specified when the Dutch language has been selected:

UALG	UFLT	USSH	UVEC
UCTL	UFOR	USTM	UVER
UDRM	UHRM	UUU	

5. When you specify a user dictionary and an IBM-supplied dictionary with the same .DL [Dictionary List] control word, specify the IBM-supplied dictionary first.
6. For a complete description of the index sort sequence, refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.
7. The dictionaries specified for spelling verification and hyphenation must be all of the same language.
8. Only one sort sequence language can be specified for a document. If a sort sequence language is not specified using .DL INDEX before the first .PI [Put Index], the default language is used. An error message is issued if .DL INDEX is specified after a .PI control word or a previous .DL INDEX.
9. The &\$IXLG system symbol indicates the index sort sequence language.
10. The &\$SVLG system symbol indicates the current spelling verification and hyphenation dictionary language.
11. SCRIPT/VS provides two algorithmic hyphenators for English (English American, English Canadian, and English United Kingdom), and two algorithmic hyphenators for German. The algorithmic hyphenator is determined at installation time. See the heading "National Use Considerations" in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for more information.
12. The &\$ENG system symbol indicates which English algorithmic hyphenator is being used. &\$ENG returns a value of "1" if the alternate English algorithmic hyphenator is used or a value of "0" if the default English algorithmic hyphenator is used.
13. The &\$GERM system symbol indicates which German algorithmic hyphenator is being used. &\$GERM returns a value of "1" if the alternate German algorithmic hyphenator is used or a value of "0" if the default German algorithmic hyphenator is used.
14. The default language is used to indicate the input language of the document on the Begin Document Structured field for AFPDS physical devices.

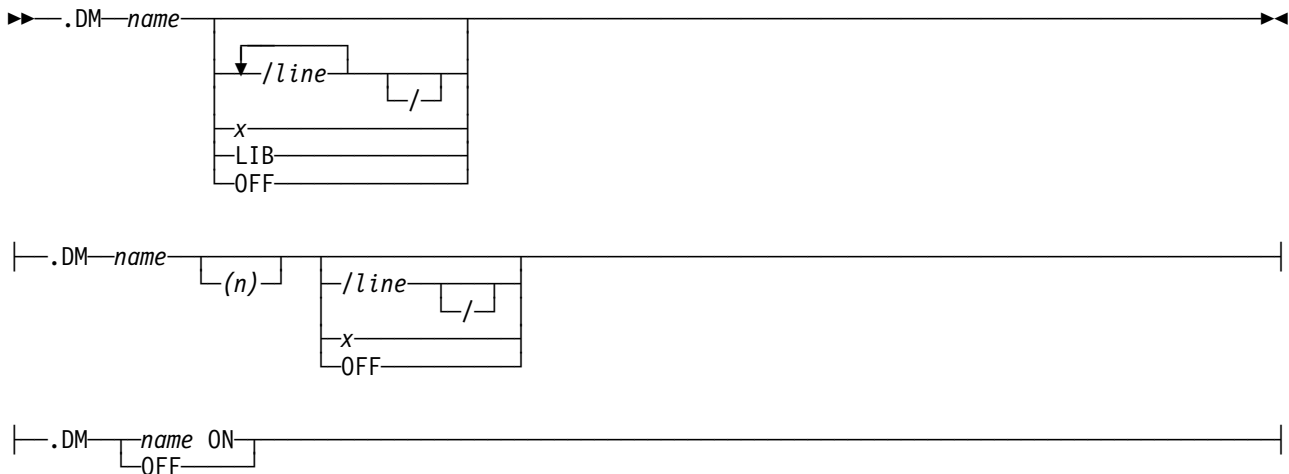
---

## .DM [Define Macro]

### Function

Use the .DM [Define Macro] control word to establish macro definitions for sequences of SCRIPT/VS control words or text lines. SCRIPT/VS macros are invoked by preceding them with periods, as if they were SCRIPT/VS control words. To use macros defined with the .DM control word, macro substitution must be enabled with the .MS [Macro Substitution] control word.

### Syntax



### Parameters

- name** Symbolic name you want to assign to the macro, so that you can invoke it with the control line:
- `. name`
- It can contain a maximum of ten nonblank characters.
- name(n)** Indicates that the line that follows is to be stored as part of the macro definition in line *n*. The values for *n* need not be sequential when the macro is defined, but if the same value for *n* is given on two uses of the .DM control word, only the latest value for the line is stored. Macro lines are executed in numerical sequence.
- When a line number is given with the name, only one line of the macro may be given. (*n*) must follow the macro name *without* intervening blanks, and must be a positive integer or zero.
- If *n* is omitted, that is, name() is specified, macro elements are assigned with line number increments of ten.
- /** Any character used to delimit the individual lines in the macro. The final delimiter may be omitted.
- line** Any SCRIPT/VS control word line or line of data that you want to include in the macro definition. It may contain symbolic names, or any of the special macro variables `&*` or `&*1` through `&*n` (see “Remarks:”). Unless you have turned symbol substitution off with .SU OFF, inline symbols are substituted during definition processing (the time the macro is initially processed). If the line parameter is omitted, the macro (or macro line if *n* is given) is stored as a null macro or macro line.



<b>x</b>	Indicates that you want the current value of a macro or macro line assigned to the symbol &x. x must be a single alphanumeric character. If you give two or more characters, SCRIPT/VS treats the first as a delimiter and the others as a line to be inserted in the macro definition.
<b>LIB</b>	Causes the macro to be defined by retrieving the lines of the macro from a library. The name of this library can be defined using the LIB option of the SCRIPT command. If LIB is used to define a macro, the definition retrieved from the library completely replaces the current definition (if one exists). If LIB is specified but no definition with the macro name given exists in the library, the macro is undefined. Because macro names are only in uppercase, names are folded to uppercase before the library is accessed. The LIB parameter sets up an entirely new macro definition; no line number can be given with the macro name. The LIB parameter can be used independently of the setting of the .LY [Library] control word.
<b>name ON</b>	Indicates that all subsequent input lines, until the next line containing only .DM OFF, should replace any existing macro definition.  The ON parameter turns off symbol substitution and GML tag processing so that symbols and GML tags are saved as part of the macro definition.
<b>name [(n)] OFF</b>	Deletes a macro definition or a line from a definition, or ends macro definition mode. The OFF parameter turns symbol substitution and GML tag processing back on. If you specify .DM OFF the following rules apply: <ul style="list-style-type: none"> <li>• It must be on a line by itself.</li> <li>• There must be a blank between the control word and the parameter.</li> <li>• A control word modifier cannot be used on this line.</li> </ul>

## Remarks

1. The following symbols have special meanings within macros:

&\*: is the line passed to the macro when it is invoked. Thus, if a macro defined with

```
.dm typeit on
.ty =====
.ty &*
.ty =====
.dm off
```

is invoked with the line

```
.typeit Hello, Nora!
```

then the symbol &\* has the value "Hello, Nora!" The processing of this macro results in the lines

```
=====
Hello, Nora!
=====
```

being displayed at your terminal.

## .DM [Define Macro]

&\*0: contains the number of tokens or words passed when the macro is called. Note that blanks, arithmetic operators, and parentheses delimit tokens. A single token can contain these and other special characters if it is enclosed in single quotation marks. Using the previous example, the value of &\*0 is 2.

&\*1 – &\*n: are the tokens passed to the macro when it is called. You can pass as many tokens to a macro as fit on the input line. If the .typeit macro is invoked

```
.typeit Processing section 5...
```

then &\*1 has a value of "Processing," &\*2 has a value of "section," and &\*3 has a value of "5..." The value of &\*0 is 3.

2. Be careful when you mix the forms of the .DM control word. You can use the single line form and the subscripted form within an inline macro definition, but you cannot use an inline macro definition within another inline macro definition.

3. Macro calls are treated as invalid control words if you have not enabled the .MS [Macro Substitution] control word:

```
.ms on
```

If the macro name is the same as a control word name and the .MS control word has not been enabled, the control word is processed.

4. Values for the symbols &\*1 through &\*n are established whenever a macro is invoked. These values are local to the current level of macro invocation and are undefined at the end of the macro.

5. Symbol substitution and GML tag processing are automatically turned off during macro definition only if you use the .DM macroname ON form of macro specification.

6. A macro name can be the same as the two-letter name of a control word. Such a macro effectively redefines the control word by getting control whenever the control word is encountered.

7. Macros can be invoked recursively. To avoid looping situations for recursive invocation, SCRIPT/VS keeps invocation counts for macros. If a given macro is opened more than 99 times, an error message is issued and processing stops. If more than 255 macros of any name are open at the same time, an error message is issued and processing stops. If you need to write recursive macros or need the ability to have more than 255 macros open at a time, see the "Tailoring SCRIPT/VS" chapter in the *DCF Text Programmer's Guide* for information on how to modify the message severity code for messages 583 and 584.

8. If macros are defined with multiple macro lines on a single line of input, for example,

```
.dm zzz /line1/line2/line3/
```

the macro is stored as if it had been entered on separate lines with an increment of ten,

```
.dm zzz(10) /line1/
```

```
.dm zzz(20) /line2/
```

```
.dm zzz(30) /line3/
```

and the new definition completely replaces any existing definition with the same name.

9. Macros defined using sequence numbers can be defined with sequence numbers in any order. However, the macro is executed as if the lines had been entered in sequence. Macro lines can be redefined at any time within a document, or lines can be added or inserted into an already existing macro definition. This addition or redefinition of lines takes place based on the sequence number specified. The largest sequence number allowed is 32767.

10. When using the .DM *name* ON form to create an inline macro definition, the macro lines are numbered in increments of 10. The largest line number allowed is 32767, therefore the largest number of lines in an inline macro definition is 3276.

11. If an entire macro is assigned to a symbol "x," it is stored in the form

```
#line1#line2#line3#linen#
```

where # represents a separator character of hexadecimal X'FF'. If only a single line of a macro is assigned to a symbol "x," it is stored in the form #line#. If you want to print this symbol, the .TR control word must be used to convert this character to one that is available on the printer being used, if this is required. When using the symbol-assignment capability, remember that the maximum length for a symbol or macro line is 252 characters.

The symbol-assignment capability can be used to test the existence of a macro or a macro line, as follows:

- If the macro (or macro line) does not exist, the symbol "&x" is assigned a null value (&L'&x=0).
- If the macro (or macro line) does exist, but has a null value, the symbol "&x" is assigned a value of hexadecimal X'FFFF', which is two consecutive separator characters (&L'&x=2).
- Else, the symbol "&x" has the value of the complete definition of the macro (or macro line) as described before (&L'&x>2).

12. Use of the LIB option of the .DM control word allows a macro definition to be explicitly retrieved from the library. Use of the .LY control word allows macro definitions to be retrieved from the library when a macro is *used* in a document where a definition for it does not currently exist.

## Examples

- The easiest means of defining a macro within a document is

```
.dm echo on
.ty ===
.ty &*
.ty ===
.dm off
```

Symbols in the macro definition are saved as part of the macro and are substituted whenever the macro is executed.

- Macros can be defined within a document in several other ways.

A short macro can be defined with a single .DM control word:

```
.su off
.dm echo /.ty ===/.ty &*/.ty ===
.su on
```

Each line can be defined separately, and each line can be given an explicit line number:

```
.su off
.dm echo(1) /.ty ===
.dm echo(2) /.ty &*
.dm echo(3) /.ty ===
.su on
```

If line numbers are not given explicitly, they are automatically generated, using an increment of ten:

```
.su off
.dm echo() /.ty ===
.dm echo() /.ty &*
.dm echo() /.ty ===
.su on
```

Whenever any of these forms are used, remember to turn symbol substitution off during the definition so that symbol names are saved as part of the macro. If your macros include GML tags, you need to turn tag processing off using .GS TAG OFF.

- There are several ways of defining a macro that resides in a macro library:

## .DM [Define Macro]

A macro can be explicitly defined from a macro library with the LIB parameter:

```
.dm echo lib
```

If library access has been enabled for macros, the macro is defined the first time it is used:

```
.ly mac
.  
.  
.  
.echo Spoon River
```

- Macros can contain conditional and iterative processing and can use local variables. For example, this macro types each of its tokens separately:

```
.dm echoem on
.se *i = 0
...loop
.se *i = &*i + 1
.if &*i gt &*0 .me
.ty Token &*i.: &*i
.go loop
.dm off
```

- Macros can be used to rename control words. For example:

```
.dm bx on
.'bx &*
.dm off
```

If you define a macro with the same name as a control word, the macro is processed instead of the control word if macro substitution is on (as set with .MS ON). For example, if you define a macro named IM, it is executed instead of the .IM [Imbed] control word:

```
.dm im on
.ty Now Imbedding: &*
.ec .im &*
.dm off
```

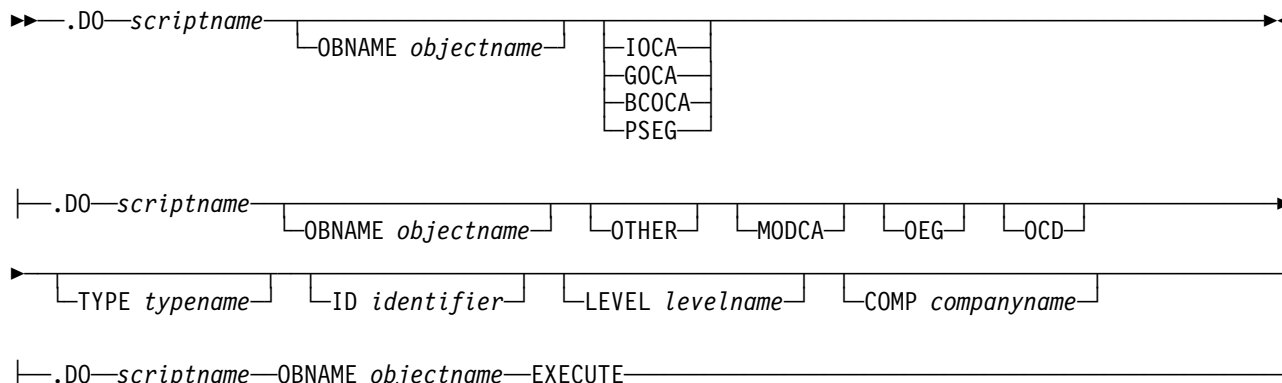
Note that if you use a control word within a macro of the same name, the .EC [Execute Control] control word or the control word modifier must be used to prevent a macro-recursion loop.

## **.DO [Define Object]**

## Function

Use the .DO [Define Object] control word to describe an external object which can then be included with the .IO [Include Object] control word when formatting for an AFP logical device. Such objects must be completely described with the .DO control word before they can be included with the .IO control word. As many .DO control words as necessary can be used to describe the object.

## Syntax



## Parameters

<i>scriptname</i>	Specifies the internal name of the object being described. <i>scriptname</i> is the name you use to include the object with the .IO [Include Object] control word. <i>scriptname</i> can be a maximum of 16 national characters. This name does not have to be the same name given with the OBNAME parameter.
-------------------	---

**OBNAME** Specifies the external name of the object. If *objectname* can be up to 64 characters. If *objectname* includes blanks, single quotes, or both, the whole name **must** be enclosed in single quotes. To use a single quote in the name, enter it as two consecutive single quotes. An object with this name must be available to the presentation system when processing the SCRIPT/VS output file. For MO:DCA objects, the *objectname* must be the same as the name in the begin and end structured fields in the object. Although this parameter is optional on a particular .DO control word, the *objectname* must be specified for each object before it is included with the .IO control word. See the documentation for your presentation system to determine what form of *objectname* you need to specify.

<b>EXECUTE</b>	Indicates that the object is executable for use in a hypertext link with the .NV [Navigate] control word.
----------------	---

<b>IOCA</b>	Identifies the <i>objectname</i> as an IOCA image object. Identifying an object as IOCA when it is not IOCA can cause an exception condition by the presentation system. Although this parameter is optional on a particular .DO control word, IOCA, GOCA, BCOCA, PSEG, or OTHER must be specified for each object before it is included with the .IO control word.
-------------	---

<b>GOCA</b>	Identifies the <i>objectname</i> as a graphics object. Identifying an object as GOCA when it is not GOCA can cause an exception condition by the presentation system. Although this parameter is optional on a particular .DO control word, IOCA, GOCA, BCOCA, PSEG, or OTHER must be specified for each object before it is included with the .IO control word.
-------------	--

## .DO [Define Object]

<b>BCOCA</b>	Identifies the <i>objectname</i> as a bar code object. Identifying an object as BCOCA when it is not BCOCA can cause an exception by the presentation system. Although this parameter is optional on a particular .DO control word, IOCA, GOCA, BCOCA, PSEG, or OTHER must be specified for each object before it is included with the .IO control word.
<b>PSEG</b>	Identifies the <i>objectname</i> as a page segment containing IOCA or GOCA. Only page segments containing IOCA or GOCA can be used with this control word. Identifying an object as a page segment when it is not a page segment can cause an exception condition by the presentation system. Although this parameter is optional on a particular .DO control word, IOCA, GOCA, BCOCA, PSEG, or OTHER must be specified for each object before it is included with the .IO control word.
<b>OTHER</b>	Identifies the <i>objectname</i> as a non-MO:DCA object. Identifying an object as OTHER when it is not OTHER can cause an exception condition by the presentation system. Although this parameter is optional on a particular .DO control word, IOCA, GOCA, BCOCA, PSEG, or OTHER must be specified for each object before it is included with the .IO control word.
<b>MODCA</b>	Indicates that this object is carried in a MO:DCA container object.
<b>OEG</b>	Indicates that this object includes an OEG structured field.
<b>OCD</b>	Indicates that this object is blocked using OCD structured fields.
<b>TYPE</b>	Specifies a generic name used to refer to the object type carried in the container object. <i>typename</i> value can be up to 32 characters. If <i>typename</i> includes blanks, single quotes, or both, the whole name <b>must</b> be enclosed in single quotes. To use a single quote in the name, enter it as two consecutive single quotes. This parameter is used only when specifying the OTHER parameter to describe a non-MO:DCA object. Although this parameter is optional on a particular .DO control word, if the OTHER parameter is specified, TYPE must be specified for each object before it is included with the .IO control word.
<b>ID</b>	<p>Specifies the Register Object Identifier of this object. This can be a 16-character hexadecimal string. This parameter is only used when specifying the OTHER parameter to describe a non-MO:DCA object. This parameter is optional.</p> <p>If the identifier includes blanks, hexadecimal values, or single quotes, the whole identifier must be enclosed in single quotes. To use a single quote in the identifier, enter it as two consecutive single quotes.</p>
<b>LEVEL</b>	Specifies the release level or version number of the object type carried in the object container. <i>levelname</i> can be up to 8 alphanumeric characters. This parameter is only used when specifying the OTHER parameter to describe a non-MO:DCA object. This parameter is optional.
<b>COMP</b>	Specifies the name of the company that owns the syntactic and semantic definition of the object type. If the object type is defined by a standards organization, specify the name of that standards organization. <i>companyname</i> value can be up to 32 alphanumeric characters. This parameter is only used when specifying the OTHER parameter to describe a non-MO:DCA object. This parameter is optional.

## Notes

- The .DO control word is only valid for AFP devices (physical devices 3820, 4028, and AFP) and is ignored for all other devices types (including the 3800-3).

## Remarks

1. The level of support of included objects depends on your presentation system.
2. The .DO control word is used to describe an existing object which can be included in the current page with the .IO control word. Since SCRIPT/VS does not read the objects when they are included with the .IO control word, the .DO control word is used to give that information. As many .DO control words as needed can be used to describe one object. If multiple values are specified for the same characteristic, the most recent specification is used. Before an object can be included with the .IO control word, the following characteristics must be specified:

*scriptname*

OBNAME *objectname*

IOCA, GOCA, BCOCA, PSEG, or OTHER.

If OTHER is specified you must also specify:

TYPE *typename*

ID *identifier* if TYPE *typename* is not a recognized TYPE

All other characteristics are optional.

Failure to provide all of the required characteristics causes an error when the object is included with the .IO control word.

3. SCRIPT/VS does not read the object and cannot verify that the description is accurate. Failure to describe the object accurately can cause unpredictable results, including exception conditions by the presentation system that processes the SCRIPT/VS output.
4. The following parameters are for use only when the OTHER parameter is specified for non-MO:DCA objects: MODCA, OEG, OCD, TYPE, ID, LEVEL, and COMP. If any of these parameters are specified for IOCA, GOCA, BCOCA or PSEG-type objects, they are ignored.
5. When using an object type of OTHER, you can specify the MODCA, OEG, or OCD parameters to supply more information about the object.  
  
Only use the MODCA parameter if you are certain that the object is carried within a MO:DCA container object. If you don't specify MODCA, and the object is within a MO:DCA container object, your object is still correctly handled by the presentation system.  
  
Only use the OCD parameter if you are sure your object is blocked using OCD (Object Container Data) structured fields. If you don't specify OCD, and the object is blocked by OCD structured fields, your object is still handled correctly by the presentation system.  
  
Only use the OEG parameter if you are sure your object includes an OEG (Object Environment Group) structured field. If you don't specify OEG, and the object contains an OEG structured field, your object is still handled correctly by the presentation system.  
  
The TYPE parameter can be used to supply a generic name for the kind of object when defining a non-MO:DCA object. The given name must be recognizable by the presentation system. An example of some given names are:  
  
TIFF - Tag Image File Format  
PCX - Paintbrush Picture File Format  
DIB - Device Independent Bitmap
6. When specifying OTHER to refer to a non-OCA object, the ID parameter is automatically filled in if the given TYPE value is recognized as one of those types that have registered identifiers. The following table shows registered identifiers for each type:

## .DO [Define Object]

Object	OID Name
PTOCA PT1	X'06072B120004010102'
GOCA DR/2V0	X'06072B120004010103'
GOCA DR/3V1	X'06072B120004010104'
IOCA FS10	X'06072B120004010105'
IOCA FS20	X'06072B120004010106'
BCOCA BCD1	X'06072B120004010107'
IM Image	X'06072B120004010108'
PTOCA PT2	X'06072B12000401010A'
IOCA FS11	X'06072B12000401010B'
EPS	X'06072B12000401010D'
TIFF (Tag Image File Format)	X'06072B12000401010E'
DIB (Device Independent Bitmap), Windows Version	X'06072B120004010111'
DIB (Device Independent Bitmap), OS/2 PM Version	X'06072B120004010112'
PCX	X'06072B120004010113'

7. If an ID *identifier* is required that is not a registered identifier, or the desired identifier is different from the registered identifier for the TYPE specified, use the &x' symbol to create an identifier of hexadecimals:
- ```
.do fred obname 'fred.xxx' type xxx id &x'0102030404.'
```
8. The external name should be encoded using EBCDIC code page 500 to represent the characters in the name. External object names should consist only of the characters A—Z, 0—9, \$, #, and @. For example, to reference an object with the name 'BOB', the external name should be encoded as X'C2D6C2'. SCRIPT/VS does no codepoint translation on the external object name specified on the .DO control word.



# .DR [Define Rule]

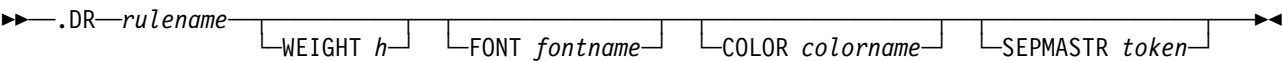
## Function

Use the .DR [Define Rule] control word to define a *named* rule to be used for drawing lines with the .HR [Horizontal Rule], .VR [Vertical Rule], .BX [Box], .UD [Underscore Definition], and .TD [Table Definition] control words.

For line devices, rules are constructed from characters representing fragments of rules and rule intersections. These are called box characters.

For page printers, rules are drawn with lines of varying thickness.

## Syntax



## Parameters

*rulename* Identifier for the rule being defined. It can contain a maximum of 16 national characters, but it must not be a valid space-unit designation. After the definition, *rulename* can be used in the .BX, .HR, .VR, .UD, and .TD control words. If no other parameters are given, the rule is defined with the default weight for page printers or the current font for line devices.

**WEIGHT** Width or thickness of the *named* rule being defined. *h* is any valid space unit. The WEIGHT parameter is ignored for line devices. If *h* is not specified, the weight specified with the default rule definition is used.

**FONT** The font of the rule being defined. The *fontname* parameter can be any *named* font defined with the .DF [Define Font] control word or specified on the CHARS option of the SCRIPT command. If *fontname* is not specified, the rule is drawn using the currently defined font. FONT is ignored for page printers.

**COLOR** Specifies the color to be used with the rule being defined. The COLOR parameter applies only to output devices with color capability. *colorname* can be any color defined previously with the .CR [Color] control word.

If COLOR is not specified, the default color for the output device is used.

**SEPMASTR** Specifies that rules are being associated with output separation masters.

The *token* is a 1- to 8-character name that identifies which separation masters this rule should be associated with. The token name corresponds to the items selected for separation masters by using the .SM control word.

The special token value of ALL is used to indicate that this rule should be contained in all separation masters, including the default.

For more information on the SEPMASTR parameter, see “.SM [Separation Master]” on page 353.

### Remarks

1. The *fontname* specified with the .DR control word can be a list of fonts previously defined with the LIST parameter of the .DF control word. A rulename defined with the LIST parameter cannot be used in the .TD [Table Definition] control word. For more information on using the LIST parameter, see “.DF [Define Font]” on page 130.
2. A rule cannot have a *name* that is also a valid space-unit designation. Table 6 on page 59 lists the valid space units. This is because rule names that are also valid space-unit designations are not interpreted as rule names on the .HR [Horizontal Rule], .VR [Vertical Rule], .BX [Box], .UD [Underscore Definition], or .TD [Table Definition] control words.
3. If you redefine an existing rule while a vertical rule is in progress, the new definition of that rule takes effect for any new rule that calls for the name, but all rules in progress continue with the old definition until they end.
4. For page printers, the default rule is named *boxrule* and is 0.3 millimeters thick. If you do not specify rulename with the .BX, .HR, .VR, and .TD control words, they are drawn with this rule. If you redefine *boxrule*, the thickness of all such rules is changed.
5. The SEPMASTR parameter is ignored for line devices.
6. The SEPMASTR parameter is ignored if the SEPMASTR command option is not specified.  
**Note:** Examples using the SEPMASTR parameter can be found in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.
7. If you use the FONT parameter to define a rule, and the definition of that font contains a color specification, the rule will not be rendered in the font color. Use the COLOR keyword on your .DR control word to change the color of the rule.

### Examples

- You can define a *named* rule to be used in documents formatted for the 3800 Printing Subsystem Model 1 by entering  

```
.dr raven font &$CHAR(2)
```

Rules drawn with RAVEN are constructed with the box characters associated with the second font specified with the CHARS option of the SCRIPT command.
  - You can define rules of varying widths for use on a page device. For example:  

```
.dr thin weight .3mm  
.dr thick weight .6mm
```
  - When a document is formatted for a page printer and a line printer, *named* rules can be defined for use on both types of devices. For example, if you specified the following:  

```
.dr dot weight .1mm font &$CHAR(2)
```

    - Then when rule DOT is used on a page printer, rules are 0.1 millimeter thick.
    - When rule DOT is used on a 3800 Printing Subsystem Model 1 and GT12 and SI12 are specified on the CHARS option, rules are constructed using the box character set associated with the font SI12.
    - When rule DOT is used on a 1403 printer and no CHARS option is specified, &\$CHAR(2) resolve to null, and rules are constructed using the default box character set of the 1403.
- If a rule is defined with a weight so small that it rounds to zero (0), no message is issued but the default rule is used instead of the newly defined rule.

- The following .DR control word is invalid, because the *name* of the rule is a valid space-unit designation:

`.dr 1mm weight 1mm`

- To designate color for a rule, specify

`.dr bluerule weight 1mm color blue`

`.hr bluerule 10mm for 20mm`

---

## **.DS [Double Space Mode]**

### **Function**

Use the .DS [Double Space Mode] control word when you want your output to be double-spaced rather than single-spaced.

### **Syntax**

►► .DS ◀◀

### **Remarks**

1. The .DS control word doubles the line spacing set by the .LS control word. When double-spacing is in effect, each space or skip caused by a .SP or .SK control word is doubled (thus, .SP 2 yields four spaces). However, if the .SP or .SK control word indicates *absolute* spaces, the space count is not doubled.
2. Additional space is placed above each output line and is discarded if the line falls at the top of the body of the page. This can result in columns being set short by the amount of discarded space.
3. If double spacing is in effect, the resumption of single spacing (when you enter .SS) is automatically deferred for one line if the next output line is a text line. For example, if you entered the following:

```
.ds
These lines are double-
spaced. They are the last
ones before we enter
the .SS [Single Space Mode]
control word and a bit
more text.
```

```
.ss
Now, we begin single-
spacing, but we have
to wait one line for the
deferred single-spacing
to begin.
```

the result looks like this:

```
These lines are double-spaced.
```

```
They are the last ones before we enter
```

```
the .SS [Single Space Mode]
```

```
control word and a bit more text.
```

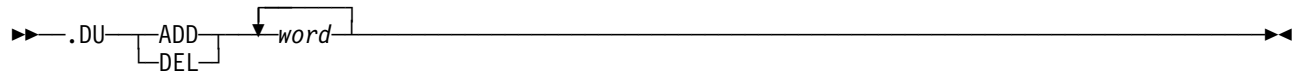
```
Now, we begin single-spacing,
but we have to wait one line
for the deferred single-spacing to begin.
```

## **.DU [Dictionary Update]**

### **Function**

Use the .DU [Dictionary Update] control word to add or delete words from an addenda dictionary. The changes to the dictionary that are specified using this control word are in effect only during the formatting of the current document.

### **Syntax**



### **Parameters**

**ADD** Specifies that the word or words given with the control word are to be added to the addenda dictionary.

**DEL** Specifies that the words given with the control word are to be deleted from the addenda dictionary.

*word* String of words delimited by blanks.

### **Remarks**

1. If you use the .DU control word to request that a word be added to or deleted from the addenda dictionary and that word is already in the addenda dictionary (for ADD), or if it is not in the addenda dictionary (for DEL), this does not cause an error message. The first ADD for a word puts the word in the dictionary, and all subsequent ADDs are ignored. The first DEL for a word deletes the word from the dictionary, and all subsequent DELs are ignored.
2. Words added to the dictionary using the .DU control word might include hyphens that indicate potential hyphenation points for the word. When hyphenation is in effect (specified by the .HY control word), these hyphenation points are used, unless the .HW control word has been used for the specific occurrence of the word or the use of the addenda dictionary has been suppressed with the NOADD option of the .HY control word.
3. Words that contain hyphens, such as lighter-than-air, should be supplied to the .DU control word with double hyphens at these hyphenation points, as described in “.HW [Hyphenate Word]” on page 195.
4. For languages listed in “Table 4” of the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*, the word can contain lowercase and uppercase alphabetic characters, the integers 0 through 9, and the accented characters specific for the current language as listed in the table. For languages not listed in the table, the word can contain all characters except hexadecimal X'00'.
5. Stem processing is used for verification against both the main and the addenda dictionaries when requested, using the .SV control word. For more details on stem processing, see the chapter on “Verifying Spelling” in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.
6. Words can be added to the addenda dictionary even when spelling verification is off or is in effect against only the main dictionary.
7. The addenda dictionary can contain a maximum of between 1600 and 1700 words. The exact number of words varies depending on the number of characters in the words. If your addenda dictionary approaches this maximum, consider creating a permanent user dictionary. See the chapter entitled “Creating and Maintaining User Dictionaries” in the *Document Composition Facility: SCRIPT/VS Text*

## **.DU [Dictionary Update]**

- | *Programmer's Guide* for information on creating a user dictionary. See “.DL [Dictionary List]” on  
| page 140 for information on using a user dictionary.
8. The addenda dictionary might not retain all hyphenation points specified. Each word placed in an addenda dictionary is divided into four three-character groups following the first vowel. Only one hyphenation point is recorded for each of the four groups.
  9. To change the hyphenation points of a word that is already in the addenda dictionary, you must delete the word with .DU DEL prior to adding the word with different hyphenation points.
  10. If a word is found in an addenda or user dictionary with no hyphenation points, DCF does not search any other dictionaries. If .HY ALG was specified and .HY ALT was not specified, the word is sent to the algorithmic hyphenator.

## **Examples**

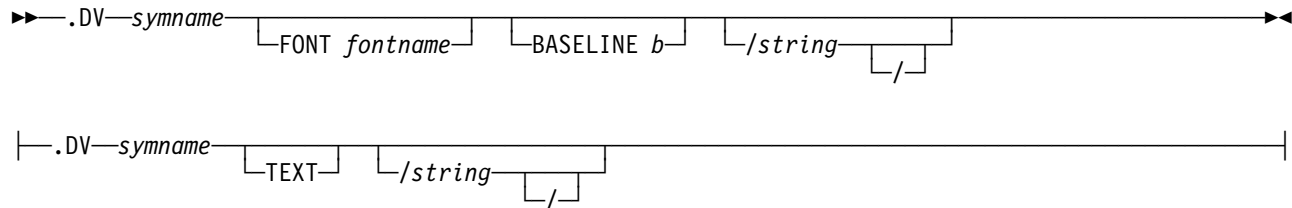
- If you want to add words to an addenda dictionary, specify  
  `.du add Happy Trails Ri-ver-front In--law`
- If you want to delete words from an addenda dictionary, specify  
  `.du del Rover Over Tele-phone`

## **.DV [Define Variable]**

### **Function**

The .DV [Define Variables] control word allows you to define and assign values to text variables.

### **Syntax**



### **Parameters**

|                 |                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>symname</i>  | Is the name of the symbol that is defined. This name must conform to all requirements of a symbol name as set with the .SE [Set Symbol] control word. Array symbols are not permitted. The symbol name can contain a maximum of 10 nonblank characters.                                                                                                  |
| <b>FONT</b>     | Is the font of the variable being defined. The fontname parameter can be any named font previously defined with the .DF [Define Font] control word or specified with the CHARS option of the SCRIPT command.                                                                                                                                             |
| <b>BASELINE</b> | Specifies a baseline position to be associated with this defined variable. <i>b</i> can be specified in any valid vertical space unit. The shift is applied to the original baseline for the line.                                                                                                                                                       |
| <b>TEXT</b>     | Specifies that the string is text characters and has no font change or baseline shift associated with it. This form of defined variable does not change fonts; it is set in the current font.                                                                                                                                                            |
| <b>/</b>        | Is any delimiter character. This character can appear within the string, because the characters within the string are not examined. A closing delimiter is required if the defined variable has trailing blanks or ends in the delimiter character. The last character is examined to see if it is the delimiter character, and if it is, it is removed. |
| <i>string</i>   | Is any string of text characters. The <i>string</i> should not contain control words, GML tags, or macro calls; if it does, the results are unpredictable.                                                                                                                                                                                               |

### **Notes**

- Defined variables should not be used in conditional processing with the .IF [If] control word.
- Characters that have special meaning to SCRIPT/VS are treated as text when used in a defined variable. For example, if the GML starter tag delimiter is defined as a colon

```

.dv colon /:
&colon.p.This is a paragraph

then the lines above print as

:p.This is a paragraph

```

- If you want to use the page number symbol (&) in a defined variable, you must also specify the TEXT parameter.

### Remarks

1. The font name specified with the .DV control word can be a list of fonts that you previously defined with the LIST parameter of the .DF control word. For more information on using the LIST parameter, see “.DF [Define Font]” on page 130.

2. Variables set with the .SE and .DV control words do not contain the same information. For example,

```
.dv setdv text '*'  
.se setse    = '*'
```

where &setdv has the length of 3 and &setse has the length of 1. (You can query the length by specifying &L'&setdv and by specifying &L'&setse, respectively.) The additional bytes in &setdv are required to indicate that the symbol was defined with the .DV control word. This additional information is returned when processing this symbol with the .TY [Type on Terminal] and .WF [Write to File] control words.

3. The string portion of a defined variable must be specified last.
4. An attempt to place control words, GML tags (if GML tag scanning is active), or macros (if macro substitution is active) in the string portion of a defined variable causes unpredictable results.
5. Use the TEXT parameter when the string portion contains a character that has special meaning to SCRIPT/VS, but you want the character interpreted as a text character. Two exceptions exist: hexadecimal X'40' is always interpreted as a blank and hexadecimal X'16' is always interpreted as a backspace, even if these two characters appear in a defined variable that uses the TEXT parameter.
6. The same symbol name can be set with either the .DV [Define Variables] or the .SE [Set Symbol] control word. The last definition given is used.
7. When using symbols in the string portion of a .DV control word, turn off symbol substitution if the value of the symbol varies.
8. If a symbol name appears at the end of the string portion of a .DV control word, it is not recognized as a symbol name unless you delimit it with a period.
9. A closing delimiter on a string is required if the string has trailing blanks or if it ends with the same character you have chosen as the delimiter. The following example uses the .DV control word to define a variable to print the ‘greater than or equal’ sign in the Pi font for the 3820 Page Printer. In this case, the string ends with the same character chosen as the delimiter:

```
.df pi type ('pi serif' 10) codepage tlgpi363  
.dv ge font pi /&X'61./
```

The delimiter character, '/', used with the .DV control word has the same hexadecimal code point, X'61', as the ‘greater than or equal’ (≥) sign in the Pi font. Thus, the closing delimiter is required to prevent the &X'61 from being interpreted as a closing delimiter and being deleted.

10. When setting a symbol to a defined variable, you must enclose the defined variable in single quotation marks:

```
.dv bullet text /&X'ba  
.se u1 = '&bullet.'  
This is u1: &u1 is the end of this.
```

Otherwise, the blank after the symbol is lost.

11. When setting a symbol to a defined variable which is a hexadecimal character, unexpected results may occur because some hex points are used internally by DCF. For example, hexadecimal X'FE' should not be used in a defined variable that is used in a complex symbol because X'FE' is needed internally for symbol substitution.



12. The FONT and TEXT and the BASELINE and TEXT sets of parameters are mutually exclusive. If either of these sets of parameters is specified at the same time, the last parameter specified is the one that is used.
13. It is a good idea to use a closing delimiter on strings that end in a symbol. If the value of the symbol ends in trailing blanks or the delimiter character, a closing delimiter is required. For example:
 

```
.se zorn 'slash/
.dv foo /&zorn./
```
14. All input lines containing .DV symbols are spell checked. If the symbols are part of a .DU control word line, they are processed correctly for the addenda dictionary update. The string assigned to a .DV symbol can cause incorrect hyphenation points on any line that is substituted for the symbol value. This can occur even if the string contains only valid words. To alleviate this problem, hyphenation can be temporarily disabled with the .HY [Hyphenate] control word.

## Examples

1. A GML profile commonly defines symbols for SCRIPT/VS special characters. The following example shows how such special characters would be defined in that profile using .DV:

```
.dv amp      text /&
.dv gml      text /:
.'dv semi    text /;
.dv period   text /.
.dv colon     text /:
.dv plus      text /+
```

Notice that the definition for &semi uses the control word modifier to prevent the semicolon from being interpreted as a control word separator in the .DV control word itself.

The symbol &period can be used for a control word example in which the period prints flush left. It is not necessary to clutter your GML markup with the .LI [Literal] control word whenever a text period is required in column one.

2. On the 3820 Page Printer, the .DV control word is a convenient way of representing characters that aren't on your keyboard or that are in a special font such as the Pi font. The following example shows how to print an arrow in your text:

```
.df pi type('pi sans serif' 8) codepage t1gpi363
.dv arrow font pi /&X'e3
Next is an arrow: &arrow
```

The result is

Next is an arrow: →

To position the arrow higher, specify

```
.df pi type('pi sans serif' 8) codepage t1gpi363
.dv arrow font pi baseline p2 /&X'e3
Next is an arrow: →
```

The result is

Next is an arrow: →

3. If a defined variable using the TEXT parameter is nested within another defined variable that uses the FONT parameter, the nested defined variable is set in the font that is current when that defined variable is encountered. In other words, it is set in the same font that the outer defined variable is set in. The following example is for page printers:

## .DV [Define Variable]

```
.df boldbig type(bold 18)
.dv outer font boldbig /outer &inner.
.dv inner text /inner variable
Next is outer: &outer
```

The result is

Next is outer: **outer inner variable**

4. If a defined variable not using the TEXT parameter is nested within another defined variable that uses the FONT parameter, the nested defined variable is set in the font that was current before the outer defined variable was encountered, even if a font change is required for the nested defined variable. The following example is for page printers:

```
.df boldbig type(bold 18)
.dv outer font boldbig /outer &inner.
.dv inner /inner variable
Next is outer: &outer
```

The result is

Next is outer: **outer** inner variable

5. Next is a defined variable that uses a type defined font and that is nested within another defined variable that uses the FONT parameter. The font of the inner defined variable is determined by the font that was current before the outer defined variable was encountered and by the modifications required for the font used by the nested defined variable. The following example is for page printers:

```
.df boldbig type(bold 18)
.dv italic type(italic)
.dv outer font boldbig /outer &inner.
.dv inner font italic /inner variable
Next is outer: &outer
```

The result is

Next is outer: **outer** *inner variable*

6. For the 3820 Page Printer and the 3812 Page Printer, the following control words define a variable to print the trademark symbol from the Pi font:

```
.df sym10 type('pi sans serif' 10) codepage T1GPI363
.dc cw off
.dv tradem font sym10 /&X'5e
&tradem.
.dc cw
```

The result is

®

**Note:** The hexadecimal code for the trademark symbol in the Symbol font is the same as the hexadecimal code for the default control word separator in the default font. Control word separator scanning is turned off while the symbol &tradem is being defined, to prevent misinterpretation of &X'5e'.

---

## **.EC [Execute Control]**

### **Function**

The .EC [Execute Control] control word is used to cause SCRIPT/VS to execute the given line as a control word line, even if there is a macro defined with the same name and macro substitution is ON.

### **Syntax**

►►—.EC—*controlwordline*—————►►

### **Parameters**

*controlwordline*      A SCRIPT/VS control word line.

### **Remarks**

1. Use the .EC control word whenever you want to cause SCRIPT/VS to execute a control word even when a macro is defined with the same name.

The .EC control word is useful within macros that have the same name as control words. Often, a macro that “redefines” a control word uses the control word function in addition to whatever other function it performs. In these cases, if the .EC function were not used, the same macro would be repeatedly invoked in a loop until SCRIPT/VS stopped it with a severe error message. Macro substitution could be turned OFF, but that would prevent any other macro from being invoked until macro substitution was turned ON again.

2. The control word modifier provides an implied .EC function. It also prevents the control word separator scan on that control word line. The control word modifier can be used with any control word; except when specifying the OFF parameter of the .CS [Conditional Section], .DM [Define Macro], .LI [Literal], and .WF [Write to File] control words. The control word modifier consists of a single quotation mark (') between the period and the name of the control word. (. 'ce center this line).

### **Examples**

- To define a macro called .IM to replace the .IM control word without using the .EC control word would require the following macro definition:

```
.dm im on
.ty imbedding &*1
.ms off
.im &*
.ms on
.dm off
```

In the above example, macro substitution needs to be turned off to avoid an infinite macro-substitution loop. Unfortunately, when you turn macro substitution off, it also turns off for the imbedded file and all files that it imbeds. In this situation, the .EC control word should be used:

```
.dm im on
.ty imbedding &*1
.ec .im &*
.dm off
```

The control word modifier can be used in the same way:

## **.EC [Execute Control]**

```
.dm im on  
.ty imbedding &1  
'im &1  
.dm off
```

The difference between the .EC form and the control word modifier form is that the .EC line is scanned for control word separators, whereas the control word modifier line is not. In this example, there are no differences between the two, because there are no control word separators in the input line.

- The .EC control word issues an error message if the subject control word line is not valid. To be a valid control word, it must start with a period and be followed by a valid control word and a blank. A line without a period in the first position is usually treated as text, but, as the subject of .EC, it is treated as an invalid control word.
- The .EC control word issues an error message if the control word line given is “valid” but refers to a nonexistent control word, even if a macro exists with the control word name given.

---

## **.EF [End of File]**

### **Function**

The .EF [End of File] control word simulates the end of the current file. When used in a profile, the contents of the profile preceding the .EF [End of File] control word are processed before the main document. The remainder of the profile, called the epifile, is processed after the main document.

### **Syntax**

►► .EF —————►  
          └─CLOSE─┘

### **Parameters**

**CLOSE** Indicates that SCRIPT/VS is not to hold your place in the current file, but to close it, so that the next time the file is imbedded, SCRIPT/VS begins processing at the top of the file, not at the line following the .EF control word.

### **Remarks**

1. The .EF [End of File] control word causes an end of file condition to be simulated on the current input file. SCRIPT/VS remembers the position of the .EF control word; if the file is used again, SCRIPT/VS begins reading at the line following the .EF control word instead of at the beginning of the file, unless the CLOSE operand is used. If a .EF control word appears in the profile, the contents of the profile before the .EF is encountered is read before reading the primary input file. When all processing in the primary input file is completed, any lines following the .EF control word in the profile (these lines are called the epifile) is read and processed.
2. When the .EF control word is processed in a profile, the data that follows that .EF, the epifile, is always started on a new page.
3. If the .EF control word is followed by a control word separator and additional control words, processing stops at the .EF control word. The control words that follow the .EF are not processed.
4. This control word should not be used in a macro that has used or is called by a macro using GML scanning.

---

## **.EL [Else]**

### **Function**

The .EL [Else] control word can be used in conjunction with the .IF [If] control word to process SCRIPT/VS input lines conditionally. The target line is processed only if the most recently performed .IF [If], .AN [And], or .OR [Or] control word resulted in a false condition.

### **Syntax**

►► .EL—*target* —————►◄

### **Parameters**

*target* Any valid SCRIPT/VS input line containing a control word, a macro, or text. If the most recently performed .IF [If], .AN [And], or .OR [Or] was false, the target line is processed next, with the first nonblank character after the .EL treated as the first position of the line. If the condition was true, the target line is ignored, and processing continues with the input line that follows the .EL control word line.

### **Remarks**

1. For readability, an optional SE can be added to the .EL control word without an intervening blank. This allows the control word to be written as .EL or .ELSE. However, the short form (.EL) is recommended for performance reasons.
2. The .TH [Then] and .EL [Else] control words, in conjunction with .IF [If], .AN [And], and .OR [Or], allow you to construct complex logic statements.
3. The .TH and period.EL control words themselves do not cause a break or change the true/false condition; a target control word might, if it is processed. For example, the input lines

```
.if &a eq &b
.el .cl 20
```

causes a break if &a is not equal to &b.  
In the following input lines

```
.if &a eq &b
.el .if &a ne &c
```

if &a is not equal to &b, then the condition is false and the .EL line is processed. If &a is not equal to &c, then the true/false condition changes to true.
4. Multiple .TH [Then] and .EL [Else] control words can follow a .IF [If], .AN [And], or .OR [Or] control word; only the .TH [Then] control words are executed if the .IF [If], .AN [And], or .OR [Or] resulted in a true comparison, and only the .EL [Else] control words are executed if the .IF [If] resulted in a false comparison.
5. If there is no most recently performed comparison, the target line is not processed.

## Examples

- The following input lines

```
.if &a ne &b .ty Yes,  
.if &a ne &b .ty still.
```

are equivalent to the following lines:

```
.if &a eq &b  
.else .ty Yes,  
.el .ty still.
```

---

## **.EM [Execute Macro]**

### **Function**

The .EM [Execute Macro] control word is used to cause SCRIPT/VS to execute the given line as a macro line, even if there is a control word with the name given and macro substitution is OFF.

### **Syntax**

►► .EM *macroline* ◄◄

### **Parameters**

*macroline*     An input line that invokes a SCRIPT/VS macro.

### **Remarks**

1. Use the .EM control word whenever you want to cause SCRIPT/VS to execute a macro when macro substitution is OFF.
2. If the .EM control word specifies a macro for which no valid macro definition exists, it is treated as an invalid control word, even if there is a control word of that name.
3. The control word modifier provides an implied .EC [Execute Control] function and prevents a control word line from being scanned for control word separators. If you want to prevent a macro line from being scanned for separators, you can use the control word modifier with the .EM control word:

```
.'EM .mymac A;B
```

The control word that is modified here is .EM, and this usage allows the macro 'mymac' to be executed, but prevents the data for the macro (A;B) from being misinterpreted as containing a control word separator.

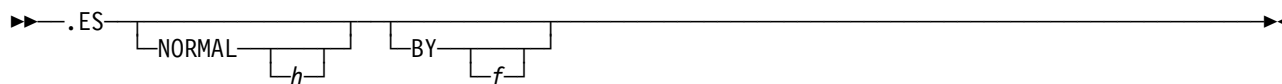


## .ES [Extra Space]

### Function

Use the .ES [Extra Space] control word to control the formatting of two or more successive blanks by specifying the width of the extra spaces. The first blank is considered to be a word space and the rest are considered to be extra spaces.

### Syntax



### Parameters

**NORMAL *h*** Establishes the width of extra spaces, until changed by another .ES control word.

*h* gives the width of the second and subsequent blanks when consecutive blanks are found in text, and it overrides the default extra space width.

If *h* is omitted, the width of subsequent extra spaces is determined by the default word space value of the current font and the BY parameter of .ES. *h* can be any valid horizontal space unit.

**BY *f*** Increases or decreases the width of extra word spaces.

*f* is the factor by which the normal width of extra spaces is multiplied to obtain the width of extra spaces. (The normal width is the default wordspace width of the current font, unless it is overridden with .ES NORMAL.)

If *f* is omitted, “1.0” is used.

**Initial Setting:** Dependent upon the default word space value of the current font.

**Default:** Restores the initial setting.

**Note:** The extra word space values are included in the active environment.

### Remarks

1. When consecutive blanks are found in text, the first blank is considered to be a word space and the rest are considered to be extra spaces. The width of the first blank is determined by the .WS [Word Space] control word. The width of each blank after the first is determined by the .ES control word.
2. Each font is designed with a default word space value appropriate to the size of the characters, and SCRIPT/VS usually uses this as the normal width of all blanks, including extra spaces. The BY parameter of .ES can be used to increase or decrease the width of extra spaces proportionally for all fonts.
3. The NORMAL parameter can be used to override the default width that comes from the current font. This establishes fixed extra spaces regardless of any change in font size.
4. The actual width of extra spaces is determined by multiplying the normal width by the factor. When neither of these has been changed with .ES, the result is 1.0 times the word space width of the current font.
5. .ES can be useful if you use the typing convention of following sentences with two blanks. SCRIPT/VS automatically places two blanks (instead of one) after full stops that occur at the end of an input line. Refer to the STOP parameter of “.DC [Define Character]” on page 112 for details. In

## **.ES [Extra Space]**

proportional typefaces, two word spaces between sentences is often excessive. The .ES control word can be used to decrease the width of the second blank without changing the width of other word spaces.

6. The width of extra spaces can be set or rounded to zero. For example,

```
.es by .33
```

decreases extra spaces to one-third their former width when processed on page printers, but sets extra spaces to zero on a line device. Values given are rounded to the capability of the device. This could result in a 0 width on line devices.

## **Examples**

- The extra spaces after a full stop are often set half as wide as word spaces in proportional typefaces. For example,

```
.es by .5
```

A full stop. A new sentence.

results in

A full stop. A new sentence.

- To set all blanks 8.5 points wide, enter

```
.ws normal p8.5
```

```
.es normal p8.5
```

- If you specify one of the parameters but give no value, the default value for that parameter is restored. If you specify .ES alone with no parameter or value, the defaults for both are restored. If you enter

```
.es
```

then any override of the default value is canceled, and the factor is reset to 1.0.

## **.EQ [Equation]**

### **Function**

The .EQ [Equation] control word identifies to SCRIPT/VS where an equation begins and ends, and it performs Formula Formatter control functions. The first time any .EQ control word is processed, the Formula Formatter is initialized. As part of initialization, the Formula Formatter defines the symbol &\$EQ to contain the level number of the Formula Formatter.

### **Syntax**



### **Parameters**

- n*** Specifies the number of input lines to be processed as a formula. If omitted, 1 is the default.
- ON** Starts an open-ended formula, where all lines are taken as Formula Formatter input until .EQ OFF ends it.
- OFF** Terminates Formula Formatter input.
- TAG** Starts an open-ended formula, just like .EQ ON except that the formula is ended by the end-tag that corresponds to the start tag whose processing macro issued the .EQ TAG control word. No other tags are processed. .EQ TAG is valid only when executed within the processing macro for a GML start tag. Any other GML markup within the scope of the .EQ is treated merely as a string, and is formatted into the formula. .EQ OFF always ends a formula, even if it was started with .EQ TAG.
- line*** The *line* can either be a line of formula input, that is, an equation, or it can be one of the .EQ control functions.

### **Remarks**

1. It makes no difference how many input lines are used to describe your formula. Spaces (blanks) and line ends are thrown away in the final output. Thus, the following notations are all equivalent:

```
.eq x=y+z
```

```
.eq 2
  x = y
  + z
```

```
.eq on
  x = y
  + z
.eq off
```

All of these produce:  $x=y+z$

Refer to the *Document Composition Facility: SCRIPT Mathematical Formula Formatter User's Guide* for more information on using the SCRIPT Mathematical Formula Formatter.

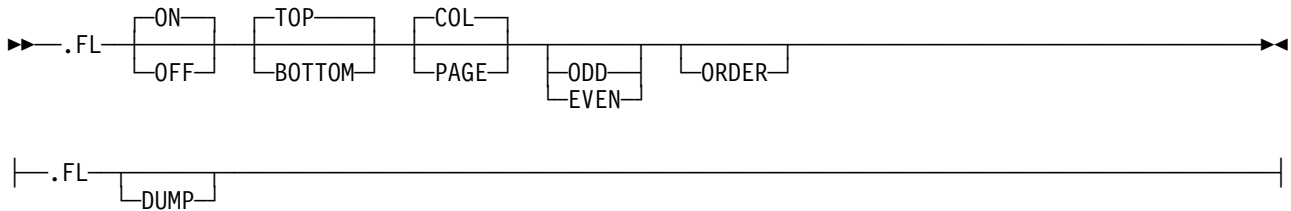
---

## .FL [Float]

### Function

Use the .FL [Float] control word to designate a block of text to be formatted and placed at the top or bottom of a subsequent output page or column. Such a block of text is called a *float*. Figure 6 on page 438 shows the relationship of the .FL [Float] to the layout of a SCRIPT/VS output page.

### Syntax



### Parameters

|               |                                                                                                                                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ON</b>     | Indicates the beginning of a float.                                                                                                                                                                                                       |
| <b>OFF</b>    | Indicates the end of a float.                                                                                                                                                                                                             |
| <b>TOP</b>    | Indicates a float that is placed at the top of a page or column.                                                                                                                                                                          |
| <b>BOTTOM</b> | Indicates a float that is placed at the bottom of a page or column. If neither TOP nor BOTTOM is specified, the float is placed at the top of a page or column.                                                                           |
| <b>COL</b>    | Specifies that the float that is starting is to be a <i>column float</i> that is formatted using the column line length.                                                                                                                  |
| <b>PAGE</b>   | Specifies that the float that is starting is to be a <i>page float</i> that is formatted using the column line length.                                                                                                                    |
| <b>ODD</b>    | Indicates a <i>page float</i> that is placed on an odd-numbered page.                                                                                                                                                                     |
| <b>EVEN</b>   | Indicates a <i>page float</i> that is placed on an even-numbered page. If neither ODD nor EVEN is specified, the float can be placed on any page.                                                                                         |
| <b>ORDER</b>  | Specifies that after all previously ordered floats have been placed, this float is to be placed in the next available top or bottom float space in the column or page. Ordered floats are placed in the order in which they were entered. |
| <b>DUMP</b>   | Causes SCRIPT/VS to put all unplaced floats in columns or on pages. As many extra columns or pages as necessary are added to place all pending floats. An automatic dump is generated at the end of the primary input file.               |

### Notes

- .FL ends a keep, float, footnote, named area, or table.
- .FL ensures that the page is started.
- This control word saves the current environment.

## Remarks

1. If a float is already in progress when the .FL control word is encountered, the first float is ended, as though .FL OFF had been processed, before the new .FL control word is processed. All .FL control words except .FL OFF and .FL DUMP cause a new float to be started.
2. If .FL OFF is encountered when no float is in progress, nothing happens.
3. If several floats are waiting to be placed, only one can be placed at the top or bottom of a column or page (except when a dump is being processed). The remaining floats are placed, one at a time, on the tops and bottoms of subsequent pages and columns.
4. If several columns are defined, each column can have a float at the top and at the bottom of it.
5. When the DUMP parameter is processed, TOP, BOTTOM, ODD, and EVEN are ignored, but ORDER, COL, and PAGE are honored.
6. Each column on a page has space available for one top and one bottom float. At the time the page is started, each pending float is placed in the first location on the page that is available and that has room for it. This process continues until the first float that does not fit on the page is encountered.
7. If COL is specified after ODD or EVEN, the float is forced to be a *column float*. Conversely, if ODD or EVEN is specified after COL, the float is forced to be a *page float* (the COL parameter is ignored).
8. A head-level one causes floats to be dumped.
9. Specification of the DUMP parameter causes SCRIPT/VS to place all floats and footnotes before resuming input text processing.
10. If you specify the DUMP parameter with other parameters, only the DUMP parameter is processed.
11. Certain control words are not allowed within a float. If one of the disallowed control words is encountered, the float is immediately ended, as though .FL OFF had been processed, and then the disallowed control word is executed. A warning message is issued, identifying the control word that ended the float. See Table 15 on page 442 for a list of the disallowed control words.
12. The first line in a top column float has the extra leading specified with the EXTRA parameter of the .LS [Line Spacing] control word removed when it is the only top float on the page.

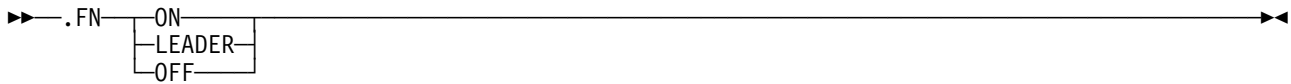
---

## .FN [Footnote]

### Function

Use the .FN [Footnote] control word to set aside lines of formatted output text to be positioned at the bottom of the current page, if possible, or at the bottom of subsequent pages. Figure 6 on page 438 shows the relationship of a .FN [Footnote] to the layout of a SCRIPT/VS output page.

### Syntax



### Parameters

- ON** Marks the beginning of the material in the footnote.
- LEADER** Defines a leader to be placed at the top of the footnotes on the page to separate the footnotes from the text of the page. The initial leader is a space of one line and a horizontal rule 16 figure spaces long.
- OFF** Marks the end of the footnote material or the footnote leader.

### Notes

- .FN ends a keep, float, footnote, named area, or table.
- .FN ensures that the page is started.
- This control word saves the current environment.
- A new .FN LEADER takes effect on the next page.

### Remarks

1. .FN ON starts a footnote. All lines until the subsequent .FN OFF control are put in the footnote. If .FN OFF is encountered when no footnote is in process, it is ignored.
2. There is no maximum size for a footnote. All text until a .FN OFF command is given is included in the footnote unless the footnote is prematurely ended by a disallowed control word.
3. The first footnote on a page is automatically started with a leader that can be redefined with .FN LEADER and .FN OFF. If the leader, plus the first line of the first footnote is larger than the page body, the footnote leader is ignored.
4. The .FN control word does not act as a break. Therefore, a footnote can be entered anywhere in the body of the page without disrupting the concatenation of the body text.
5. Footnotes run across the page in a single column. The line length can be changed in the footnote.
6. When the footnote is started, offsets are cleared, indentation is set to the current .IN (indent) value, and the column line length is set to the line length. You must include a .OF control word if you want the footnote offset.

When the footnote ends, any changes within the footnote to the indentation, font, or certain other values in the formatting environment, are automatically restored to the values in effect before the footnote started. See “The SCRIPT/VS Formatting Environment” on page 451 for a list of all the values that are automatically saved and restored after a footnote.

7. Widow-zone processing is not performed in footnotes.
8. When you define a footnote leader, the same restrictions that apply to running headings and footings also apply to footnote leaders.
9. Control words that begin a new page should not be used within a footnote leader. These control words cause a message to be issued and are then ignored. The ignored control words are:

- .CB [Column Begin]
- .CC [Conditional Column Begin]
- .CP [Conditional Page Eject]
- .IX [Index]
- .PA [Page Eject]
- .TC [Table of Contents]

In addition, any of the .H0 [Heading Level 0]—.H6 [Heading Level 6] control words that start a new page are ignored. Note that .CB [Column Begin] and .CC [Conditional Column Begin] are ignored only if they would cause a page eject.

10. Footnotes can contain labels and .GO [Goto] control words for iterative processing, can invoke macros, and can contain GML tags. The label must be contained within the footnote.
11. The .FN [Footnote], .Hn [Heading Level n],<sup>7</sup> .RH [Running Heading], or .RF [Running Footing] control words end the definition phase of a footnote leader. If you invoke a macro whose name begins with the characters “fn,” “hn,” “rh,” or “rf” and have not specified the .EM [Execute Macro] control word, the definition phase is ended.
12. If a skip occurs at the top of the first footnote on a page, the depth of that skip is set to zero.
13. Certain control words are not allowed within a footnote. If one of the disallowed control words is encountered, the footnote is immediately ended, as though .FN OFF had been processed, and then the disallowed control word is executed. A warning message is issued, identifying the control word that ended the footnote. See Table 15 on page 442 for a list of the disallowed control words.

---

<sup>7</sup> The .Hn [Heading Level n] control word is an obsolete control word. See Appendix A, “Unsupported Control Words” on page 461 for a description of the .HN control word.

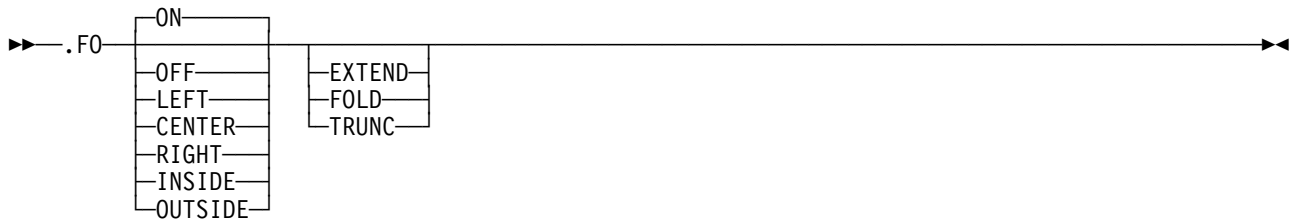
---

## **.FO [Format Mode]**

### **Function**

Use the .FO [Format Mode] control word to cancel or restore concatenation of input lines and justification of output lines. The .FO control word also controls whether lines can be extended beyond the column boundary.

### **Syntax**



### **Parameters**

- ON** Causes both concatenation of input lines and justification of output lines. Each output line is filled with text and aligned on both the left and right margins.
- OFF** Cancels both concatenation of input lines and justification of output lines. Subsequent text is printed as entered.
- LEFT** Specifies that input lines are to be concatenated but not justified. The resulting output lines are left-aligned in the column. This format is sometimes called “ragged right.”
- CENTER** Specifies that input lines are to be concatenated but not justified. The resulting output lines are centered in the column.
- RIGHT** Specifies that input lines are to be concatenated but not justified. The resulting output lines are right-aligned in the column. This format is sometimes called “ragged left.”
- INSIDE** Specifies that input lines are to be concatenated but not justified; the resulting output lines are aligned against the inside margin of the column—toward the presumed binding edge of the duplexed page. This is equivalent to .FO LEFT for odd pages and .FO RIGHT for even pages.
- OUTSIDE** Specifies that input lines are to be concatenated but not justified; the resulting output lines are aligned against the outside margin of the column—away from the presumed binding edge of the duplexed page. This is equivalent to .FO RIGHT for odd pages and .FO LEFT for even pages.

When concatenation is in effect and a single word is wider than the column, or when concatenation is not in effect and an input line is wider than the column, the text is processed according to the current overdraw option:

- EXTEND** Specifies that if the text does not fit in the column, it is extended beyond the column line length. This is the initial setting.
- FOLD** Specifies that if the text does not fit in the column, it is broken and the remainder is to be placed on the next output line. The text is broken at the last character that fits in the column.
- TRUNC** Specifies that if the text does not fit in the column, it is truncated at the last character that fits in the column.



**Initial Setting:** ON EXTEND

**Default:** ON

## Notes

- .FO causes a break.
- The formatting mode is included in the active environment.

## Remarks

1. The .FO control word determines whether words are shifted from one input line to another to fill each output line (concatenation), and whether any extra horizontal white space left at the end of the output line is distributed between the words on the line (justification).
2. The overdraw options TRUNC, FOLD, and EXTEND may be specified as the only options of the .FO control word. In this case, the current formatting mode is unchanged, although a break is done.
3. The .NF control word is similar to .FO; both control words affect concatenation and justification, and both control words can set the overdraw option.
4. Text that exceeds the column line length is processed according to the current overdraw option, as established with a .FO or with the .NF [No Formatting] control word. When concatenation is in effect, this can happen when a single word is wider than the column line length or when a word following a tab crosses the right margin. When concatenation is not in effect, this can happen when an input line is wider than the column line length.
5. Options can be specified in any sequence. If contradictory options are specified, the latest one is used.
6. When you use the EXTEND overdraw option, you must ensure that the width of extended output lines does not exceed the width of the page.

## Examples

- .fo off

```
The sun was shining on the sea,
Shining with all his might:
He did his very best to make
The billows smooth and bright--
And this was odd, because it was
The middle of the night.
```

- .fo on

```
The moon was shining sulkily, Because she thought the sun Had got no business to be there
After the day was done-- "It's very rude of him," she said, "To come and spoil the fun!"
```

## .FO [Format Mode]

- .fo left

The sea was wet as wet could be,  
The sands were dry as dry. You  
could not see a cloud because No  
cloud was in the sky: No birds  
were flying overhead-- There were  
no birds to fly.

**Note:** A column line length of 16p was specified for this example.

- .fo center

The Walrus and the Carpenter Were walking close at hand: They wept like anything to see Such  
quantities of sand: "If this were only cleared away," They said, "it would be grand!"

**Note:** An indention of -2 was specified for this example.

- .fo right

"If seven maids with seven mops  
Swept it for half a year, Do you  
suppose," the Walrus said, "That  
they could get it clear?" "I doubt  
it," said the Carpenter, And shed  
a bitter tear.

**Note:** A column line length of 16p was specified for this example.

- .fo outside

"O Oysters come and walk with us!"  
The Walrus did beseech.  
"A pleasant walk, a pleasant talk,  
Along the briny beach:  
We cannot do with more than four,  
To give a hand to each."

- .fo inside

The eldest Oyster looked at him,  
But never a word he said:  
The eldest Oyster winked his eye,  
And shook his heavy head--  
Meaning to say he did not choose  
To leave the oyster-bed.

- This poem is continued under ".NF [No Formatting]" on page 249.

---

## **.FV [Format Vertically]**

### **Function**

Use the .FV [Format Vertically] control word to indicate how formatted lines of text should be placed within the columns of the section.

### **Syntax**



### **Parameters**

- TOP** Specifies that text should be placed at the top of the columns, just below any top column float.
- CENTER** Specifies that text should be centered in the columns between any column floats.
- BOTTOM** Specifies that text should be placed at the bottom of the column, just above any bottom column float.
- JUSTIFY** Specifies that the text in all columns of a section should extend to the same depth, if possible. This is achieved by increasing or decreasing the amount of vertical white space in each column within the given limits. If necessary, the spacing between lines of text may be increased or decreased, within the limits given by the .LS [Line Spacing] control word.

**Initial Setting:** TOP

**Default:** TOP

### **Notes**

- .FV ends a keep, float, footnote, named area, or table.
- .FV takes effect at a section break.
- The vertical formatting mode is included in the active environment.

### **Remarks**

1. Starting at the top, columns are filled with lines of text and vertical white space until the column is explicitly ended or until the next line does not fit in the room remaining in the column. Normally, any extra room in the column is left at the bottom. If you specify .FV BOTTOM, the text in the column is moved to the bottom, leaving the extra room at the top.
2. .FV CENTER centers the text in the column.
3. Floats, footnotes, floating keeps and *named* areas are always top aligned.
4. Even if .FV JUSTIFY has been specified, the columns may not all be the same length if some columns have more vertical white space than can be redistributed within the limits set by the .LS [Line Spacing] control word.
5. If a page body consists of more than one section, only the last section is justified toward the bottom of the page.

## **.FV [Format Vertically]**

6. Vertical formatting formats the text only within a section. If a section is ended unconditionally, the bottom of the section is at the bottom of the text. The section, therefore, is as deep as the text, and the text appears the same before and after vertical formatting. For more details on section endings, refer to the *Document Composition Facility: SCRIPT/VS User's Guide*.

## **Examples**

- To cause text to be placed at the bottoms of columns, instead of at the tops, enter  
    .fv bottom
- For more examples on how to control justification values and increments, see “.LS [Line Spacing]” on page 235.

---

## **.GO [Goto]**

### **Function**

The **.GO [Goto]** control word causes SCRIPT/VS to branch to another part of the current SCRIPT/VS input file or macro.

### **Syntax**

►► **.GO** TO *label* —————►►

### **Parameters**

**TO** An optional parameter, which is ignored if present; its only purpose is to allow the alternative forms **“.GOTO”** and **“.GO TO.”** However, the short form **(.GO)** is recommended for performance reasons.

*label* The name of a line identified elsewhere in the current file or macro with the **.... [Set Label]** control word.

### **Remarks**

1. Use the **.GO** control word to branch to another place in a **SCRIPT** file or macro. If the label designated on the **.GO** control word is not defined elsewhere in the current file or macro, an error message is issued and processing terminates.
2. The input line preceding the **.GO** control and the line at the label designated in the **.GO** control word are processed as though they were two sequential lines from the **SCRIPT** file.
3. Every **.GO** control word must refer to a label defined with the **... [Set Label]** control word. More than one **.GO** control word can refer to the same label.
4. **.GO** is particularly useful when performed conditionally as the object of an **.IF** statement.
5. The use of labels and the **.GO [Goto]** control word for conditional looping within an imbed file may require an **.EF CLOSE [End of File]** control word at the end of the imbed file in order to explicitly close it.

### **Performance Considerations:**

The **.GO** control word can be relatively inefficient when used in files. When processing includes extensive iteration, a considerable performance improvement can be achieved by moving loops into macros.

For example, suppose you have created a large, sparse array. The following sequence of control words types the index and values of the elements that actually exist:

```
.se i = 1
...loop
.if &E'&array(&i) eq 1
.th .ty Element &i is &array(&i)
.se i = &i + 1
.if &i le 1000 .go loop
```

## **.GO [Goto]**

The performance of this sequence can be greatly improved by defining the loop as a macro and then executing the macro:

```
.dm sparse on
.se i = 1
...loop
.if &E'&array(&i) eq 1
.th .ty Element &i is &array(&i)
.se i = &i + 1
.if &i le 1000 .go loop
.dm off
.ms on
.sparse
```

See “... [Set Label]” on page 60 for a discussion of the best placement of labels when .GO must be used within a file.

## **Examples**

Suppose you had a SCRIPT file that was designed to recognize the variable SYSVAR5. In this example, if SYSVAR5 is set to SMALL, you want SCRIPT/VS to format the output at 36 lines per page and 4.2 inches per line. Otherwise, the default values are to be used. This could be done with the following control words:

```
.if /&SYSVAR5 ne /SMALL .go default
.pl 36
.ll 4.2i
...default
(etc.)
```

## .GR [Group]

### Function

Use the .GR [Group] control word to specify the beginning and end of a group of pages to be used for navigating through a document with the IBM AFP Workbench for OS/2 and Windows.

### Syntax

►► .GR *groupname* ON

— .GR OFF

### Parameters

*groupname* The *groupname* of this group. The *groupname* can be up to 64 characters. If the name includes blanks, single quotes, or both, the whole *groupname* **must** be enclosed in single quotes. To use a single quote in the *groupname*, enter it as two consecutive single quotes.

**ON** Starts a group, including the current page as its first page.

**OFF** Ends the current group, including the current page as its last page.

### Remarks

1. .GR is valid for only AFPDS devices (physical devices 3820, 4028, and AFP). It is ignored for all other physical devices (including the 3800-3).
2. The characters used in *groupname* are displayed in AFP Workbench for OS/2 and Windows using the codepage appropriate for the default language of SCRIPT/VS. See the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for information on changing the default language.
3. After defining the start of a group, use the .NV GROUP control word to define group-level navigation elements and instances of those navigation elements within the group. Navigation through groups, based on navigation elements, is done with the **Find Group** function of AFP Workbench.
4. .GR is not allowed inside a keep, float, footnote, table, running heading or running footing.
5. OFF is not allowed as a group name. If OFF is specified as a group name, it is interpreted as the OFF parameter, and the most recently started group is ended, or the control word is ignored, if there is no active group.
6. The page which contains a .GR NAME ON control word is included as the first page in the group. The page which contains a .GR OFF control word is included as the last page in that group. Ensure that the page is started prior to using the .GR control word, to ensure the group starts and ends on the correct pages. For example:  

```
.pa
.gr name on
.
.
.
.gr off
.pa
```
7. You can create more than one group with the same name, but AFP Workbench for OS/2 and Windows only allows you to navigate to the first one. If you process the SCRIPT/VS output through

## **.GR [Group]**

- |     Advanced Function Presentation Conversion and Indexing Facility (ACIF), unique group names are
- |     created, such that all are navigable through AFP Workbench.
- |     8. Only one group can be active at a time. Any attempt to start a second group before the first is ended
- |     results in an error message.

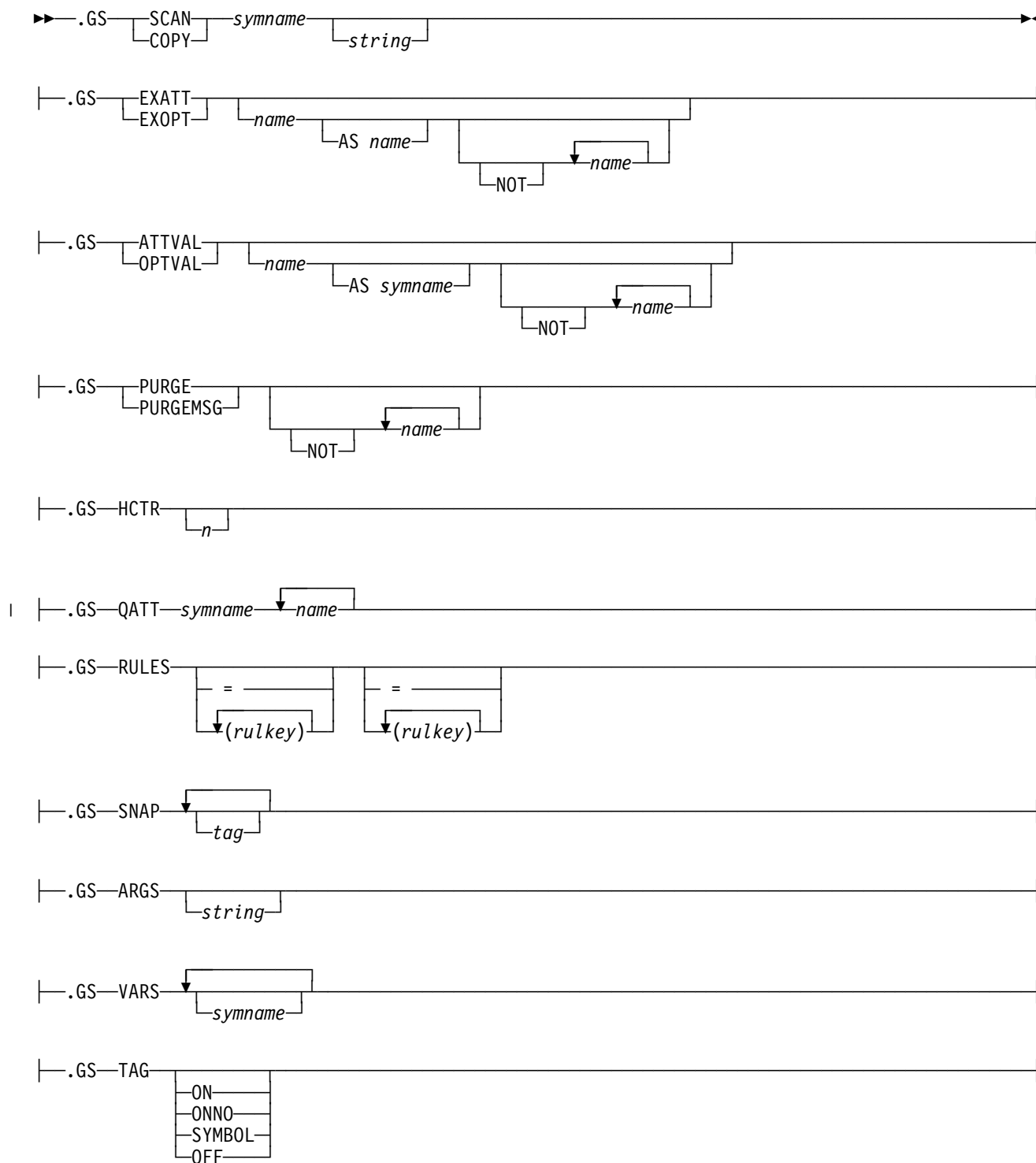


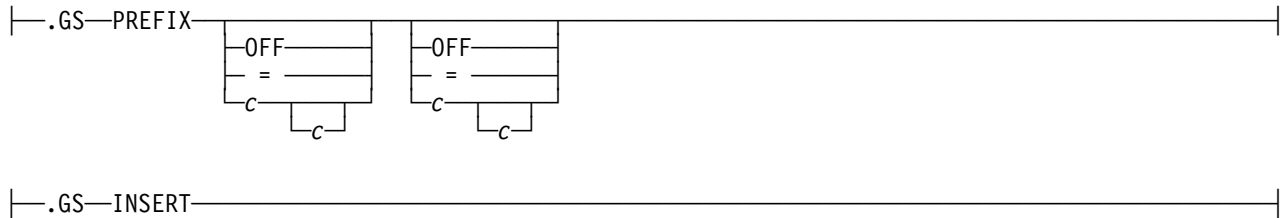
## .GS [GML Services]

### Function

Use the .GS [GML Services] control word to perform various services that may be required when writing GML APFs.

### Syntax





## Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SCAN</b>    | <p>Specifies that the string is to be scanned for GML attributes and residual text. Any attributes found are added to the current attribute stack; any residual text becomes the value of the symbol &amp;symname.</p> <p>If no string is given, any residual text following the current GML tag is removed from the document and becomes the value of the symbol &amp;symname.</p> <p>The scan is performed according to the current rules for start tags. See the following description of the various scanning rules.</p> |
| <b>COPY</b>    | <p>Identical to the SCAN parameter, except that if no string is given, the residual text following the current GML tag is not removed from the document: only a copy of the residual text is made in &amp;symname.</p>                                                                                                                                                                                                                                                                                                       |
| <i>symname</i> | <p>Name of a symbol to contain the residual text to the right of the markup/content separator. If the NOATT or STOP rule for scanning is in effect, the residual line can be found without any markup/content separator. See the following description of the scanning rules.</p>                                                                                                                                                                                                                                            |
| <i>string</i>  | <p><i>String</i> to be scanned for attributes and residual text. Any attributes found are added to the attribute stack; any residual text becomes the value of &amp;symname.</p>                                                                                                                                                                                                                                                                                                                                             |
| <b>EXATT</b>   | <p>Specifies that any attributes that have been found and placed in the attribute stack are to be executed. If a list of names is given, then only the attributes listed are executed. If the list of names is preceded by the parameter NOT, then all attributes <i>except</i> those listed are executed. After an attribute has been executed, it is deleted from the attribute stack.</p>                                                                                                                                 |
| <b>EXOPT</b>   | <p>Specifies that any user-defined options specified on the SCRIPT command are to be executed. If a list of names is given, then only the options listed are executed. If the list of names is preceded by the parameter NOT, then all options <i>except</i> those listed are executed. After an option has been processed, it is deleted from the attribute stack. User-defined options are only available in the CMS environment.</p>                                                                                      |
| <b>ATTVAL</b>  | <p>Retrieves the value of an attribute and puts the value into a symbol. When you specify ATTVAL with no names, all attributes that have been found and placed in the attribute stack are to be processed. If a list of names is given, only the attributes listed are processed. If the list of attributes is preceded by the parameter NOT, then all attributes <i>except</i> those listed are processed. After an attribute has been processed, it is deleted from the attribute stack.</p>                               |
| <b>OPTVAL</b>  | <p>Retrieves the value of a user-defined option specified on the SCRIPT command and puts the value into a symbol. If a list of names is given, then only the options listed are processed. If the list of names is processed by the parameter NOT, then all options <i>except</i> those listed are processed. After an option has been processed, it is deleted from the attribute stack. User-defined options are only available in the CMS environment.</p>                                                                |

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>name</i>     | Specifies the name of the attribute or option to be processed. The maximum number of characters in an attribute name is 8. Only alphabetic, numeric, @, #, and \$ characters can be used in the names.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>AS</b>       | With EXATT or EXOPT, specifies explicitly the APF to be executed for a given attribute.<br><br>With ATTVAL or OPTVAL, specifies explicitly the name of the symbol to receive the value of the attribute or option.<br><br>If AS is not specified, the name of the macro or symbol is the same as the name of the attribute or option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>symname</i>  | Specifies that any attributes or options that are retrieved are set in this symbol. The symbol name cannot be a symbol array element.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>NOT</b>      | Specifies that the list of names that follows is not to be acted upon.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>PURGE</b>    | Specifies that any attributes that have been found and placed in the attribute stack are to be deleted. If a list of names is given, then only the attributes listed are purged. If the list of names is preceded by the parameter NOT, then all attributes <i>except</i> those listed are purged.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>PURGEMSG</b> | Same as PURGE, except that a message is issued to notify you what attributes were purged from the stack. This is useful if you want to be informed of unused attributes for a particular tag.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>HCTR</b>     | Specifies that the decimal numbering contained in the symbol &@xref is to be either set with the value specified, or is to be incremented. If <b>n</b> is omitted, the symbol &@xref is set to a null value, and numbering is reinitialized.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <i>n</i>        | Specifies whether the head counter is to be incremented at the level given or is to be set with the value given. <i>n</i> can also be used to indicate whether numbering is to be in Arabic or alphabetic form. If <b>n</b> is a decimal number, the counter at that level is incremented, and the symbol &@xref contains the counter value for that level. If <b>n</b> is of the form n1.n2.n3...(up to 32 levels of numbering are supported), the heading counter is reset with the value given. If less than 32 numbers are given, those given are assumed to relate to the <i>leftmost</i> positions. For example, if the value 1.3.2 is supplied, &@xref is set to '1.3.2'.<br><br>A value of zero reinitializes the head counter to 1 for the first level and zero for all other levels. A value of zero in any specific position resets the values of all subsequent positions. If a single digit is specified, the corresponding head counter is incremented, and all sublevels are reset to zero. The first request to increment a head counter after the counter has been explicitly set with .GS HCTR is ignored. |
| <b>QATT</b>     | Specifies that the attribute stack is to be checked for the presence of specific attributes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <i>symname</i>  | Specifies that any attributes that are checked and found to be absent from the attribute stack are to be set in this symbol as array elements.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <i>name</i>     | Specifies an attribute name to be checked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>RULES</b>    | Specifies up to four rules to be used for GML attribute scanning. You can specify two lists of rule parameters, each enclosed in parentheses. The first list sets the rules for scanning start tags, and the second list sets the rules for scanning end tags. The start-tag rules are used for scanning using .GS SCAN. The rules set by .GS RULES are used for scanning any tag that does not have its own rules set using .AA [Associate APF].<br><br>The recognized rule parameters are:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

- ATT** Specifies that attributes are allowed. A regular attribute is in the form “name=value”, where the name is a maximum of 8 characters, and the first character is not a number. The other characters can be alphabetic, numeric, and the characters @, #, and \$. The value can be in either of two forms: a single word consisting entirely of the same restricted character set that is allowed for names, with no embedded blanks, or a string delimited with single quotation marks.
- Any character is allowed within single quotation marks. (A single quotation mark that is part of the string must be expressed as two single quotation marks.) Leading and trailing blanks are ignored.
- In the line being scanned, any item that does not conform to this description of a valid attribute is dealt with according to the VAT, STOP, and MSG rules that are currently in effect.
- All regular attributes found during the scan are placed in the attribute stack, and they are available to the APF using .GS EXATT and .GS QATT.
- NOATT** Specifies that no regular attributes are allowed on this tag. If NOATT is in effect, the other rules are immaterial. No scan is done for a tag that has the NOATT rule in effect, but what follows the tag in the input is treated as text (unless it is another tag, of course).
- VAT** Specifies that “value attributes” are allowed for this tag. A value attribute is a single word composed of the same restricted character set that is allowed for attribute names, with no “name=” before it. Quoted strings are not allowed as value attributes. All value attributes found during the scan are placed on the APF invocation line and are available to the APF in the macro local symbol &\*.
- NOVAT** Specifies that no value attributes are allowed for this tag. In this case, a word that would, with the VAT rule in effect, be recognized as a value attribute is considered an invalid attribute.
- STOP** Specifies that when an invalid attribute is found during the scan, the scan is stopped at that point, and the invalid attribute, and everything to the right of it, are treated as text.
- NOSTOP** Specifies that an invalid attribute does not stop the scan. The invalid attribute is skipped and the scan continues.
- MSG** Specifies that when an invalid attribute is found, a message is to be issued. If the STOP rule is in effect, then the message shows the beginning of the string that was not an attribute, and is treated as text. If the NOSTOP rule is in effect, the message shows the entire invalid attribute.
- NOMSG** Prevents a message from being issued when an invalid attribute is found. The scan stops or continues, according to the STOP or NOSTOP rule, with no message.

An equal sign in place of a rules list means to leave that set of rules unchanged. An empty list, that is, left and right parentheses with nothing in between, means to restore the default rules.

The default rules for start tags are ATT, NOVAT, STOP, and NOMSG. The default rule for end tags is NOATT.

- SNAP** Displays the current rules for start- and end-tag scanning. If a list of tag names is given, .GS SNAP also displays the .AA [Associate APF] association in effect for each tag in the list, along with the scanning rules for the tag.

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|------------------------------------------------|-------------|------------------------------------------------------------------------------|---------------|-----------------------------------------------------------------------------------------------------------------|------------|------------------------|
| <b>ARGS</b>   | Resets the macro local symbols using the string that follows the ARGS parameter. For example, if a macro issued the control word<br><br><pre>.gs args one two three</pre> then the symbol <code>&amp;*</code> is set to the value “one two three.” The symbol <code>&amp;*0</code> is set to the value “3,” <code>&amp;*1</code> would contain “one,” <code>&amp;*2</code> would contain “two,” and <code>&amp;*3</code> would contain “three.”<br><br>The control word “.GS ARGS,” with no string, resets <code>&amp;*0</code> to 0, and all the others to null. |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>VARs</b>   | Assigns the values of the current macro parameters, <code>&amp;*1</code> , <code>&amp;*2</code> , ... , to the specified symbols. Array symbols are not allowed.                                                                                                                                                                                                                                                                                                                                                                                                  |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>TAG</b>    | Specifies the GML tag-parsing technique to be used. <table> <tr> <td><b>ON</b></td><td>Enables GML scanning.</td></tr> <tr> <td><b>ONNO</b></td><td>Enables scanning and suppresses warning messages concerning unresolved tags.</td></tr> <tr> <td><b>SYMBOL</b></td><td>Enables simulated GML scanning using symbols rather than tags.<sup>8</sup></td></tr> <tr> <td><b>OFF</b></td><td>Disables GML scanning.</td></tr> </table>                                                                                                                              | <b>ON</b>  | Enables GML scanning.                          | <b>ONNO</b> | Enables scanning and suppresses warning messages concerning unresolved tags. | <b>SYMBOL</b> | Enables simulated GML scanning using symbols rather than tags. <sup>8</sup>                                     | <b>OFF</b> | Disables GML scanning. |
| <b>ON</b>     | Enables GML scanning.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>ONNO</b>   | Enables scanning and suppresses warning messages concerning unresolved tags.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>SYMBOL</b> | Enables simulated GML scanning using symbols rather than tags. <sup>8</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>OFF</b>    | Disables GML scanning.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>PREFIX</b> | Specifies a class mapping of GML tags to APFs. The mapping may be specified separately for start tags (first parameter) and end tags (second parameter). <table> <tr> <td><b>OFF</b></td><td>Disables automatic mapping of tag name to APF.</td></tr> <tr> <td><b>=</b></td><td>APF mapping option is unchanged.</td></tr> <tr> <td><b>'c[c]'</b></td><td>Specifies one or two characters that are to be used as a prefix to the tag name to create the mapping APF name.</td></tr> </table>                                                                      | <b>OFF</b> | Disables automatic mapping of tag name to APF. | <b>=</b>    | APF mapping option is unchanged.                                             | <b>'c[c]'</b> | Specifies one or two characters that are to be used as a prefix to the tag name to create the mapping APF name. |            |                        |
| <b>OFF</b>    | Disables automatic mapping of tag name to APF.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>=</b>      | APF mapping option is unchanged.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>'c[c]'</b> | Specifies one or two characters that are to be used as a prefix to the tag name to create the mapping APF name.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |
| <b>INSERT</b> | Intended for use by an APF that needs to insert some new text into the output. The given line, which can include imbedded blanks, is placed in the output before any residual text line that might exist for this tag, using appropriate continuation. If the residual text associated with the tag does not immediately follow the tag, continuation might not be done correctly. Also, the use of this function from a file or a macro might not result in correct continuation.                                                                                |            |                                                |             |                                                                              |               |                                                                                                                 |            |                        |

## Notes

- The setting of the TAG parameter is included in the page environment.

## Remarks

1. When a GML tag is found, as many lines as necessary are read to obtain all of the associated attributes plus the residual text. Scanning is ended if one of the following is found:

- The residual text
- Another GML tag
- A control word line.

After scanning has been completed, the APF for the GML tag is executed.

2. The residual text following a GML tag is treated as literal text; no special processing is performed for leading blanks or tabs.

<sup>8</sup> This is provided for compatibility with SCRIPT/VS Release 1 only.

## Examples

- Suppose a GML tag with the following attributes is entered:

```
:critter type=sloth toes=3
  name='Warren Jr.' food=leaves
  family=Bradypodidae food=fruit.
```

Within the APF which processes the :CRITTER tag, the .GS EXATT function may be used to selectively process the attributes. For example,

```
.gs exatt type
```

results in the execution of the TYPE macro. The value of the attribute is provided to the macro as its parameters.

The name of the macro to be executed can be given explicitly when it is not the same as the attribute name. For example,

```
.gs exatt food as diet
```

results in the execution of the DIET macro twice; once with the parameter “leaves,” and again with the parameter “fruit.”

Once an attribute is processed, it is no longer available. For example, if the two preceding .GS control words are followed by

```
.gs exatt
```

the following macros are executed:

```
.TOES 3
.NAME Warren Jr.
.FAMILY Bradypodidae
```

- Suppose the following GML tag is entered:

```
:tagname id='nora' place='raleigh'.
```

The APF for ‘tagname’ could contain

```
.gs attval id place
```

which sets the symbol &ID to “nora” specified in the ID attribute, and it sets the symbol &PLACE to “raleigh.” Note that the symbol name is always uppercase.

If you want to set a different symbol, use the AS parameter:

```
.gs attval id as *xyz
```

This sets the macro local symbol &\*xyz to the value of the ID attribute (in this case, “nora”). If you specify a symbol name with the AS parameter, it is not folded to uppercase; the name is used exactly as you specified it.

- The symbol name on ATTVAL and OPTVAL are folded to uppercase unless the AS parameter and a specific symbol name are given. In that case, the symbol name is used exactly as given on the control word.
- The head-level counter is initially set to “1.0.0.0.0.....” The .GS HCTR control word can be used to change the head-level counter at any time. For example,

```
.gs hctr 4.7.7.4
```

sets the head-level counter to “4.7.7.4.0.0.....” If the next head-level control word is .H4, it is numbered “4.7.7.4”; if it is .H5, it is numbered “4.7.7.4.1”; if it is .H3, it is numbered “4.7.7.”

When only a single number is given with .GS HCTR, the corresponding head-counter level is incremented, and all sublevels are reset. For example,

```
.gs hctr 2
```

sets the head-level counter to “4.8.0.0....”

- The .GS ARGS and .GS VARS control words provide a convenient means of setting a number of symbols simultaneously when the current macro parameters are not needed. For example,

```
.gs args 1 7 7 6
.gs vars x y z t
```

is equivalent to

```
.se x = 1
.se y = 7
.se z = 7
.se t = 6
```

- Value attributes are presented to the APF that processes a tag in the same way attribute values are presented to the macro that processes an attribute. For example, if a tag is entered as

```
:figure big.
```

the macro local symbol &\* of the macro that processes the :FIGURE tag is set to “big.” If the tag is entered as

```
:figure size=big.
```

the macro local symbol &\* of the macro that processes the SIZE attribute is set to “big.”

- Use .GS INSERT to insert text from an APF:

```
.gs insert Figure 7
```

The text, Figure 7, is inserted prior to processing the residual text.

- Use .GS SCAN to obtain the residual text in a symbol for purposes of processing it. For example,

```
.gs scan textline
```

puts the residual text into the &textline symbol for later use. The APF must now process this text. SCRIPT/VS does not automatically process this text.

- Use .GS COPY to obtain the residual text in a symbol for purposes of examining it. For example,

```
.gs copy textline
```

puts the residual text into the &textline symbol. SCRIPT/VS still processes the residual text automatically.

---

## **.H0–.H6 [Head Levels 0–6]**

### **Function**

The control words .H0 through .H6 automatically format topic headings in SCRIPT/VS output. The definition of a particular head level can also result in an entry in the table of contents for that heading. The definition of a head level can be changed with the .DH [Define Head Level] control word, or a macro can be defined to perform whatever function you want for .H0 through .H6, using the .DM [Define Macro] control word.

### **Syntax**

►►—.Hn—text—————►◄

### **Parameters**

*n*        The number of the head level from 0 to 6.

*text*     The data to be formatted as a subject head and optionally placed in the table of contents.

### **Notes**

- Head levels H0–H1 start the page, even if they do not appear in the body of the text.
- Head levels H0–H6 cause a break.

### **Remarks**

1. The .Hn control words provide several automated functions. They can provide a topic heading that is underscored or capitalized with a specified number of skips or spaces before it and conditional line spaces after it. They can cause the unformatted topic head to be saved, along with the current page number and revision code character, in the table of contents utility file for automatic table of contents generation. They can also cause the heading to be numbered with a decimal number that reflects the level of the heading. These functions can be redefined using the .DH [Define Head Level] control word.

Whether you use the default values or redefine them, the topic head that is generated gives you the function of a keep for the heading line and the size of the space after the heading plus two additional lines. This is true only for headings that do not cause page or section breaks. This keep is of the form “.KP v + v.” See the discussion of the .KP [Keep] control word for information about which forms of keep can cancel or supersede this form.

2. If a head level control word calls for an entry in the table of contents, the text goes into the table of contents as entered. You control how the table of contents entry is capitalized by how you enter the associated head level control word text or by using an uppercase font for the table of contents.
3. See Table 8 on page 138 for information about the default head level definitions.
4. If you want to define a head level, such as .H3, to include function not within the scope of the .DH [Define Head Level] control word, you can:
  - a. Define a .H3 macro that would provide all the function you wanted for .H3. Your macro would then be processed whenever .H3 was encountered in the input file, assuming macro substitution was ON.
  - b. Provide a .DH macro that creates and maintains head level macros and then augment the function of that existing .H3 by adding or deleting lines from the macro that provides the function for .H3.



In either case, see the description of the `.DM` [Define Macro] control word for information about defining macros.

5. The head level text is limited to 243 characters.
6. Do not enter a footnote immediately following a heading. This may cause the heading inline keep to be prematurely ended.
7. If the heading occupies two or more output lines, it may be formatted in long indent style if an offset is not already in effect. Refer to the description of the `.DH` [Define Head Level] control word for details.
8. When headings occur without intervening text, and the heading levels for those headings are not defined to cause a page or section break, those headings are kept together on the same page due to the keep function described in Remark 1. The exception is when a higher level heading follows a lower level heading. For example, if several level 4 headings with no intervening text are followed by a level 3 heading, all the level 4 headings are kept together on the same page, but the level 3 heading is not placed in the keep with the previous level 4 headings, and is not forced to stay on the same page as the previous level 4 headings.

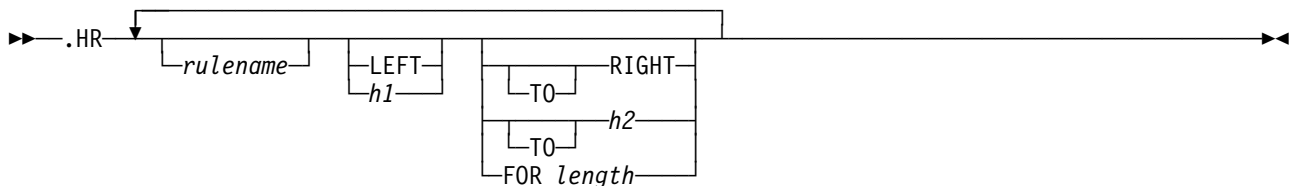
---

## .HR [Horizontal Rule]

### Function

Use the .HR [Horizontal Rule] control word to place one or more horizontal rule segments in the current column.

### Syntax



### Parameters

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>rulename</i> | Any <i>named</i> rule defined with the .DR [Define Rule] control word. It can contain a maximum of 16 national characters. If no name is given, the horizontal segment is drawn with the same rule as was used for the previous segment of this .HR control word. If no previous rulename was specified, the default rule is used. If you specify rulename you must also specify at least one set of horizontal displacements. |
| <i>h1</i>       | Horizontal displacement of the left end of the rule segment, specified in any valid space unit.                                                                                                                                                                                                                                                                                                                                |
| <b>LEFT</b>     | Specifies that the left end of the rule is to be flush left in the column.                                                                                                                                                                                                                                                                                                                                                     |
| <b>TO</b>       | An optional parameter that indicates that the next parameter is the horizontal displacement of the right end of the horizontal rule segment.                                                                                                                                                                                                                                                                                   |
| <i>h2</i>       | Horizontal displacement of the right end of the rule segment, specified in any valid space unit.                                                                                                                                                                                                                                                                                                                               |
| <b>RIGHT</b>    | Specifies that the right end of the rule is to be flush right in the column.                                                                                                                                                                                                                                                                                                                                                   |
| <b>FOR</b>      | Indicates that the next parameter is the length of the rule segment. Length can be expressed in any valid space unit.                                                                                                                                                                                                                                                                                                          |

### Notes

- .HR causes a break.
- .HR ensures that the page is started.

### Remarks

1. Several separate horizontal rule segments can be drawn with a single .HR control word. For each segment, you can specify three things:
  - The *name* of the rule to be used in drawing the segment
  - Where the segment starts (the left end of the rule)
  - Where the segment ends (the right end of the rule), given either as the length of the segment or as the horizontal displacement of the right end of the rule.
2. All horizontal positions on a single .HR control word must be sequentially increasing. In other words, you must specify all rules from left to right.

3. If *rulename* is not specified, the rule is drawn with a default rule appropriate for the logical device:
  - For line devices, the rule is constructed using the box character set of the current font.
  - For page printers, the rule is drawn with the default rule, as described under “.DR [Define Rule]” on page 151.
4. On page printers, the depth of the lines containing horizontal rules created by .HR depends upon the linespacing currently in effect:
  - If fixed linespacing has been specified with .LS NORMAL, lines containing horizontal rules have the specified depth, even if this is less than the thickness of the rule.
  - If fixed linespacing is not in effect, lines containing horizontal rules have a depth equal to the thickness of the thickest rule plus the difference between the current line spacing value and the capital height in the current font. This preserves a uniform “leading” between text lines and horizontal rule lines.
5. To create a horizontal rule that does not contain the extra vertical space which is explained above, use the NORMAL parameter of the .LS [Line Spacing] control word to set the linespacing to the thickness of the rule being used on the .HR control word. Restore the previous line spacing after the .HR control word.
6. If .HR is coded with no parameters at all, the previous horizontal rule is repeated.
7. If the rulename is a valid space unit, it is recognized as displacement and not as a rulename.
8. The output from DCF when formatting for a PostScript or 4028 physical device is a resolution-independent data stream. Therefore, when you format for one of these devices, there may be one pel rounding errors at the printer that are evident in rule widths and rule intersections. Because of the variations in printer resolutions, the same SCRIPT/VS output file might produce different results on different devices.

## Examples

- The .HR control word can be used to draw a simple rule the width of the column:

```
.hr left right
```



- Several horizontal rules can be drawn with a single .HR control word:

```
.hr 5mm for 8mm 22mm to 39mm
```



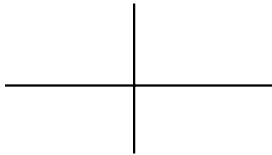
- If you define *named* rules with the .DR [Define Rule] control word, for a page printer, you can use them with .HR:

```
.dr thin weight .4mm  
.dr thick weight .6mm  
.hr thin 5mm for 8mm thick 22mm to 39mm
```



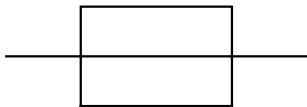
- Horizontal rules can be combined with vertical rules:

```
.vr 15  
.sp 2  
.hr 5 for 20  
.sp 2  
.vr off
```



- Horizontal rules can intersect with boxes drawn with the .BX [Box] control word.:

```
.bx 3cm 5cm  
.sp  
.hr 2cm to 6cm  
.sp  
.bx off
```



---

## .HW [Hyphenate Word]

### Function

Use the .HW [Hyphenate Word] control word to specify how a single occurrence of a word should be hyphenated if needed.

### Syntax

►►—.HW—*text-word*—————►◄

### Parameters

*text-word*    The word that you want to hyphenate. It should be entered with hyphens showing where you want it broken. The word is printed in your document.

### Notes

- .HW ensures that the page is started.

### Remarks

1. The .HW control word is a separate function from the hyphenation facility; it works regardless of whether hyphenation is ON or OFF (via the .HW [Hyphenate] control word).
2. Whether the hyphenation point specified with the .HW control word is used depends on the settings of the MINPT, MAXPT, MINWORD, and LADDER parameters of the .HY control word.
3. The .HW control word does not define how a word should be hyphenated each time it is encountered. It specifies how to handle that word only for this particular instance. If you want a word hyphenated each time it occurs (if hyphenation is in use), then use the .DU control word to define hyphenation points for the word in the dictionary.
4. If, while SCRIPT/VS is formatting the line, it is not necessary to break the word, the hyphens are compressed out, and they do not appear in the output. Use two hyphens to indicate a hyphen that should remain in a compound-word

```
.cl 11m
.hy minpt 3
This is a
.hw com-pound--word
that can be broken in
either of two places.
```

and the word is printed in your text as follows:

```
This is a compound-
word that can be broken
in either of two places.
```

Use the .HW [Hyphenate] control word to control automatic hyphenation and word space adjustment ranges.

```

graph LR
    subgraph Line1 [Line 1]
        direction LR
        L1_1[.HY] --- L1_2[ON]
        L1_2 --- L1_3[OFF]
        L1_3 --- L1_4[SUP]
        L1_4 --- L1_5[ADD]
        L1_5 --- L1_6[NOADD]
        L1_6 --- L1_7[DICT]
        L1_7 --- L1_8[NODICT]
        L1_8 --- L1_9[ALG]
        L1_9 --- L1_10[NOALG]
        L1_10 --- L1_11[ALT]
        L1_11 --- L1_12[NOALT]
        L1_12 --- L1_13[LADDER n]
        L1_13 --- L1_14[MAXPT n]
    end
    subgraph Line2 [Line 2]
        direction LR
        L2_1[MINPT n] --- L2_2[MINWORD n]
        L2_2 --- L2_3[RANGE]
        L2_3 --- L2_4[range]
    end

```

|           |                                                                                                                                                                                                                                                                                |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ON</b> | Begins automatic hyphenation of SCRIPT/VS output lines. Addenda dictionaries created with .DU [Dictionary Update] are searched when a word is to be hyphenated. If the word is not found, the dictionaries specified with the .DL [Dictionary List] control word are searched. |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**OFF** Causes hyphenation to be turned off.

**ADD** Specifies that addenda dictionaries created with the .DU control word are also to be searched for words to be hyphenated.

**NOADD** Specifies that addenda dictionaries are not to be searched.

|             |                                                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DICT</b> | Specifies that the dictionaries specified with the .DL [Dictionary List] control word are also to be searched for words to be hyphenated. |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------|

**NODICT** Specifies that the dictionaries specified with the .DL control word are not to be searched.

**ALG** Specifies that an algorithmic hyphenation routine is also to be used. Algorithmic hyphenators are available for all languages supported by SCRIPT/VS. ALG is the default when .HY ON is specified.

**NOALG** Specifies that the hyphenation algorithm is not to be used.

196 DCF: SCRIPT/VS Language Reference

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NOALT</b>   | Specifies that SCRIPT/VS should try algorithmic hyphenation, if it is enabled, if a useable hyphenation point cannot be found in the current dictionaries.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>LADDER</b>  | Specifies a positive number indicating the maximum number of successive hyphenated lines. The initial value of <i>n</i> is 2, which means that no more than two consecutive output lines are hyphenated.                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>MAXPT</b>   | Specifies a positive number indicating the minimum number of characters following a hyphen. The initial value of <i>n</i> is 2, which means that a hyphenation point in a word must leave at least two characters at the end of the word.                                                                                                                                                                                                                                                                                                                                                                       |
| <b>MINPT</b>   | Specifies a positive number indicating the minimum number of characters preceding a hyphen. The initial value of <i>n</i> is 4, which means that the hyphenation point in a word must leave at least four characters at the beginning of the word.                                                                                                                                                                                                                                                                                                                                                              |
| <b>MINWORD</b> | Specifies a positive number indicating the minimum number of characters in a hyphenated word. The initial value of <i>n</i> is 6, which means that a word must have at least six characters to be hyphenated.<br><br>Note that the sum of MINPT and MAXPT values may not exceed the MINWORD value.                                                                                                                                                                                                                                                                                                              |
| <b>RANGE</b>   | Specifies the maximum or minimum factor by which word spaces may be compressed or expanded to avoid hyphenation, respectively.<br><br>The RANGE parameter may be specified as one or two factors. A factor of less than 1.0 establishes the amount of compression allowed; a factor of greater than 1.0 establishes the amount of expansion allowed. A factor equal to 1.0 sets the maximum and the minimum if no other value is given later. If only one factor is given and it is "1.0," both the maximum and minimum are reset.<br><br>If RANGE is omitted, both the maximum and minimum are reset to "1.0." |

**Initial Setting:** OFF

Parameter initial settings:

```
.HY LADDER 2
.HY MAXPT 2
.HY MINPT 4
.HY MINWORD 6
.HY RANGE 1.0
```

**Default:** Restores the initial setting.

## Notes

- The values for MAXPT, MINPT, MINWORD, and LADDER are part of the page environment. The ON, OFF, ALG, and ALT settings are part of the active environment.

## Remarks

- When SCRIPT/VS is concatenating input lines to fill output lines, but the next word does not fit on the line, it ordinarily moves the word onto the next output line.

Whether hyphenation is on or off, SCRIPT/VS first attempts to compress the word spaces in the line to make room for the last word. If the line cannot be sufficiently compressed, SCRIPT/VS then attempts to expand the word spaces to fill all of the extra room at the end of the line. Use the RANGE parameter to set limits on how much compression or expansion is acceptable.

If the line cannot be compressed or expanded enough to avoid hyphenation, SCRIPT/VS attempts to hyphenate the last word. Hyphenation is performed only if:

## .HY [Hyphenate]

- The number of characters in the word is equal to or greater than the number given with MINWORD.
- A hyphenation point can be found so that:
  - The number of characters before the hyphenation point is equal to or greater than the number given with MINPT.
  - The number of characters after the hyphenation point is equal to or greater than the number given with MAXPT.
- A usable hyphenation point is found, considering LADDER, MAXPT, MINPT, MINWORD, or RANGE values in effect. SCRIPT/VS searches the candidate word for a usable hyphenation point only as far as the number of characters that would fit on the line. Unless the NOALG parameter was specified, the algorithmic hyphenator is used, even if hyphenation points occur later in the word.

Use the ALT parameter to prevent SCRIPT/VS from calling the algorithmic hyphenator if the word is found in one of the current dictionaries, even if a useable hyphenation point is not found. The entire word is moved to the next line.

2. When a word is found in a dictionary, even if no useable hyphenation points are found, no additional dictionaries are searched; additionally, if ALT is specified, the algorithmic hyphenator is not called to create hyphenation points.
3. As many options as necessary may be specified. If contradictory options are given, the last is used.
4. If you specify a parameter that takes a value but you do not provide a value, the value for that parameter is reset to its default.
5. IBM-supplied dictionaries exist for all of the supported languages but the following dictionaries are empty: Danish, Finnish, Icelandic, Norwegian, Portuguese, and Swedish. However, you can create user dictionaries for spelling verification for these languages using the Dictionary Maintenance Program supplied with DCF.

**Note:** Refer to the *Document Composition Facility: SCRIPT/VS Text Programmers's Guide* for more information.

6. Hyphenation results might not always be accurate. Refer to the chapter on “Verifying Spelling” in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for details.

7. The algorithmic hyphenator does *not* look for hyphenation points in words that contain the following hexadecimal codepoints. The International Codepage 500 character is in parentheses:

- X'00' — X'3F' (unprintable characters)
- X'4D' (plus sign)
- X'5F' (circumflex accent)
- X'6A' (broken vertical line)
- X'79' (grave accent)
- X'7C' (at sign)
- X'8A' (left angle quotes)
- X'8B' (right angle quotes)
- X'8F' (plus or minus sign)
- X'90' (degree symbol)
- X'9A' (feminine ordinal indicator)
- X'9B' (masculine ordinal indicator)
- X'9D' (cedilla)
- X'9F' (international currency symbol)
- X'A0' (micro symbol)
- X'AF' (registered trademark symbol)
- X'B0' (cent sign)



- X'B1' (pound sterling sign)
- X'B2' (yen sign)
- X'B4' (copyright symbol)
- X'B5' (USA section symbol)
- X'B6' (USA paragraph symbol)
- X'B7' (one-quarter)
- X'B8' (one-half)
- X'B9' (three-quarter symbol)
- X'BA' (logical not sign)
- X'BB' (vertical line)
- X'BC' (overline)
- X'BD' (umlaut accent)
- X'BE' (acute accent)
- X'BF' (multiply sign)
- X'C0' (left brace)
- X'D0' (right brace)
- X'E0' (backslash)
- X'E1' (divide sign)
- X'EA' (two superscript)
- X'FA' (three superscript)
- X'FF' (unprintable)

8. Words containing hyphens might be incorrectly hyphenated with the algorithmic hyphenator. Use the .DU ADD control word to add such words to the addenda dictionary with correct hyphenation points.
9. If NOALT is active, SCRIPT/VS looks in the current dictionaries until the word is found. If that entry for the word contains a useable hyphenation point, it is used. If the hyphenation point is not useable, the dictionary search is stopped and the algorithmic hyphenator is called if it is enabled. If the word is not found, the algorithmic hyphenator is called, if it is enabled, to try to determine a hyphenation point.  
  
If ALT is active, SCRIPT/VS looks in the current dictionaries until the word is found. The algorithmic hyphenator is not called, even if the hyphenation point is not useable or there is no hyphenation point. If the word is not found in the active dictionaries, the algorithmic hyphenator is called, if it is enabled, to try to determine a hyphenation point.
10. A word ended by a required blank will not be hyphenated if the word itself fits on the output line even if it no longer fits with the attached required blank and subsequent characters.

## Examples

- You can enable hyphenation using the SCRIPT/VS dictionaries, but not the algorithmic hyphenation routine, by entering

```
.hy on noalg minpt 3
```

Hyphenation is considered successful if at least three characters precede the hyphen point.

- You can control the size of the word fragments resulting from hyphenation by entering

```
.hy minpt 2 maxpt 2 minword 5
```

Word fragments of only two characters might be left at the beginning or end of an output line, but only words of five characters or more are hyphenated.

- You can specify the range within which the line can be compressed to avoid hyphenation by entering

```
.hy range .75
```

The word spaces in the line can be compressed by as much as 25 percent in the attempt to make enough room for the last word. If the word still does not fit on the line, the ALT parameter attempts to hyphenate it.

## **.HY [Hyphenate]**

If instead you enter

```
.hy range .75 1.5
```

SCRIPT/VS still attempts to make enough room for the last word by compressing the word spaces; if this fails, SCRIPT/VS then attempts to fill the extra room at the end of the line by expanding the word spaces up to 50 percent. Only if this also fails does SCRIPT/VS attempt to hyphenate the word.

- If you specify 1.0 as the only value on the RANGE parameter of the .HY control word

```
.hy range 1.0
```

then both the minimum and maximum factors are 1.0.

When 1.0 is encountered as a factor, both the minimum and maximum factors are set to 1.0, therefore, if you specify two values on the RANGE parameter and one of those is 1.0 and the other is not, be careful of the order in which you specify these values. For example, if you specify

```
.hy range 1.0 .8
```

then the minimum factor is .8 and the maximum factor is 1.0.

If you specify

```
.hy range 1.0 1.1
```

then the minimum factor is 1.0 and the maximum factor is 1.1.

If you specify

```
.hy range .8 1.0
```

then the minimum and maximum factors are both reset to 1.0.

If you specify

```
.hy range 1.1 1.0
```

then the minimum and maximum factors are both reset to 1.0.

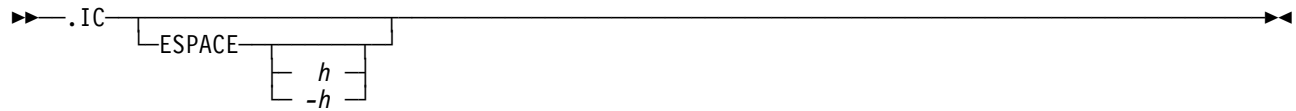
---

## **.IC [Intercharacter Space]**

### **Function**

Use the .IC [Intercharacter Spacing] control word to insert extra white space between characters.

### **Syntax**



### **Parameters**

**ESPACE** Specifies that a fixed amount of extra horizontal space be inserted *between* characters of each word. *h* is the amount of additional horizontal space to be inserted.

**Initial Setting:** 0

**Default:** Restores the initial setting

### **Notes**

- .IC is ignored for line devices.
- The intercharacter space value is included in the active environment.

### **Remarks**

1. The extra intercharacter space inserted *between* characters is in addition to the normal horizontal white space that is designed as part of each character. The spacing between words is *not* affected by the .IC control word.
2. Large negative intercharacter spacing values may result in formatting problems if the spacing requested with .IC causes text to be placed beyond the left edge of the column or page.
3. Negative values of intercharacter spacing are ignored for page printers.

### **Examples**

Use the .IC control word to stretch text slightly for emphasis. For example, if you enter

```

This is an
.ic space p3
important
.ic space 0
word.
  
```

the results are

```

This is an
i m p o r t a n t
word.
  
```

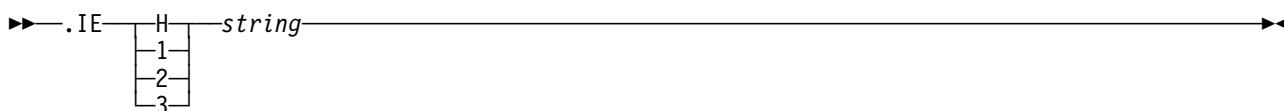
---

## .IE [Index Entry ]

### Function

The .IE [Index Entry] control word formats a single index entry. The .IE control word is normally used only by the .IX [Index] control word to format individual index entries constructed from .PI [Put Index] control words.

### Syntax



### Parameters

**H** Indicates that an index heading is to be generated.

**1 2 3** Defines the level of the index entry as a primary, secondary, or tertiary entry. Appropriate formatting for spacing, indention, and so on, is provided for each level of index entry.

**string** The text of the index entry.

### Notes

- .IE causes a break.
- .IE ensures that the page is started.

### Remarks

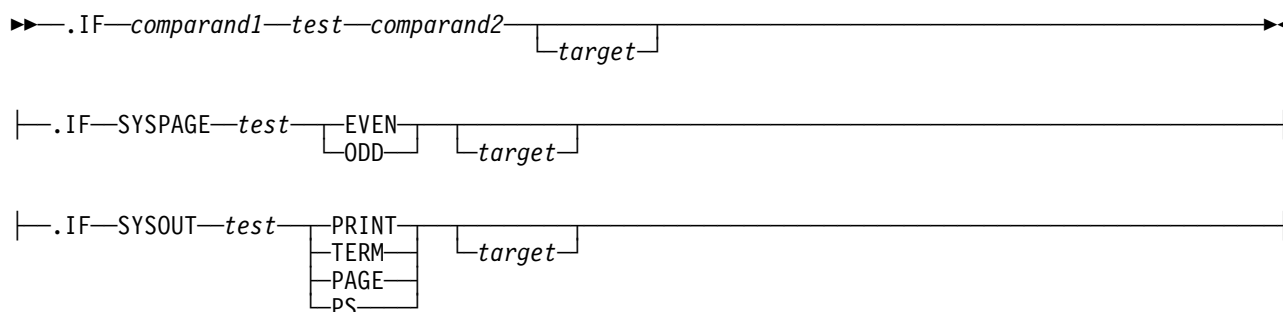
1. The .IX [Index] control word creates .IE control words to format the index. The text of index entries generated by .IX [Index] consists of the index term and the page numbers on which the term appears, separated by two required blanks. The .IX [Index] control word generates an .IE H control word when the first character of an index entry differs from the previous index entry.
2. Because an .IE control word is executed for each index entry by the .IX [Index] control word, you can replace it with a macro of the same name to change the formatting provided for index entries. However, note that the .IX [Index] control word calls for the .IE control word in the form .IE1, .IE2, and so on. This means, for example, that to replace the default index headings with your own, more sophisticated headings, you need supply only an .IEH macro; your macro processes all index headings, but level 1, 2, and 3 index entries are processed by the .IE control word.

## .IF [If]

### Function

The .IF [If] control word allows a SCRIPT/VS input line to be processed conditionally.

### Syntax



### Parameters

**comparand1** Any string to be used as the first *comparand*. This *comparand* can be the value of a set symbol.

**comparand2** Any string to be used as the second *comparand*. This *comparand* can also be the value of a set symbol.

**test** A 1- or 2-character code that indicates to SCRIPT/VS how to determine whether the comparison between the two comparands is true. The following codes are recognized by SCRIPT/VS:

| Codes |    | Meaning               |
|-------|----|-----------------------|
| eq    | =  | equal                 |
| ne    | ≠  | not equal             |
| gt    | >  | greater than          |
| lt    | <  | less than             |
| ge    | >= | greater than or equal |
| le    | <= | less than or equal    |

**target** Any valid SCRIPT/VS input line containing a control word, a macro, or text. If the condition is true, then the target line is processed next, with the first nonblank character after the second comparand treated as the first position of the subject line. If the condition is not true, the target line is ignored, and processing continues with the input line that follows the .IF control line.

**SYSPAGE** Determines whether the page currently being processed is an even- or odd-numbered page.

**EVEN** Indicates that the test is for an even-numbered page.

**ODD** Indicates that the test is for an odd-numbered page.

**SYSOUT** Determines whether SCRIPT/VS output is being formatted for a printer or a terminal.

**PRINT** Indicates that the test is to determine if output is being formatted for a 1403 or a 3800.

**TERM** Indicates that the test is to determine if output is being formatted for a 2741 or a 3270.

## .IF [If]

|             |                                                                                               |
|-------------|-----------------------------------------------------------------------------------------------|
| <b>PAGE</b> | Indicates that the test is to determine if output is being formatted for a page printer.      |
| <b>PS</b>   | Indicates that the test is to determine if output is being formatted for a PostScript device. |

In SCRIPT/VS, more variety is possible in output formatting than can be determined with the SYSOUT parameter alone. You can also use the SCRIPT/VS system symbols '&\$LDEV' and '&\$PDEV' to determine the actual logical and physical devices.

## Remarks

1. The .IF [If] control word, in conjunction with .TH [Then], .EL [Else], .AN [And], and .OR [Or], allows you to construct complex logic statements.
2. The .IF control word itself does not cause a break; the target control word might, if it is processed.
3. Two special sets of comparands are recognized by the IF processing routine. These are SYSPAGE and SYSOUT. You can use SYSPAGE to determine whether the current page is even- or odd-numbered. When you use these two special comparands, you must capitalize the parameters; SYSPAGE is recognized, but sypage is not. You can use any of the test codes with SYSPAGE and SYSOUT:  

```
.if SYSPAGE eq EVEN (do this)
```

is the same as

```
.if SYSPAGE ne ODD (do this)
```
4. Each of the comparands can be up to 255 characters in length, and the shorter comparand is extended to the length of the longer with trailing blanks.
5. If substitution is off when the .IF control word is processed, all valid symbols in the comparands are resolved before the comparison is made.
6. If either comparand contains an imbedded blank, you must place the comparand in a symbol. Symbols containing imbedded blanks must be compared with substitution off so that the test to be performed and the target of the .IF can be identified.

## Examples

- The target of an .IF can be another .IF. Suppose you wanted to imbed a file called ABC if it is Monday afternoon. You could use the following:

```
.se H = &SYSHOUR  
.se D = &SYSDAYOFW  
.if &H ge 12 .if &D eq 2 .im ABC
```

This is the same as saying, IF the hour is 12 or more, AND IF today is Monday, THEN imbed the file; ELSE, go on to the next line.

- If one of the comparands can be a null symbol, another technique should be used:

```
.if X&answer eq Xyes (do this)
```

Now, if the symbol answer is null, the line becomes

```
.if X eq Xyes (do this)
```

Otherwise, if you had not included the Xs, a null symbol could shift the fields over like this

```
.if eq yes (do this)
```

and yes is not a recognized condition. Note that the symbol is null only if so set by the .SE or .RV control words.

- You can use the .IF control word as part of a conditional sequence that allows you to specify different fonts for a particular physical device:

```
.if &$pdev = 1403
.th .df font1 us
.th .df font2 up
.el .df font1 &$CHAR(1)
.el .df font2 &$CHAR(2)
```

---

## .IL [Indent Line]

### Function

Use the .IL [Indent Line] control word to indent the next output line.

### Syntax



### Parameters

*h* Specifies the amount of horizontal space to shift the next output line from the current margin. *+h* shifts text to the right, and *-h* shifts text to the left.

**Initial Setting:** 0

### Notes

- .IL causes a break.
- Line indention is included in the active environment.

### Remarks

1. The .IL [Indent Line] control word provides a way to indent only the next output line. The line is shifted to the right or the left of the current margin (which includes any indent or offset values in effect).
2. The value of **h** represents the amount of blank space just before the text. Thus, .IL .5i indents only the next line one-half inch, and the text begins *after* this blank margin area.
3. The .IL control word and the .UN [Undent] control word are complements; thus, the control words .UN 5 and .IL -5 are equivalent.
4. The .IL control word is useful for beginning new paragraphs.
5. When successive .IL and .UN control words are encountered without intervening text, or when positive or negative increments are specified for .IL control words entered without intervening text, the indent amount is newly set for the next output line, using only the last specified .IL or .UN (any previously specified .IL or .UN is canceled). Thus the lines

```
.il 4m
.il +6m
```

result in the next line being indented 6 em-spaces, not 10 em-spaces.
6. An attempt to indent a line to the left of the real left margin or to the right of the real right margin results in an error message, and all indentions are reset to zero.
7. The .IL control word is triggered by the next text, skip, or space line.



## Examples

- .il 3m

This line is preceded by the control word .il 3m. The purpose of this example is to demonstrate the .IL control word. The .IL control word indents the first subsequent output line. In this case the first subsequent output line is indented 3m.

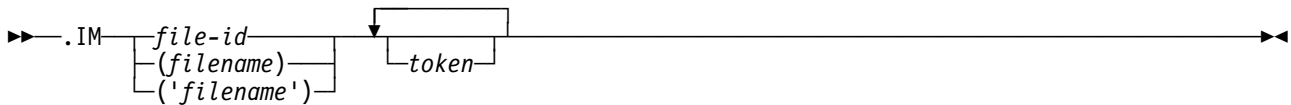
---

## .IM [Imbed]

### Function

Use the .IM [Imbed] control word to process the contents of a specified file at this point in the current file. Processing continues as though the material in the imbedded file were part of the current file.

### Syntax



### Parameters

*file-id* An 8-character SCRIPT/VS name for the file to be imbedded.

An 8-character name can be associated with an external file or a data set with the .DD [Define Data File-id] control word. If no .DD has been executed for the name, in CMS and ATMS-III, an external file or data set name is built by SCRIPT/VS from the given name, using the rules for the current environment. See “Naming the Primary Input File” on page 8 for information about ATMS-III, and see “SEARCH: Specify a Library” on page 42 for information about other environments. In TSO, if the *file-id* to be imbedded is not the real name of a member of a PDS named on the SEARCH option or allocated to TEXTLIB, then you must define the *file-id* with a .DD [Define Data File-id] control word because you cannot specify a fully qualified dataset name on the .IM [Imbed] control word in TSO.

*filename* The real name of the file or data set to be imbedded. The *filename* must be enclosed in parentheses.

If no .DD has been executed for the file or data set, SCRIPT/VS assigns an 8-character name to be henceforth associated with that file or data set.

If the *filename* contains lowercase or special characters, it must be enclosed in single quotation marks ( ' ') and parentheses.

*tokens* Positional values with a maximum length of 8 characters to be passed to the file to be imbedded.

The first *token* (word) becomes the value of the symbol &1, the second *token* becomes the value of the symbol &2, and so on. The symbol &0 contains the number of tokens that were passed; up to 14 can be specified.

### Remarks

1. When a .IM control word is encountered, the contents of the file named on that .IM control word are processed immediately. When processing is completed in the imbedded file, processing continues in the outer file at the input line following the .IM control word.
2. Error messages, trace output, and identifiers provided by the NUMBER option of the SCRIPT command all use the internal 8-byte file-id to describe a file.
3. Any SCRIPT/VS control word or text can be in an imbedded file. Files can be imbedded to a maximum nesting level of 16, but no more than 16 files can be active at one time. If you have many files that are open because of the .EF [End of File] control word, the nesting limit can be reduced. After .EF has been processed, that file is left open, but it is not in the list of currently imbedded files.

4. The .IM and .AP control words perform similar functions, but .IM allows the contents of a second file to be inserted into the processing of an existing file, rather than appended to it.
5. The maximum number of tokens that can be passed to the imbed file are reduced if you use the .IM ('filename') or .IM (filename) form of the imbed control word. The maximum number of tokens is reduced by two, one for each parenthesis, and an additional one for each additional word in the file name. SCRIPT/VS issues a warning message if more than 16 tokens are on the control word line.
6. The symbols &1 through &14 are reset whenever an .IM or .AP control word is processed, and the token &0 is reset to the number of nonnull tokens. If you want to leave token &1 unset but set token &2, you can use a percent sign (%) in place of token1 (or any other token you want left unset).
7. You cannot imbed any file that is already active.

## Examples

- .im common chap4

The contents of the SCRIPT file whose file-id is COMMON are inserted into the processing sequence of the current SCRIPT file; when the end of the COMMON file is reached, processing of the current file resumes. The token CHAP4 is set as the value of the symbol &1. The file COMMON might have in it another imbed in the form

```
.im &1
```

and this would be substituted as

```
.im chap4
```

A different file could contain the control word

```
.im common CHAP5
```

so &1 in COMMON is substituted for CHAP5 instead.

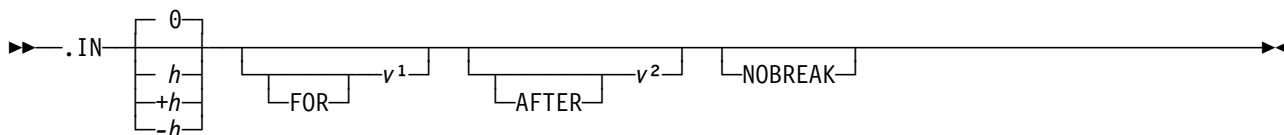
---

## .IN [Indent]

### Function

Use the .IN [Indent] control word to change the left margin displacement of SCRIPT/VS output. Figure 6 on page 438 shows the relationship of the .IN [Indent] to the layout of a SCRIPT/VS output page.

### Syntax



### Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>h</i>       | Specifies the amount of space to be indented. If omitted, 0 is assumed, and indentation reverts to the left margin. If you use <i>+h</i> or <i>-h</i> , the current left margin is incremented or decremented accordingly.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>FOR</b>     | An optional parameter that signifies that the following parameter on the line specifies the vertical duration of the indentation. If FOR is omitted, the number after <i>h</i> on the line is taken as <i>v1</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>v1</i>      | <p>Specifies the vertical duration of the indentation. The indentation changes temporarily for the vertical distance specified in <i>v1</i> and then reverts to the original indentation.</p> <p>If <i>v1</i> is specified as 0, the new indentation is temporary, but it remains in effect until changed by another .IN control word. If <i>v1</i> is specified as *, the new indentation is permanent until changed with another .IN control word. See the remarks below for a discussion of the difference between a temporary and a permanent change to the indentation. If <i>v1</i> is not specified at all, * is assumed, and the new indentation is permanent.</p> |
| <b>AFTER</b>   | An optional parameter that signifies that the following parameter on the line specifies the vertical distance until the new indentation takes effect. If AFTER is omitted, the number after <i>v1</i> on the line is taken as <i>v2</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>v2</i>      | Specifies the vertical distance until the new indentation takes effect. The previous indentation remains in effect until the vertical distance given has been formatted, and then the new indentation takes effect. If <i>v2</i> is specified as 0, the new indentation takes effect immediately. If <i>v2</i> is not specified at all, 0 is assumed, and the new indentation takes effect immediately.                                                                                                                                                                                                                                                                    |
| <b>NOBREAK</b> | Specifies that a break is not to be performed when the .IN control word is encountered.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

**Initial Setting:** 0

**Default:** Restores the initial setting.

### Notes

- .IN causes a break unless the NOBREAK parameter is specified.
- Indention is included in the active environment.

## Remarks

1. The .IN control word resets the current left margin. The new indentation can be permanent or temporary, depending upon whether a duration was given in *v1*. The current indentation for any line is composed of the permanent indentation plus the temporary indentation. If you specify the horizontal distance to be indented with a plus (+) or minus (–) sign, the new indentation is calculated by adding or subtracting your specified increment to the *permanent* indentation. That is, before a new indentation is calculated, any temporary indentation is removed from the total, leaving only the permanent indentation as the current value. (This is the only difference between a permanent change to the indentation and one that is temporary until changed.)

Any new .IN control word can change the current indentation value. This indentation remains in effect for all following lines (including new paragraphs and pages), until another .IN control word is encountered or until the vertical duration has been fulfilled. “.IN 0” cancels the indentation, and output continues at the original left-margin setting.

2. The value of *h* represents the amount of blank space just before the text. Thus, .IN .5i sets a left margin of one-half inch, and the text begins *after* this blank margin area.
3. If a vertical distance is given in *v2*, any blank skip or space lines inserted before text is formatted are not counted. Counting begins with the first nonblank text line, and thereafter includes any subsequent blank lines.
4. The value of the system symbol &\$IN reflects the composite net indentation for the next output line, including permanent and temporary components.
5. An attempt to set the indentation to the left of the real left margin or to the right of the real right margin results in an error message, and all indentation is reset to zero.
6. If FOR is not specified, the next number after *h* is taken as *v1*, unless preceded by AFTER. If no *v1* is given, \* (permanent indentation) is assumed.
7. If AFTER is not specified, the next number after *v1* is taken as *v2*, unless preceded by FOR.  
If no *v2* is given, 0 (immediate new indentation) is assumed.
8. If *v2* is specified as 0 or defaults to 0 (immediate new indentation) and if NOBREAK was specified, the new indentation takes effect on the next line to be started. If no line is started, the new indentation takes effect immediately.

## Examples

- .in 5

All lines processed after this request are indented 5 spaces from the left. This indentation continues until another .IN control word is encountered.

- .in 0

The effect of any current .IN and .OF control words is canceled, and output is formatted flush left.

- If you specify indentation for a specific vertical duration and then enter a .IN 0 before that vertical duration has been reached, indentation is returned to the left margin immediately, because .IN causes a break. For example, if you specify

## **.IN [Indent]**

```
.in 5 for 1i
```

All lines processed after this request are indented 5 character spaces from the left. This indention continues for one inch or until

```
.in 0
```

another .IN control word is encountered. And the text goes on from here and is placed back at the left margin.

The result is

All lines processed after this request are indented 5 character spaces from the left.

This indention continues for one inch or until

another .IN control word is encountered. And the text goes on from here and is placed back at the left margin.

- The .OF [Offset] control word

```
.of .7i
```

is similar to the .IN [Indent] control word

```
.in .7i 0 1
```

(or .in .7i for 0 after 1)

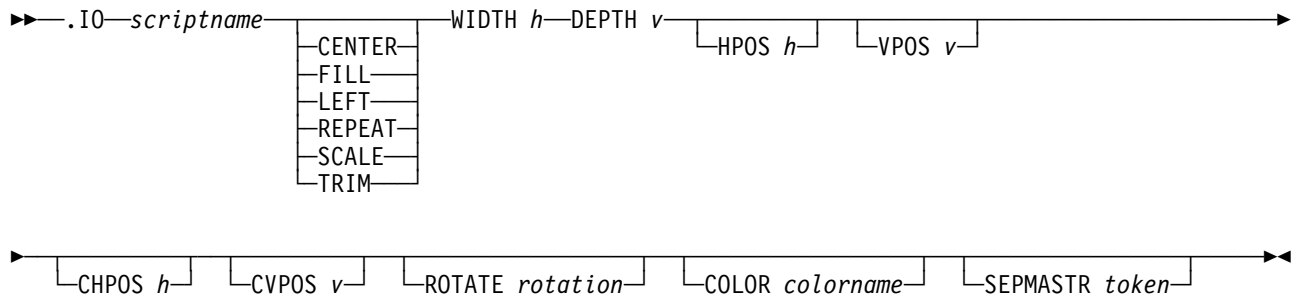
The difference is that if + or – is specified, the .IN control word calculates the new indention based on the permanent component of the total indention, whereas .OF uses the temporary component. Thus, .OF –*n* cannot reduce the current indention more than the temporary component, but .IN –*n* can.

## .IO [Include Object]

### Function

Use the .IO [Include Object] control word to include IOCA, GOCA, BCOCA, IOCA/GOCA page segments, and certain non-MO:DCA objects into the output and override different characteristics of the object when formatting for an AFP logical device type.

### Syntax



### Parameters

- scriptname* Specifies the name of the object to be included. *scriptname* is the same name used on the .DO [Define Object] control word to describe the object. *scriptname* can be up to 16 national characters.
- CENTER** Specifies the object is centered in the object block as defined by the WIDTH and DEPTH parameters. Any portions of the object that fall outside of the object block are trimmed away. No scaling is performed.
- FILL** Specifies the object should be asymmetrically scaled to fit in the object block as defined by the WIDTH and DEPTH parameters. The scale amount for the horizontal and vertical dimensions may be different to exactly fill the object area as defined with the WIDTH and DEPTH parameters.
- LEFT** Specifies the object is positioned at the upper, left-hand corner of the presentation space as defined by the HPOS, VPOS, CHPOS, and CVPOS parameters. Any portion of the object that falls outside of the object block as defined by the WIDTH and DEPTH parameters is **not** trimmed and could cause an exception condition by the presentation system.
- REPEAT** Specifies the object is positioned at the upper, left-hand corner of the presentation space defined by the HPOS and VPOS parameters and repeated as often as necessary to fill the object area as defined by the WIDTH and DEPTH parameters. Any portion of the object that falls outside of the object block as defined by the WIDTH and DEPTH parameters is **not** trimmed and could cause an exception condition by the presentation system.
- SCALE** Specifies the object be scaled to fit within the object block as defined by the WIDTH and DEPTH parameters. The center of the object is placed in the center of the object block and the object is scaled up or down to fit the block. Scaling in the horizontal and vertical directions is symmetrical. The SCALE parameter ensures that all of the data in the object is presented in the object block at the largest possible size. The object is not trimmed.

## .IO [Include Object]

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TRIM</b>  | Specifies the object is positioned at the upper, left-hand corner of the presentation space as defined by the HPOS, VPOS, CHPOS, and CVPOS parameters. Any portion of the object that falls outside of the object block as defined by the WIDTH and DEPTH parameters is trimmed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>WIDTH</b> | Specifies the width of the object block. The WIDTH parameter is required. The <i>h</i> value can be any valid horizontal unit of space. The <i>h</i> value indicates the width of the object block regardless of its rotation. The value must be greater than zero. The WIDTH and DEPTH parameters along with mapping parameters of LEFT, SCALE, CENTER, and TRIM determine how much of the object is presented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>DEPTH</b> | Specifies the depth of the object block. The DEPTH parameter is required. The <i>v</i> value can be any valid vertical unit of space. The <i>v</i> value indicates the depth of the object block regardless of its rotation. The value must be greater than zero. The DEPTH and Width parameters along with the mapping parameters of LEFT, SCALE, CENTER, and TRIM determine how much of the object is presented.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>HPOS</b>  | Specifies the horizontal origin of the upper left corner of the object block regardless of the rotation of the object. The <i>h</i> value can be any valid horizontal unit of space. If HPOS is not specified, a value of +0 is used. If a "+" or "-" is specified, the value is a relative value based on the current horizontal position in the document. Space is reserved for the object in the SCRIPT/VS output as if the object was a single word. The object is placed as if it were a word, positioning its bottom left corner relative to the current position. If a "+" or "-" is not specified, the value is an absolute value figured from the upper left corner of the logical page to the upper left corner of the object. Space is not reserved for the object and it overlays any data placed in the same position on the page. If a relative value is specified with the HPOS parameter, but an absolute value is defined with the VPOS parameter, the HPOS value is viewed as an absolute value and no space is reserved for the object.                                                         |
| <b>VPOS</b>  | Specifies the vertical origin from the upper left corner of the page to the object block regardless of the rotation of the object. The <i>v</i> value can be any valid vertical unit of space. If VPOS is not specified, a value of +0 is used. If a "+" or "-" is specified, the value is a relative value based on the current vertical position in the document. Space is reserved for the object in the SCRIPT/VS output as if the object was a single word. The object is placed relative to the current position as if it were a word, positioning its bottom left corner to the current position. If a "+" or "-" is not specified, the value is an absolute value figured from the upper left corner of the logical page to the upper left corner of the object. Space is not reserved for the object and it overlays any data placed in the same position on the page. If a relative value is specified with the VPOS parameter, but an absolute value is defined with the HPOS parameter, the VPOS value is viewed as an absolute value and no space is reserved for the object in the SCRIPT/VS output. |
| <b>CHPOS</b> | Specifies the horizontal offset of the object contents within the presentation space defined with the WIDTH and DEPTH parameters. The <i>h</i> value can be any valid horizontal unit of space. If CHPOS is not specified, CHPOS defaults to zero. If CHPOS is specified, the value is relative to the upper left corner of the object block regardless of how the object is rotated within that object block.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>CVPOS</b> | Specifies the vertical offset of the object contents within the presentation space defined with the WIDTH and DEPTH parameters. The <i>v</i> value can be any valid horizontal unit of space. If CVPOS is not specified, CVPOS defaults to zero. If CVPOS is specified, the value is relative to the upper left corner of the object block regardless of how the object is rotated within that object block.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |



- ROTATE** Specifies the angle of rotation of the object within the object block. Rotation is relative to the current rotation. The ROTATE parameter is given in degrees and must be specified in one of these multiples of 90: -270, -180, -90, 0, 90, 180, and 270. Rotation is clockwise.
- A rotated, included object can be placed in another SCRIPT/VS object (such as an area or table) which is itself rotated. The rotation of the included object is relative to the upper left corner of the area or table. If the ROTATE parameter is not specified, a rotation of zero is used.
- COLOR** Specifies the color used with this object. The COLOR parameter applies only to output devices with color capability. *colorname* can be any color previously defined with the .CR [Color] control word.
- SEPMASSTR** Specifies the object is being associated with output separation masters. *token* is a 1- to 8-character name that identifies which separation master the object is associated with. *token* corresponds to the items selected for separation masters specified with the .SM [Separation Master] control word. A token value of ALL is used to indicate that the object be contained in all separation masters, including the default. For more information on the SEPMASSTR parameter, see “.SM [Separation Master]” on page 353.

## Notes

- .IO ensures the page is started.
- .IO is only valid for AFP device types (physical devices of 3820, 4028, and AFP). It is ignored for all other device types (including 3800-3).

## Remarks

1. The level of support for included objects depends on your presentation system.
2. An object included with the .IO control word must be fully described by one or more .DO [Define Object] control words. See “.DO [Define Object]” on page 147 for information about fully and accurately describing an object.
3. The *scriptname* of the included object is required and must be the first parameter specified.
4. SCRIPT/VS doesn't search for or read the object. You must ensure that the object exists and that it is accessible by the presentation system when it is needed.
5. The CENTER, FILL, LEFT, REPEAT, SCALE, and TRIM parameters are options that determine how the presentation system maps the object to the presentation space. Whatever you specify is used in the SCRIPT/VS output file. Some presentation systems may ignore some of these options for particular object types. If one of these parameters is not specified, the option specified in the object is used. If an option is not specified in the object, the default option for the presentation system is used. The architected default is SCALE. If other than FILL is specified, it may appear that the bottom of the object is above the baseline of the current output line. This is because the upper, left-hand corner of the presentation space, as defined by the HPOS, VPOS, WIDTH, and DEPTH parameters, is used to place the object. The object may also have white space at the bottom. Use the .SB [Shift Baseline] control work to fine tune the positioning of the object if necessary.
6. If vertical justification is active, lines containing included objects are subject to vertical justification in the same way as text lines.
7. No line breaks are performed around the object. The included object is treated as a word. Word spaces eligible for justification are added before and after the included object (unless the .CT [Continued Text] control word is used to suppress the interword blanks). Use the .BR [Break] control word before or after the .IO control word to get the desired breaks.

## **.IO [Include Object]**

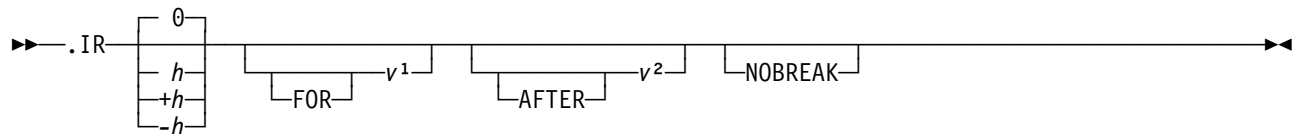
8. If implicit line spacing is active (the line spacing value of the current font is being used), the line spacing of the output line containing the included object is adjusted to allow for the depth as specified with the DEPTH parameter. Extra white space is added above the object to separate it from the previous line. The white space is equal to the leading in the current font. Leading in the current font is the linespacing of the current font minus the em height in the current font.
9. If explicit line spacing is active, the depth of all of the output lines is determined by the explicit line spacing given on the .LS [Line Spacing] control word and is not affected by the value specified with the DEPTH parameter.
10. The horizontal space reserved by SCRIPT/VS must fit on one line. If not enough space is available on the current line to contain the included object, the entire object is placed on the next line, regardless of the overdraw option is in effect. If the object is the only thing on the line and its width exceeds the column line length, the space for the object extends beyond the column line length.
11. Specifications for the HPOS and VPOS parameters that fall outside the boundaries of the page cause an error message.
12. The SEPMASTR parameter is ignored if the SEPMASTR command option is not specified. The SEPMASTR parameter specifies whether or not the object should be included in an output separation master rather than the default master. Objects are included only if the value on the SEPMASTR parameter corresponds to the value selected with the .SM [Separation Master] control word.
13. A value of "HPOS +2i" specifies that the horizontal position of the upper left corner of the object block is positioned two inches to the right of the current position. A value of "VPOS -2i" specifies that the vertical position of the upper left-hand corner of the object block be positioned two inches up from the current position. If neither HPOS nor VPOS is specified, the object block is positioned at the current position. This is the same as specifying:  
HPOS +0 VPOS +0
14. The values for HPOS and VPOS must both be either relative or absolute. If a relative value for HPOS is specified, and an absolute value for VPOS is specified, the HPOS value is viewed as an absolute value. The value is still based on the current horizontal location, but as an absolute value and no space is reserved for the object on the page. The same is true if a relative value for VPOS and an absolute value for HPOS is specified.
15. If an included object is placed in an object, such as an area that is rotated, the included object is rotated with the area.
16. By MO:DCA architectural standards, the only objects included with the .IO control word that can have more than one internal object per physical object are page segments. For such objects, the characteristics specified on the .IO control word override corresponding values in all objects contained in the page segment.
17. Page segments used with the .IO control word can contain IOCA, GOCA, or PTOCA objects. Using page segments that contain IM1-type objects can cause exception conditions by the presentation system. Use the .SI control word instead for those types of page segments.
18. Don't include objects with the .IO control word that contain MCF (Map Coded Font) structured fields. Including such objects causes unpredictable results in the presentation system.
19. For more information about the Include Object (IOB) structured field, see the *Mixed Object Document Content Architecture Reference*, SC31-6802-03.
20. If you are specifying a color for IOCA, BCOCA, GOCA or page segment objects, the color must be defined with the IBM OCA color model.

## .IR [Indent Right]

### Function

Use the .IR [Indent Right] control word to change the right margin displacement of SCRIPT/VS output. Figure 6 on page 438 shows the relationship of the .IR [Indent Right] to the layout of a SCRIPT/VS output page.

### Syntax



### Parameters

- h*** Specifies the amount of space to be indented. If omitted, 0 is assumed, and indentation reverts to the right column boundary. If you use *+h* or *-h*, the current right margin is incremented or decremented accordingly.
- FOR** An optional parameter that signifies that the following parameter on the line specifies the vertical duration of the indentation. If FOR is omitted, the number after *h* on the line is taken as *v1*.
- v1*** Specifies the vertical duration of the indentation. The indentation changes temporarily for the vertical distance specified in *v1* and then reverts to the original indentation.
- If *v1* is specified as 0, the new indentation is temporary, but it remains in effect until changed by another .IR control word. If *v1* is specified as \*, the new indentation is permanent until changed with another .IR control word. See the remarks below for a discussion of the difference between a temporary and a permanent change to the indentation. If *v1* is not specified at all, \* is assumed, and the new indentation is permanent.
- AFTER** An optional parameter that signifies that the following parameter on the line specifies the vertical distance until the new indentation takes effect. If AFTER is omitted, the number after *v1* on the line is taken as *v2*.
- v2*** Specifies the vertical distance until the new indentation takes effect. The previous indentation remains in effect until the vertical distance given has been formatted, and then the new indentation takes effect. If *v2* is specified as 0, the new indentation takes effect immediately. If *v2* is not specified at all, 0 is assumed, and the new indentation takes effect immediately.
- NOBREAK** Specifies that a break is not to be performed when the .IR control word is encountered.

**Initial Setting:** 0

**Default:** Restores the initial setting.

### Notes

- .IR causes a break unless the NOBREAK parameter is specified.
- Indention from the right margin is included in the active environment.

## Remarks

1. The .IR control word resets the current right margin. The new indentation can be permanent or temporary, depending upon whether a duration was given in *v1*. The current indentation for any line is composed of the permanent indentation plus the temporary indentation. If you specify the horizontal distance to be indented with a plus (+) or minus (–) sign, the new indentation is calculated by adding or subtracting your specified increment to the *permanent* indentation. That is, before a new indentation is calculated, any temporary indentation is removed from the total, leaving only the permanent indentation as the current value. (This is the only difference between a permanent change to the indentation and one that is temporary until changed.)

Any new .IR control word can change the current indentation value. This indentation remains in effect for all following lines (including new paragraphs and pages), until another .IR control word is encountered or until the vertical duration has been fulfilled. “.IR 0” cancels the indentation, and output continues at the original right-margin setting.

2. The value of *h* represents the amount of blank space just before the right margin.
3. If a vertical distance is given in *v2* describing when the new indentation is to take effect, any blank skip or space lines inserted before text is formatted are not counted. Counting begins with the first nonblank text line, and thereafter includes any subsequent blank lines.
4. If FOR is not specified, the next number after *h* is taken as *v1*, unless preceded by AFTER.  
If no *v1* is given, \* (permanent indentation) is assumed.
5. If AFTER is not specified, the next number after *v1* is taken as *v2*, unless preceded by FOR.  
If no *v2* is given, 0 (immediate new indentation) is assumed.
6. If *v2* is specified as 0 or defaults to 0 (immediate new indentation) and if NOBREAK was specified, the new indentation takes effect on the next line to be started. If no line is started, the new indentation takes effect immediately.
7. The value of the system symbol &\$IR reflects the composite net indentation for the next output line, including permanent and temporary components.
8. An attempt to set the indentation to the left of the real left margin or to the right of the real right margin results in an error message, and all indentation is reset to zero.

## Examples

- If you enter

```
.ir .5i
```

all lines processed after this request are indented one-half inch from the right side of the column. This indentation continues until another .IR control word is encountered.

- If you enter

```
.ir 0
```

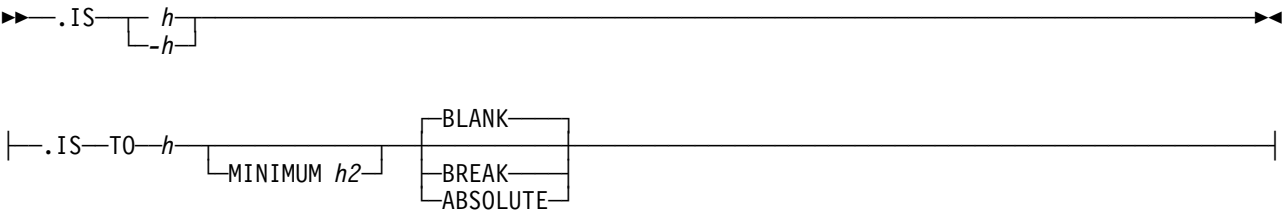
the effect of any .IR control word is canceled, and subsequent lines are formatted to the right margin.

# .IS [Inline Space]

## Function

Use the .IS [Inline Space] control word to insert a specified amount of horizontal white space between two words or to perform a tab to a specific position on the line.

## Syntax



## Parameters

***h*** Specifies an amount of horizontal white space to be inserted into the line. The inserted space, which may be positive or negative, is not subject to justification, and there are no word spaces between the inserted space and the surrounding text.

**TO** Specifies that an immediate tab to a specific position on the line should be performed. *h* gives the position on the line.

**MINIMUM** Specifies the amount of space used to determine whether an immediate tab has been missed. The default value is one horizontal device unit.

When the TO parameter is given and the width of the text on the line exceeds *h* minus *h2*, the immediate tab is considered missed, and processing depends upon the following parameters:

**BLANK** If an immediate tab is missed, a word space is inserted.

**BREAK** If an immediate tab is missed, a break is performed, and the immediate tab is performed on a new line.

**ABSOLUTE** If an immediate tab is missed, a negative horizontal space is inserted to perform the immediate tab.

## Notes

- .IS ensures that the page is started.

## Remarks

1. The .IS control word can be used to insert a fixed amount of horizontal space between two characters, and it can be thought of as a required blank or backspace character of arbitrary width, depending upon whether the inserted space is positive or negative, respectively.
2. The .IS control word can also be used to perform a tab to a specific point on the line, without disturbing the current tab settings established with the .TB [Tab Setting] control word.

## Examples

- The .IS control word can be used to include a fixed amount of horizontal white space in text:

```
Forewarned
.is .5i
is forearmed.
```

The inserted space is treated as a single character and is not subject to justification:

```
Forewarned      is forearmed.
```

- Inserted horizontal white space can be negative:

```
An underscored
.is -.5i
_____ word.
```

The inserted space is treated as a single backspace character:

```
An underscored word.
```

- The .IS control word can also be used to perform an immediate tab: it positions subsequent text at a specific position in the output line without disturbing the tabs established with the .TB [Tab Setting] control word. For example:

```
Enter your name:
.is to 10p
```

\_\_\_\_\_

The amount of inserted space is the difference between the amount specified and the width of the text preceding it in the output line:

```
Enter your name: _____
```

- An immediate tab can be missed if the width of the text already placed on the output line exceeds the value specified with the .IS control word. If the BLANK parameter has been specified, then a missed immediate tab is treated as an ordinary word space. For example,

```
Enter your name:
.is to 1i blank
```

\_\_\_\_\_

results in

```
Enter your name: _____
```

If the BREAK parameter has been specified, a break is performed, and the immediate tab is processed on a new output line:

```
Enter your name:
_____
```

If the ABSOLUTE parameter is specified, the immediate tab is always processed on the current output line; if the tab is missed, negative horizontal space is inserted:

```
Enter your name:_____
```

## .IT [Input Trace]

### Function

The .IT [Input Trace] control word allows trace information about input lines to be displayed at your terminal or written to the same file as error messages.

### Syntax



### Parameters

- ON** Traces any control word that has been specified previously with CTL. This is the default.
- OFF** Terminates tracing.
- MAC** Causes each line coming out of a macro to be traced.
- SUB** Causes each stage of symbol substitution to be traced for lines that contain symbols or GML tags.
- GML** Causes various stages of GML processing to be traced. The number of different stages a GML tag goes through depends upon the attribute scanning rules that are in effect for the tag. GML tracing can show:
- The tag and the APF that are executed
  - Each line that is scanned for attributes
  - Each regular attribute that was found in the scan
  - All value attributes that were found in the scan
  - The residual text line that was found in the scan.
- ALL** Causes macro and symbol substitution, GML processing, and all control word lines to be traced.
- CTL** Causes the control words specified to be traced just before they are executed. If no control words are given with .IT CTL, then the list of control words to be traced is cleared. If some control words are given, they are added to the list. Nonexistent control words can be added to the list without causing an error, but they are never traced because they are detected as invalid control words before tracing would be done. The list remains intact when .IT OFF is executed and is resumed if .IT ON is subsequently executed.

## .IT [Input Trace]

- STEP** Causes SCRIPT/VS to *single step* through all control words that are being traced. If .IT ALL is in effect, all control words are traced. Otherwise, just those control words specified with .IT CTL are traced.
- When .IT STEP is in effect, SCRIPT/VS displays the control word line and then pauses to read a line from the terminal before executing it. The line you enter at this point can simply allow the control word execution to proceed, or you can enter another input line to be processed before, after, or instead of, the traced control word.
- RUN** Cancels .IT STEP mode, while allowing all tracing to continue. (.IT OFF stops STEP mode and stops tracing.)
- SNAP** Displays the current definitions for any symbol or macro that exists by the name or names given. If no names are given, the entire symbol and macro table is displayed. The SNAP is done without changing any other tracing that may be in effect.

**Initial Setting:** OFF

**Default:** ON

## Remarks

1. All trace information is written out as messages. If the MESSAGE (DELAY) option of the SCRIPT command is in effect, the trace information is written to the same SCRIPT/VS utility file as error messages.
2. .IT STEP mode can only take effect if messages (and trace information) are actually being displayed at your interactive terminal. Thus, STEP mode is not available in the batch environment or when the MESSAGE (DELAY) option of the SCRIPT command is in effect. When SCRIPT/VS reads a line from the terminal after tracing a control word line in STEP mode, it can have any of the following formats:

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>null line</b>       | Continue processing                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>?</b>               | Verifies who is reading from the terminal. If you are stepping through control words and you are also using .TE [Terminal Input], it is easy to lose sight of which kind of read is being done from the terminal. While in .IT STEP mode, the single character ? is recognized by the control trace module and by the terminal input module; the message TERMINAL INPUT: or CONTROL TRACE: is displayed; and another read is done. If the read comes from some other source, such as .RV [Read Variable] or .RD [Read Terminal], the ? is taken as ordinary data, just as it would be from terminal input when not in .IT STEP mode. |
| <b>STK 'data line'</b> | The data line entered is stacked and processed after the traced control word has been processed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>PRE 'data line'</b> | The data line entered is processed before the traced control word (the tracing is done before the control word is actually processed).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>REP 'data line'</b> | The data line entered replaces the traced control word line and is processed instead of it.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>'data line'</b>     | The data line is treated like a data line entered with the 'PRE' parameter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

If the new line to be entered is also a control word line that is being traced, it is traced before being processed, giving another opportunity to enter a line. If the line entered causes the original line to be reprocessed later, it can be traced again.

3. The trace function is initially OFF.
4. The SNAP parameter provides a selective printout of all currently defined set symbols and their values. It does not affect the current ON/OFF status of the trace control.



5. On all trace lines, the first three characters indicate which type of trace it is, as follows:

- \*S\* symbol substitution trace
- \*M\* macro substitution trace
- \*G\* GML trace
- \*C\* control word trace
- \*\*\* symbol or macro SNAP line

If .IT ALL is in effect, control word lines can be traced several times. Each line can be traced to show the various stages of symbol substitution, then traced again as a control word line after it has been completely substituted.

## Examples

If you wanted each stage of symbol substitution in a file called TEST to be traced for lines that contain symbols or GML tags, in TEST you could enter

```
.se x = &y
.se yt = help!
.it sub
&x.t is a complex symbol.
```

and the trace would appear like this

```
*S* TEST  <&x.t is a complex symbol.>
          <&yt is a complex symbol.>
          <help! is a complex symbol.>
```

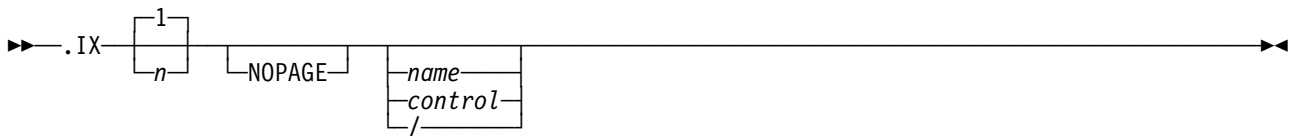
---

## .IX [Index]

### Function

The .IX [Index] control word causes an index to be created, (from entries specified with .PI [Put Index]) formatted, and printed. The index entries are sorted according to either the sort sequence for the language requested with the .DL INDEX control word or your installation's default language. For more information about index sort sequences, refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.

### Syntax



### Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>n</i>       | The number of page numbers that are reserved for the index. If <i>n</i> is omitted, 1 is assumed.                                                                                                                                                                                                                                                                      |
| <b>NOPAGE</b>  | Specifies that SCRIPT/VS does not perform a page eject before and after the index. The index is placed on the current page, and text following is continued on the last page of the index.                                                                                                                                                                             |
| <i>name</i>    | An optional line to be used as the title of the index. If no <i>name</i> is given and the system symbol (&SYSINDEX) is not set, the word INDEX is used, by default. A head-level 1 (.H1) is generated at the top of the index, using either the <i>name</i> given or the word INDEX. The first word in the name cannot be NOPAGE, unless NOPAGE was already specified. |
| <i>control</i> | A control word or macro to be processed at the top of the index instead of the .H1 control word. If this parameter begins with a period, it is assumed to be a control word and not a <i>name</i> .                                                                                                                                                                    |
| <i>/</i>       | Signals SCRIPT/VS not to generate any head-level 1 for the index, but a page eject is still performed, unless NOPAGE is also specified. Use “/” when you want no name on the index, you have no control word to be executed, and you do not want the default name INDEX to be generated.                                                                               |

### Notes

- .IX causes a break
- .IX ends a keep, float, footnote, named area, or table
- .IX ensures that the page is started.

### Remarks

1. Although all of the parameters are optional, if used, they must appear in the order given in the description above.
2. When the .IX [Index] control word is encountered, a head-level 1 is processed (unless overridden by the control parameter). All entries that have been saved with previous .PI [Put Index] control words are then formatted and printed.

When the NOPAGE parameter is specified, the head-level 1 used to process the name parameter does not cause a page eject. There is no space between the last line of text on the page and the head-level 1 that follows, because the default for a head-level 1 has no space or skip before the heading. To put space before the heading, use .SP or .SK before the .IX, or redefine the space or skip value for level-1 headings using the SPBF or SKBF parameters on .DH.

3. When the NOPAGE parameter is specified, the pending floats are not dumped. However, keeps and floats that fit on the current page are placed before starting the index. To get a total float dump before the index, use .FL DUMP.
4. The index is formatted according to the line and page dimensions in effect at the time the .IX [Index] control word is encountered, not those in effect when the .PI [Put Index] control words were processed. Each entry in the index has the page numbers that were in effect when the .PI [Put Index] control words were processed.
5. When the index is completely formatted, subsequent .PI control words are placed in a new index.
6. When the index is completely formatted, a conditional page eject is done (unless NOPAGE was specified), and the new page is numbered as though the index had occupied exactly  $n$  pages. If the index takes more or less than  $n$  pages, there is either an overlap or a gap in page numbering.

When you specify  $n$ , you reserve  $n$  page numbers for the index. For example, if you specify .IX 5 and the index starts on page 20, then the first page after the index is numbered 25. If you do not specify an  $n$  value on .IX and the index starts on page 20, the  $n$  value defaults to one (1) and the first page after the index is numbered 21, regardless of how many pages the index actually occupies.

7. If the NOPAGE parameter is specified along with the  $n$  parameter, the pages that the index starts and ends on must be included in the number of pages reserved for the index ( $n$ ), even if text precedes or follows the index on the same page.
8. If NOPAGE was specified, the index could start at the bottom of a page, and there might not be enough room for the heading or first line of text. Therefore, the text of the index starts on the next page. If this happens, the page number for the page following the index is calculated while still on the bottom of the first page. Therefore, that page needs to be included in the number of pages reserved on .IX in order for the page following the index to be numbered correctly.
9. If you need a page break either before or after the index, but not in both places, use .PA either before or after the .IX NOPAGE.

If you add pages to the current page number, an error can result on the .PA control word if the page number exceeds 99,999,999.

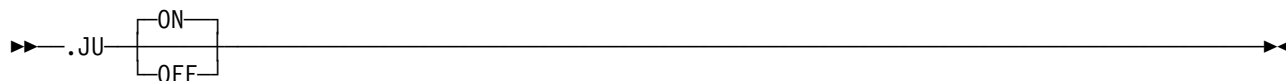
10. When NOPAGE is not specified on .IX, the index contains entries only from pages that precede the .IX control word. Entries created with .PI control words that refer to pages on which the index is formatted are not included. However, if NOPAGE is specified, any .PI control words that are processed before the .IX NOPAGE is encountered appear in that index. This includes .PI control words within items that are placed at the beginning of a page, such as running headings, running footings, and floats.

Page and body areas are normally placed when the page is ended. When an .IX NOPAGE is specified, however, any page and body areas that are placed (using the .AR [Area] control word) prior to the .IX NOPAGE are placed as part of NOPAGE processing. Any .PI control words in those areas are processed at that time and appear in the NOPAGE index.

**Note:** .PI is a deferred control word, which means that it is not processed until the text it is associated with is actually placed on a page. For example, if you define a float before the .IX NOPAGE, but it is placed on the page following the index, .PI control words within that float do not appear in the index.

The .JU [Justify Mode] control word is provided for compatibility with earlier releases of SCRIPT/VS. Most of the same functions are provided by the .FO [Format Mode] control word.

## Syntax



## Parameters

**ON** Restores justification of output lines. If neither ON nor OFF is specified, ON is the default.

**OFF** Cancels justification of output lines. If concatenation is still in effect, .JU OFF results in ragged right output.

## Remarks

Concatenation and justification are controlled by the .FO [Format Mode] control word. Ragged right output results from concatenation ON and justification OFF. The control word .FO LEFT provides this combination. Full formatting, with concatenation and justification both ON, is provided by .FO ON. “As is” output, with concatenation and justification both OFF is provided by .FO OFF. The only combination not covered by the .FO control word is concatenation OFF and justification ON, and if you need this combination, you can use the .JU control word to control it separately.

## **.KP [Keep]**

### **Function**

The .KP [Keep] control word allows you to designate blocks of text that must be kept together in the same column. There are several different ways of designating keeps, and each form has different functions. When .KP is encountered inside another keep, it might end the first keep before starting the new one. If the new keep is of a form that cannot end the current keep, it is ignored, and the text is kept together because it is part of the larger keep.

### **Syntax**



### **Parameters**

- ON** Starts a regular keep. The text within a regular keep is separate from the text outside of it. No output line can be built from text that came from both inside and outside the keep. A regular keep is put in the current column if it fits; otherwise an immediate column eject is done. The regular keep appears in the output in the same relative location as it was in the input. A regular keep ends any other keep before starting. A break is done at the start and the end of a regular keep.
- FLOAT** Starts a floating keep. A floating keep is put in this column if it fits; otherwise, it goes in the next column. The current column is filled using the text following the floating keep. .KP FLOAT ends any other keep before starting. Text that is placed after the float is not rearranged during the text distribution process (as specified with the .BC [Balance Column] control word). A break is done at the start and the end of a floating keep.
- DELAY** Starts a delayed keep. A delayed keep is always printed in the next column, even if there is room for it in this column, and the current column is filled using the text following the delayed keep. A delayed keep, in effect, acts like a floating keep that did not fit in the current column. .KP DELAY ends any other keep before starting. A break is done at the start and the end of a delayed keep.
- INLINE** Starts an inline keep. An inline keep flows with the preceding and following text. No break occurs, and formatting continues as though no keep were designated. All lines that contain text from within the keep are then kept together, and if column balancing is done, the entire keep is moved as a block from one column to another. .KP INLINE ends an inline keep or a keep in the form .KP v + v or .KP v before the inline keep is started, but if a regular, floating, or delayed keep is in process, .KP INLINE is ignored.
- OFF** Marks the end of a regular, floating, delayed, or inline keep. .KP OFF also ends a keep of a designated depth, but it is not required.
- v + v** Starts a keep of a designated vertical depth. The depth of the keep is determined by adding up all the separate v's given. If a previous .KP v or v + v is still active, then the depth is compared to the current remaining depth of the active keep. The larger of the two depths then becomes the active depth.

## .KP [Keep]

For example, .KP 3 + 2 would start a keep for 5 lines, and .KP 2i + 3 would start one for 2 inches plus 3 lines. (This is the only control word that allows you to add up different space units to get a single result.) This form of keep is used by the head level control words .H0 [Heading Level 0]. A keep for a designated depth need not be explicitly ended with .KP OFF. It is ended automatically when its depth has been filled. If an inline or higher keep is in process when .KP  $v + v$  is encountered, the .KP  $v + v$  is ignored. The  $v + v$  parameter does not cause a break.

- $v$  Starts a keep of a designated depth specified by  $v$ . When the designated depth has been filled, a keep of the  $v$  form is automatically ended. If any other form of keep is in process except  $v + v$  or  $v$ , .KP  $v$  is ignored. If a .KP  $v$  or  $v + v$  is already active, then  $v$  is compared to the current remaining depth of the active keep. The larger of the two depths becomes the current remaining depth. The  $v$  parameter does not cause a break.

## Notes

- .KP ends a keep, float, footnote, named area, or table
- .KP ensures that the page is started
- All keeps except inline keeps save the current environment.

## Remarks

1. Keeps started with .KP ON, .KP FLOAT, and .KP DELAY all operate with a separate environment from ordinary text. No output line can be formed by concatenating text from inside the keep to text from outside of it. When the keep is started, the active environment is saved. Offsets and undents are cleared so that the indentation at the beginning of the keep is set to the basic indentation currently in effect, and the maximum column line length is set to the width of the current column.

When the keep is ended, the original text values are restored automatically. This means that if you change the indentation, formatting mode, hyphenation, double-spacing, or certain other things, you need not restore them when the keep is ended. See “The SCRIPT/VS Formatting Environment” on page 451 for a list of the active environment values that are saved and restored for these keeps.

2. Keeps started with .KP INLINE, .KP  $v + v$ , or .KP  $v$  are not separated from the surrounding text. Output lines can be formed by concatenating text from inside the keep to text from outside of it. No environment values are saved or changed, and the text within the keep flows with neighboring text. Formatting continues for these keeps as though no keep had been started, but all output lines encompassed by the keep are kept together in the same column of output.
3. Certain control words are not allowed within a keep. If one of the disallowed control words is encountered, the keep is immediately ended, as though .KP OFF had been processed, and then the disallowed control word is executed. A warning message is issued, identifying the control word that ended the keep. See Table 15 on page 442 for a list of the disallowed control words.
4. Once a floating keep or a delayed keep has been placed in a column, it is considered to be a regular keep. As a result, either may then be moved in the process of text distribution (.BC ON). A floating keep could be moved forward, away from its original placement in a column. A delayed keep could be moved forward or even backward into a previous column.
5. If a keep is too large to be placed on the page, an error message is issued.
6. Vertical keeps of the form .KP  $v$  and .KP  $v + v$  do not suspend widow-zone processing. Instead, the keep is considered to be the first two lines of a widow. This means that one additional line can be kept in order to satisfy widow-zone requirements.
7. Keeps of the  $v$  and  $v + v$  form can combine to form a single keep that is very large. If this keep becomes larger than a page, the keep is ended and a message is issued.

8. If an inline keep is followed by a footnote (.FN ON) or an area (.AR ON), the last line containing text from within the keep might not be kept with the rest of the keep text. This occurs because footnotes and areas do not cause a break, but they do cause any pending keep to be placed on the page. Because a break does not occur, there could be a partially built output line that contains text from within the keep that is not placed in the keep. If inline keeps (.KP INLINE) and footnotes (.FN) or areas (.AR) are closely mixed in a document, a break (.BR) should be inserted prior to the footnote or area control words to cause any partially built line to be placed in the keep.
- | 9. The first line in a keep has the extra leading specified with the EXTRA parameter of the .LS [Line  
| Spacing] control word removed when the keep is placed at the top of the body of the page.

---

## **.LB [Leading Blank]**

### **Function**

The .LB [Leading Blank] control word is generated by SCRIPT/VS and executed whenever an input line that starts with a blank is processed.

### **Syntax**

►► .LB ◀◀

### **Notes**

- .LB causes a break.

### **Remarks**

1. A leading blank on an input line causes a break. This is done by generating and executing a .LB control word whenever a line with a leading blank is processed. Such an input line can only be up to 252 characters long. The function of the .LB control word is identical to that of the .BR control word.

If you want to have leading blanks perform some other function, you can define a .LB macro with .DM [Define Macro], and assuming macro substitution is ON, your .LB macro is executed whenever a leading blank is processed. Note, however, that after the .LB control word or macro is processed, the leading blank is still on the line, and it is processed as part of that text input line. In other words, you cannot use the .LB macro to remove leading blanks from a line.

2. No .LB function is performed for lines processed in literal mode (.LI [Literal]).

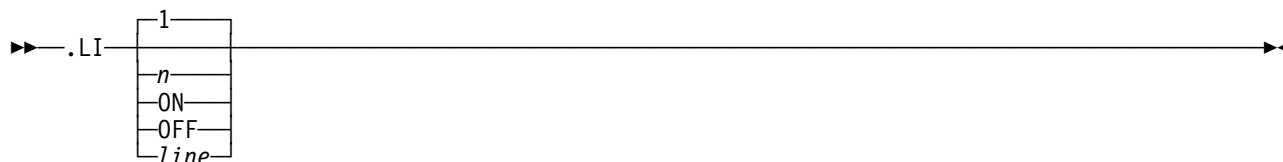


## .LI [Literal]

### Function

The .LI [Literal] control word allows all input lines, including those that begin with periods, to be processed as text.

### Syntax



### Parameters

- n* Specifies the number of lines to be treated literally. If *n* is omitted, 1 is assumed.
- ON** Starts an open-ended literal mode, in which a period in column one in subsequent lines is treated as text. After this control word is processed, SCRIPT/VS recognizes only the .LI OFF parameter.
- OFF** Terminates literal mode if it was ON, or if *n* was given and has not been exhausted.
- line* The line to be treated as literal text.

**Initial Setting:** OFF

**Default:** 1

### Notes

- This is a type 1 control word.
- The line form of .LI starts the page.

### Remarks

1. Ordinarily, any SCRIPT/VS input line that begins with a period is interpreted as a SCRIPT/VS control word or macro. Entering .LI *n* causes SCRIPT/VS control words and macros in the next *n* lines to be processed as normal text even if the control words and macros are the result of GML tag processing. If .LI ON is encountered, all control words and macros in subsequent lines except .LI OFF (which must be recognized to cancel literal mode) are treated as normal text.
2. Literal mode has no effect on GML scanning or symbol substitution. When in literal mode, GML tags are recognized and processed if GML scanning is enabled (.GS TAG ON). However, the resulting control words and macros are treated as text. Symbols are substituted for their associated values if symbol substitution is in effect.
3. When literal mode is in effect, null lines, lines with leading blanks, and lines with leading tabs do not cause a break. Null lines, however, do cancel continuation if the previous line ended with a continuation character. Blank lines also cause the .BL control word to be processed.
4. If you specify .LI OFF the following rules apply:
  - It must be on a line by itself
  - There must be a blank between the control word and the parameter

## **.LI [Literal]**

- A control word modifier can not be used on this line.

## **Examples**

- If a text line must begin with a period,

Study the following control words:

```
.fo off
.li on
.LB [Leading Blank]
.LT [Leading Tab]
.NL [Null Line]
.LI [Literal]
.li off
```

These lines are formatted as

Study the following control words:

```
.LB [Leading Blank]
.LT [Leading Tab]
.NL [Null Line]
.LI [Literal]
```

- If formatting mode had not been turned OFF with .FO OFF, the same lines would be processed as

Study the following

```
control words:      .LB
[Leading Blank]      .LT
[Leading Tab]         .NL
[Null Line]          .LI
[Literal]
```

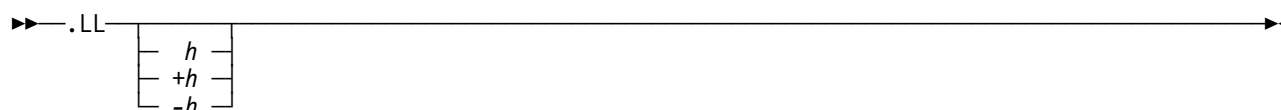
---

## .LL [Line Length]

### Function

The width of all formatted lines is always determined by the setting of the .CL control word. The .LL [Line Length] control word specifies the width used to set the value of the .CL control word at the beginning of running headings, running footings, and footnotes. It also changes the column line length in the body of the page, if the latter has never been set explicitly with .CL [Column Line Length]. Figure 6 on page 438 shows the relationship of the .LL [Line Length] to the layout of a SCRIPT/VS output page.

### Syntax



### Parameters

*h* Specifies an output line length. If no value is specified for *h*, the default value established for the device being used is taken. The value of the output line length plus any page margin must not exceed the maximum page width of the device.

**Initial Setting:** Dependent upon the logical device specified.

**Default:** Restores the initial setting.

### Notes

- .LL causes a break
- .LL ends a keep, float, footnote, named area, or table.
- The line length is included in the page environment.

### Remarks

1. The .LL control word value sets the initial column line length for running headings, running footings, and footnotes. The .CL [Column Line Length] control word sets the line length for text in the body of the page, but if the column line length has never been explicitly set, changing the output line length with the .LL [Line Length] control word also changes the column line length.
2. The .LL control word is also used to determine where reference numbers are placed. See the description of the .RN [Reference Numbers] control word for more details.
3. A new .LL value first takes effect the next time it is used to set the column line length at the start of a new running heading, running footing, or footnote. Because running headings and footings are automatically placed on a page when the page starts, changing the .LL value does not affect the column line length of a running heading or footing until the next page is started. If you reformat the running heading or footing with .RH EXECUTE or .RF EXECUTE, then the column line length of the new running heading or footing remains unchanged, because the new line length value has not taken effect yet.

---

## **.LO [Lead-Out]**

### **Function**

The .LO [Lead-Out] control word identifies a place in a column where up to a specified amount of extra vertical white space can be placed during vertical justification.

### **Syntax**

►► .LO *v* ◀◀

### **Parameters**

*v* Maximum amount of extra vertical white space that can be added to the column where the .LO control word appears.

### **Notes**

- .LO causes a break.

### **Remarks**

1. The amount specified by *v* is rounded to an even multiple of increments as specified with .LS INCR *v*.
2. Lead-out points at the top or bottom of columns are ignored.

### **Examples**

If you specified the following:

A heading or beginning of paragraph.

.lo 1

.sk 1

The rest of the text continues.

the result, after justification, might be

A heading or beginning of paragraph.

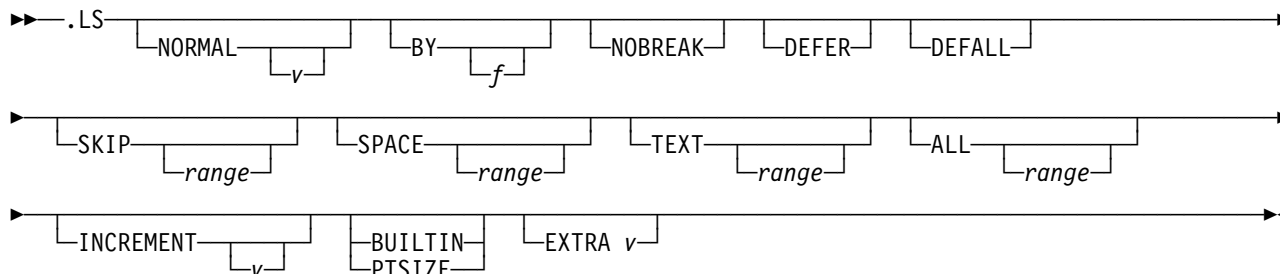
The rest of the text continues.

## .LS [Line Spacing]

### Function

Use the .LS [Line Spacing] control word to change the vertical space separating lines of text and to establish thresholds for increasing or decreasing line spacing for vertical justification.

### Syntax



### Parameters

- NORMAL** Establishes the depth of subsequent output lines, until changed by another .LS control word. If NORMAL is specified, EXTRA and PTSIZE are ignored.
- $v$  gives the space between lines, that is, the vertical distance from one baseline to the next. If  $v$  is specified, the depth of output lines is fixed, regardless of the sizes of the fonts used to print the text on the lines.  $v$  can be any valid vertical space unit.
- If  $v$  is omitted, the spacing of lines are determined dynamically: the depth of each output line is determined by the largest line spacing value of all the fonts used to print the text<sup>9</sup> on the line multiplied by the value specified with the BY parameter.
- BY** Increases or decreases the space between lines.  $f$  is a factor by which the normal line spacing is multiplied to obtain the actual line spacing. (The normal line spacing is the default for the current font, unless it is overridden with .LS NORMAL.)
- If  $f$  is omitted, “1.0” is used.
- NOBREAK** Specifies that this .LS control word is not to cause the usual break.
- DEFER** Specifies that the transition from the old line space factor to the new factor given with this .LS control word takes into account whether the new factor is larger or smaller than the old one. The DEFER parameter is meaningful only if the BY parameter is also specified.
- If DEFER is not specified, then the new factor takes effect on the output line that has the first text following the .LS control word. If NOBREAK was also specified, then this line can also contain text that was before the .LS control word in the input file.
- If DEFER is specified, then the effect of the new factor is delayed for one output line if the new factor is smaller than the old one. If a block of text has a different line space value than the adjacent text, DEFER has the effect of ensuring that the larger of the two line space values is used at the beginning and at the end of that block. This provides visual separation of this block of text from adjacent text.

<sup>9</sup> Excluding text which is formatted with a nonzero baseline shift, as set with the .SB [Shift Baseline] control word.

## **.LS [Line Spacing]**

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DEFALL</b>         | Is the same as DEFER, except that the line spacing change is deferred until the next full output line, regardless of the size of the line space factor.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>SKIP</b>           | Establishes a range of limits within which vertical white space created with the .SK [Skip] control word can be compressed or expanded during vertical justification.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>SPACE</b>          | Establishes a range of limits within which vertical white space created with the .SP [Space] control word can be compressed or expanded during vertical justification.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>TEXT</b>           | Establishes a range of limits within which the depth of lines of text can be compressed or expanded during vertical justification.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>ALL</b>            | Sets the ranges for SKIP, SPACE, and TEXT to the same values.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>range</i>          | <p>The maximum or minimum factor by which skips, spaces, or text lines can be compressed or expanded during vertical justification, respectively.</p> <p>The exact maximum or minimum amount is determined by multiplying the original depth by the specified factor. The result is rounded to multiples of the value specified with the INCREMENT parameter.</p> <p>The range parameter can be specified as one or two factors. Factors of less than 1.0 establish the amount of compression that is allowed; factors of greater than 1.0 establish the amount of expansion that is allowed. If 1.0 is given, both the minimum and the maximum factors are set to 1.0.</p> <p>If range is omitted, the values that are not specified are reset to the default value (1.0).</p> |
| <b>INCREMENT</b>      | <p>Establishes the size of the adjustments in line spacing that can be made during vertical justification. <i>v</i> is the size of the amounts of space that can be added or removed.</p> <p>If <i>v</i> is omitted, one vertical device unit is used.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>BUILTIN</b>        | Specifies the default line spacing value of the current font. If the PTSIZE parameter has been specified, then BUILTIN must be explicitly specified in order to return to the default line spacing value of the current font.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>PTSIZE</b>         | Specifies that line spacing is to be set to a particular point size in the current font. Once PTSIZE has been specified, the BUILTIN parameter must be specified in order to return to the default line spacing value of the current font. PTSIZE is ignored for line mode devices.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>EXTRA <i>v</i></b> | Specifies a fixed amount ( <i>v</i> ) of <i>extra</i> leading to be added to the depth of any output object such as a text line. <i>v</i> can be any valid vertical unit of measure. You cannot specify a <i>-v</i> value that is larger than the normal line spacing value. If <i>v</i> is not specified, then it is set to 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

**Initial setting:** Normal depends on the fonts used in each line. The BY setting is 1.0. The RANGE setting is 1.0.

Parameter defaults:

```
.LS SKIP 1.0 1.0
.LS SPACE 1.0 1.0
.LS TEXT 1.0 1.0
.LS ALL 1.0 1.0
.LS INCREMENT 1
.LS BUILTIN
.LS EXTRA
```

## Notes

- This control word causes a break unless the NOBREAK parameter is specified.
- The line space values are included in the active environment.

## Remarks

1. On line devices, SCRIPT/VS uses the nearest multiple of the fixed line space value for the device for the depth of each line.
2. On page printers, each font is designed with a default line space value appropriate to the size of the characters. SCRIPT/VS uses the largest line space value of all the fonts used on a line (except for those fonts whose baselines have been shifted) as the depth of that line. The BY parameter of .LS can be used to decrease or increase this value to set text more densely or sparsely.
3. For PostScript devices, the default line space value comes from the logical device table. The default line space value is 110% of the current point size. Refer to “Tailoring SCRIPT/VS for your Installation” in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* to learn how to change the logical device table.
4. The NORMAL parameter can be used to override the default line spacing value that comes from the current font. This establishes fixed line spacing, regardless of any font size change.
5. When the .FV [Format Vertically] control word is used to justify columns vertically, the .LS control word determines which kinds of vertical white space is compressed or expanded and sets limits on the amount of adjustment.
6. Vertical white space that is created with .SK [Skip] and .SP [Space], and that uses qualified space units (inches, centimeters, picas, and so on) or the A parameter, is not affected by .LS NORMAL or .LS BY.
7. As many parameters as necessary can be specified with .LS.
8. If negative extra leading results in a negative line spacing value, the negative extra leading is:
  - Ignored for nontext lines
  - Reset to zero for text lines.
9. Any additional space necessary to achieve an output line of a specified size is placed above the line. However, if the output line is at the top of a column in the first section of the body, it is discarded. For example,

```
.rh on
Your Heading
.rh off
.fo off
.ls by 2.0
Body line one.
Body line two.
```

results in

```
Your Heading
Body line one.
```

```
Body line two.
```

On the other hand,

## .LS [Line Spacing]

```
.rh on
Your Heading
.rh off
.fo off
.sp 1
.ls by 2.0
.cd
Body line one.
Body line two.

results in

Your Heading
```

Body line one.

Body line two.

In the above example, the .CD [Column Definition] control word ended the *first* section of the body (which has a space in it) and started the *second* section of the body. Therefore, you get the “extra” space between the line “Your Heading” and the line “Body line one.”

10. You cannot specify a .LS EXTRA value if it causes negative line spacing before a .LS BY value has been applied. Also, .LS EXTRA is either reset or ignored if it causes a negative text rule or page segment line spacing value after a .LS BY value has been applied.
11. If line spacing is set to a small value or to zero, causing two output lines to overlap, text and rules might be horizontally misaligned when your output is formatted for the 3800 Printing Subsystem Model 1.
12. The extra leading created with the EXTRA parameter is removed from the first lines in running headings, running footings, body areas, and page areas when that line contains text or horizontal rules. The extra leading is removed from the first lines in top column floats, keeps, and section areas when those items occur at the top of the body of the page. The extra leading is removed from the first line in the body of a page.

## Examples

- You can double-space your output by entering

```
.ls by 2.0 defer
```

Each line of text (and .SK and .SP, when given in line spaces) is twice as deep as normal.

The .DS [Double Space Mode] control word is equivalent to

```
.ls by 2.0 defer nobreak
```

- You can return to single-spacing by entering

```
.ls by 1.0 defer
```

The .SS [Single Space Mode] control word is equivalent to

```
.ls by 1.0 defer nobreak
```

- You can establish fixed line spacing by entering

```
.ls normal p15
```

On page printers and PostScript devices, each line has a depth of 15 pica points, regardless of the size of the characters on the line.

On line devices, the depth of each line is the nearest multiple of the vertical device unit:



- At 6 lines per inch, each line of text occupies one output print line. That is, 6 lines per inch equals 12 points per line; 15 points is rounded to the nearest multiple of 12, giving 1 output line.
- At 8 lines per inch, each line of text occupies two output print lines. That is, 8 lines per inch equals 9 points per line; 15 points is rounded to the nearest multiple of 9, giving 2 output lines.
- You can cause columns to be vertically justified by expanding vertical white space by entering

```
.fv justify
.ls skip 2 space 2 text 1.0
```

The depth of output produced by .SK and .SP can be doubled, if necessary, to make all columns in the section the same depth, but text lines cannot be expanded or compressed at all.

- You can cause columns to be vertically justified by compressing or expanding lines of text by entering

```
.fv justify
.ls text .90 1.33
```

Text lines can be compressed by as much as 10% or expanded by as much as 33%, but this does not happen unless the full adjustment allowed for skips, then spaces, is done, and justification is still not achieved.

- The “granularity” of vertical justification can be controlled with the INCREMENT parameter. Vertical adjustments are made in even multiples of INCREMENT. For example, if you specify

```
.fv justify
.ls skip 2.5 increment p4
```

The values given with .SK can be increased as much as 150%, in increments of 4 pica points, to make all the columns in the section the same depth. The increments must be made in whole numbers of the minimum vertical escapement of the particular output device. As a result, rounding can occur, and this might cause the values and ranges you specified to be exceeded.

- The following example illustrates the use of the PTSIZE and BUILTIN parameters of .LS on page printers. The difference shown in the examples below is very slight because if you make these line spacing changes in a 10 point font, for example, the difference is only 6 pels, or 1/100th of an inch. The lines

```
.cl 2.8i
Now is the time for all good salamanders to play.
Now is the time for all good salamanders to play.
.ls ptsize
Now is the time for all good salamanders to play.
Now is the time for all good salamanders to play.
.ls builtin
Now is the time for all good salamanders to play.
Now is the time for all good salamanders to play.
```

## .LS [Line Spacing]

result in

Now is the time for all good  
salamanders to play. Now is the time  
for all good salamanders to play.

Now is the time for all good  
salamanders to play. Now is the time  
for all good salamanders to play.

Now is the time for all good  
salamanders to play. Now is the time  
for all good salamanders to play.

- In the following example, 2.5 pica points are added to the line spacing for page printers. The value of the EXTRA parameter is added to the line spacing value determined by the font. The lines

```
.cl 2.8i
.df small type(4)
Now is the time for all good snipes to snip away.
Now is the time for all good snipes to snip away.
.ls extra p2.5
Now is the time for all good snipes to snip away.
Now is the time for all good snipes to snip away.
.br
.bf small
Now is the time for all good snipes to snip away.
Now is the time for all good snipes to snip away.
.ls extra
Now is the time for all good snipes to snip away.
Now is the time for all good snipes to snip away.
.pf
.br
Now is the time for all good snipes to snip away.
Now is the time for all good snipes to snip away.
```

result in

Now is the time for all good snipes to  
snip away. Now is the time for all good  
snipes to snip away.

Now is the time for all good snipes to  
snip away. Now is the time for all good  
snipes to snip away.

**Now is the time for all good snipes to  
snip away. Now is the time for all  
good snipes to snip away.**

**Now is the time for all good snipes to  
snip away. Now is the time for all  
good snipes to snip away.**

Now is the time for all good snipes to  
snip away. Now is the time for all good  
snipes to snip away.

- The EXTRA parameter can also be used to reduce the line spacing, as determined by the font, by a specific value. In the following example, the value of the EXTRA parameter is used to reduce the line spacing by 2.7 pica points for page printers. The lines

```
.cl 2.8i
.df big type(14)
Now is the time for all good horses to run away.
Now is the time for all good horses to run away.
.ls extra -p2.7
Now is the time for all good horses to run away.
Now is the time for all good horses to run away.
.br
.bf big
Now is the time for all good horses to run away.
Now is the time for all good horses to run away.
.ls extra
Now is the time for all good horses to run away.
Now is the time for all good horses to run away.
.pf
.br
Now is the time for all good horses to run away.
Now is the time for all good horses to run away.
```

result in

Now is the time for all good horses to  
run away. Now is the time for all good  
horses to run away.  
Now is the time for all good horses to  
run away. Now is the time for all good  
horses to run away.  
**Now is the time for all good horses to  
run away. Now is the time for all  
good horses to run away.**  
**Now is the time for all good horses to  
run away. Now is the time for all  
good horses to run away.**  
Now is the time for all good horses to  
run away. Now is the time for all good  
horses to run away.

- When vertically justifying text, the adjustments of vertical white space is attempted in the following order:
  1. Lead-out points are used (only for expansion).
  2. Skips are adjusted within the range specified on the SKIP parameter of the .LS control word.
  3. Spaces are adjusted within the range specified on the SPACE parameter of the .LS control word.
  4. Text is adjusted within the range specified on the TEXT parameter of the .LS control word.

---

## **.LT [Leading Tab]**

### **Function**

The .LT [Leading Tab] control word is generated by SCRIPT/VS and is executed whenever an input line that starts with a tab is processed.

### **Syntax**

►► .LT ◀◀

### **Notes**

- .LT causes a break.

### **Remarks**

1. A leading tab on an input line causes a break. This is done by generating and executing a .LT control word whenever a line with a leading tab is processed. Such an input line can only be up to 252 characters long. The function of the .LT control word is identical to that of the .BR control word.

If you want to have leading tabs perform some other function, you can define a .LT macro with .DM [Define Macro], and assuming macro substitution is ON, your .LT macro is executed whenever a leading tab is processed. Note, however, that after the .LT control word or macro is processed, the leading tab is still on the line, and it is processed as part of that text input line. In other words, you cannot use the .LT macro to remove leading tabs from a line.

2. No .LT function is performed for lines processed in literal mode (.LI [Literal]).

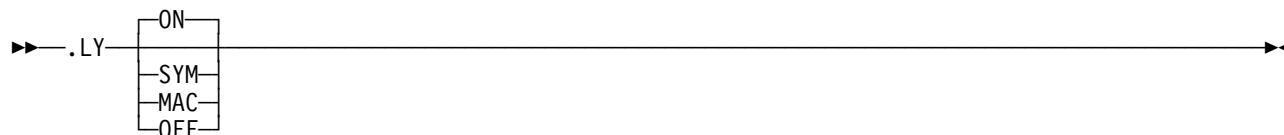
---

## **.LY [Library]**

### **Function**

Use the .LY [Library] control word to cause symbol values and macro definitions to be retrieved from a library defined with the LIB option on the SCRIPT command.

### **Syntax**



### **Parameters**

- ON** Causes unresolved symbol values and macro definitions to be retrieved from a library. This is the default.
- SYM** Causes unresolved symbol values to be retrieved from the library. If SYM is specified, the library is not used to resolve undefined macros from the library (unless MAC or ON was previously specified).
- MAC** Causes unresolved macro definitions to be retrieved from the library. If MAC is specified, no undefined symbol values are resolved from the library (unless SYM or ON was previously specified).
- OFF** Indicates that use of the library for symbol values and macro definitions is to stop. This is the initial setting.

**Initial Setting:** OFF

**Default:** ON

### **Remarks**

1. Use of the library to resolve symbol values and macro definitions is expensive in processing time. This is especially true for forward referencing of symbol values where there are normally many potentially unresolved symbols. For this reason, the .LY control word is provided to control library lookup. If unresolved symbols or macros are *used* in a document, the .LY control word can be used to indicate that SCRIPT/VS is to attempt to retrieve these from a library.
2. Symbol values or macro definitions can be explicitly *set* from the library, regardless of the setting of the .LY control word, using the LIB option of the .SE and .DM control words.

---

## **.MC [Multicolumn Mode]**

### **Function**

The .MC [Multicolumn Mode] control word restores multiple-column processing after it has been temporarily suspended by .SC [Single Column Mode].

### **Syntax**

►► .MC ◀◀

### **Notes**

- .MC ends a keep, float, footnote, named area, or table.
- .MC causes a break and an unconditional section break.

### **Remarks**

1. The .MC control word cancels a temporary single-column mode that was put into effect by the .SC [Single Column Mode] control word. If there was no .SC control word preceding this control word, it has no effect, other than to cause a break.
2. The .SC control word *saves* the current column definition and starts a temporary, single-column processing mode. The column definition that was in effect when .SC saved it might have been either a multiple-column definition or a single-column definition. The .MC control word is, perhaps, misnamed. What .MC actually does is to *restore* the column definition that was saved by .SC, however many columns that definition called for. The column definition saved by .SC and restored by .MC includes the number of columns, their positions, and the column line length. If two .SC control words are processed without an intervening .MC control word, then two .MC control words are needed to restore the original column definition that existed before the first .SC control word. The first .MC control word restores the single-column definition that existed, by virtue of the first .SC control word, when the second .SC control word was processed.
3. The .CD [Column Definition] control word starts an entirely new column definition and cancels any .SC and .MC control words that might be in effect.

---

## **.ME [Macro Exit]**

### **Function**

Use the .ME [Macro Exit] control word to cause SCRIPT/VS to end processing of a macro.

### **Syntax**

►► .ME line ◄◄

### **Parameters**

*line* Any valid SCRIPT/VS input line. If present, it is saved until after the macro is closed and executed in the macro caller's environment.

### **Remarks**

1. The .ME control word has meaning only when the current input source is a macro. If the current input source is not a macro when .ME is encountered, the line is processed as if “line” had been specified instead of “.ME line.”
2. The .EF [End of File] control word, when encountered in a macro, causes that macro and all macros that called it to be closed, up to the parent file that first invoked the top macro in the chain. In other words, the .EF control word ends nested sources until it has ended the current file. The .ME control word, by contrast, ends only the current macro— no other nested input source.

### **Examples**

The .ME control word can be used to set a macro caller's local symbol. For example, suppose macro OUTER calls macro INNER, which contains

```
.me .se *radio = &*clash
```

The .ME ends macro INNER, returns control to macro OUTER, and causes the .SE control word to be executed as if it were part of macro OUTER. The value of &\*clash is taken from macro INNER's local symbol, because symbol substitution is performed before control word execution. But the symbol to which it is assigned is macro OUTER's local symbol &\*radio, because that is where the control word is executed.

---

## .MG [Message]

### Function

The .MG [Message] control word is used to deliver a message. It can be used to provide diagnostic information from macros.

### Syntax

►► .MG / code / message / / ►►

### Parameters

**code** The message identifier. The code parameter must not be longer than 16 characters, and the last character must be R, I, W, E, S, or T. This final letter is used to establish the severity of the message and to set the SCRIPT/VS return code; the same meanings apply as for regular SCRIPT/VS messages:

**R** Response required (RC=0)  
**I** Informational (RC=0)  
**W** Warning (RC=4)  
**E** Error (RC=8)  
**S** Severe error (RC=12)  
**T** Terminating error (RC=16)

If a null code is specified, the message is considered to be an informational message. If the code is **R**, you must provide the terminal read using .RV or .TE for the response; .MG does not do this for you. Message code is not printed unless the MESSAGE(ID) option of the SCRIPT command has been specified.

**message** Text of the message. The *message* parameter can be any string of characters.

**/** An arbitrary delimiter. Any nonblank character that does not appear in the code or the message can be used.

### Remarks

1. Messages generated by .MG affect the return code from SCRIPT/VS, and they can cause SCRIPT/VS processing to terminate. **S** (Severe) and **T** (Terminating) messages always terminate processing, and **E** (Error) messages terminate processing unless the CONTINUE option of the SCRIPT command is in effect.
2. When a message is displayed, a prefix of “+++” appears before the code or message parameter to indicate the message was generated by .MG. If the entire message is null, only this prefix is displayed. This can happen if you do not specify any code or message, or if you specify code and no message, but the MESSAGE(ID) option is not in effect.
3. If the .MG line has no data at all, it is ignored.

### Examples

- To issue an informational message, enter  
    .mg //Your terminal is on.  
The message is displayed as  
+++ Your terminal is on.



- To issue a warning message, enter  
`.mg /W/Your terminal is smoking.`

If you have specified MESSAGE(ID) with the SCRIPT command, the code is included with the message when it is displayed:

`+++W Your terminal is smoking.`

- To issue an error message, enter  
`.mg /001e/Your terminal is on fire.`

If you have specified MESSAGE(ID) with the SCRIPT command, the code is included with the message when it is displayed:

`+++001E Your terminal is on fire.`

If you have not specified the CONTINUE option, SCRIPT/VS terminates its processing.

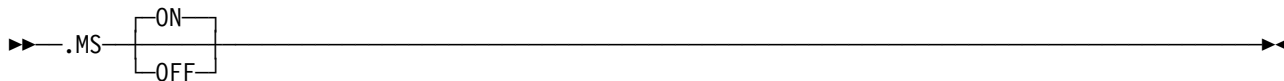
---

## **.MS [Macro Substitution]**

### **Function**

Use the .MS [Macro Substitution] control word to enable or disable macro processing.

### **Syntax**



### **Parameters**

**ON** Initiates a search for macro names during processing.

**OFF** Ends the search for macro names during processing.

**Initial Setting:** OFF

**Default:** ON

### **Remarks**

1. SCRIPT/VS macros can be defined with the .DM control word. However, SCRIPT/VS does not recognize and process macros until macro processing has been started with the .MS control word. When macro substitution is OFF (the initial setting for SCRIPT/VS processing), macros that have been defined with the .DM [Define Macro] control word are treated as invalid control words unless the macro name is the same as a control word name.
2. Even when macro substitution is OFF, a macro can be explicitly invoked using the .EM [Execute Macro] control word.

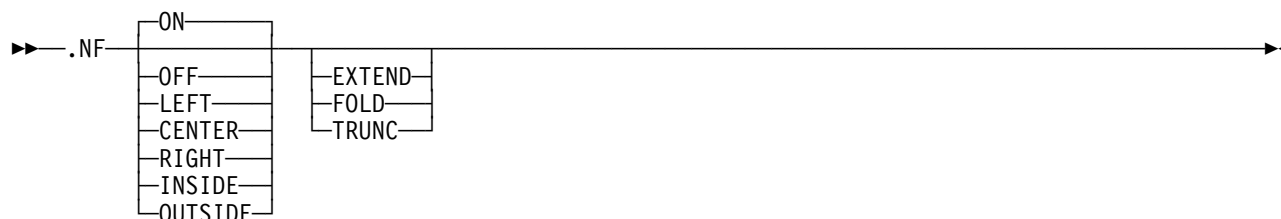
---

## **.NF [No Formatting]**

### **Function**

Use the .NF [No Formatting] control word to cancel or restore concatenation of input lines and justification of output lines. The .NF control word also controls whether lines can be extended beyond the column boundary.

### **Syntax**



### **Parameters**

- ON** Cancels both concatenation of input lines and justification of output lines. Subsequent text is printed as entered.
- OFF** Causes both concatenation of input lines and justification of output lines. Each output line is filled with text and aligned on both the left and right margins.
- LEFT** Specifies that input lines are not to be concatenated but are to be left-aligned in the column. (This is equivalent to .NF ON.)
- CENTER** Specifies that input lines are not to be concatenated but are to be centered in the column.
- RIGHT** Specifies that input lines are not to be concatenated but are to be right-aligned in the column.
- INSIDE** Specifies that input lines are not to be concatenated but are to be aligned against the inside margin of the column—toward the presumed binding edge of the duplexed page. This is equivalent to .NF LEFT for odd pages and .NF RIGHT for even pages.
- OUTSIDE** Specifies that input lines are not to be concatenated but are to be aligned against the outside margin of the column—away from the presumed binding edge of the duplexed page. This is equivalent to .NF RIGHT for odd pages and .NF LEFT for even pages.

When concatenation is not in effect and an input line exceeds the column line length, or when concatenation is in effect and a single word exceeds the width of the column, the text is processed according to the current overdraw option:

- EXTEND** Specifies that if the text does not fit in the column, it is to be extended beyond the column line length. This is the initial setting.
- FOLD** Specifies that if the text does not fit in the column, it is to be broken and the remainder is to be placed on the next output lines. The text is broken at the last character that fits in the column.
- TRUNC** Specifies that if the text does not fit in the column, it is to be truncated at the last character that fits in the column.

**Initial Setting:** OFF EXTEND

**Default:** ON

## Notes

- .NF causes a break.
- The formatting mode is included in the active environment.

## Remarks

1. The .NF control word determines whether words are shifted from one input line to another to fill each output line (concatenation), and how any extra horizontal white space left at the end of the output line is distributed between the words on the line (justification).
2. The overdraw options TRUNC, FOLD, and EXTEND can be specified as the only options of the .NF control word. In this case, the current formatting mode is unchanged, although a break is done.
3. The .FO control word is similar to .NF; both control words affect concatenation and justification, and both control words can set the overdraw option.
4. Text that exceeds the column line length is processed according to the current overdraw option, as established with .NF or with the .FO [Format Mode] control word. When concatenation is not in effect, this can happen when an input line is wider than the column line length. When concatenation is in effect, this can happen when a single word is wider than the column line length, or when a word following a tab crosses the right margin.
5. Options can be specified in any sequence. If contradictory options are specified, the latest one is used.
6. When you use the EXTEND overdraw option, you must ensure that the width of extended output lines does not exceed the width of the page.

## Examples

- .nf off  
But four young Oysters hurried up, All eager for the treat: Their coats were brushed, their faces washed, Their shoes were clean and neat--And this was odd, because, you know, They hadn't any feet.
- .nf on  
Four other Oysters followed them,  
And yet another four;  
And thick and fast they came at last,  
And more, and more, and more--  
All hopping through the frothy waves,  
And scrambling to the shore.
- .nf left  
The Walrus and the Carpenter  
Walked on a mile or so,  
And then they rested on a rock  
Conveniently low:  
And all the little Oysters stood  
And waited in a row.
- .nf center

“The time has come,” the Walrus said,  
“To talk of many things:  
Of shoes--and ships--and sealing-wax--  
Of cabbages--and kings--  
And why the sea is boiling hot--  
And whether pigs have wings.”

- .nf right

“But wait a bit,” the Oysters cried.  
“Before we have our chat;  
For some of us are out of breath,  
And all of us are fat!”  
“No hurry!” said the Carpenter.  
They thanked him much for that.

- .nf outside

“A loaf of bread,” the Walrus said,  
“Is what we chiefly need:  
Pepper and vinegar besides  
Are very good indeed--  
Now, if you're ready, Oysters dear,  
We can begin to feed.”

- .nf inside

“But not on us!” the Oysters cried,  
Turning a little blue.  
“After such kindness, that would be  
A dismal thing to do!”  
“The night is fine,” the Walrus said.  
“Do you admire the view?”

- This poem is concluded under “.RC [Revision Code ]” on page 301.

---

## **.NL [Null Line]**

### **Function**

The .NL [Null Line] control word is generated by SCRIPT/VS and executed whenever a null line is processed.

### **Syntax**

►► .NL ◀◀

### **Notes**

- .NL resets line continuation.

### **Remarks**

1. Whenever SCRIPT/VS encounters a null input line, that is, a line whose length is zero, it generates and executes a .NL control word. The .NL control word does nothing, except to reset line continuation if the previous line ended with a continuation character.

If you want to have null lines perform some other function, you can define a .NL macro with .DM [Define Macro] and, assuming macro substitution is ON, your .NL macro is executed whenever a null line is processed.

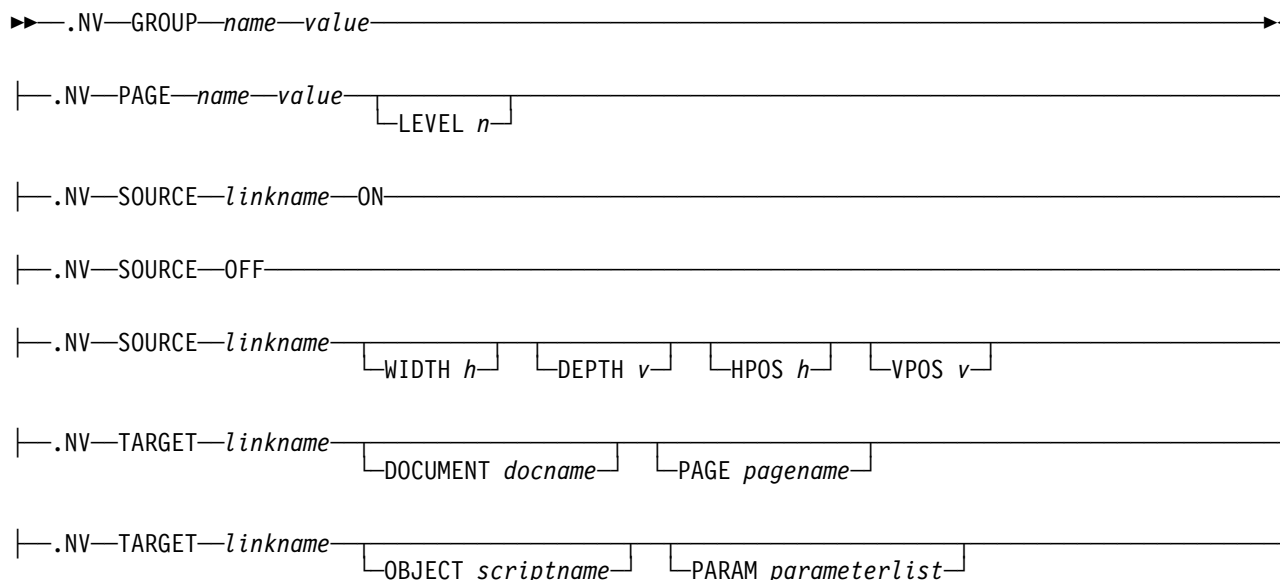
2. No .NL function is performed for lines processed in literal mode (.LI [Literal]).
3. A null line can originate from a number of sources. Because of this, you should define a .NL macro only when a specific use in a certain part of a document requires it. Null lines might originate from:
  - A source input file
  - Terminal input (.TE)
  - A nonnull line that becomes null as a result of substitution
  - A macro line that is null.
4. A null line is not the same as a blank line. Blank lines contain at least one character and are processed by the .BL [Blank Line] control word.

## .NV [Navigate]

### Function

Use the .NV [Navigate] control word to define group and page level element/value pairs and links between sources and targets to aid you in navigating through a document and between documents with the IBM AFP Workbench for OS/2 and Windows product. The source and target of a hypertext link can also be defined.

### Syntax



### Parameters

|              |                                                                                                                                                                                                                                                                                                |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GROUP</b> | Specifies that this <i>name</i> and <i>value</i> pair is attached to the current group and is a selectable attribute using the <b>Select Group</b> function of AFP Workbench for OS/2 and Windows. A group must be active (started with the .GR [Group] control word) or else an error occurs. |
| <b>PAGE</b>  | Specifies that this <i>name</i> and <i>value</i> pair is attached to the current page and is selectable from the <b>Go To</b> function of AFP Workbench for OS/2 and Windows.                                                                                                                  |
| <i>name</i>  | Can be up to 64 characters. If the name includes blanks, single quotes, or both, the whole name <b>must</b> be inclosed in single quotes. To use a single quote in the name, enter it as two consecutive single quotes.                                                                        |
| <i>value</i> | Can be up to 64 characters. If the value includes blanks, single quotes, or both, the whole value <b>must</b> be inclosed in single quotes. To use a single quote in the value, enter it as two consecutive single quotes.                                                                     |
| <b>LEVEL</b> | Defines the level of this navigation element. AFP Workbench for OS/2 and Windows uses the level of a navigation element to include appropriate left indentation when displaying the navigation element. LEVEL is only valid when used with the PAGE parameter.                                 |
| <i>n</i>     | Specifies the level of the navigation element. A number from 0 to 99999999 can be used.                                                                                                                                                                                                        |

## .NV [Navigate]

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SOURCE</b>   | Specifies the source of the link. A target for this link with the same <i>linkname</i> must be specified somewhere in the same document. <i>linkname</i> can be up to 16 national characters and is not case sensitive.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>ON</b>       | Specifies the start of a link source. Use the OFF parameter to indicate the end of the source. A source started with the ON parameter must be ended by another .NV SOURCE control word with the OFF parameter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>OFF</b>      | Specifies the end of the source that was started with the ON parameter. If the OFF parameter is used, but no source is active, the control word is ignored.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>WIDTH</b>    | Specifies the width of the rectangle for the source. Any valid space unit notation can be used. The WIDTH, DEPTH, VPOS, and HPOS parameters are mutually exclusive with the ON and OFF parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>DEPTH</b>    | Specifies the depth of the rectangle for the source. Any valid space unit notation can be used. The DEPTH, WIDTH, VPOS, and HPOS parameters are mutually exclusive with the ON and OFF parameters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>HPOS</b>     | Specifies the horizontal position of the upper left corner of the source. Any valid space unit notation can be used. If a relative value (+ <i>h</i> or - <i>h</i> ) is specified, it is added to or subtracted from the current horizontal position and the resultant value establishes the left side of the source rectangle. If an absolute value is specified, that value establishes the left side of the source rectangle, relative to the upper left corner of the page. If ON or OFF are not specified, and if HPOS is not specified, the current horizontal position is used. The HPOS, VPOS, DEPTH, and WIDTH parameters are mutually exclusive with the ON and OFF parameters.                                                                                                                                                                                             |
| <b>VPOS</b>     | Specifies the vertical position of the upper left corner of the source. Any valid space unit notation can be used. If a relative value (+ <i>v</i> or - <i>v</i> ) is specified, it is added to or subtracted from the vertical position of the top of the current output line and the resultant value establishes the top of the SOURCE rectangle. If an absolute value is specified, that value establishes the top of the source rectangle relative to the upper left corner of the page. If the ON, OFF, and VPOS parameters are not specified, the current vertical position is used. The VPOS, HPOS, DEPTH, and WIDTH parameters are mutually exclusive with the ON and OFF parameters.                                                                                                                                                                                         |
| <b>TARGET</b>   | Specifies the target of the link. If a source with the same <i>linkname</i> is not found, this target is ignored. The <i>linkname</i> can be up to 16 national characters and is not case sensitive.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>DOCUMENT</b> | Specifies the document which contains the target. If not specified, the current document is assumed to contain the target. <i>docname</i> can be up to 64 characters. If <i>docname</i> includes blanks, single quotes, or both, it <b>must</b> be enclosed in single quotes. To use a single quote in the <i>docname</i> , enter it as two consecutive single quotes. <i>docname</i> should be the external name of the document to which you are referring. This corresponds to the name on the BDT structured field. In the VM environment the name on the BDT structured field is usually the same as the filename of the file; in MVS it is usually the member name. If neither DOCUMENT nor PAGE is specified, the target is placed on the current page. If DOCUMENT is specified, and PAGE is not specified, the target is placed on the first page of the specified document. |
| <b>PAGE</b>     | Specifies the name of the page on which the target is placed. Use this keyword to place the target on a specific page in the current document or on a specific page in the document named on the DOCUMENT keyword. If neither PAGE nor DOCUMENT is specified, the target is placed on the current page. If PAGE is not specified and DOCUMENT is specified, the target is placed on the first page of the document specified.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |



- OBJECT** Specifies this target is an external object with a name of *scriptname*. *scriptname* is the name specified on the .DO [Define Object] control word for the object. The object must be defined on a .DO control word before it can become the target of a navigation link.
- PARAM** Specifies the parameter list to be sent to the object. *parameterlist* can be up to 128 characters. If *parameterlist* includes blanks, single quotes, or both it **must** be enclosed in single quotes. To use a single quote in the *parameterlist*, enter it as two consecutive single quotes.

## Notes

- .NV starts a page and a line.
- .NV is only valid for true AFP devices (physical devices 3820, 4028, and AFP). It is ignored for all other device types (including the 3800-3).

## Remarks

1. An error occurs if .NV GROUP is specified but a group has not been started with the .GR control word.
2. The characters used in *name* are displayed in AFP Workbench for OS/2 and Windows using the codepage appropriate for the default language of SCRIPT/VS. See the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for information on changing the default language.
3. Many instances of the same page-level navigation element and value pair are allowed and are navigable by AFP Workbench for OS/2 and Windows.
4. Multiple instances of the same group-level navigation element and value pairs in the same group are allowed, but are redundant because navigation in AFP Workbench for OS/2 and Windows using group-level navigation elements only jumps to the first page in the group.
5. A possible use of the LEVEL parameter would be to distinguish between different navigation elements for headings. For example, page-level navigation elements created for level 1 headings could specify LEVEL 1, and those for level 2 headings could specify LEVEL 2, and so on. When displayed in AFP Workbench for OS/2 and Windows, the different levels cause corresponding levels of indentation making it easier to determine the level of the heading.
6. If you overlap the source of two links that have different targets, the presentation system may not be able to determine which target to link to when you request a jump to one of the overlapping sources.
7. For each link both a source and a target must be defined. Links are only placed into the output file if both a source and target with the same *linkname* are available. If the definition of the target occurs after the definition of the source, multiple formatting passes are required.
8. When defining a source you can either specify an origin and size of the source rectangle with the HPOS, VPOS, WIDTH, and DEPTH parameters, or you can establish the origin for the source with the ON parameter and later end the SOURCE with the OFF parameter on another .NV SOURCE control word.  
  
If the ON and OFF parameters are not used, the SOURCE is a rectangle with an origin and a size. In this case, the WIDTH and DEPTH parameters must be specified to give the size of the rectangle. If the HPOS parameter is not used to specify the horizontal origin of the rectangle, the current horizontal position is used (just as if HPOS +0 was specified). If the VPOS parameter was not used to specify the vertical origin of the rectangle, the top of the current outline line is used (just as if VPOS +0 was specified).  
  
If the ON or OFF parameter is used on the same input line with WIDTH, DEPTH, HPOS, and VPOS, the first one specified is used.

## .NV [Navigate]

9. If the ON parameter is used to start a source, the source is started at the current horizontal position and the top of the current output line. The source must be ended with another .NV SOURCE control word using the OFF parameter.
10. When using implicit linespacing, all output lines contain extra white space at the top of the output line to prevent the top of this line from overprinting the bottom of the previous output line. This extra vertical space is called the leading and is the difference between the current line spacing value (based on the current font) and the height of an em space in the current font. If you don't want this extra white space included in your source, you must explicitly set your linespacing such that it doesn't include this space.
11. If a source created with ON and OFF extends to the end of at least one outline line, it extends all the way to the right of the current line length value for the current section. If your source needs to span multiple output lines and needs to be narrower than the current column line length, you can place the source inside a section area as wide as your source.
12. There are three different types of navigational targets. If DOCUMENT is used, the link is an interdocument link, linking a source from the current document to a target in a different document. If DOCUMENT is not used, the link is an intradocument link, linking the source from the current document to a target on another page in the same document. If OBJECT is used, the link is an executable link, linking the source in the current document to an executable object. The object must be defined using the EXECUTE parameter on the .DO control word.
13. If a .AM [Adjust Margin] control word value is different at the end of a source created with the ON and OFF parameters than it was at the beginning, the width of the source may be wider or narrower than anticipated. To avoid this, change your .AM value before or after the lines that include navigational sources, but not within those lines.
14. When creating a target using the PAGE parameter, make sure the specified page name is a unique page name (specified with the .PN [Page Numbering Mode] control word and automatic incrementation of page numbers). If there are multiple pages with the same name, it is unpredictable which of those pages the presentation system uses when linking to the target.
15. If you start a source with ON inside a keep and end the source outside the keep, you may get unexpected output if the keep moves to the next page or column. The same is true if the source is started with the ON parameter outside the keep, but ended inside the keep.
16. When you view your document with navigational links at your presentation system, it may seem that the source area doesn't accurately correspond to the portion of the page you intended. This is probably due to the degree of fidelity you are using with your presentation system or because the fonts your presentation system is using are different from the fonts SCRIPT/VS used to format the document.
17. If you specify either HPOS or VPOS as a relative value and VPOS or HPOS as an absolute value respectively, the relative value is calculated based on the current position, but the value is treated as an absolute number, based on the upper left corner of the page, not on the current position. If a relative value is used and the source is in a rotated object, the source origin is relative to the rotated object's origin. An absolute value establishes the origin relative to the page origin not the object origin.
18. When using SEPMASTR to create separations masters, navigation sources on a particular page are placed on all masters for that page.
19. If a large number of .NV control words are placed in the same output line, it is possible to fill up the output buffer and receive a severe error message: **DSMLNB550S TEXT EXCEEDS BUFFER SIZE**. To remedy this situation, rearrange your .NV control words so that they are spread out among multiple output lines.

## Examples

- The following example shows an interdocument navigational link:

```
See heading reference
.nv source ref on
"Here We Go Again:"
.nv source off
in Appendix A
...
.h1 Appendix A
...
.h2 Here We Go Again
.nv target ref
```

- The following example shows an intradocument navigational link:

```
For more information, read the description of aardwolf in
.nv source external on
Extinct Animals of Australia
.nv source off
.nv target external document tecno page 10
```

- The following example shows a link to an executable object:

```
The
.do mplayer obname mplayer execute
.nv target croc object mplayer param 'play croc.avi'
.nv source croc on
crocodile
.nv source off
can move very fast in water to capture its prey.
```

- To make a source the same size as a certain image included with the .IO [Include Object] control word, do the following:

```
.do b obname button type tiff
.da area 0 0 section width 2i
.ar area on
.ls normal 2i
.nv source button on
.io b width 2i depth 2i fill
.nv source button off
.ar off
.ar put
...
.nv target button
...
```

The FILL parameter is used on the .IO [Include Object] control word to cause the image to exactly fill the 2 inch by 2 inch object block defined with the WIDTH and DEPTH parameters. The width of the section area is defined as 2 inches, the same as the width specified with the .IO control word. The .LS [Line Spacing] control word is used to cause the output line containing the image to be the same depth as the image. No extra leading is added at the top of the outline line because the .LS control word was used to explicitly set the line spacing. If the line spacing is not explicitly set, the extra leading included with all output lines is included in the navigational source as well.

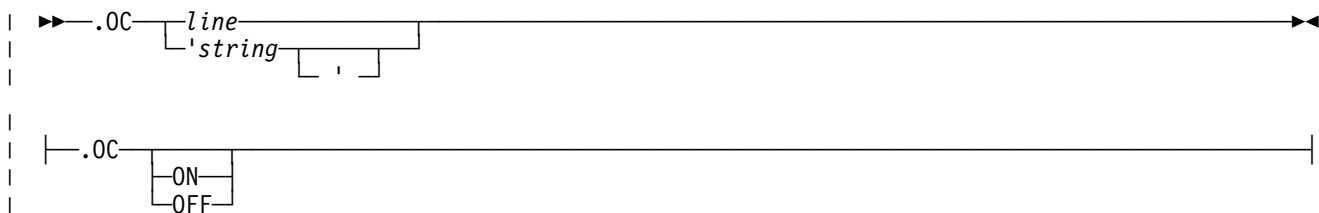
---

## .OC [Output Comment]

### Function

Use the .OC [Output Comment] control word to place comments and carriage control characters in the output data stream. Such comments are not examined by the formatter. This control word is designed for the systems programming user of SCRIPT/VS and must be used with caution.

### Syntax



### Parameters

**line** For line mode, can be anything, because it is not used in formatting the output. However, because this is a control word, the input line is scanned for control word separators. The *line* given is not preceded by a carriage control character when output is being directed to a printer. The first character in the given *line* is taken as a carriage control character, and you must ensure that a valid carriage control character for the output device is provided.

Output comments of the line form are not synchronized with the formatted text; they are written to the output destination immediately before the page on which the text surrounding them appears.

**string** Anything delimited by single quotation marks. However, because this is a control word, the input line is scanned for control word separators. The *string* is placed in the output exactly as given, in the same place relative to the input text that it was specified in the input. For this reason you should be extremely careful to control the contents of the *string*.

The width of output comments is considered to be zero. The depth of output comments is considered to be equal to the normal line spacing for the device.

**ON** All subsequent input lines are interpreted as an output comment until the next .OC control word is encountered.

**OFF** Stops interpreting input lines as output comments.

### Notes

- The ON and OFF parameters are only recognized as parameters for AFP devices (physical devices 3820, 4028, and AFP). These parameters are recognized as the line form of .OC for all other devices.
- .OC is ignored for 4250 printers.
- .OC takes effect on the current page.
- The string form of .OC starts the page.

## Remarks

1. The .OC control word allows comments to be placed in the output data stream. They are not examined by the formatter and thus, unless they are correctly interpreted by the output device, the output is disrupted.
2. The string form of output comments is processed with front continuation on. That is, concatenation can occur even though, for example, .FO OFF is in effect. If this concatenation is undesirable, you can avoid it by inserting the .BR [Break] control word around your output comments.
3. If a line or string form of the .OC control word line, including the .OC control word and any intervening blanks, is longer than 124 characters, an error results.
4. For PostScript printers, all occurrences of the .OC control word are processed as the string form. The comment line is preceded by a percent sign (%).
5. For AFP printers, all occurrences of .OC are processed as the line form. All occurrences of .OC that are found in the input before the page is ended are placed in the data stream in the order that they were entered. The comments are placed as NOP structured fields immediately preceding the Begin Page Structured Field.  
  
For AFP printers, output comments occurring after the last page of a document are placed as NOP structured fields immediately preceding the End Document structured field. A document that is empty (except for output comments) does not produce an AFPDS output file. An output comment that appears on a page that is not printed due to the PAGE command option does not appear in the AFPDS output file.
6. For the line and string forms of the .OC control word, the text of the output comment is assumed to start in the 4th position of the control word line. Therefore, any blanks starting in column 4 of the input line are treated as part of the comment text.
7. If symbol substitution is active, symbols within output comments are substituted. Input translation is done for all output comments, but output translation is not. If separation masters are being generated, an output comment on a particular page appears in all masters for that page.
8. When formatting for AFP printers the ON and OFF parameters can be used to cause multiple input lines to be combined into a single output comment and placed into one NOP structured field. Intervening blanks are not added between the input lines. If you want blanks between the input lines, you must add them at the beginning of the input lines. If the resulting output comment is longer than 8183 characters, it is split into multiple output comments with the second output comment starting with the input line that causes the total number of characters to exceed the 8183-character limit.

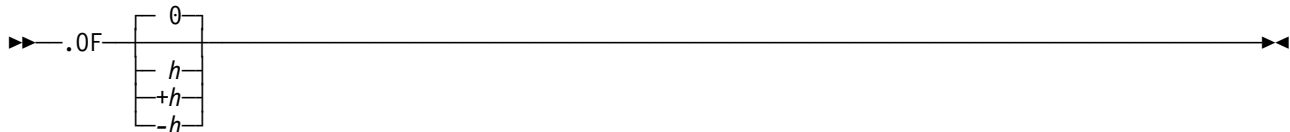
---

## .OF [Offset]

### Function

Use the .OF [Offset] control word to indent all but the first line of a block of text.

### Syntax



### Parameters

*h* Specifies the amount of horizontal space to offset subsequent output lines. If you specify *+h* or *-h*, the old offset value is incremented or decremented the specified amount to establish the new offset size. If *h* is omitted, the new offset size is 0.

The next output line to be formatted after the .OF control word has been processed is formatted at the left margin established by the .IN [Indent] control word, with no added offset. For all subsequent lines, the left margin is established by adding the offset (.OF) to the size of the indent (.IN).

**Initial Setting:** 0

**Default:** 0

### Notes

- .OF causes a break.
- Offsetting lines is included in the active environment.

### Remarks

1. The .OF control word does not take effect until after the next line is formatted. The offset remains in effect until a .IN [Indent] control word or another .OF control word is encountered.  
The .OF control word can be used within a section that is also indented with the .IN control word. Note that .IN settings take precedence over .OF, however, and any .IN request clears all offsets.
2. The .IL [Indent Line] and the .UN [Undent] control words can be used to shift only the next line to the left or right of the current margin.
3. An attempt to set the indentation to the left of the real left margin or to the right of the real right margin results in an error message, and all indentation is reset to zero.
4. The .OF control word is triggered by the next text, skip, or space line.
5. The value of *h* represents the amount of blank space just before the text. Thus, .OF .5i sets a left margin of one-half inch, and the text begins *after* this blank margin area.

## Examples

- This example uses the .OF control word to specify an offset of 7:

.of 7

The line immediately following the .OF control word is printed at the current left margin.

The following lines are indented seven spaces from the current margin setting.

This offset setting remains effective until another indentation or offset specification is encountered.

results in

The line immediately following the .OF control word is printed at the current left margin. The following lines are indented seven spaces from the current margin setting. This offset setting remains effective until another indentation or offset specification is encountered.

- This example shows the use of the .OF control word to end the offset currently in effect. Specifying

.of

The effect of any previous .OF request is canceled, and all output after the next line continues at the current left margin setting.

results in

The effect of any previous .OF request is canceled, and all output after the next line continues at the current left margin setting.

# .OI [Overlay Include]

## Function

Use the .OI control word to specify and position an overlay to be included on the page. For device types that do not support overlays, this control word can be used to reserve space for cut-and-paste artwork.

.OI [Overlay Include] is used for one of two purposes:

- Including an overlay and reserving space for it
- Positioning an overlay on a page without reserving space.

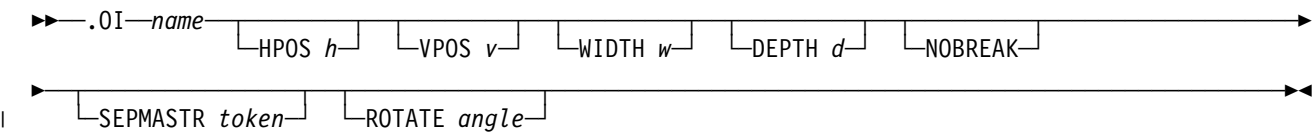
The physical device types that support overlays are 3820, 4224, 4028, and AFP.

The WIDTH and DEPTH parameters can be used to reserve space for an overlay to be included at the current print position.

The HPOS and VPOS parameters can be used to position an overlay at a specific location on a page. If HPOS or VPOS are specified, WIDTH and DEPTH are ignored.

If HPOS and VPOS or WIDTH and DEPTH are not specified, an overlay is included at the current print position with no space reserved.

## Syntax



## Parameters

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>name</i>          | <p>The <i>name</i> of the overlay to be placed in the document. The <i>name</i> can be up to 8 characters. If the <i>name</i> is more than 8 characters long, it is truncated.</p> <p>No checking is done by SCRIPT/VS to ensure that the specified overlay exists. PSF checks to see if the overlay exists when the document is printed by checking the following:</p> <ul style="list-style-type: none"><li>• In CMS, the default filetype for overlays is OVLY38PP.</li><li>• In MVS, the overlay name must be a member of the partitioned data set defined as the overlay library.</li></ul> |
| <b>HPOS</b> <i>h</i> | <p>Horizontal position in any valid space units for the top left corner of the overlay relative to the physical page origin. If the overlay is not rotated, the origin is the top left corner of the overlay on the physical page.</p> <p>If HPOS is not specified, but VPOS is, the default is zero. If neither HPOS nor VPOS is specified, the default is the current print position on the logical page.</p>                                                                                                                                                                                  |
| <b>VPOS</b> <i>v</i> | <p>Vertical position in any valid space units for the origin of the overlay relative to the physical page origin. If the overlay is not rotated, the origin is the top left corner of the overlay on the physical page.</p> <p>If VPOS is not specified, but HPOS is, the default is zero. If neither VPOS nor HPOS is specified, the default is the current print position on the logical page.</p>                                                                                                                                                                                             |



**WIDTH** *w* The amount of horizontal space in any valid space units that SCRIPT/VS reserves for the overlay. If WIDTH is not specified, but DEPTH is, the default is the current column line length minus the current left and right indentions. If neither WIDTH nor DEPTH is specified, the default is zero.

**DEPTH** *d* The amount of vertical space in any valid space units that SCRIPT/VS reserves for the overlay. If DEPTH is not specified, the default is zero, whether WIDTH is specified or not.

**NOBREAK** Indicates that a line break should not occur either before or after the overlay. A break occurs if NOBREAK is not specified.

**SEPMASSTR** Specifies that overlays are being associated with output separation masters.

The *token* is the 1- to 8-character name that identifies which separation masters this overlay should be associated with. The *token* name corresponds to the items selected for separation masters with the .SM control word. You cannot specify OLD COLOR as a *token* name.

The special *token* value of ALL is used to indicate that this overlay should be contained in all separation masters, including the default.

For more information on the SEPMASSTR parameter, see “.SM [Separation Master]” on page 353.

**ROTATE** Specifies that the overlay has a specific rotation relative to the physical page when printed. The rotation value is given in degrees, and must be one of the following multiples of 90 degrees:

–0  
–90  
–180  
–270  
0  
90  
180  
270

If ROTATE is not specified, the rotation is zero degrees.

If HPOS and VPOS are used to give a location on the physical page, the overlay rotates around that point. If HPOS and VPOS are not used, and WIDTH and DEPTH are not used, then rotation is around the current print position.

If WIDTH and DEPTH are specified with an overlay rotated either 90 or 270 degrees, the width and depth values are automatically swapped to maintain their correspondence to the dimensions of the overlay. The rotated overlay is placed inside the reserved space with the origin at the appropriate corner of the space.

## Notes

- .OI ensures that the page is started.
- .OI causes a break unless NOBREAK is specified.

## Remarks

1. Overlays included with .OI [Overlay Include] are included only on the current page.
2. The parameters on the .OI control word and their abbreviations cannot be used as the overlay name.
3. SCRIPT/VS does not attempt to verify that the overlay fits on the page at the specified location. If you attempt to place an overlay that is too large or too close to an edge of the page, the overlay extends off the page, and this might cause PSF errors depending on how PSF is set up.
4. If WIDTH and DEPTH are specified, SCRIPT/VS does not check to see that the space reserved is adequate to accommodate the overlay.
5. DCF issues a warning message if more than 127 unique overlays are encountered on any one page and ignores any subsequent .OI [Overlay Include] with new names.
6. Whether an overlay prints as expected depends on the contents of the overlay, the physical device type, and the operating environment.
7. If you are printing on a 4224 Printer and the overlay you are including contains shading, raster data, or a page segment created for a 3820, the image expands in length and width by 66%.
8. The SEPMASTR parameter is ignored for line devices.
9. The SEPMASTR parameter is ignored if the SEPMASTR command option is not specified.

**Note:** Examples using the SEPMASTR parameter can be found in the *Document Composition Facility: Text Programmer's Guide*.

10. Overlay rotation is clockwise relative to the physical page and is **not** affected by the rotation of the text, areas, or tables used around it.
11. Overlay rotation is optional. If it is not specified, the rotation is zero.
12. Rotation is independent of the other keywords.  
If you specify HPOS and VPOS, the overlay will rotate around the point given. If HPOS and VPOS is not specified, the overlay will rotate around the current point in the text. If WIDTH and DEPTH are specified to reserve a space for the overlay, the current position will be shifted so that the orientation of the overlay tends to put it into that reserved space.

## Examples

In the following examples, the pages have been reduced so that rules could be shown representing the page origin and coordinates for placing the overlays.

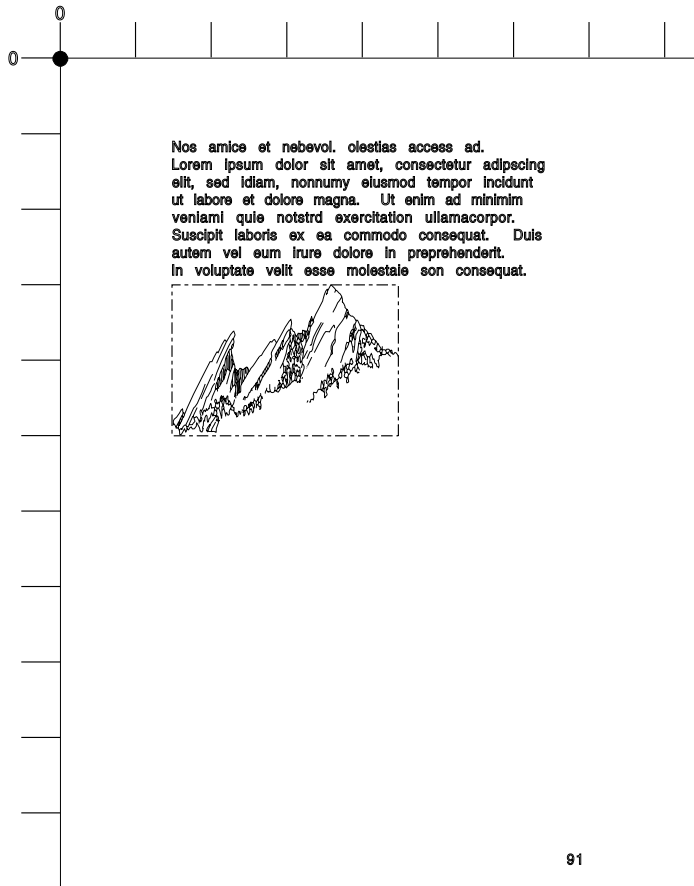
- **Positioning Overlays**

You can use the HPOS and VPOS parameters to position overlays on a page.

- The following overlay example shows what happens when you do not specify any parameters on the .OI [Overlay Include] control word.

```
.oi o1flat
```

The overlay “o1flat” is positioned at the current print position relative to the logical page origin. Because no parameters were specified, the overlay overlays any text that follows the current line.

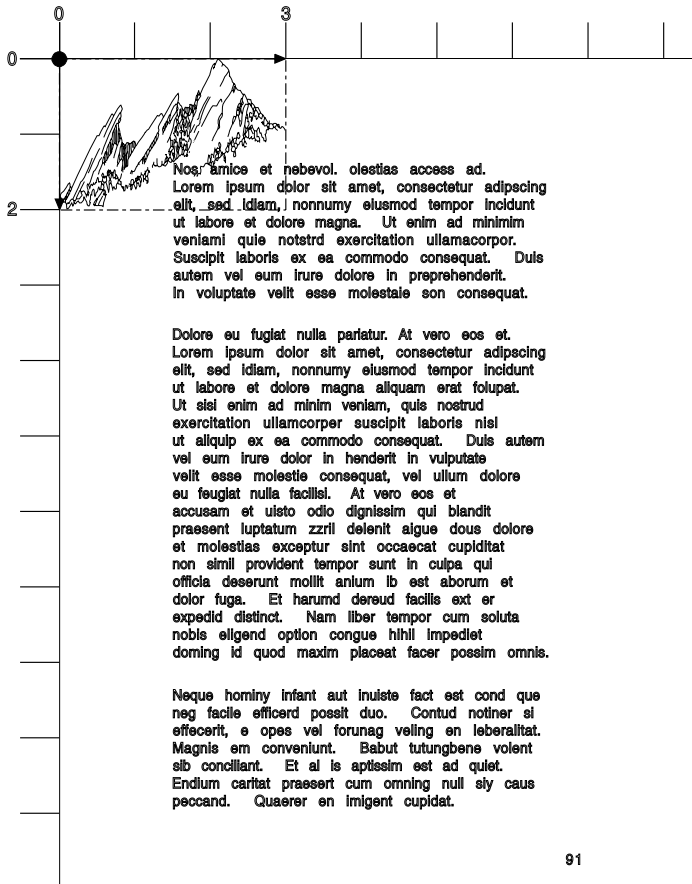


## .OI [Overlay Include]

- The following overlay example is positioned on the logical page at coordinates HPOS 0, VPOS 0.  
`.oi o1flat hpos 0 vpos 0`

Because the overlay “o1flat” is actually 2 inches by 3 inches in size and no space is reserved, it overlays the text.

If HPOS and VPOS values are not specified, it defaults to the current print position. You must specify HPOS and VPOS to place an overlay at the logical page origin.

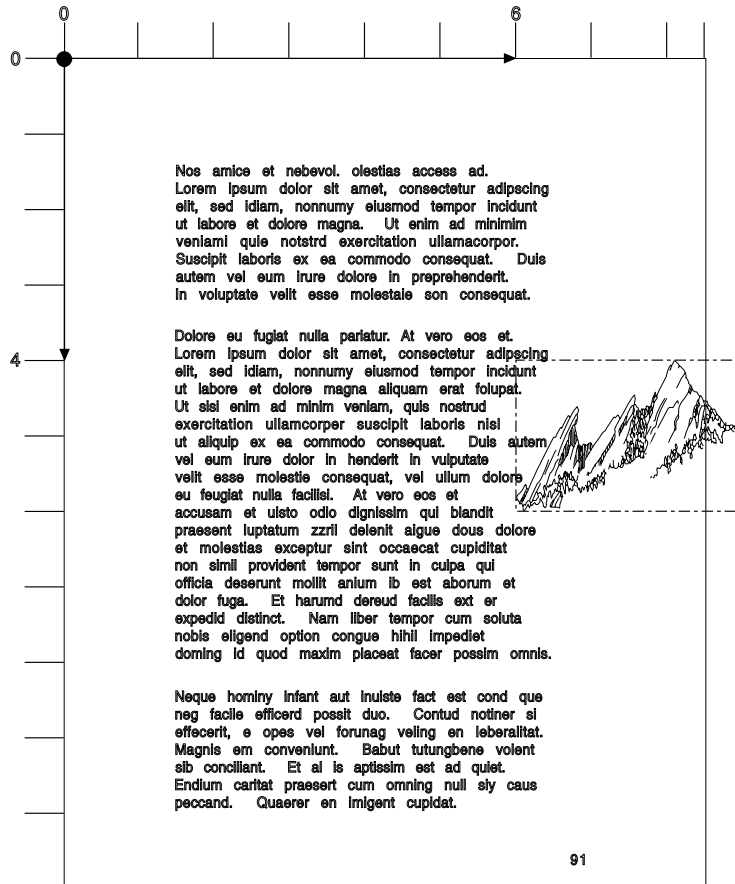


- The following overlay example is positioned four inches horizontally and six inches vertically from the logical page origin.

```
.oi o1flat hpos 6i vpos 4i nobreak
```

The overlay “o1flat” overlays the text and extends off the page.

This example might cause PSF errors, depending on the type of printer and the operating environment.

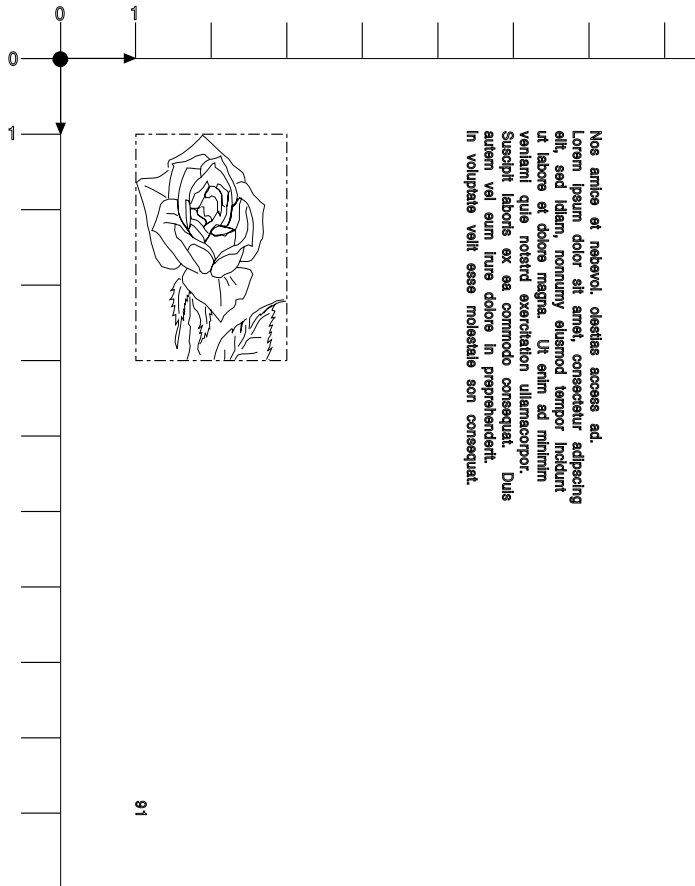


## .OI [Overlay Include]

- The following overlay example is positioned 1 inch horizontally and 1 inch vertically from the physical page origin, which is the top left corner of the physical page. The overlay was included on a rotated page.

```
.oi olrose hpos 1i vpos 1i
```

Because the overlay “olrose” was included on a rotated page, it is placed relative to the physical page origin, and the text is placed relative to the logical page origin. Notice that the overlay did not rotate.



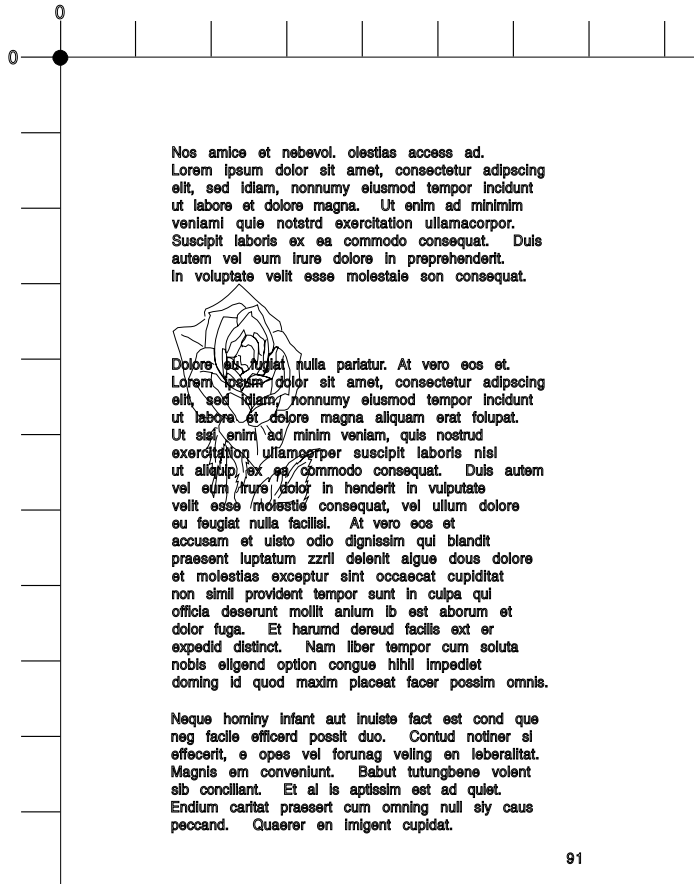
- **Reserving Space for Overlays**

You can use the WIDTH and DEPTH parameters to reserve space for overlays.

- The following overlay example reserves a 1 inch vertical space at the current print position.

```
.oi o1rose depth 1i
```

The overlay “o1rose” is larger than the space reserved for it, and overlays the text.



## .OI [Overlay Include]

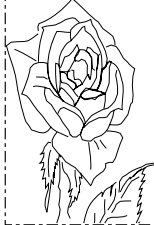
- The following overlay example reserves a space measuring two inches by three inches.

```
.oi olrose width 2i depth 3i nobreak
```

The requested width exceeds the space remaining on the current line. Therefore, space is reserved starting on the next line (a break occurs). Because NOBREAK is specified, a break does not occur after the overlay is placed, and the text continues on the current line.

Duis autem vel eum irure dolore

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed idiam, nonummy eiusmod tempor incididunt ut labore et dolore magna. Ut enim ad minimim veniam quise nostrud exercitation ullamcorpor. Suscipit laboris ex ea commodo consequat. Duis autem vel eum irure dolore in reprehenderit. In voluptate velit esse molestiae non consequat.



per suscipit laboris nisi ut aliquip ex ea commodo consequat. Duis autem vel eum irure dolor in hendrerit in vulputate velit esse molestiae consequat, vel ullam dolore eu feugiat nulla facilisi. At vero eos et accusam et uisto odio dignissim qui blandit praesent luptatum zzril delenit augue dous dolore et molestias exceptur sint occaecat cupiditat non simil provident tempor sunt in culpa qui officia deserunt mollit animum ib est aborum et dolor fuga. Et harum doreud facilis ext er expedit distinct. Nam liber tempor cum soluta nobis eligend option congue hihil impedit doming id quod maxim placeat facer possim omnis voluptas assumenda est, omnis repellend. Temporibus auteui quibusd et aur office.



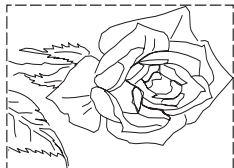
- The following overlay example is rotated 90° and reserves a space measuring two inches by three inches.

.oi olrose width 2i depth 3i nobreak rotate 90

---

#### Duis autem vel eum irure dolore

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed idiam, nonnumy eiusmod tempor incididunt ut labore et dolore magna. Ut enim ad minimim veniam quie notstrd exercitation ullamacorpor. Suscipit laboris ex ea commodo consequat. Duis autem vel eum irure dolore in reprehenderit. In voluptate velit esse molestiae son consequat.



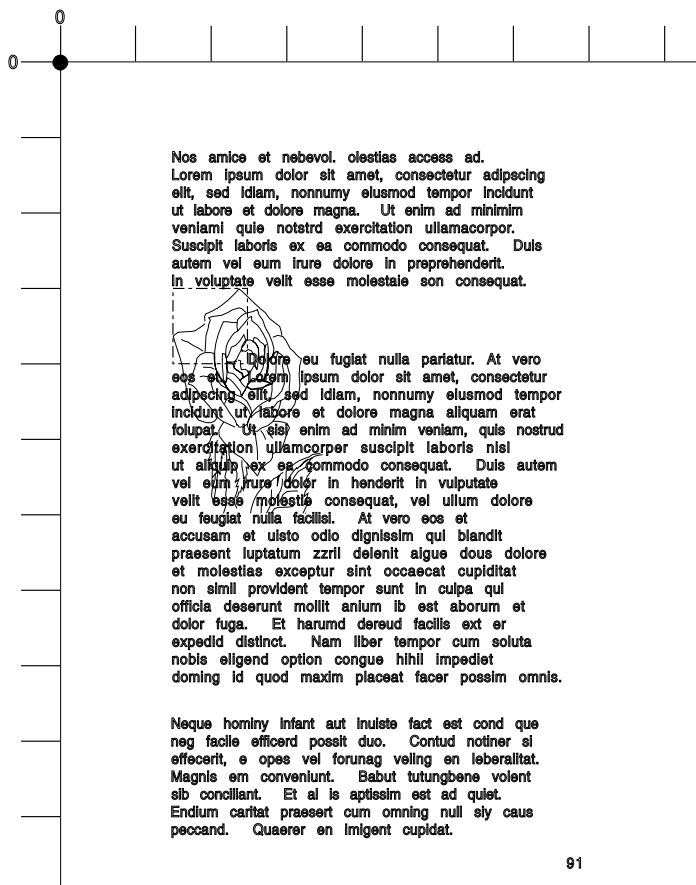
per suscipit laboris nisi  
ut aliquip ex ea commodo consequat. Duis autem vel eum irure dolor in hendert in vulputate velit esse molestie consequat, vel ullum dolore eu feugiat nulla facilisi. At vero eos et accusam et uisto odio dignissim qui blandit praesent luptatum zzril delenit aigue dous dolore et molestias exceptur sint occaecat cupiditat non simil provident tempor sunt in culpa qui officia deserunt mollit anium ib est aborum et dolor fuga. Et harumd deraud facilis ext er expedit distinct. Nam liber tempor cum soluta nobis eligend option congue hihil impedit doming id quod maxm placeat facer possim omnis voluptas assumenda est, omnis repellend. Temporibus auteul quinuised et aur officæ.

## .OI [Overlay Include]

- The following overlay example reserves a space at the current print position measuring one inch by one inch.

```
.oi olrose width 1i depth 1i nobreak
```

The requested width exceeds the space remaining on the current line. Therefore, space is reserved starting on the next line (a break occurs). Because NOBREAK is specified, a break does not occur after the overlay is placed, and the text continues on the current line. Because the overlay does not fit in the space reserved, it overlays the text.



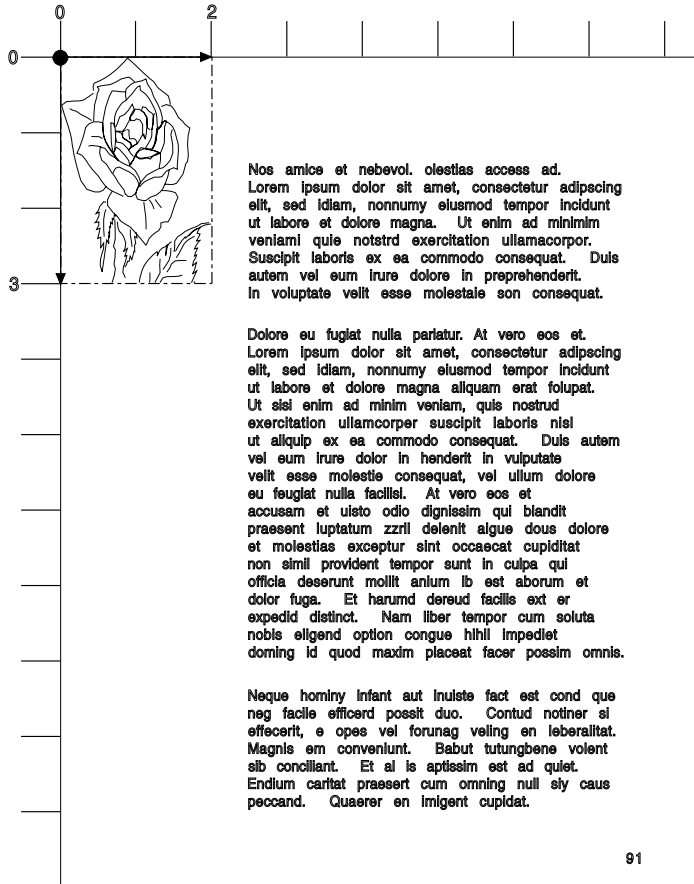
- **Placing an Overlay in a Named Area**

You can position the overlay on the page with the .DA control word.

- The following example places the overlay in a named area.

```
.da rose 0 0
.ar rose on
.oi olrose
.ar rose off
.in 1.5i
```

The overlay “olrose” is placed in a named area at coordinates 0,0.



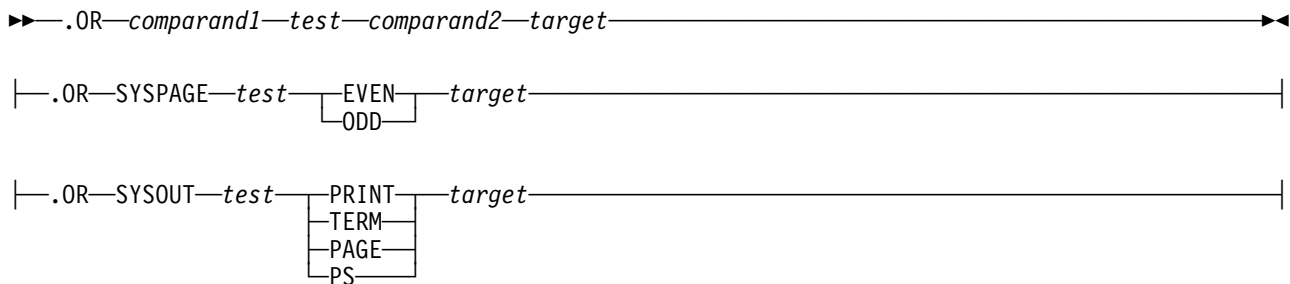
---

## .OR [Or]

### Function

The .OR [Or] control word can be used in conjunction with the .IF [IF] control word to process SCRIPT/VS input lines conditionally. The result of the test performed is logically ORed to the result of the most recently performed .IF [IF], .AN [And], or .OR [Or] control word to determine whether the target is to be processed.

### Syntax



### Parameters

- comparand1* Any string to be used as the first comparand. This comparand can be the value of a set symbol.
- comparand2* Any string to be used as the second comparand. It, too, can be the value of a set symbol.
- test* A 1- or 2-character code that tells SCRIPT/VS what comparison to make between the comparands. The following codes are recognized by SCRIPT/VS:
- | Codes        | Meaning               |
|--------------|-----------------------|
| eq =         | equal                 |
| ne $\neq$ <> | not equal             |
| gt >         | greater than          |
| lt <         | less than             |
| ge $\geq$    | greater than or equal |
| le $\leq$    | less than or equal    |
- target* Any valid SCRIPT/VS input line containing a control word, a macro, or text. If this condition or the result of the most recently performed .IF [IF&brk, .AN [And], or .OR [Or] is true, then the target line is processed next, with the first nonblank character after the second comparand treated as the first position of the line. Otherwise, the target line is ignored, and processing continues with the input line that follows the .OR control line.
- SYSPAGE** Determines whether the page currently being processed is an even- or odd-numbered page.
- EVEN** Indicates that the test is for an even-numbered page.
- ODD** Indicates that the test is for an odd-numbered page.
- SYSOUT** Determines whether SCRIPT/VS output is being formatted for a printer or a terminal.
- PRINT** Indicates that the test is to determine if output is being formatted for a 1403 or a 3800.
- TERM** Indicates that the test is to determine if output is being formatted for a 2741 or a 3270.

**PAGE** Indicates that the test is to determine if output is being formatted for a page printer.

**PS** Indicates that the test is to determine if output is being formatted for a PostScript device.

In SCRIPT/VS, more variety is possible in output formatting than can be determined with the SYSOUT parameter. The SCRIPT/VS system symbols '&\$LDEV' and '&\$PDEV' can be used to determine the actual logical and physical devices.

## Remarks

1. The .AN [And] and .OR [Or] control words, in conjunction with .IF [If], .TH [Then], and .EL [Else], allow you to construct complex logic statements.
2. The .OR control word itself does not cause a break; the target control word might, if it is processed.
3. Each of the comparands can be up to 255 characters long, and the shorter comparand is extended to the length of the longer with trailing blanks.
4. If substitution is off when the .OR control word is encountered, all valid symbols in the comparands are resolved before the comparison is made. (Symbols containing imbedded blanks must be compared with substitution off, so that the test to be performed and the target of the .OR can be identified.)

## Examples

- The following input line

```
.if &a eq &b .or &c eq &d .ty Yes.
```

is equivalent to the input lines

```
.if &a eq &b
.else .if &c eq &d
.then .ty Yes.
```

- You can use .OR with .IF, .TH, .AN, and .EL to construct such complex logical statements as the following:

```
.if &$sysmonth = 12
.or &$sysmonth = 1
.or &$sysmonth = 2
.th .se season = winter
.th .go start
.if &$sysmonth ≥ 3
.an &$sysmonth ≤ 5
.th .se season = spring
.th .go start
.if &$sysmonth ≥ 6
.an &$sysmonth ≤ 8
.th .se season = summer
.el .se season = fall
... start
.fo off
```

Season is &season.. It should be:

```
fall if Sept, Oct, Nov
winter if Dec, Jan, Feb
spring if March, April, May
summer if June, July, Aug
```

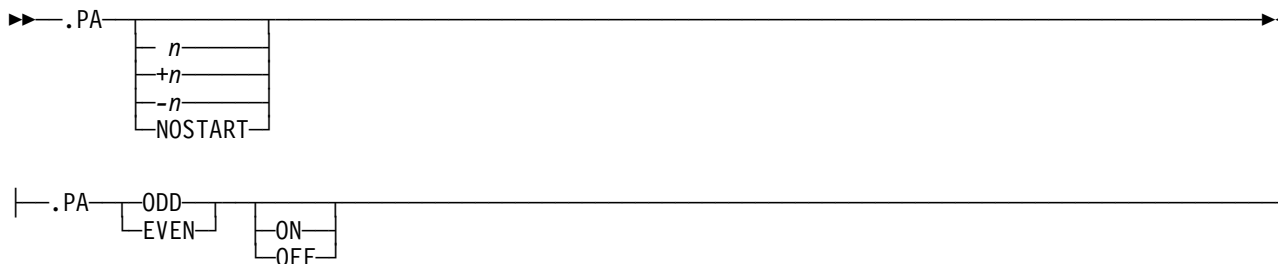
---

## .PA [Page Eject]

### Function

Use the .PA [Page Eject] control word to force subsequent text onto a new page of output, even if the current page has not been filled.

### Syntax



### Parameters

|                |                                                                                                                                                                                                                                                                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>n</i>       | Specifies the page number of the next page. If <i>n</i> is not specified, sequential page numbering is assumed, and the next page number is one greater than the current page number. <i>n</i> must be a nondecimal arabic integer.                                                                                                                         |
| <i>+n</i>      | Specifies that the next page should have a number that is equal to the normal next sequential page number plus <i>n</i> . <i>n</i> must be a nondecimal arabic integer.                                                                                                                                                                                     |
| <i>-n</i>      | Specifies that the next page should have a page number that is equal to the next sequential page number minus <i>n</i> . If subtracting <i>n</i> from the next page number yields a negative number, an error message is issued, and the control word is ignored.<br><br>The maximum allowed page number is 99,999,999.                                     |
| <b>NOSTART</b> | Causes the current page to be ended, but the next page is not started until some data causes it to be started or a control word that requires the page to be started is processed. After .PA NOSTART, the page definition (including running headings and footings) can be changed until the page is started.                                               |
| <b>ODD</b>     | Causes one or two page ejects, such that the new page is odd numbered.                                                                                                                                                                                                                                                                                      |
| <b>EVEN</b>    | Causes one or two page ejects, such that the new page is even numbered.                                                                                                                                                                                                                                                                                     |
| <b>ON</b>      | Defines the start of odd or even page eject mode. This mode is ended by specifying OFF or by the start of another .PA even or odd mode, or <i>n</i> . In odd or even page eject mode, output is formatted only on odd pages or only on even pages, whichever the case may be, and the other pages are left blank, except for running headings and footings. |
| <b>OFF</b>     | Defines the end of odd or even page eject mode. When .PA OFF (even or odd mode) is used, it does <i>not</i> cause a page eject.                                                                                                                                                                                                                             |

### Notes

- .PA causes a break and a section break.
- .PA ends an area, keep, float, or footnote.
- .PA starts a page unless the NOSTART parameter has been specified.
- The .PA [Page Eject] control word causes a short section.

## Remarks

1. The minimum page number is 1, and the maximum page number is 99,999,999. If a .PA control word attempts to set the page number outside this range, a message is issued and the control word is ignored.  
  
If the page number becomes 100,000,000 because of either a .PA control word being specified with no parameters or a page eject occurring because the page is full, a warning message is issued and the page number is set to 1.
2. Whenever a .PA control word is encountered, the rest of the current page is skipped after printing any text lines accumulated thus far. The next page is started, unless .PA NOSTART was specified. Starting a page includes formatting running headings and running footings for the page and establishing the page dimensions. These parameters are then fixed for the duration of the page, and changes to them do not take effect until the next page is started.
3. If you use the STOP option of the SCRIPT command on a 2741 terminal in CMS, SCRIPT/VS waits for you to enter a null line (with the Return or Enter key) before starting the new page.
4. If you want to change any page dimensions or define new running titles or running headings and footings for a new page, the appropriate control words must be processed before the .PA control word (except when NOSTART is specified). These control words are listed in Table 14 on page 441. Note that at the beginning of SCRIPT/VS processing, the first page has not yet been started.
5. If .PA *n* (or *+n* or *-n*) is specified after .PN FRAC is specified, the page eject occurs; however, page number is not reset, because the page number change to fractional pagination is pending.
6. Table 16 on page 442 lists the control words that require a page to be started; they cause one to start if one is not already started.

## Examples

- To start the next sequential page enter

.pa

The rest of the current page is skipped. Running headings and running footings are put on the next page.

- To repeat a page number enter

.pa -1

The new page has the same page number as the preceding page. The calculation is done *after* establishing the next sequential page number.

---

## **.PF [Previous Font]**

### **Function**

Use the .PF [Previous Font] control word to resume the use of the font whose ID was last saved using the .BF [Begin Font] control word.

### **Syntax**

►► .PF ◀◀

**Note:** The previous font and the font save stack are included in the active environment.

### **Remarks**

1. The .PF [Previous Font] control word restores the previous font; however, if the definition of a previous font has changed so that internal formatting functions (such as underscoring, capitalization, and overstriking as specified with the .US [Underscore], .UP [Uppercase], and .UC [Underscore and Capitalize] control words or with the OS RPT or OS CHAR parameters of the .DF [Define Font] control word) have been changed in that definition, then these changes are reflected when the font is restarted. See “.DF [Define Font]” on page 130 for a description of what can be changed in a font definition.
2. If the .PF control word is used when there is no previously saved font, an error message is issued, and the default font for the output device remains effective.
3. The font save stack is 16 entries deep, and the stack is saved and restored by the .SA [Save Environment] and .RE [Restore Environment] control words as well as by all other control words that cause the current environment to be saved and restored, such as .KP [Keep] and .FL [Float]. See “The SCRIPT/VS Formatting Environment” on page 451 for a list of these control words.

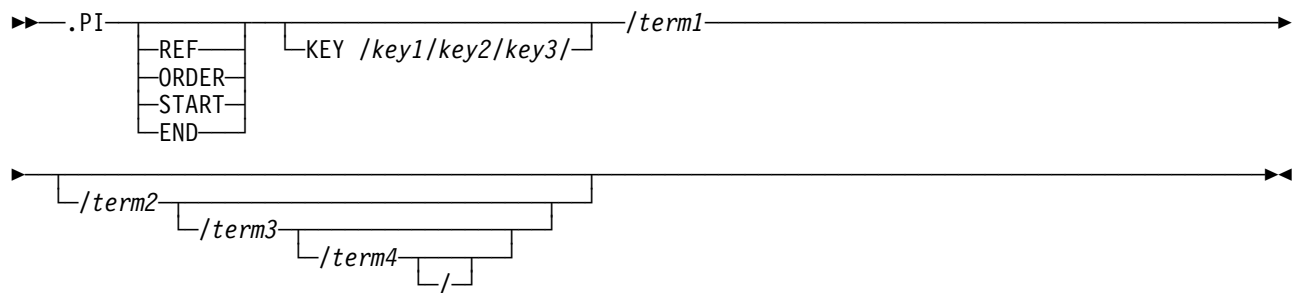


**.PI [Put Index]**

## Function

The .PI [Put Index] control word saves the specified lines for use in building an index. The .IX [Index] control word causes this index to be inserted into the document. The index entries are sorted according to either the language specified with the .DL INDEX control word or the default language for your installation. For more information about indexing sort sequences, refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*. The .PI [Put Index] control word is ignored unless enabled by the INDEX option of the SCRIPT command.

## Syntax



## Parameters

**REF** Specifies that this control word refers to an index reference. The first term given with the REF parameter is the index term entry, the second term is the “See” or “See also” entry. An index reference differs from an index term in that the words “See” or “See also” are prefixed to the index entry supplied, and no page number is printed in the index. “See” is prefixed if this is the only entry at this level under the preceding higher level entry; “See also” is prefixed if there are other entries. Use the system symbols, &SYSSEE and &SYSSEEALSO, to specify the character string to use as the cross-reference prefix to related terms in your index. “See” and “See also” are the default cross-reference prefixes used in this book.

**ORDER** Specifies that the page number of the entry in the index is to be placed in front of all previous page numbers for the term given.

**START** Specifies that the index entry being defined is the start of a reference to a range of pages. The START parameter is ignored when the REF parameter is specified.

**END** Specifies that the index entry being defined is the end of a reference to a range of pages. The END parameter is ignored when the ORDER or REF parameters are specified.

**KEY** Indicates that the sort keys for the index entries are given explicitly. If KEY is omitted, the sort keys are developed from the index terms. You should not use the KEY parameter with the REF parameter.

**key1–key3** Specifies the sort keys to be used for the index terms. If any key is null, that key is developed from the index term.

*term1*            The level 1, or primary, index term.

*term2*      The level 2, or secondary, index term.

*term3*            The level 3, or tertiary, index term.

## .PI [Put Index]

|              |                                                                                                                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>term4</i> | The text to be used in the index in place of the current page number. If a null <i>term4</i> is specified, no page number is printed in the index. If no <i>term4</i> is specified, the current page number is printed in the index. |
| <i>/</i>     | Any delimiter character that does not appear in any term or key.                                                                                                                                                                     |

## Remarks

1. The .PI [Put Index] control word defines an index entry. When entering the .PI control word, you can specify an index term that appears in the index and can specify a sort key by using the KEY parameter of the .PI control word.

If you do not specify a sort key with the index term, SCRIPT/VS builds a sort key by:

**Step 1** Translating the index term to uppercase

**Step 2** Ignoring all characters specified on the .DC IXI control word

**Step 3** Translating to blank all characters specified on the .DC IXB control word.

If you do supply a key, the key is used as specified, and these steps are not performed. Although Step 1 is not performed on user-defined keys, the lowercase and uppercase letters sort the same. The only time you see differences as a result of Step 1 is if the sort sequence at your installation was modified, such that the uppercase characters sort differently from the lowercase characters.

The index keys are sorted according to the sort sequence for the language specified, using .DL with the INDEX parameter. If no language is specified, the default language is used. The specific sort sequence for each language is described in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.

2. The entire .PI control word cannot be longer than 256 characters. This includes the .PI control word, the space after .PI, and the text that follows the control word. If the string is longer than 256 characters, SCRIPT/VS truncates the control word at 256 characters, but it does not issue an error message. Also, a single key or a term cannot exceed 120 characters.
3. Index entries are sorted on keys developed from the index terms, except when explicitly given with the KEY parameter. The key is developed by folding the term to uppercase, removing any characters to be ignored (as specified by the IXI parameter of the .DC [Define Character] control word), and translating any characters to be considered blanks (as specified by the IXB parameter of the .DC [Define Character] control word).
4. Because the .PI control word is a deferred control word, the index entry is not placed into the index until the output line with which it is associated is placed on the page. The index entry is associated with the next output line placed after the .PI control word is encountered. The next output line placed may not be the next line after the .PI control word in the input document. At least one word of text should follow a .PI control word to help avoid undesirable index page number references. However, if text following a .PI control word is never placed (for example, text in an unplaced area), no index entry is made.
5. The .PI control word is ignored while an index is actually being formatted.
6. A .PI control word is never associated with a footnote leader, top margin, or bottom margin. If a .PI control word is encountered preceding a .TM or .BM control word, the .PI control word is associated with some other output line. This may cause an undesirable entry in the index.
7. The .PI control word is processed to correctly resolve page numbers at the time the text line with which it is associated is placed on the page. Consider the following:

None of the existing fonts  
are available;  
.pi /existing fonts  
new fonts, however, are  
available.

If the text preceding the .PI control word appears on page four and the text following the .PI control word appears on page five, the index entry is listed as page four.

8. Use the system symbols, &SYSPRS and &SYSPLS, to specify range of pages or a list of pages in your index. By default, &SYSPRS is a '-' (hyphen), and &SYSPLS is a ',' (comma followed by a blank).

## Examples

- If you want a term to appear in the index, specify  
.pi /cowabunga

The term, *cowabunga*, is placed in the index in alphabetical order followed by the page numbers for each occurrence of a .PI [Put Index] control word that specifies that term in your document.

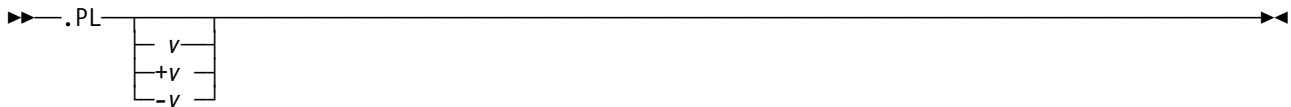
---

## .PL [Page Length]

### Function

The .PL [Page Length] control word specifies the vertical length (depth) of output pages. The value specified overrides the standard page length that is established for each logical device. Figure 6 on page 438 shows the relationship of the .PL [Page Length] to the layout of a SCRIPT/VS output page.

### Syntax



### Parameters

- $v$  Specifies the vertical length, or depth, of output pages. If no value is specified for  $v$ , the default value for the device is used. This number should be the same as the physical size of the printable part of the paper being used. However, when formatting for a printer logical device, it can be different, as explained below. The minimum value for the page length is the sum of the top margin (.TM) and the bottom margin (.BM) plus one line. If  $+v$  or  $-v$  is specified, the current page length is incremented or decremented accordingly.

**Initial Setting:** Dependent upon the logical device specified.

**Default:** Restores the initial setting.

### Notes

- .PL takes effect on the next page.
- .PL ends an area, keep, float or footnote.
- The page length is included in the page environment.

### Remarks

- The .PL control word allows varying paper sizes to be used for output. (The logical device specified in the DEV option of the SCRIPT command implies a default page length, but this can be overridden with the .PL control word.) Page length can be changed anywhere in a file, with the change effective on the page after the control word is encountered.
- If the output is in line printer format, the page length value need not be the same as the actual depth of the real paper, because SCRIPT/VS causes the printer paper to be ejected to the top of the next real page whenever a new SCRIPT/VS page is started. Thus, a SCRIPT/VS page can occupy less than a real page or more than a real page, and the output is newly aligned to the paper each time a SCRIPT/VS page is started.
- The previous rule notwithstanding, if you define a top margin (.TM) such that SCRIPT/VS needs to print data within the first three lines on a page, no printer page ejections can be done. Instead, SCRIPT/VS uses the page length value to find the top of the next page. It is, therefore, good practice to keep the .PL value accurate, so that it reflects the true depth of the page under SCRIPT/VS control.
- If the running headings and footings that are defined for a page fill up the page so that no room is left for text, SCRIPT/VS stops with an error message. The depth of running headings and footings cannot be predicted at the time they are defined, because they are formatted to the current line length (.LL)

when a page is started. The same running heading can occupy differing amounts of vertical space on different pages if the line length changes.

- | 5. In order to ensure that the ability to print your page printer output on AFP printers with different units
- | of measure bases, you should not exceed page length requests in excess of 22.75 inches.

## **Examples**

- Setting page length:

.p1 84

Page length is set to 84 lines. This is the correct size for 14 inch printer paper when printing at six lines per inch.

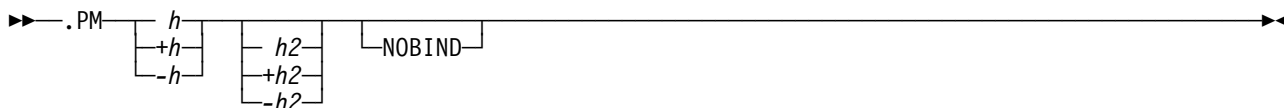
---

## .PM [Page Margins]

### Function

The .PM [Page Margins] control word causes SCRIPT/VS to shift the formatted output of each page to the right, and is used in conjunction with the BIND option of the SCRIPT/VS command and the .AM [Adjust Margin] control word. Figure 6 on page 438 shows how this control word interacts with other control words when formatting a page.

### Syntax



### Parameters

- h*** Specifies the amount of horizontal space (binding) to shift odd-numbered output pages. If *+h* or *-h* is specified, pages are shifted to the right or left, respectively, of the current binding.
- h2*** Specifies a binding for even-numbered pages. If *h2* is omitted, *h* applies to all pages.
- NOBIND** Indicates this control word should be processed only if the BIND option of the SCRIPT command was *not* specified. If the BIND option was specified, the control word is ignored.

**Initial Setting:** Dependent upon the logical device and the binding specified.

**Default:** Restores the initial environment.

### Notes

- .PM causes a break.
- .PM takes effect on the next page.
- .PM ends a keep, float, footnote, named area, or table.
- The page margin size is included in the page environment.

### Remarks

1. If the NOBIND parameter is not specified, the .PM control word unconditionally resets the page margins.
2. If the NOBIND parameter is specified, the .PM control word resets the page margins only if the BIND option of the SCRIPT command was not specified.
3. If the BIND option is not specified, the initial setting for the page margins is established by the logical device.
4. The actual (or potential) page number of the output page is controlled by the .PA [Page Eject] and .PN [Page Numbering Mode] control words, which are used to specify even and odd page numbers. Consequently, you can have two or more even-numbered (or odd-numbered) pages in a row.

5. Left-aligned revision codes, defined with the .RC [Revision Code] control word, for the first column are placed in the binding; revision codes for subsequent columns are placed in the gutter between columns. If sufficient space is not provided for revision codes, they are discarded.
6. Ensure that you do not specify so large a value for .PM that formatted output exceeds the page width for the logical device.

# .PN [Page Numbering Mode]

## Function

The .PN [Page Numbering Mode] control word allows you to control various aspects of page numbering including the format of the page number, and whether it is to be shown in running headings and footings that call for it.

## Syntax



## Parameters

- OFF** Suppresses the display of page numbers in running headings and footings, although pages are still sequentially numbered internally. Symbols set with the .SE [Set Symbol] control word to the current page number contain the correct number of the page on which they were processed.
- OFFNO** Suppresses both page number display and internal page numbering. The current page number for use with .SE remains the same for all pages until .PN OFFNO is ended with .PN ON.
- ON** Cancels .PN OFF or .PN OFFNO, so that internal numbering of pages is resumed, and the current page number can be displayed in running headings and footings.
- ARABIC** Causes the following page numbers to be represented as standard Arabic numerals. The ARABIC parameter can be abbreviated as AR.
- ROMAN** Causes page numbers to be represented as lowercase Roman numerals. Page numbers greater than 3999 are not supported with the ROMAN option. The ROMAN parameter can be abbreviated as RO.
- ALPH** Causes alphabetic page numbering to be started. In this mode, the number 1 is converted to a, 2 to b, 26 to z, and 27 to aa. The number 1978 is represented as bxb. The ALPH parameter can be abbreviated as AL.
- FRAC** Causes fractional pagination to begin. The next time a page eject occurs that would normally increment from an even to an odd number, the even number (for example, 20) is saved, and numbering starts with a fractional sequence (in this case, 20.1, 20.2, 20.3, and so forth).
- NORM** Causes an immediate page eject to occur and normal pagination to be resumed. In the previous example, the new page would be numbered 21. If .PN FRAC is not in effect, .PN NORM is ignored, and does nothing.
- PREF** Specifies a 1- to 8-character string to be used as a prefix in front of all page numbers. This is useful for page numbers printed in running headings and running footings, in tables of contents, or in front of set symbols set with the value of the current page number (&). The



string cannot contain imbedded blanks. To cause the prefix to be omitted from the page number, specify .PN PREF, with no string. This clears the previously defined prefix string.

**SUFFIX** Specifies a 1- to 8-character string to be used as a suffix at the end of all page numbers. This string cannot contain imbedded blanks. To cause the suffix to be omitted from the page number, specify .PN SUFFIX with no string. This clears the previously defined suffix string.

*n* Specifies the number of the next page. When the next page eject occurs, either naturally because the page becomes full, or as a result of .PA, the new page has the page number specified in *n*, as though this page eject had been caused by .PA *n*. If the next page really is started with .PA *n*, the number given on the .PA control word supersedes the number previously specified with .PN *n*. The maximum value of *n* is 99,999,999.

## Notes

- .PN ends a keep, float, footnote, named area, or table.
- .PN takes effect on the next page.
- The page number mode, suffix, and prefix are included in the page environment.

## Remarks

1. The .PN control word can be used to control page numbering. If the OFF parameter is specified, page numbering is discontinued on output, although the page numbers continue to be incremented internally. The page number symbol in running headings and running footings is not resolved.

The OFFNO parameter discontinues page numbering on output and stops the internal incrementing of page numbers. When the ON parameter is specified, page numbering resumes from the last internal page number.

The actual page numbers can appear in either Arabic numerals (which is the default), Roman numerals, or alphabetic letters, depending upon whether .PN ARABIC, .PN ROMAN, or .PN ALPH was most recent.

2. The .PN OFF and .PN OFFNO control words suppress the display of page numbers by resolving the page number symbol to null.
3. If FRAC is specified while the page numbers are represented in ROMAN or ALPHA numerals, the page number that is printed is in lowercase Roman or alpha numerals, but the fractional part is in Arabic.
4. Table of contents entries generated by .H0 [Heading Level 0] or the .PT [Put Table of Contents] control words show the page numbers in the same format they appear on the page, that is, if a prefix is used, it is shown in the table of contents; if Roman numerals are in effect, the contents entry has a Roman numeral, and so on.
5. Whenever the page number symbol is substituted, its prefix is also included. Care must be taken therefore when using the page number symbol as a part of an arithmetic operation on the right-hand side of a .SE statement.
6. The .PN control word takes effect on the page after it is encountered. However, if the .PN control word is encountered while SCRIPT/VS is determining page boundaries, it may appear to take effect on the current page. To avoid this, use a .BR control word before the .PN control word.
7. The NORM parameter is the only parameter that causes a page eject.
8. .PN *n* cancels all other options.
9. The total number of characters for the page number prefix, suffix, and the page number itself should not exceed 24.

## Examples

- `.pn off`

The internal page count continues to be incremented for each page printed. If the default running footing is still active, you should also cancel or suppress it using `.RF CANCEL` or `.RF SUPPRESS`.

- `.pn offno`

No page numbers appear on SCRIPT/VS output, and the internal page count remains at its current setting without further incrementing.

- `.pn on`

Page numbering on SCRIPT/VS output resumes, using the current internal page count; this count is incremented for each page printed.

- `.pn roman`

The page number in the running footing at the bottom of the page after this one appears as a Roman numeral.

The control word

`.pn arabic`

restores Arabic numbering on the next page.

- `.pn pref`

You can use the PREF parameter of the `.PN [Page Numbering Mode]` control word to create a compound page numbering scheme. For example, if you specify

`.pn 1`

`.pn pref 1-`

for the first chapter of a document, then the pages of that chapter are numbered 1-1, 1-2, 1-3, and so on. If you then specified

`.pn 1`

`.pn pref 2-`

for the second chapter, then the pages of that chapter are numbered 2-1, 2-2, 2-3, and so on.

- `.pn suffix`

You can use the SUFFIX parameter of the `.PN` control word to create a compound page number string. For example, if you specify

`.pn 1`

`.pn suffix -Intro`

then the pages are numbered 1-Intro, 2-Intro, and so on.

## .PO [PostScript]

### Function

The .PO [PostScript] control word allows you to include an ASCII file that contains PostScript commands in a SCRIPT/VS file. The .PO control word also allows you to specify how much vertical and horizontal white space to reserve in the output for the PostScript image, whether you want the PostScript image to be scaled, and whether to include the PostScript image on the current line. Refer to the *Document Composition Facility: SCRIPT/VS User's Guide* for more information on using PostScript devices with SCRIPT/VS.

**Note:** PostScript files are referred to as *image* files because SCRIPT/VS treats them as images; however, the files can contain any combination of images and text.

### Syntax

➡.PO *file-id* [WIDTH *h*] [DEPTH *v*] [SCALE] [NOBREAK] [SEPMASR *token*] ➡

### Parameters

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>file-id</i>        | <p>The SCRIPT/VS <i>file-id</i> for the PostScript file to be included. The <i>file-id</i> must be 1- to 8-characters. If the <i>file-id</i> is more than 8-characters, it is truncated to 8.</p> <p>You can use the .DD [Define Data File-id] control word to associate the <i>file-id</i> with an external filename or data set. If no .DD [Define Data File-id] control word is found for the <i>file-id</i>, SCRIPT/VS uses the <i>file-id</i> to derive the external name of the file or data set, based on the file naming conventions appropriate for the operating system.</p> <p>You can also use the SEARCH option of the SCRIPT command to alter the way SCRIPT/VS builds the <i>file-id</i> of the included PostScript file. See Chapter 3, "SCRIPT Command Options" on page 17 for more information about the SEARCH option.</p> |
| <b>WIDTH</b> <i>h</i> | <p>Amount of horizontal space (in any valid space unit) SCRIPT/VS reserves in the output.</p> <p>The default width value is the current column line length minus the current left and right indentation.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>DEPTH</b> <i>v</i> | <p>Amount of vertical space (in any valid space unit) SCRIPT/VS reserves in the output.</p> <p>The default depth value is zero.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>SCALE</b>          | <p>Specifies to scale the PostScript image to the specified DEPTH or WIDTH values. If only one of the parameters (DEPTH or WIDTH) is specified, the PostScript image is scaled proportionally using the one value.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>NOBREAK</b>        | <p>Specifies that SCRIPT/VS should not cause a line break before or after the space reserved for the PostScript image. The space is formatted as if it were a word of text, with the bottom of the space positioned on the current baseline.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>SEPMASR</b>        | <p>Specifies that PostScript images are being associated with output separation masters.</p> <p>The <i>token</i> is a 1- to 8-character name that identifies which separation masters this PostScript image should be associated with. The token name corresponds to the items selected for separation masters with the .SM control word. You cannot specify OLD COLOR as a token name.</p> <p>The special token value of ALL is used to indicate that this graphic should be contained in all separation masters, including the default.</p>                                                                                                                                                                                                                                                                                                 |

## .PO [PostScript]

For more information on the SEPMASSTR parameter, see “.SM [Separation Master]” on page 353.

### Notes

- .PO causes a break unless NOBREAK is specified.
- .PO ensures that the page is started.

### Remarks

1. Before using .PO to include the PostScript image in your document, you should verify that it prints without errors on a PostScript device. DCF does not interpret the PostScript commands. Therefore, errors in the included PostScript file cause errors at print time. Refer to the *Document Composition Facility: SCRIPT/VS User's Guide* for more information.
2. You must use a *binary* upload procedure to transfer a PostScript image file from an IBM PC to a host system. Refer to the section on “Using SCRIPT/VS with PostScript Devices” in the *Document Composition Facility: SCRIPT/VS User's Guide* for more information on uploading PostScript files.

**Note:** Do not specify ASCII conversion when uploading the PostScript files.

3. The PostScript file specified must be in ASCII and cannot exceed a logical record length of 256 bytes.
4. The PostScript file must follow Version 2.0 Adobe Document Structuring Conventions, and the first line of the file must be a comment that begins with the characters %!.

The file must also be “well-behaved” as defined in Adobe Systems' rules for creating encapsulated PostScript documents. Files that do not follow these rules may print alone but may cause printer errors when included in DCF output.

PostScript files that claim to follow Adobe Document Structuring Conventions Version 2.0 include “Adobe–2.0” in the first comment in the file. A PostScript file that also claims to be encapsulated (an EPS file) also includes “EPSF–” in the first comment.

**Note:** If warning message 660 is issued indicating that the PostScript file %%BoundingBox: comment contained real numbers that have been truncated, the reserved space and the positioning of the image may be affected. To reposition the image, use the .IS [Inline Space] control word and the .SB [Shift Baseline] control word. To increase or decrease reserved space, use the DEPTH and WIDTH parameters.

5. You must specify a PostScript device with the DEVICE option of the SCRIPT command for the PostScript image files to be included in your formatted output.
6. If a .PO control word is encountered when formatting for a device that is not a PostScript device, white space is reserved. If the DEPTH and WIDTH parameters are not specified, the defaults are used.
7. The PostScript image file should not generate more than one page. If multiple pages are generated, all of those pages overlay each other in the space reserved for the image. SCRIPT/VS disables the PostScript *showpage* command if it is encountered in a PostScript input file.
8. The space reserved for a PostScript image must fit on one page. If the PostScript image is larger than the space reserved, use the SCALE parameter to make the image smaller.
9. If NOBREAK is specified, the space reserved for the PostScript image must fit on one line. If not enough room is left on the current line to contain the space, the image is moved to the next line. If the PostScript image is the only object on the line and its width exceeds the column line length, the image extends beyond the column line length, regardless of the overdraw option in effect.
10. If NOBREAK is specified, the bottom of the PostScript image is positioned on the current baseline. If the image does not appear to be positioned correctly because of white space inside the image, you can use the .SB [Shift Baseline] control word to shift the image up or down.

11. When NOBREAK is specified, the image is treated as a word; therefore, word spaces eligible for justification appear before and after the image.
12. If DEPTH or WIDTH is not specified, SCRIPT/VS calculates the necessary vertical or horizontal space for the PostScript image, using the values on the %%BoundingBox: comment in the PostScript file. SCRIPT/VS also uses the %%BoundingBox: comment values to translate the origin of the PostScript image so that the bottom-left corner of the PostScript image is positioned at the bottom-left corner of the space reserved. If the %%BoundingBox: comment is not found, the PostScript image may not be positioned correctly.  
  
If DEPTH or WIDTH is not specified and a %%BoundingBox: comment is not found, the default DEPTH or WIDTH value is used.
13. If DEPTH is not specified and a %%BoundingBox: comment is not found, no vertical space is reserved, but a break occurs unless NOBREAK is specified. Formatting continues with the next output line after the .PO control word.
14. If SCALE is specified, either the DEPTH parameter, the WIDTH parameter, or both DEPTH and WIDTH must also be specified. If neither DEPTH nor WIDTH is specified, a warning message is issued and the SCALE parameter is ignored.
15. If SCALE is specified, the PostScript file must have a %%BoundingBox: comment. The values given in the %%BoundingBox: comment are used to determine the size of the PostScript image. The PostScript image is then scaled (reduced or enlarged) to fit into the space requested by either the DEPTH parameter, the WIDTH parameter, or both. A warning message is issued if a %%BoundingBox: comment is not found, and if the SCALE parameter is ignored.
16. Scaling may affect the position of the image. If the image does not appear to be positioned correctly, you can use .IS [Inline Space] and .SB [Shift Baseline] to reposition the image.
17. If the PostScript image is placed in a rotated area, the image is rotated.
18. Including large PostScript files or including many PostScript files on one page could cause you to run out of storage.
19. SCRIPT/VS automatically saves and restores the PostScript environment around the included PostScript file. This has no effect on the SCRIPT/VS formatting environment.
20. The space reserved for the PostScript image includes the depth of the PostScript image or the value specified on the DEPTH parameter plus extra leading calculated as follows: the current line space value minus the height of the em space in the current font.
21. The SEPMASTR parameter is ignored for line devices.
22. The SEPMASTR parameter is also ignored for all devices if the SEPMASTR command option is not specified.
23. When used with the .PO control word, the SEPMASTR parameter indicates whether or not the PostScript image should be included in an output separation master other than the default master. PostScript images are included only if the value on the SEPMASTR parameter corresponds to a value selected with the .SM control word.

**Note:** Examples using the SEPMASTR parameter can be found in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.

## Examples

- The example below shows how the .PO control word can be used to include a PostScript image file in a DCF document. The WIDTH and DEPTH parameters are not specified on the .PO control word, so the space reserved for the PostScript image is determined by the %%BoundingBox: comment values.

```
.po test nobreak
```

Figure 2 shows the image on the page. Because the NOBREAK parameter has been specified, the image is on the same line as the text.

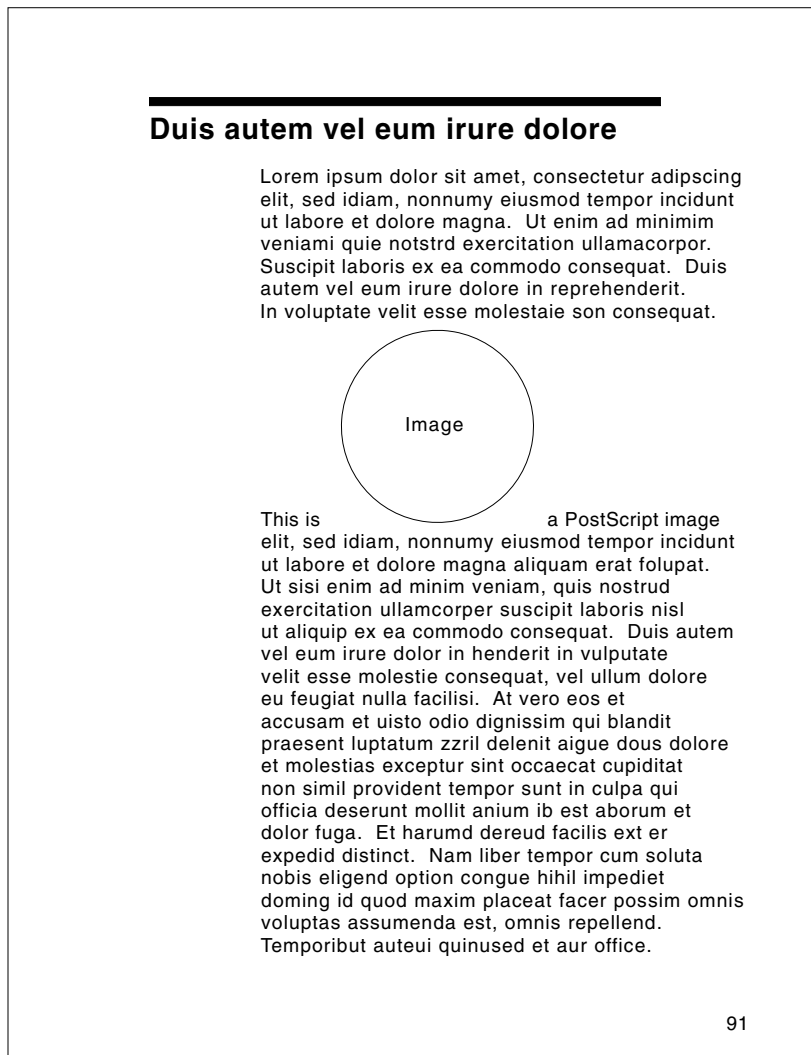


Figure 2. .PO Example Using NOBREAK

- The example below shows how the .PO control word can be used to scale a PostScript image and to include the image in a DCF document. The example scales a two-inch by one-inch PostScript image. If you specify SCALE and .5i on the DEPTH parameter, the image is scaled to one-fourth of its original size.

```
.po test depth .5i scale
```

Figure 3 on page 293 shows the image before and after it has been scaled. When only one value is specified for WIDTH or DEPTH, the image is always scaled proportionally. Note that the image below retains its original shape after scaling to a .5 inch depth because the width was scaled proportionally.

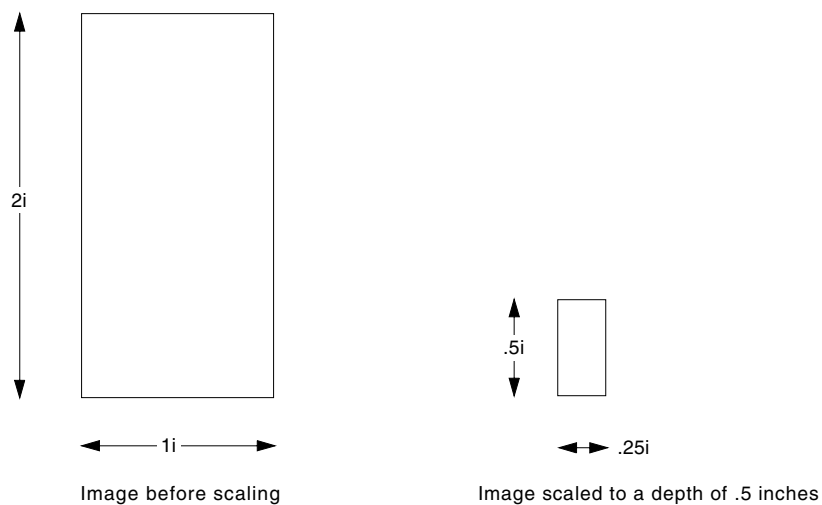


Figure 3. .PO Example Showing Proportional Scaling

Refer to the *Document Composition Facility: SCRIPT/VS User's Guide* for more examples using the .PO control word.

---

## **.PT [Put Table of Contents]**

### **Function**

Use the .PT [Put Table of Contents] control word to add lines or control words to the file that is used to generate the automatic table of contents.

### **Syntax**

►► .PT—*line*—————►►

### **Parameters**

*line* Any text line or control word line that you want in the table of contents.

If *line* is text, it is written to the file DSMUTTOC as part of a .SX [Split Text] control word, which causes it to be formatted as a table of contents line when DSMUTTOC is processed. The current page number is put in as the right part of the .SX string. (For more details, see the discussion of the .SX control word.)

If *line* starts with a period, it is assumed to be a control word or macro. It is written into the DSMUTTOC file directly and is executed when the DSMUTTOC file is processed.

If *line* is specified with extra leading blanks, it is taken as a line of text, even if the first nonblank character is a period. The extra leading blanks are removed, and a .SX control word is built for the DSMUTTOC file, using the first nonblank character as the beginning of the data.

### **Remarks**

1. For text lines, the .PT control word generates a .SX control word to be written into the table of contents utility file in the form  

```
.sx f /text line/ ./33/
```

where the page number used is the actual page number when the .PT is processed, and the delimiter used is actually hexadecimal X'00'. The .PT control word does not accept lines that begin with hexadecimal X'00' as valid lines; such lines result in an error message.
2. This control word is especially useful when defining macro heading levels with the .DM [Define Macro] control word. The internal processing for the head level control words .H0 [Heading Level 0] generate a .PT to write the required information into the table of contents file.
3. The .PT control word is ignored while a table of contents is actually being formatted.
4. Because the .PT control word is a deferred control word, the table of contents entry is not placed into the table of contents until the output line with which it is associated is placed on the page. The table of contents entry is associated with the next output line placed after the .PT control word line is encountered. The next output line placed may not be the next line after the .PT control word line in the input document. At least one word of text should follow a .PT control word line; otherwise, page number references or the order of the entries in the table of contents may be incorrect. However, if text following a .PT control word line is never placed (for example, text in an unplaced area), no table of contents entry is made.
5. A .PT control word is never associated with a footnote leader, top margin, or bottom margin. If a .PT control word is encountered preceding a .TM or .BM control word, the .PT control word is associated with some other output line. This may cause an incorrect table of contents entry.
6. The .PT control word is processed to correctly resolve page numbers at the time the text line with which it is associated is placed on the page. Consider the following:



```

Start each sentence on a
new input line
.pt How to Create a File
Once the file is created...

```

If the text preceding the .PT control word appears on page four and the text following the .PT control word appears on page five, the entry is listed as page four in the table of contents.

## Examples

- .pt .pa

This line places the .PA control word in the table of contents, so that when the table of contents file is being processed, a page eject occurs at this point. You can do this if you want separate content sections to appear on different pages.

- .pt    .pa

Because the line given has extra leading blanks, it is a text line not a control word. The leading blanks are removed, and a .SX control word is built, using the characters “.pa” as the data:

```
.sx f /.pa/ ./33/
```

(The head level control words insert a leading blank in front of a line to be written to the table of contents with .PT when it is known to be text.)

- .pt .h3 this is a head level 3

In this case, the control word .h3 is written into the table of contents file because the period appears in the first available position with no extra leading blanks. Any head level that is written into the table of contents file in this way is processed as a heading when the table of contents is actually formatted.

A normal head level 3 is generated at that point in the table of contents, but no attempt is made to write any more information into the table of contents utility file. In other words, the .PT function of the .H3 control word is ignored while the table of contents is actually being formatted.

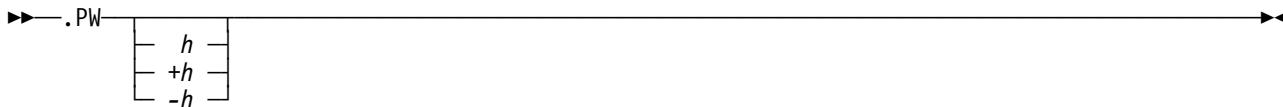
---

## .PW [Page Width]

### Function

The .PW [Page Width] control word specifies the width of the output page.

### Syntax



### Parameters

*h* Specifies the width of subsequent output pages.

If *+h* or *-h* is specified, the width of subsequent pages is increased or decreased. *h* can be any valid horizontal space unit.

**Initial Setting:** Dependent upon the logical device specified.

**Default:** Restores the initial setting.

### Notes

- .PW ends a keep, float, footnote, named area, or table.
- .PW takes effect on the next page.
- The size of the page width is included in the page environment.

### Remarks

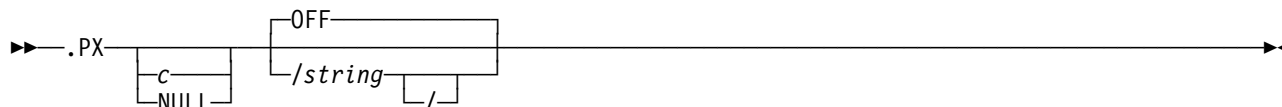
1. The page width includes both the left margin, as determined by the .PM [Page Margins] and .AM Adjust Margin] control words (or binding, as established with the BIND option of the SCRIPT command), and the line length, as determined by the .LL [Line Length] control word. These relationships are illustrated in Figure 6 on page 438. Page width should not be less than the page margin (as set with the .PM and .AM control words) plus the line length (as set with the .LL [Line Length] control word).
2. All text must be placed within the page, as defined with the .PW control word. If text exceeds the boundaries of any page, SCRIPT/VS issues an error message. If the CONTINUE option of the SCRIPT command has been specified, formatting proceeds with text exceeding the boundaries of the page. This might result in loss of text on some devices.
3. The default values for each logical device are listed in the logical device tables on page 27 through 28.
4. The maximum value that can be specified for non-3800 line printers is 255DH or the equivalent in other units of measure.

## **.PX [Prefix]**

### **Function**

Use the .PX [Prefix] control word to replace control characters at the beginning of lines read from input files with control words, macros, or other strings.

### **Syntax**



### **Parameters**

- c** Specifies a control character, either as a single character or as a 2-character hexadecimal code. Whenever *c* appears at the front of a line, it is replaced with *string*.
- NULL** Indicates that if an input line does not begin with any of the currently defined prefix control characters, *string* should be added to the front of the line, without disturbing the first character.
- OFF** Indicates that input line prefixing for the given control character should no longer be performed. If no control character is specified, all input line prefixing is canceled. If you specify .PX OFF, the following rules apply:
- It must be on a line by itself
  - A control word modifier cannot be used on this line.
- string* Adds a control word, macro, symbol, or other text string to the beginning of each line that begins with the specified control character.
- /* An arbitrary delimiter. Any nonblank character that does not appear in the string can be used. If no ending delimiter is used, the string is suffixed with a single blank. If you want a null string or a single blank, you must include an ending delimiter.

**Initial Setting:** OFF

### **Remarks**

1. Only lines read from an input file are examined for prefixing; macro lines are never prefixed.
2. Prefixing is performed before symbol substitution, and it is performed even when symbol substitution is off.
3. The .PX control word allows output files that have been prepared for 1403-like output to be processed as SCRIPT input files by providing a profile that defines macros to process each carriage control character.
4. If a NULL prefix has been defined, *all* lines are prefixed with some *string*. If the NULL prefix is the *only* one defined, all lines are prefixed with the same string until .PX NULL OFF or .PX OFF is specified.
5. An error occurs if input prefixing causes an input line to be too long.

## Examples

- To treat input lines beginning with an asterisk (\*) as comments, you can prefix them with the SCRIPT/VS comment:

```
.px * /.*
```

Any input line beginning with “\*” is prefixed; the “\*” is replaced with “.\*”

- Prefixing can be used to create symbol arrays. For example, to define an array containing the names of the days of the week, enter

```
.px # /.px off  
.px null /&periodse days() = '/  
Sunday  
Monday  
Tuesday  
Wednesday  
Thursday  
Friday  
Saturday  
#
```

The value of the SCRIPT/VS system symbol &SYSDAYOFW is an integer indicating the day of the week, and it can be used as an index value for the &days array. If you enter

```
Today is &days(&SYSDAYOFW)..
```

the result of symbol substitution is

```
Today is Thursday.
```

---

## **.QQ [Quick Quit]**

### **Function**

The .QQ [Quick Quit] control word causes SCRIPT/VS processing to end immediately, without the usual final page eject.

### **Syntax**

►► .QQ ◄◄

### **Remarks**

1. Because SCRIPT/VS does not perform a final page eject after encountering the .QQ control word, some output that has been formatted may never be displayed.
2. The .QQ control word is useful when you are using the .TE [Terminal Input] control word to enter lines from the terminal and you want to terminate processing quickly.
3. The .QQ control word causes an end of processing no matter where or when it is encountered, including within imbedded files (see the .IM control word). All open SCRIPT files are closed before processing ends.
4. Because the .QQ control word causes SCRIPT/VS to terminate processing immediately, when formatting for a page mode device, some required structured fields may not be included, such as the EDT (end of document structured field). This may cause an error message to be issued from the print services program. To avoid this, use the .QU [Quit] control word which allows final page processing to occur prior to termination.

---

## **.QU [Quit]**

### **Function**

The .QU [Quit] control word causes processing to end with a final page eject.

### **Syntax**

►►—.QU—————►◄

### **Notes**

- .QU causes a break.

### **Remarks**

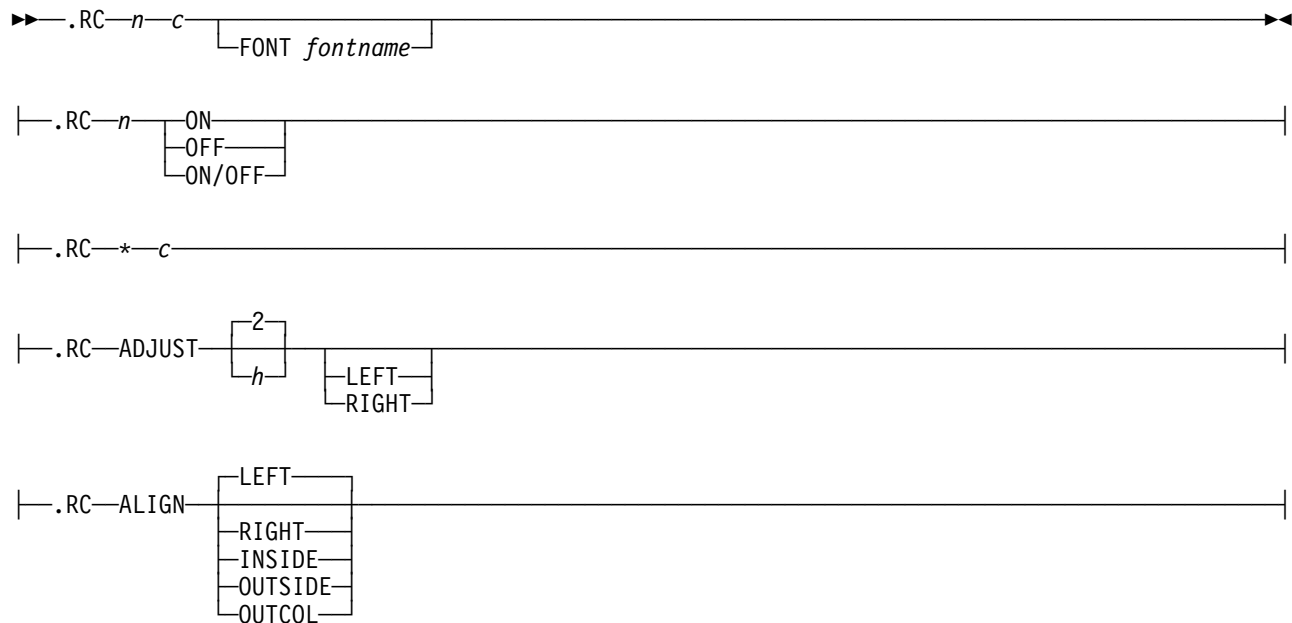
1. The .QU control word causes a final page eject so that the last partial page of formatted text can be printed.
2. The .QU control word is useful when you are using the .TE &lrbk.Terminal Input] control word to enter lines from the terminal and you want to terminate processing quickly.
3. The .QU control word causes an end of processing no matter where or when it is encountered, including within imbedded files (see the .IM control word). All open SCRIPT files are closed before processing ends.

## .RC [Revision Code ]

### Function

The .RC [Revision Code] control word allows you to designate a revision marker to be printed to the left or right of the column.

### Syntax



### Parameters

- n* Specifies the revision number. *n* must be an integer from 1 to 9.
- c* Specifies a revision character to be printed in the margin to the left or right of the text associated with revision *n*. *c* can be any single character.
- FONT fontname** Specifies the name of the font to be used for printing the revision code character associated with revision code *n*. The *fontname* must have been previously defined with the .DF control word. If FONT is not specified, the revision code character prints in the default font for the device being used.
- The special value of (=) can be specified as *fontname* to indicate that the revision character should be printed using the font that is current at the time the revision character is placed.
- The FONT parameter is ignored for line devices.
- ON** Specifies the beginning of text to be marked with revision number *n*.
- OFF** Specifies the end of text to be marked with revision number *n*.
- ON/OFF** Specifies that only the next output line should be marked with revision number *n*.
- \*** Marks the next output line with the specified revision character *c*. Unlike the ON/OFF parameter, the \* option allows you to specify any character for one occasion without associating it with a revision code number. When using \*, the character is printed in the default font.

## **.RC [Revision Code ]**

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ADJUST</b>  | <p>Specifies the amount of horizontal space to be used for revision characters.</p> <p>If LEFT is also specified with this .RC control word, the <i>h</i> value applies only to revision codes on the left of the output.</p> <p>If RIGHT is specified, the <i>h</i> value applies only to revision codes on the right of the output.</p> <p>If neither LEFT nor RIGHT is specified, the <i>h</i> value applies to both types of revision codes.</p>                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>h</i>       | <p>The amount of horizontal space to leave between the revision character and the start or end of the output line. If <i>h</i> is omitted, 2 is the default. If a value of 0 is specified, no revision characters are printed. If the width of revision codes is greater than <i>h</i>, no revision codes are printed.</p> <p>If LEFT is specified, the value specified is the amount of space between the beginning of the revision character and the beginning of the text line. The value should not be so great that it extends into the bind area.</p> <p>If RIGHT is specified, the value applies only to revision codes on the right of the output. The value specified is the amount of space between the end of the text line and the end of the revision character. The value should not be so great that it extends off the page.</p> |
| <b>LEFT</b>    | Specifies that the adjust value given pertains to left revision codes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>RIGHT</b>   | Specifies that the adjust value given pertains to right revision codes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>ALIGN</b>   | Indicates the alignment of subsequent revision characters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>LEFT</b>    | Specifies that all revision characters are placed to the left of all output lines. This is the default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>RIGHT</b>   | Specifies that all revision characters are placed to the right of all output lines.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>INSIDE</b>  | Specifies that all revision characters are placed to the right of output on even pages and to the left of output lines on odd pages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>OUTSIDE</b> | Specifies that all revision characters are placed to the right of output lines on odd pages and to the left of output lines on even pages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>OUTCOL</b>  | Specifies that for two column document formatting (in the body of the document), the revision characters are placed to the left of column 1 and to the right of column 2 for all pages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

**Note:** Revision codes are included in the active environment.

## **Remarks**

1. The .RC control word has four functions:
  - a. To define a revision code and select its font
  - b. To activate or deactivate the revision code
  - c. To set the size of the revision-code field
  - d. To determine the position of the revision character.

You can have as many as nine revision codes defined at any time, and each revision code can be assigned a different character. The ON and OFF parameters activate and deactivate the actual revision-code marking. The ON/OFF parameter has the effect of applying revision code *n* to only one line: the line that is printed after the ON/OFF parameter is processed.



2. By assigning different symbols (including the blank) to different revision code numbers, it is possible to selectively print specific revision-code markers, or to differentiate between various levels of revision.
3. Because the .RC control word does not cause a break, revision-code markings can be turned on and off within a paragraph or even a sentence without disrupting normal SCRIPT/VS formatting. An explicit .BR [Break] control word can be necessary under certain circumstances to cause the last unrevised line to be finished before formatting begins on the revised material.
4. When you are printing a document with left revision codes in more than one column, the revision code for the leftmost column is placed in the page margin that is specified with the BIND option of the SCRIPT command or the .PM [Page Margins] control word. The revision codes for the other columns are placed in the gutter between the columns. If there is not enough space for the revision-code field in the binding or gutter, the revision code is omitted.

When you are printing a document with right revision codes in more than one column, either the revision code for the rightmost column must fit within the page value or the revision code is omitted.

5. Revision codes can be nested. This is useful in circumstances where revisions are made to sections that already have a revision code active. If a revision code is turned on while another is active, the first is stacked. It is neither on nor off. When the second revision code is turned off, the stacked revision code becomes active again. Only one revision code is active at a time.
6. If you attempt to redefine a revision code character while that revision code is on, an error message is issued.
7. Revision codes are part of the active environment. For example, if you enter

```
.rc 3 on
.sa
.rc 3 off
.re
```

the revision code 3 is on after the .RE [Restore Environment] control word is executed.

8. Revision codes are not applied to running headings, footings, or page segments. Because revision codes are intended to identify revised text, they are not applied to blank lines; other non-text objects included with the .PO [PostScript], .SI [Segment Include], .OI [Overlay Include], or .IO [Include Object] control words; or to lines consisting of only vertical rules, horizontal rules, or equations.
9. Revision codes are not placed on blank lines created with the .SP [Space], .SK [Skip], .BL [Blank], or .DS [Double Space Mode] control words.
10. The &\$RB system symbol can be used to insert a required blank onto lines with equations enabling the output line to have a revision code. An output line with the &\$RB system symbol can be used instead of a blank line to enable that output line to have a revision code.
11. When your output is formatted for the 3800 Printing Subsystem Model 1 and an attempt is made to place a revision character on top of text, blanks, or rules from another line, column, or named area, the revision character is not printed.
12. When revision codes are placed on the right, and the column line length plus the amount of the adjust value would exceed the page width, no revision characters are printed.
13. No INSIDE/OUTSIDE parameters are needed when .RC ADJUST is specified, because these revision codes alternate from left to right, based on the page type (even or odd). The left and right adjust values are used appropriately.
14. The .RC ADJUST control word does not actually cause revision codes to be placed on the page. It only specifies the adjust values for left or right revision codes.
15. The ALIGN parameter does not cause revision codes to be placed on the page. It only specifies the position where all revision codes (those that are in effect or will be in effect) should be placed.

## .RC [Revision Code ]

16. Revision codes used within a table print to the left or right, outside the table. Revision codes for 90°, 180°, or 270° rotated cells are ignored. If you have revision codes active in two cells that are horizontally adjacent, the revision codes can overlap.
17. Left revision codes on tables are ignored for the 3800 Printing Subsystem Model 1.
18. The OUTCOL parameter is intended for documents formatted in two text columns. If body text is being formatted in other than two columns, and OUTCOL is specified, revision codes for the leftmost column are placed on the left and revision codes for all of the other columns are placed on the right. A single column is considered the leftmost column.
19. OUTCOL is valid only in the body of the document. If OUTCOL is specified at the time you place text into footnotes, page-type floats, page-type or body-type named areas, it is ignored and the revision code is RIGHT aligned, even though, some of these items appear in columns later on.  
  
For section areas, the revision character is LEFT aligned if OUTCOL is specified because section areas are placed relative to column 1. For the most predictable results, specify a different ALIGN value when you place text in areas other than the document body.
20. For text inside of tables, widows, keeps, and column floats, the text is evaluated for revision code placement *after* it is placed on the page, and it is treated the same as other text in that column while the OUTCOL parameter is processed.
21. The **fontname** that is specified can be a list of fonts previously defined with the LIST parameter of the .DF control word.
22. Revision codes are defined and placed using the default or initial font. When formatting for the 3800 Printing Subsystem Model 3 or 6, you must be careful to avoid placing revision codes on rotated tables and areas that are in a different rotation than the default font. If this is done, a PSF error message occurs, and the job terminates when printing.
23. The definition of the font is the one used at the time the revision code character is encountered in the input, not the definition at the time the character is placed on the output page. For example, if you enter

```
.df example type(itc avant garde gothic semibold)
.rc 1 A font example
.fo off
line of text
.rc 1 on
.df example type(itc avant garde gothic italic)
line of text with revision character A in gothic bold
.rc 1 off
line of text
```

is formatted as

```
line of text
A line of text with revision character A in gothic bold
line of text
```

The definition of **rbold** at the time the revision code is started determines that (bold) is used, even though the second .df might have been processed before the line when the revision character is placed on the page.
24. If the equal sign (=) is specified as the *fontname*, the revision character is placed in the current font. In this context, the current font is defined as the font that is active at the end of the output line with which the revision character is associated regardless of whether the character is placed on the left side or right side of the output line.
25. Any revision code that appears in the Table of Contents is in the default font regardless of the font that was printed within the body of the document.

26. To ensure that the revision code font you have selected is the font that prints in your document, it is necessary to select a codepage that contains the code point identifier associated with the character graphic ID you wish to print. For example, if you specify:

```
.df rcfont type(pi sans serif) codepage t1gpi363
.rc 1 | font rcfont
```

unexpected results occur when you start .rc 1. The code point for (I) is X'4f'. Codepage T1GPI363 (PI Font) does not have the character graphic (I) defined at hexadecimal code point (X'4f').

Therefore, you do not get the results that you expect.

27. The revision character size does not affect line spacing, therefore, the revision codes might overlap if the font size you select is large enough. For example:

```
.df rcbig type('sonoran sans serif' 24 bold) codepage t1dcdcf
.rc 2 | font rcbig
.fo off
```

Some text prints here before the revision codes.

```
.rc 2 on
```

Line one prints one revision character.

Line two then prints another.

```
.rc 2 off
```

And then another line prints after the revision codes.

Prints as follows:

Some text prints here before the revision codes.

```
| Line one prints one revision character.
```

```
| Line two then prints another.
```

And then another line prints after the revision codes.

## Examples

- When several revision codes are on at the same time, only the most recent is active. For example, you can define revision codes 1 and 2, and associate the characters “\*” and “?” respectively, with them by entering

```
.rc 1 *
.rc 2 ?
```

You can nest revision codes when entering text, like this

```
.rc 1 on
.rc 2 on
“It was so kind of you to come!
And you are very nice!”
The Carpenter said nothing but
.rc 2 off
“Cut us another slice.
I wish you were not quite so deaf--
I've had to ask you twice! .”rc 1 off
```

The formatted output looks like this

```
? “It was so kind of you to come!
? And you are very nice!”
? The Carpenter said nothing but
* “Cut us another slice.
* I wish you were not quite so deaf--
* I've had to ask you twice!”
```

## .RC [Revision Code ]

- A single line can be marked with a revision character with the ON/OFF parameter, or with the \* parameter. For example,

```
"It seems a shame," the Walrus said,  
.rc 2 on/off  
"To play them such a trick.  
After we've brought them out so far,  
And made them trot so quick! ."rc * ?  
The Carpenter said nothing but  
"The butter's spread too thick!"
```

is formatted as

```
? "It seems a shame," the Walrus said,  
? "To play them such a trick.  
? After we've brought them out so far,  
? And made them trot so quick!"  
! The Carpenter said nothing but  
! "The butter's spread too thick!"
```

- The width of the revision field can be changed with the ADJUST parameter. For example,

```
.fo off  
.rc 1 /  
.rc 1 on  
.rc align right  
.rc adjust 4  
"O Oysters," said the Carpenter,  
.rc adjust 5  
"You've had a pleasant run!  
.rc adjust 6  
Shall we be trotting home again? ."rc adjust 7  
But answer came there none--  
.rc adjust 8  
And this was scarcely odd, because  
.rc adjust 9  
They'd eaten every one.  
.rc 1 off
```

is formatted as

```
"O Oysters," said the Carpenter,  
"You've had a pleasant run!  
Shall we be trotting home again?"  
But answer came there none--  
And this was scarcely odd, because  
They'd eaten every one.
```

```
/   
/   
/   
/   
/   
/
```

Revision codes are not printed if there is not room for the revision field in the margin.

- Revision codes are not placed on blank lines created with the .SP [Space], .SK [Skip], or .BL [Blank Line] control words. If you want a revision code to print on a blank line, use system symbol &\$RB to create the blank line. For example,

```
.rc align left
.rc 1 *
.rc 1 on
There is no revision code on the next line.
.sp 1
But there is one on the following line.
.br
&$RB
.br
.rc 1 off
```

is formatted as

```
*   There is no revision code on the next line.
```

```
*   But there is one on the following line.
```

```
*
```

- If ALIGN OUTCOL is specified, two column revision codes that alternate by column are placed to the left of all lines in the first column of all pages, and they are placed to the right of all lines in the second column of all pages. For example,

```
.cd 2 0 22p
.cl 20p
.rc align outcol
.rc 1 *
.rc 1 on
This text is in the first column of a two-column example.
Margin revision characters are placed to the left of all text
lines in the first column and to the right of all text lines
in the second column.
.cb
This text is in the second column of a two-column example.
Margin revision characters are placed to the left of all text
lines in the first column and to the right of all text lines
in the second column.
.sp 1
.rc 1 off
```

is formatted as

```
*   This text is in the first column of a two-column
*   example. Margin revision characters are placed
*   to the left of all lines in the first column and to
*   the right of all lines in the second column.
```

```
   This text is in the second column of a
   two-column example. Margin revision characters
   are placed to the left of all lines in the first
   column and to the right of all lines in the second
   column.
```

```
*
*
*
*
*
```

---

## **.RD [Read Terminal]**

### **Function**

The .RD [Read Terminal] control word allows you to stop a typewriter terminal during SCRIPT/VS output so that you can type something. SCRIPT/VS does not process this text in any way, but resumes its output when you signal ATTENTION.

### **Syntax**

►► .RD ◀◀

### **Notes**

- .RD ensures that the page is started.
- .RD is ignored for non-2741 devices.

### **Remarks**

1. The .RD control word is meaningful only when the formatted output is actually being typed at your terminal in interactive environments. The text typed is not processed by SCRIPT/VS, but appears in the output exactly as it was typed.
2. The .RD control word causes continuation of text lines before and after it, if any. Because SCRIPT/VS does not examine the text entered with .RD in any way, any formatted text to the right of the .RD on the same line is shifted to the right by the width of whatever you enter.
3. If the output is not being typed at a terminal, SCRIPT/VS does not stop in the middle of an output line to accept more text. In this case, the only effect of the .RD control word is to cause continuation of the text surrounding it.
4. When using the .RD control word under CMS, specify  
cp term attn off  
before invoking SCRIPT/VS, to suppress CP's normal attention acknowledgment.
5. The .RD [Read Terminal] control word is recognized only in the CMS environment.
6. Typed terminal input can be read from a disk file if the terminal input file name DSMTERMI has been associated with the file or data set name with the .DD [Define Data File-id] control word. See the descriptions of the .DD control word for more information.

---

## **.RE [Restore Environment]**

### **Function**

The **.RE [Restore Environment]** control word restores the **SCRIPT/VS** formatting environment as previously saved by the **.SA [Save Environment]** control word.

### **Syntax**

►► **.RE** envname ◀◀

### **Parameters**

*envname*     Name of an environment, as saved by the **.SA [Save Environment]** control word. It can contain a maximum of 16 national characters.

If *envname* is not specified, the most recently saved *unnamed* environment is restored.

### **Remarks**

1. The **.RE** control word restores the **SCRIPT/VS** formatting environment from a *named* saved environment or from the last-in-first-out stack created by the **.SA** control word. The **.RE** control word restores the **SCRIPT/VS** variables to values that were in effect at the time of the corresponding **.SA** control word. Refer to “**.SA [Save Environment]**” on page 323 for additional information.
2. If no **.SA** control word is currently active the **.RE** control word restores the initial values.
3. Ordinarily, **.RE** restores the designated environment without causing a break. Exceptions to this rule are as follows:
  - If the column definition changes as a result of a **.RE** control word, then a break occurs as if a new column definition had been specified with the **.CD [Column Definition]** control word.
  - If the format changes as a result of a **.RE** control word, then a break occurs as if a new format mode had been specified with the **.FO [Format Mode]** control word.
4. If the formatting mode (set by the **.FO [Format Mode]** and **.NF [No Formatting]** control words) or text alignment (set by the **.CE [Center]** or **.RI [Right Adjust]** control words) changes as a result of the **.RE** control word, a break is performed before the environment is restored.
5. When the environment is saved with the **.SA [Save Environment]** control word, many formatting values are also saved for the current line being built. If these values are restored while another line is being built, **SCRIPT/VS** might give unexpected results. You can avoid this problem by entering a **.BR** control word before the **.SA** control word that saves the environment and before the **.RE** control word that restores it.

### **Examples**

- To restore the most recently saved *unnamed* environment, enter  
`.re`
- To restore the *named* environment **RACHEL**, enter  
`.re rachel`

---

## .RF [Running Footing]

### Function

Use the .RF [Running Footing] control word to identify lines of text to be saved as a running footing for subsequent pages.

### Syntax



### Parameters

- ODD** Specifies that the following lines are to be saved as the running footing only for odd-numbered pages. If neither ODD nor EVEN is specified, the running footing is formatted on every page.
- EVEN** Specifies that the following lines are to be saved as the running footing only for even-numbered pages. If neither ODD nor EVEN is specified, the running footing is formatted on every page.
- ON** Identifies the following lines as a running footing to be saved and placed on an even-numbered page, odd-numbered page, or on every subsequent page, depending on whether EVEN or ODD is specified. ON is the default.
- OFF** Indicates that the definition of the running footing is ended.
- CANCEL** Can be used with the ODD or EVEN parameter, or by itself to cancel running footings defined with the ODD, EVEN, or ON parameters.
- SUP** Can be used with the ODD or EVEN parameter, or by itself to suppress a running footing definition. A suppressed definition still exists, but it is not printed on any page until restored. A suppressed footing definition is deleted if a new one is defined with .RF ODD, EVEN, or ON. A new footing is not suppressed by a previous .RF SUP. To suppress the new definition, you must issue another .RF SUP after the definition has been completed.
- RES** Can be used with the ODD or EVEN parameter or by itself to restore a running footing that was previously suppressed with the SUP parameter.
- EXECUTE** Specifies that the running footing is to be reformatted using the most current definition.

### Initial Setting:

```
.RF ON
.SP 2
.CE - &$PS -
.RF OFF
```

### Notes

- .RF causes a break.
- .RF takes effect on the next page.



## Remarks

1. When using control words that do not cause breaks (for example, .SE and .PN) to change data in the running heading and footing, the effect of those control words may occur earlier than expected if they are encountered while SCRIPT/VS is determining page boundaries. To avoid this, use a .BR before the control words.
2. The running footing is placed on the page immediately above the space that is defined by the .BM [Bottom Margin] control word.

| Description              | Control Word | System Symbol    |
|--------------------------|--------------|------------------|
| Page number symbol       | .DC PS       | &\$PS            |
| Control word separator   | .DC CW       | &\$CW            |
| Continuation character   | .DC CONT     | &\$CONT          |
| GML delimiter            | .DC GML      | &\$GML           |
| GML first end delimiter  | .DC GML      | &\$EGML          |
| GML second end delimiter | .DC GML      | &\$EGML          |
| Markup content separator | .DC MCS      |                  |
| GML prefix               | .GS PREFIX   | First parameter  |
| GML end prefix           | .GS PREFIX   | Second parameter |

*Table 9. Fields of the Active Environment. These fields are carried over from the current environment to the running heading and footing environment. Changes made to these fields in the running headings and footings are not carried back to the normal text environment.*

3. Running footings are formatted using the initial formatting environment, as modified by control words in the running footing. See Table 9 for a list of the fields of the active environment that are carried over from the current environment to the running heading and footing environments.
4. The .RH [Running Heading], .RF [Running Footing], .FN [Footnote], and .Hn [Head Level n].<sup>10</sup> control words are not allowed within a running footing. If one of these control words is encountered within a running footing, the running footing is ended. Macros within a running footing that begin with the characters “rh,” “rf,” “fn,” or “hn” cannot be processed correctly unless the .EM [Execute Macro] control word is specified in front of the macro call. If .EM is not used, the macro causes the running footing to end.
5. Control words that begin a new page should not be used within a running footing. These control words cause a message to be issued and are then ignored. The ignored control words are:
  - .CB [Column Begin]
  - .CC [Conditional Column Begin]
  - .CP [Conditional Page Eject]
  - .IX [Index]
  - .PA [Page Eject]
  - .TC [Table of Contents]

In addition, any of the .H0–.H6 [Head Level 0–6] control words that start a new page are ignored. Note that .CB [Column Begin] and .CC [Conditional Column Begin] are ignored only if they would cause a page eject.
6. Running footings can contain labels and .GO [Goto] control words for iterative processing, can invoke macros, and may contain GML tags.

<sup>10</sup> The .Hn [Head Level n] control word is an obsolete control word. See Appendix A, “Unsupported Control Words” on page 461.

## .RF [Running Footing]

7. Symbols that appear in running footing definitions are replaced with the symbol's value when the running footing is placed on the page. Symbol substitution is automatically disabled while a running footing is being defined.
8. Page number symbols that appear in the running footing are replaced with the current page number. The default page number symbol is an ampersand (&); it can be changed with the .DC [Define Character] control word.

**Note:** If the ampersand (&) is being used as the page number symbol and substitution is turned off inside the running footing, the ampersand in any imbedded symbols is replaced with the page number. This could produce errors or incorrect output.

9. The .RF control word parameters, CANCEL and RES, do nothing if there is no running footing definition to cancel or restore.
10. The .RF control word parameter SUP suppresses the default running footing if a running footing has not been defined. The default running footing contains the page number.
11. When you specify the EXECUTE parameter, the current running footing definition is used to reformat the running footing. The reformatted running footing replaces any existing one. If the running footing contains variable information, the current values are used.
12. If the depth of the new running footing specified with the EXECUTE parameter is different from the original running footing and if the new running footing fits on the current page, it is placed on the current page, and the page body is adjusted accordingly. If the new running footing does not fit, a page eject is performed.
13. The depth of the current running footing can be found in system symbol &\$RF. The value of &\$RF is returned in line spaces of the current font. If &DV' is appended to the front of the symbol name, the depth is returned in vertical device units. For more information on system symbol &\$RF, refer to the "Processing Symbols" chapter in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.
14. The .GR [Group] control word is not allowed inside a running footing.
15. The first line of the running footing has the extra leading specified with the EXTRA parameter of the .LS [Line Spacing] control word removed.

## Examples

- A simple running footing, that places the current page number in the right-hand corner of each page can be entered as

```
.rf on
.sp 2
.ri Page &
.rf off
```

The page number symbol (&) is replaced with the current page number on each page.

- Separate running footings can be defined for odd and even pages. If a document is to be duplexed, these running footings place the page number in the outside corner of each page:

```
.dr thick weight .5mm font &$CHAR(2)
.
.
.
.rf even
.sp
.hr thick left right
Page &
.rf odd
.sp
.hr thick left right
.ri Page &
.rf off
```

The footing for even pages will look like this:

---

Page 312

The footing for odd pages will look like this:

---

Page 313

---

## **.RH [Running Heading]**

### **Function**

Use the .RH [Running Heading] control word to identify lines of text to be saved as a running heading for subsequent pages.

### **Syntax**



### **Parameters**

- ODD** Specifies that the following lines are to be saved as the running heading only for odd-numbered pages. If neither ODD nor EVEN is specified, the running heading is formatted on every page.
- EVEN** Specifies that the following lines are to be saved as the running heading only for even-numbered pages. If neither ODD nor EVEN is specified, the running heading is formatted on every page.
- ON** Identifies the following lines as a running heading to be saved and placed only on an even-numbered, odd-numbered, or on every subsequent page, depending on whether EVEN or ODD is specified. ON is the default.
- OFF** Indicates that the definition of the running heading is ended.
- CANCEL** Can be used with the ODD or EVEN parameter, or by itself to cancel running headings defined with the ODD, EVEN, or ON options.
- SUP** Can be used with the ODD or EVEN parameter, or by itself to suppress a running heading definition. A suppressed definition still exists, but it is not printed on any page until restored. A suppressed heading definition is deleted if a new one is defined with .RH ODD, EVEN, or ON. A new heading is not suppressed by a previous .RH SUP. To suppress the new definition, you must issue another .RH SUP after the definition has been completed.
- RES** Can be used with the ODD or EVEN parameter or by itself to restore a running heading that was previously suppressed with the SUP parameter.
- EXECUTE** Specifies that the running heading is to be reformatted using the most current definition.

### **Initial Setting:**

```
.RH ON
.SP 3
.RH OFF
```

### **Notes**

- .RH causes a break.
- .RH takes effect on the next page.

## Remarks

1. When using control words that do not cause breaks (for example, .SE and .PN) to change data in the running heading and footing, the effect of those control words may occur earlier than expected if they are encountered while SCRIPT/VS is determining page boundaries. To avoid this, use a .BR before the control words.
2. The running heading is placed on the page immediately below the space that is defined by the .TM [Top Margin] control word.
3. Running headings are formatted using the initial formatting environment, as modified by control words in the running heading. See Table 9 on page 311 for a list of the fields of the active environment that are carried over from the current environment to the running heading and footing environments.
4. The .RH [Running Heading], .RF [Running Footing], .FN [Footnote], and .Hn [Heading Level n]<sup>11</sup> control words are not allowed within a running heading. If one of these control words is encountered within a running heading, the running heading is ended. Macros within a running heading that begin with the characters “rh,” “rf,” “fn,” or “hn” cannot be processed correctly unless the .EM [Execute Macro] control word is specified in front of the macro call. If .EM is not used, the macro causes the running heading to end.
5. Control words that begin a new page should not be used within a running heading. These control words cause a message to be issued and are then ignored. The ignored control words are:
  - .CB [Column Begin]
  - .CC [Conditional Column Begin]
  - .CP [Conditional Page Eject]
  - .IX [Index]
  - .PA [Page Eject]
  - .TC [Table of Contents]

In addition, any of the .H0–H6 [Head Level 0–6] control words that start a new page are ignored. Note that .CB [Column Begin] and .CC [Conditional Column Begin] are ignored only if they would cause a page eject.
6. Running headings can contain labels and .GO [Goto] control words for iterative processing, and can invoke macros, and can contain GML tags.
7. Symbols that appear in running heading definitions are replaced with the symbol's value when the running heading is placed on the page. Symbol substitution is automatically disabled while a running heading is being defined.
8. Page number symbols which appear in the running heading are replaced with the current page number. The default page number symbol is an ampersand (&); it can be changed with the .DC [Define Character] control word.
 

**Note:** If the ampersand (&) is being used as the page number symbol and substitution is turned off inside the running heading, the ampersand in any imbedded symbols is replaced with the page number. This could produce errors or incorrect output.
9. The .RH control word parameters CANCEL, SUP, and RES do nothing if there is no running heading definition to cancel, suppress, or restore.
10. When you specify the EXECUTE parameter, the current running heading definition is used to reformat the running heading. The reformatted running heading replaces any existing one. If the running heading contains variable information, the current values are used.

---

<sup>11</sup> The .Hn [Heading Level n] control word is an obsolete control word. See the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for more information.

## .RH [Running Heading]

11. If the depth of the new running heading specified with the EXECUTE parameter is different from the original running heading and if the new running heading fits on the current page, it is placed on the current page, and the page body is adjusted accordingly. If the new running heading does not fit, a page eject is performed.
12. The depth of the current running heading can be found in system symbol &\$RH. The value of &\$RH is returned in line spaces of the current font. If &DV' is appended to the front of the symbol name, the depth is returned in vertical device units. For more information on system symbol &\$RH, refer to the "Processing Symbols" chapter in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.
13. The .GR [Group] control word is not allowed inside a running heading.
14. The first line in a running heading has the extra leading specified with the EXTRA parameter of the .LS [Line Spacing] control word removed.

## Examples

- A simple running heading, that places a security classification at the top of each page can be entered as

```
.rh on
.ce Top Secret
.sp 2
.rh off
```

- The following running heading communicates the same message as the preceding example, but a bit more emphatically:

```
.rh on
.bx left right
.bf hi2
.ce Top Secret!
.pf
.bx off
.sp
.rh off
```

This security classification is placed at the top of each page:

|                    |
|--------------------|
| <b>Top Secret!</b> |
|--------------------|

- Unresolved symbols included in a running heading definition are substituted on each page on which the running heading appears. For example,

```
.dr thick weight .5mm font &$char(2)
.
.
.
.rh on
.sx /Section &section.//Page &/
.hr thick left right
.sp
.rh off
```

The unresolved symbol &section is included in the running heading definition. On each page, &section is replaced with the value of the symbol at the time the page is started. The page number symbol (&) is replaced with the current page number.

For example, if the value of &section is “&@xref(1).,” the running heading is

Section 4

Page 317

---

---

## .RI [Right Adjust]

### Function

Use the .RI [Right Adjust] control word to position an output line flush with the right margin.

### Syntax



### Parameters

- n*** Specifies the number of input lines to be right adjusted. If *n* is omitted, 1 is assumed. If .RI *n* is specified when .RI ON is in effect, right adjusting is turned off when *n* lines have been right adjusted, or when .RI OFF is encountered.
- ON** Specifies that subsequent text lines are to be right adjusted.
- OFF** Terminates right adjust mode if it was ON, or if *n* has been specified and has not been exhausted.
- line*** A line of text to be right adjusted. The *line* is considered to start with the first nonblank character after the .RI control word.

### Notes

- .RI causes a break.
- This is a type 1 control word.
- The line form of .RI starts the page.
- The right adjustment mode is included in the active environment.

### Remarks

1. The parameter ON or OFF, or the number of input lines to be right adjusted (*n*), must be the only parameter on the control word line. A string of words that happens to start with one of these is interpreted as a single line to be right adjusted. For example, the control word lines  
.ri on top of old smokey  
.ri 555 Bailey Ave.  
are taken to be of the “.RI *line*” form, not requests to turn right adjustment on or for 555 lines to be right adjusted.
2. When right adjusting is in effect, no formatting is done on the line. That is, the line is right adjusted as it stands, and it is not filled from other input lines or justified. If a tab character appears in the line to be right adjusted, the tab is resolved before the line is right adjusted.
3. If the line to be right adjusted is longer than the current column line length, the excess words are right adjusted on a separate output line.
4. The use of “.RI *line*” while a “.RI *n*” is still in effect resets to zero the number of lines to be right adjusted.



5. The .CE [Center]. control word is a variant of .RI. If a .CE control word is issued, it cancels a specified .RI; if a .RI control word is issued, it cancels a specified .CE control word.
6. Contrast this control word with .FO RIGHT. The latter allows lines to be formatted by concatenating words until the line is nearly full, but then the filled line is right adjusted instead of being justified, as would be the case with .FO ON.

## Examples

If you wanted to right-adjust some lines, you could enter

```
.ri 3
```

```
These three lines are  
right-adjusted,  
as you can see.
```

and the result would be

```
These three lines are  
right-adjusted,  
as you can see.
```

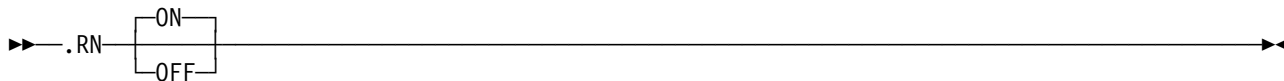
---

## .RN [Reference Numbers]

### Function

Use the .RN [Reference Numbers] control word to control output line reference numbering.

### Syntax



### Parameters

**ON** Starts reference numbering.

**OFF** Stops reference numbering.

**Initial Setting:** OFF

**Default:** ON

### Notes

- .RN ends a keep, float, footnote, named area, or table.
- This control word is ignored for page printers and PostScript devices.

### Remarks

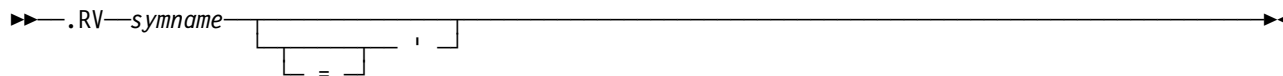
1. This control word takes effect on the current page and affects the entire page. The last .RN control word encountered on a page is used, and any earlier .RN control words on the page are ignored. For example, if .RN OFF is encountered at the top of the page and .RN ON is encountered at the end of the page, reference numbers appear on *all* nonblank lines on the page.
2. Lines are numbered in increments of one starting at one. Numbers are placed about half an inch (12.7mm) to the right of the line length value.
3. Only nonblank lines on the body of the page are numbered. Lines in the running headings and footings, footnotes, and top and bottom margins are not numbered.

## .RV [Read Variable]

### Function

The .RV [Read Variable] control word is similar to the .SE [Set Symbol] control word, except that the value of the symbol is read from the terminal.

### Syntax



### Parameters

*symname* Name of the symbol to be set. *symname* can be any name that would be allowable on the left side of the equal sign in a .SE [Set Symbol] control word.

= ' Indicates that the value set into *symname* is to be treated as a quoted string. If you do not specify the equal sign, SCRIPT/VS provides it automatically and processes whatever string is entered according to the rules for the value on the right-hand side of the equal sign in .SE [Set Symbol] control words. Any value that requires a beginning single quotation mark must have the quotation mark supplied with the .RV control word or as a part of the value entered from the terminal.

### Remarks

1. When the .RV control word is encountered, a line is read from your terminal. This line is used as the right-hand side of the equal sign to set the value of the symbol named in the .RV control word. Any expression that would be allowable as the value in a .SE control word is allowable here. If no parameters are given on the .RV control word, the control word is ignored, and no line is read from the terminal.
2. No message is displayed before the terminal is unlocked to accept the input line. You can use the .TY [Type on Terminal] control word to issue a prompting message before the .RV control word issues its terminal read.
3. The .RV [Read Variable] control word is not supported in the CICS/ATMS-III environment and results in a null value. In batch environments, the .RV control word is ignored unless the file DSMTERMI can be read.
4. When terminal input is from a file, the end-of-file condition is reflected by undefining the symbol *symname*. This is most easily checked with the existence symbol attribute (&E').
5. Variable input can be read from a disk file if the terminal input file name DSMTERMI has been associated with the file or data set name with the .DD [Define Data File-id] control word. See the description of .DD for your environment for more information. If the variable input is to be read from disk, follow the characteristics of an input file.
6. The entire read variable (.RV) input line must not be greater than 256 characters, including the control word, *symname*, equal sign, quotation mark, embedded blanks, and text to be read. For example,

```
.rv inrec = '
|---+---1---+
```

contains 15 characters, which allows a maximum of 241 characters to be read in to complete the entire read variable input line. If the entire input line is greater than 256 characters, truncation occurs from the right, and no message is issued.

## Examples

- A symbol called “name” could be set with the following control word:

```
.se name = 'Sydney Stencil'
```

The same symbol could also be set this way:

```
.rv name = '
```

At this point, SCRIPT/VS issues a read to your terminal, and you can enter the material to be used as the value of the symbol. In this example, you would enter

```
Sydney Stencil
```

You must use single quotation marks in the same circumstances where they would be required in a .SE control word.

- When terminal input is from a file, the end-of-file condition can be detected by testing the existence of the symbol named with .RV. For example:

```
.dd dsmtermi lib datafile
.se count = 0
... loop
.se count = &count. + 1
.rv record&count = '
.if &E'&record&count ne 0 .go loop
```

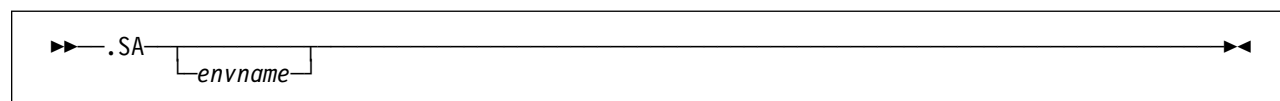
---

## **.SA [Save Environment]**

### **Function**

The .SA [Save Environment] control word saves the SCRIPT/VS formatting environment, which consists of the values and dimensions of certain control words.

### **Syntax**



### **Parameters**

*envname* Specifies the identifier of the environment being saved by the .SA control word. After *envname* has been saved, it may be restored by name using the .RE control word. It can contain a maximum of 16 national characters.

If *envname* is not specified, the current environment is saved as an *unnamed* environment on a last-in-first-out stack.

### **Remarks**

1. The .SA control word saves environments in a stack or by *name*. The .RE [Restore Environment] control word restores the SCRIPT/VS environment to the values that were in effect at the time of the corresponding .SA control word.
2. The .SA control word only *saves* a copy of the values of these SCRIPT/VS variables, it does not *change* any of these variables.
3. Because the .SA control word does not change any of the SCRIPT/VS variable settings, all variables should be explicitly set to the values appropriate unless the current settings are known.
4. The environment saved by a .SA control word is divided into three parts, the active environment, page environment, and translate tables. See “The SCRIPT/VS Formatting Environment” on page 451 for more details on these parts of the environment. The active environment is automatically saved and restored for some formatting functions. It is not necessary to use the .SA and .RE control words within keeps and footnotes unless you want to save and restore values that are not in the active environment, such as .TR [Translate Character] specifications.
5. When the environment is saved with the .SA control word, many formatting values are also saved for the current line being built. If these values are restored while another line is being built, SCRIPT/VS might give unexpected results. You can avoid this problem by entering a .BR control word before the .SA control word that saves the environment and before the .RE control word that restores it.

## Examples

- To save the current environment, enter

```
.sa
```

The current environment is saved in a last-in-first-out stack.

- To temporarily reset the current indentation, enter

```
.sa  
.in 0  
...  
.re
```

The **.RE [Restore Environment]** control word restores the indentation to its value at the time the **.SA** control word was executed.

- To save the current environment by *name*, enter

```
.sa rachel
```

The current environment is saved as RACHEL.

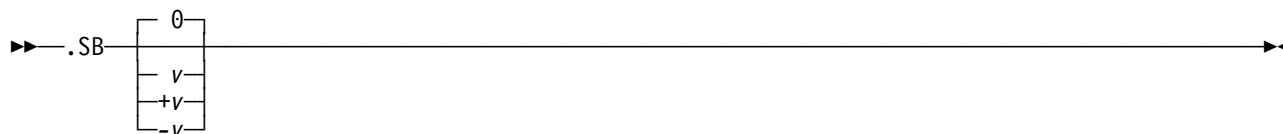
---

## **.SB [Shift Baseline]**

### **Function**

Use the .SB [Shift Baseline] control word to place text above or below the normal baseline.

### **Syntax**



### **Parameters**

- $v$  Specifies the displacement above the *normal* baseline where subsequent text should be placed.  
 $v$  can be any valid vertical space unit.  $+v$  or  $-v$  specifies a displacement above or below the *current* baseline.

**Initial Setting:** 0

**Default:** Return current baseline to normal baseline.

### **Notes**

- .SB is ignored for line devices.
- The baseline offset is included in the active environment.

### **Remarks**

1. Fonts are designed so that the characters appear to rest on the normal baseline. Use the .SB control word to place characters above or below the normal baseline, for example, to create subscripts or superscripts.
2. If no parameters are given, the current baseline is returned to the normal baseline.
3. It is possible, with the .SB control word, to shift text completely off the top or bottom edge of the page. This results in an error.
4. For the 4250 printer, if large baseline shift values are specified during formatting, large amounts of virtual storage are required during printing. This should be avoided.
5. The .SB control word does not affect the depth of a line. As a result, if the vertical displacement is large, text on other output lines might be overlaid.
6. The depth of a line containing shifted text is based on the maximum line spacing for any unshifted text on the line. If the entire line is being shifted, the depth of the line is based on the last line of text that was printed. Text, in this case, refers to actual character text, not blank lines, horizontal rules, and so on.
7. The .LS [Line Spacing] control word, not the .SB control word, should be used to change the amount of vertical space separating output lines.
8. Underscores are not affected by the .SB control word. Use the .UD control word to specify the vertical position of underscores relative to the baseline.

## Examples

- Superscripts can be formatted with baseline shifts. For example, the expression “x-squared” can be formatted as

```
x
.sb .15cm
.ct 2
```

The “2” is placed 1.5 millimeters higher than the “x”:

$x^2$

To restore the normal baseline for subsequent text, enter

```
.sb
```

- Subscripts can also be formatted with baseline shifts. For example, the expression “x-sub-i” can be formatted as

```
x
.sb -p3
.ct i
```

The “i” is placed three pica points lower than the “x”:

$x_i$

- Sub-subscripts can be formatted by further shifting the baseline. For example, the expression “x-sub-i-sub-3” can be formatted as

```
x
.sb -p3
.ct i
.sb -p2
.ct 3
```

The “i” is placed three points lower than the “x,” and the “3” is placed 2 points lower than the “i”:

$x_{i_3}$



---

## **.SC [Single Column Mode]**

### **Function**

The .SC [Single Column Mode] control word saves the current column definition and starts a temporary single-column format. The .MC [Multicolumn Mode] control word restores the column definition that was saved by .SC.

### **Syntax**

►► .SC ————— ►◀

### **Notes**

- .SC ends a keep, float, footnote, named area, or table.
- .SC causes a break and an unconditional section break.

### **Remarks**

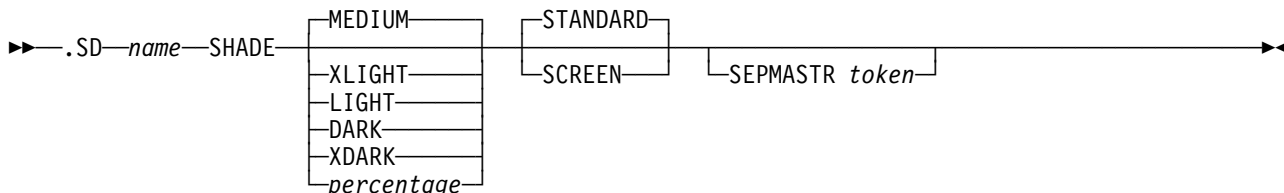
1. The .SC [Single Column Mode] control word temporarily starts formatting in a single column that is the same width as the current .LL [Line Length] specification. The .MC [Multicolumn Mode] control word restores the column definition that was in effect before the .SC was processed.
2. More than one .SC control word can be processed without an intervening .MC. Each .MC control word clears one .SC control word, and, until the first .SC control word in the list is cleared, the column definition restored by each .MC control word is a single-column definition that was set up by an earlier .SC control word.
3. The .CD [Column Definition] control word starts an entirely new column definition and clears all .SC and .MC control words that may be in effect.
4. The .SC control word starts a new section. Therefore, skips inserted by the .SK [Skip] control word before any text is put in the column are discarded, because they would appear at the top of a column.

## .SD [Shading Definition]

### Function

Use the .SD [Shading Definition] control word to define a *named* shade to be used for shading parts of the page with the .BX [Box], .DA [Define Area], and .TD [Table Definition] control words.

### Syntax



### Parameters

**name** Assigns the *name* of the shading definition. The *name* can be a maximum of 16 national characters, and it is not case sensitive.

**SHADE** Shading values can be specified either as parameters or as percentages given as an integer ranging from 0 to 100 (0 means no shading). Enter only the percentage number and not a percent sign.

Valid parameter values are:

| Parameter | Percentage Value |
|-----------|------------------|
| XLIGHT    | 5                |
| LIGHT     | 26               |
| MEDIUM    | 50               |
| DARK      | 74               |
| XDARK     | 100              |

**STANDARDISCREEN** For AFP page printers, select one of two shading patterns (STANDARD or SCREEN). STANDARD is the default. See Figure 4 on page 330 and Figure 5 on page 331 for samples of the shading patterns.

This parameter is ignored for PostScript devices.

**SEPMASTR** Specifies that the shading definition is being associated with output separation masters.

The *token* is a 1- to 8-character name that identifies which separation masters this named shade should be associated with. The token name corresponds to the items selected for separation masters with the .SM control word. You cannot specify OLD COLOR as a token name.

The special token value of ALL is used to indicate that this item should be contained in all separation masters, including the default.

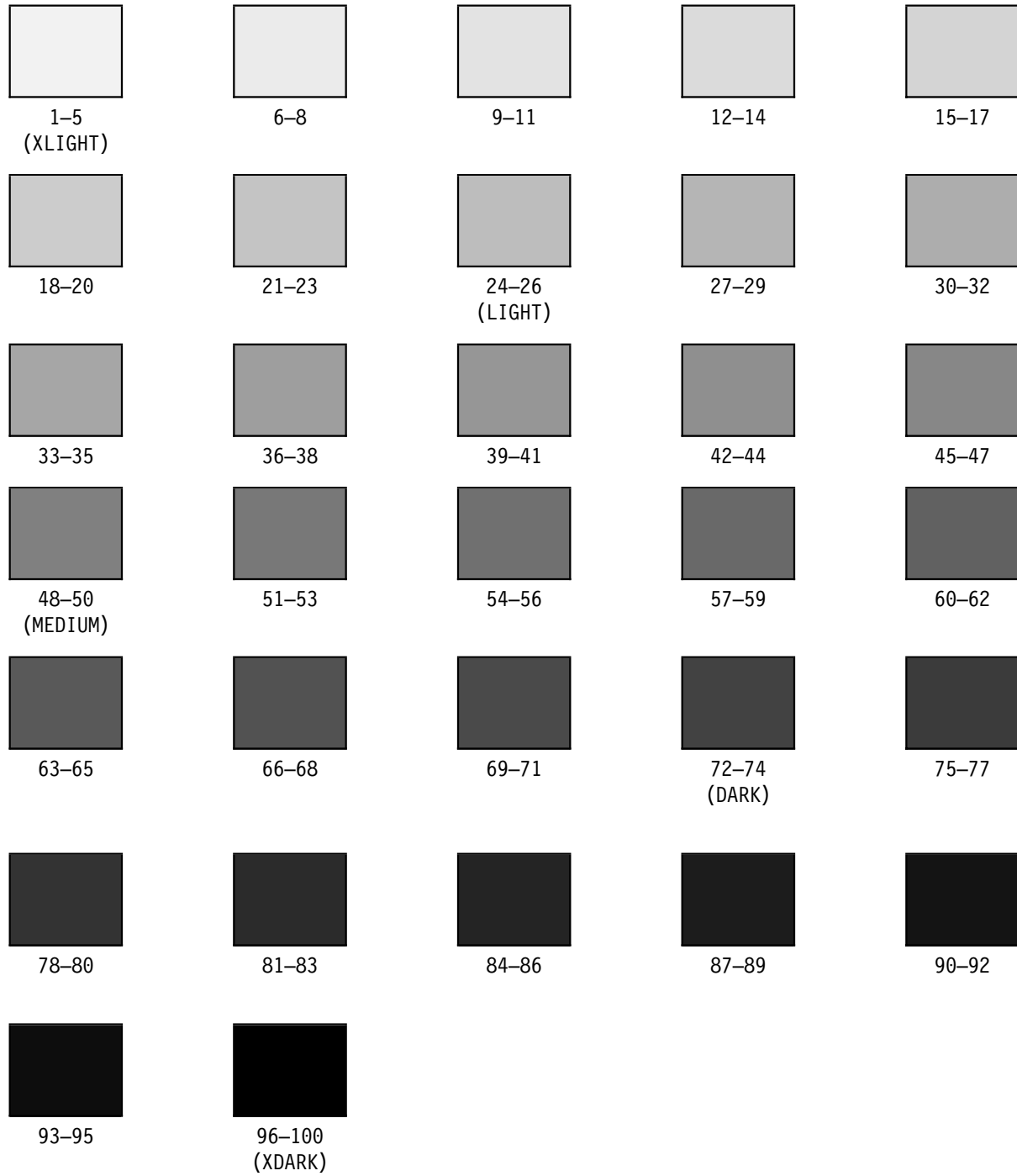
For more information on the SEPMASTR parameter, see “.SM [Separation Master]” on page 353.

## Remarks

1. The .SD [Shading Definition] control word is ignored if you are formatting for a line device or a 4250 printer.
2. Even though you can specify percentage values from 0 to 100, only 32 shading variations are available if you are formatting for an AFP page printer. All 100 values are available for PostScript devices.
3. If you position a shade image close to the right edge or bottom of a page, the shade image does not print if the FORMDEF does not start the logical page at coordinates 0,0.
4. The SEPMASTR parameter is ignored for line devices.
5. The SEPMASTR parameter is also ignored for all devices if the SEPMASTR command option is not specified.

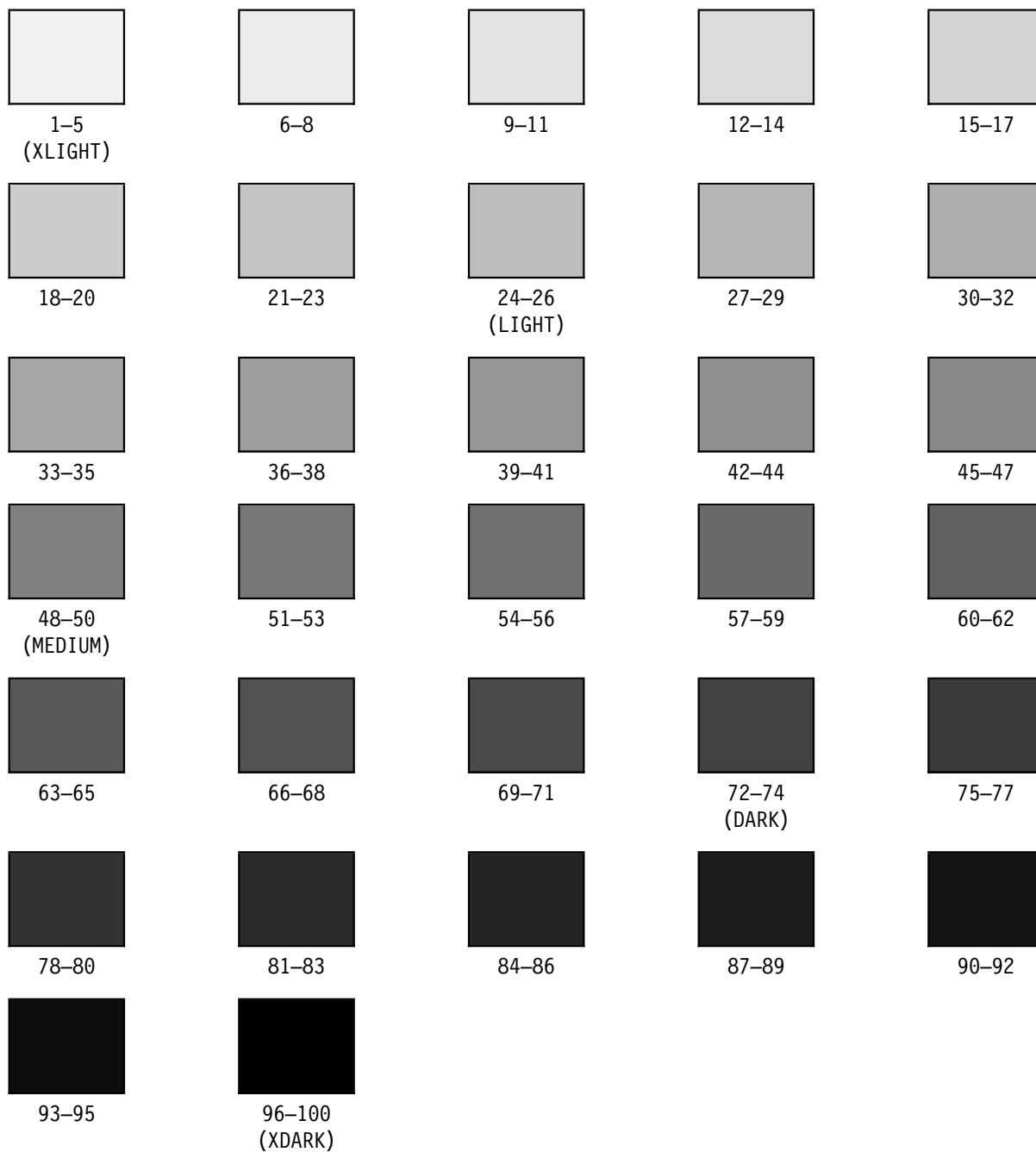
**Note:** Examples using the SEPMASTR parameter can be found in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.

6. Your shading results might vary depending on the output device you are using and the resolution of the device.
7. If you overlap different shading patterns, the result depends on your output device. For PostScript devices, the last shade placed replaces everything under it. Some AFP printers accumulate the different patterns and for others the last shade replaces everything under it. In SCRIPT/VS, such areas are placed on the page from top-to-bottom, left-to-right order, **not** the order they were receive in the input.
8. For PostScript printers, shading can sometimes replace text, rules, other shading, or images if a shift baseline (.SB [Shift Baseline]) control word is used to move vertical rules, boxes, or tables upwards in an area, box, or table that is shaded. Likewise, shifting the baseline upwards and then placing a shaded object on the page could cause that object to replace whatever the shift baseline is shifting over.
9. When formatting for an AFP2xxx logical device, the STANDARD and SCREEN parameters are ignored. The shading pattern is acheived using the HIGHLITE color model.
10. For AFP printers, if you overlap different patterns, accumulation occurs. For PostScript printers, if shades are overlapped, the last shade placed replaces everything under it.



---

Figure 4. Standard Shade Patterns for AFP Devices



---

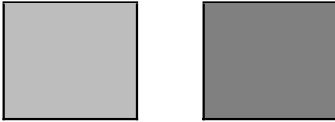
Figure 5. Standard Screen Patterns for AFP Devices

## Examples

- If you define two boxes named `shadbox1` and `shadbox2` as follows

```
.sd shadbox1 standard shade light
.sd shadbox2 screen shade medium
.bx shade shadbox1 5 15 / shade shadbox2 20 30
.sp 3
.bx off
```

the result is



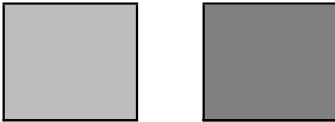
`Shadbox1` has a `STANDARD` pattern with a `LIGHT` shade.

`Shadbox2` has a `SCREEN` pattern with a `MEDIUM` shade.

- The shading values can be specified in two ways. Specify either the parameter value name, such as `LIGHT` or `MEDIUM`, shown in the prior example or specify the percentage value assigned to the parameter, such as 26 or 50, shown in the following example:

```
.sd shadbox1 standard shade 26
.sd shadbox2 screen shade 50
.bx shade shadbox1 5 15 / shade shadbox2 20 30
.sp 3
.bx off
```

which results in



The shading results are the same whether you specify the parameter or its associated percentage.

- If you define an area named `area` and a shade named `sdarea` as follows

```
.sd sdarea standard shade light
.da area 5p0 54p page shade sdarea
.ar area on
.bx left right
.ce This is a defined area shaded with a STANDARD pattern and LIGHT shading.
.bx off
.ar off
```

the result is

This is a defined area shaded with a `STANDARD` pattern and `LIGHT` shading.

- In the following example, the .TD table cell named sdcell1 is shaded with a STANDARD pattern and LIGHT shading, and the .TD table cell named sdcell2 is shaded with a STANDARD pattern and 14 percent shading:

```
.sd sdcell1 standard shade light
.sd sdcell2 standard shade 14
.td table xmp5
.td row xmp5 arrange (1 2)
.td cell 1 xmp5 shade sdcell1
.td cell 2 xmp5 shade sdcell2
.ta table xmp5 on
.in .li
.ir .li
.ta row xmp5 on
.ta cell 1
The first cell in the table shaded
with a STANDARD pattern and LIGHT shading.
.ta cell 2
The second cell in the table shaded
with a STANDARD pattern and 14 percent shading.
.ta table off
```

The above input results in

|                                                                                  |                                                                                        |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| The first cell in the table shaded with a<br>STANDARD pattern and LIGHT shading. | The second cell in the table shaded with a<br>STANDARD pattern and 14 percent shading. |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|

## .SD [Shading Definition]

- For AFP printers, if you define a table with a cell with a SCREEN pattern and MEDIUM shading placed over a box with a SCREEN pattern and LIGHT shading, as follows

```
.sd sc31 screen sh light
.bx sh sc31 .5i 4.5i
.sp 2
.sd sdcell1 screen shade medium
.td table xmp5
.td row xmp5
.td cell 1 xmp5 shade sdcell1
.ta table xmp5 on
.in .1i
.ir .1i
.ta row xmp5 on
.ta cell 1
Screen xlight
.ta table off
.sp 2
.bx off
```

the result is



The patterns have been created so that lighter variations of the same patterns overlay darker variations with no effect. In other words, they are not accumulative.



- For AFP printers, if you define a table with a cell with a SCREEN pattern and MEDIUM shading placed over a box with a STANDARD pattern and MEDIUM shading, as follows

```
.sd sc31 standard sh medium
.bx sh sc31 .5i 4.5i
.sp 2
.sd sdcell1 screen shade medium
.td table xmp5
.td row xmp5
.td cell 1 xmp5 shade sdcell1
.ta table xmp5 on
.in .1i
.ir .1i
.ta row xmp5 on
.ta cell 1
Screen medium
.ta table off
.sp 2
.bx off
```

the result is



For AFP printers, when shaded portions with different patterns overlap, accumulation occurs. In the portions of the table that overlap the box, accumulation occurs because a STANDARD pattern overlapped a SCREEN pattern.

## .SD [Shading Definition]

- For PostScript printers, if you define a table with one cell shaded with a STANDARD pattern and LIGHT shading and another cell with a SCREEN pattern and 11 percent shading placed over a box with differing patterns and shades, as follows

```
.sd sc31 sh 31
.sd st16 sh 16
.sd sc11 sh 11
.sd st23 sh 16
.sd sc9 sh 9
.bx sh sc31 .5i sh st16 1 .5i sh sc11 2 .5i sh st23 3 .5i sh sc9 4i 5i
.sp1
.sd sdcell1 shade 11
.sd sdcell2 shade 20
.td table xmp5
.td row xmp5 arrange (1 2)
.td cell 1 xmp5 shade sdcell1
.td cell 2 xmp5 shade sdcell2
.ta table xmp5 on
.in .1i
.ir .1i
.ta row xmp5 on
.ta cell 1 The first cell in the table shaded with 11% shading.
.ta cell 2 The second cell in the table shaded 20% shading.
.ta table off
.sp 1
.bx off
```

the result is

|                                                      |  |  |                                                  |  |
|------------------------------------------------------|--|--|--------------------------------------------------|--|
|                                                      |  |  |                                                  |  |
| The first cell in the table shaded with 11% shading. |  |  | The second cell in the table shaded 20% shading. |  |
|                                                      |  |  |                                                  |  |

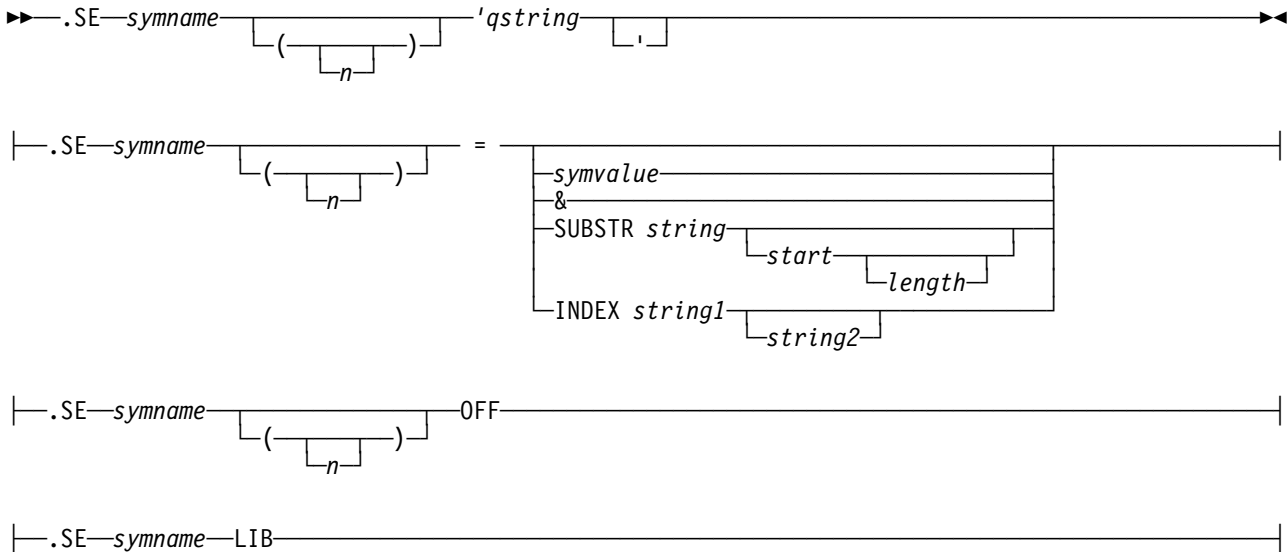
In the portions of the table that overlap the box, accumulation occurs where the patterns types are different and the darker shade prevails where the patterns types are the same.

## .SE [Set Symbol]

### Function

The .SE [Set Symbol] control word allows you to define and assign values to symbols or arrays of symbols.

### Syntax



### Parameters

- symname** The name to which you want to assign a symbolic value to be substituted during SCRIPT/VS processing. It can contain a maximum of 10 nonblank characters.
- n** You can specify a line number in parentheses for array symbols, except when you use the LIB parameter. An array line number is also called an element number or a subscript.
- qstring** Any string preceded by a single quotation mark. If the string has trailing blanks, the string must be terminated by a quotation mark. The *qstring* parameter can contain single quotation marks; these should *not* be doubled, as in the *symvalue* form of .SE.
- symvalue** Assigns a value to the symbol name; it can be a character string or an arithmetic expression.
- &** Assigns the symbol name a value equal to the current page number string.
- SUBSTR** Obtains the specified characters (substring) from a given string and assigns them to the symbol provided.
- string** The *string* from which the substring is to be extracted.
- start** A positive integer that defines the position of the beginning character in the string that is to be assigned to the symbol. If both *start* and *length* are omitted, the symbol is assigned the entire value of *string*.
- length** Specifies the number of characters to be extracted from string, in other words, the length of the substring. If *length* is omitted, the remainder of the *string*, from the specified start to the end, is assigned to the symbol.
- INDEX** Searches the string *string1* to see if it contains the string *string2*. If it does, the symbol value is set to the position of the starting character of *string2* within *string1*.

## .SE [Set Symbol]

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>string1</i> | A string that is to be searched to see if it contains the string <i>string2</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <i>string2</i> | A string that is to be searched for in the string <i>string1</i> . If <i>string2</i> is omitted or has a null value, the symbol is set to 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>OFF</b>     | Unsets the symbol so that, to SCRIPT/VS, it was never set. An entire array symbol is set off if no subscript is provided; however, only the specified element is set off if a subscript is provided.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>LIB</b>     | Causes the symbol to be set by retrieving its value from a library. The name of this library can be defined using the LIB option on the SCRIPT command. If LIB is used to set a symbol, the value retrieved from the library replaces the current value. If no entry with the symbol name given exists in the library, the symbol is undefined. Because symbol names in the library are only in uppercase, the same member of the library is used to define all symbols of the same name that differ only in the case of the characters used. Subscripted symbol names cannot be used with LIB. LIB can be used regardless of the most recent specification of the .LY control word. |

### Symbol Names:

The character “\*” has a special meaning as the first character of a symbol name, because it denotes that the symbol is a local *symbol*. This means that it has only the value that is set at the current level of macro nesting. Every macro that is called has its own set of local symbols for the duration of that macro's execution.

During SCRIPT/VS processing, a symbol name is recognized when it is preceded by an ampersand (&) and followed by a blank or a period:

&symname

If the symbol name appears in any of the following forms:

&symname()  
&symname(\*)  
&symname(*n*)  
&symname(&symbol.)

it is an array symbol.

SCRIPT/VS also recognizes symbol names that are preceded by the GML delimiter, initially and by default, the colon (:). The name of a symbol defined with the .SE control word as a GML tag must be all in uppercase characters, and it cannot be an array symbol. The GML delimiter causes SCRIPT/VS to search for a symbol name that is all uppercase, regardless of the case of the name in the input line.

### Symbol Values:

If a symbol value is set to a character string that contains any imbedded blanks or any special characters, it must be enclosed in single quotation marks. For example,

```
.se dog = cat  
.se end = '.qu'  
.se sentence = 'This is a sentence.'
```

are all valid character strings. A special character is any character *other* than a–z, A–Z, 0–9, #, @, and \$.

If you want a character string to contain a single quotation mark ('), you must enter two of them, for example

```
.se title = 'Mrs. O''Grady''s Cat'
```

unless you use the form:

```
.se title 'Mrs. O'Grady's Cat'
```

If you want to use the INDEX or SUBSTR parameter of the .SE control word to operate on a portion of the string “Mrs. O’Grady’s Cat,” which is the value of the symbol &title, you should turn substitution OFF with the .SU [Substitute Symbol] control word before issuing the .SE control word. If substitution is ON, the control word line

```
.se syma = index &title Cat
```

is substituted for

```
.se syma = index Mrs. O'Grady's Cat Cat
```

before the .SE function is performed. Therefore, the string *Mrs.* is searched for an occurrence of *O’Grady’s* and the remainder of the line (Cat Cat) is ignored. Because the string *O’Grady’s* does not occur in the string *Mrs.*, the symbol &syma is set to zero. If substitution is OFF, the .SE control word receives the line in its unsubstituted form. Even though substitution is OFF, the .SE control word can retrieve symbol values when it recognizes symbol names on the right side of the equal sign. In this case, the .SE control word knows that the entire value of the symbol &title constitutes the string1 parameter for this .SE control word.

If the symbol value is an arithmetic expression, it must be in the form:

```
[op1] n [op2 n op2 n op2 n...]
```

where:

**op1** is an unary + or – sign.

**op2** is an arithmetic operator:

```
+  addition
-  subtraction
*  multiplication
/  division
```

*n* is a valid integer of fewer than 9 digits. Lengths greater than this can produce unpredictable results. The integers might have been assigned their values as a result of a symbol substitution (including the page number symbol).

For example,

```
.se nextpage = & + 1
.se current = -100
.se addit = &current + 25
.se answer = 15 - 42
```

are all valid arithmetic expressions.

## Remarks

1. If you are setting symbols that are going to occur in the running heading or running footing, and the .SE control word is encountered while SCRIPT/VS is determining the page boundaries, the symbol value may change earlier than you expect. To avoid this, use a .BR control word before the .SE control word.
2. In symbol names, uppercase and lowercase letters are considered to be different; thus the symbols symbol1, Symbol1, and SYMBOL1 are three distinct symbols. Symbols whose names start with the dollar sign (\$) are system symbols, and they exist only in an uppercase form. These symbols can be entered in lowercase, but they are folded to uppercase before being resolved. The reserved system symbols that cannot be set by the user, such as &\$RET, are in this category.

## **.SE [Set Symbol]**

Although you can set a symbol whose name starts with \$ if that name is not in use as a read-only system symbol, this is discouraged; confusion can occur because of the name folding.

3. The SYSVAR option of the SCRIPT command causes certain symbols to be preset. These variables are named &SYSVARx, where “x” is the single letter given with the SYSVAR option, capitalized. For example, if you specify the following SYSVAR option with the SCRIPT command:

```
script ... sysvar(1 6I m 11I) ...
```

this is equivalent to specifying:

```
.se SYSVARL = '6I'  
.se SYSVARM = '11I'
```

4. An iterative substitution, as described in the .SU [Substitute Symbol] control word section, is automatically performed on all character string symbol values.
5. If the symbol value is omitted, the symbol's value is set to a null character string (length zero).
6. If a symbol name is set equal to the current page number

```
.se refer = &
```

the symbol is actually set twice. The symbol is first set when it is encountered in the input document. The symbol is set again when the output line with which it is associated is placed on the page. The symbol is associated with the next output line placed after the .SE control word is encountered. The next output line placed might not be the next line after the .SE control word in the input document. At least one word of text should follow the .SE control word to help avoid undesirable page number references. However, if text following a .SE control word is never placed (for example, text in an unplaced area), the symbol is not reset. If the symbol is referred to before the second setting, the value may be inaccurate.

7. The symbol for the current page number, &, remains the same even if the page number symbol that is used in running headings and footings is changed with the .DC PS [Define Character] control word.
8. Arithmetic expressions in set statements are evaluated strictly from left to right, and no operator takes precedence over another. For example, the expression

```
.se x = 1 + 2 * 4 + 6
```

sets the symbol x to the value 18.

9. The LIB option of the .SE control word allows a symbol to be explicitly retrieved from the library. The .LY [Library] control word allows a symbol value to be retrieved from the library when it is *used* in a document and when a value for it does not currently exist. When a symbol value is once retrieved from the library, it is stored in the symbol table for future use.
  10. You should be careful when using local symbols and page number symbols in arithmetic set statements with the multiplication operand (\*). The expression
- ```
.se a = &*3+1
```
- would be taken as a request to add 1 to the value of the local symbol &\*3, *not* as a request to multiply the page number by 3 and then add 1. To achieve the latter effect, the page number symbol must be delimited:
- ```
.se a = &.*3+1
```
11. When symbol substitution is ON, a .SE control word line is completely substituted before it is processed. When substitution is OFF, the .SE control word can still perform individual substitutions on symbolic values in the control word.

**Note:** Be cautious when you use the .SE control word in running headings and footings if symbol substitution is turned off. Such a condition results in errors if a symbol is referred to with the .SE control word. For example, if substitution was off and you specified

```
.se a = &b
```

then the ampersand (&) would be replaced with the current page number and you would get either incorrect output or an error message.

See the following remarks for examples.

12. Be careful of the effects of substitution on arithmetic set statements when symbols that contain negative numbers are used. For example,

```
.se a = -3
.se b = 5+&a
```

results in an invalid expression if substitution is on, as the line is substituted as

```
.se b = 5+-3
```

which is invalid. However, if substitution is OFF, the .SE control word processor can add 5 to -3 and can do it correctly.

13. Even when symbol substitution has been turned off with the .SU [Substitute Symbol] control word, symbols on a .SE control word line are substituted if they are not enclosed in single quotation marks. For example:

```
.se s = zot!
.su off
.se foo = &s
.se baz = '&s.'
.su on
.ty &v'&foo
.ty &v'&baz
```

The value of the symbol &foo is “zot!,” but the value of the symbol &baz is “&s.”

14. Substitution also has an effect on the .SE control word if strings are to be set that are longer than 16 characters. SCRIPT/VS treats a single character string without special characters or blanks as a character string even if it is not enclosed in quotation marks, if it is not more than 16 characters long. If the string is longer than this, an error results. For example:

```
.se a = '12345678901234567890'
.se b = index &a 1
```

results in an error if substitution is on because the symbol &a is not enclosed in quotation marks. The error does not happen if substitution is off.

15. The SCRIPT/VS system symbol &\$C256 contains all 256 character code points in collating sequence, including the control word separator character and single quotation mark (&ssq). It cannot, therefore, be used with the .SE control word while symbol substitution is on. For example, to extract the 112th character from &\$C256, you must turn off symbol substitution:

```
.su off
.se qmark = substr &$C256 112 1
.su on
```

16. See the description of the .SU [Substitute Symbol] control word for more information about symbol substitution.

## .SE [Set Symbol]

17. To cause an ampersand (&) to be treated as text on the right-hand side of a .SE control word, enclose it in single quotation marks as follows:

```
.se sym = '&'
```

Otherwise, the ampersand is replaced with the current page number, regardless of the setting of the page number symbol (.DC PS). If the symbol, as defined above, is to be used as the value for another symbol, the above symbol name must be enclosed in single quotation marks when given on the right-hand side of the .SE control word used to define the new symbol.

18. There can be up to 32767 elements in a symbol array. Attempting to set any higher numbered array elements causes an error.

## Examples

- Symbol substitution is performed iteratively to resolve intermediate symbols created by symbol substitution. For example, the value of the SCRIPT/VS system symbol &SYSMONTH is an integer indicating the month of the year. &SYSMONTH can be used with the following symbols:

```
.se month01 = January
.se month02 = February
...
.se month12 = December
```

The name of the current month can be assigned to a symbol by entering

```
.se thismonth = &month&SYSMONTH
```

Symbol substitution is performed twice for this control word, because the first substitution yields an intermediate result containing another symbol:

```
.se thismonth = &month08
```

The result of the second symbol substitution contains no further symbols; the .SE control word is then processed to create the symbol &thismonth:

```
.se thismonth = August
```

- The INDEX and SUBSTR parameters can be used together to process control information. For example, the SYSVAR option of the SCRIPT command can be used to specify formatting parameters, and you can define SYSVAR “C” to establish the number of columns. You can validate that a permissible value has been given with the INDEX parameter:

```
.se x = index '-one-two-1-2-' '-&SYSVARC.-'
.if &x eq 0 .mg /e/SYSVAR C invalid.
```

If the value given in &SYSVARC is valid, you can use the symbol &x, set with the INDEX parameter, and the SUBSTR parameter to convert synonyms of valid values to a standard value:

```
.se cols = substr '1---2---1-2' &x 1
```

- When the values of two symbols are concatenated, as in the following example,

```
.se a1 = 'aard'
.se a2 = 'vark'
&a1.&a2
```

which results in the string *aardvark*, the period must be used as the symbol delimiter.



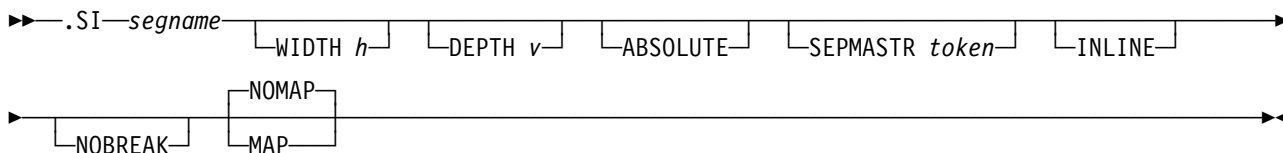
## .SI [Segment Include]

### Function

Use the .SI [Segment Include] control word to identify a segment to be included in the column if the document is printed on a page printer, or to reserve space for cut-and-paste artwork if the document is printed on a line device or PostScript device.

**Note:** The .SI [Segment Include] control word does not allow you to imbed PostScript images. You must use the .PO [PostScript] control word to imbed PostScript images.

### Syntax



### Parameters

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>segname</i>  | <p>An external name that identifies the segment. The SEGLIB option of the SCRIPT command can be used to override the default host system segment library.</p> <p>SCRIPT/VS has different restrictions for <i>segname</i> in the supported environments as follows:</p> <ul style="list-style-type: none"> <li>• In CMS, the <i>segname</i> must be a CMS file name in a segment library.</li> <li>• In TSO, ATMS-III, and MVS, the <i>segname</i> must be the name of a member in a segment library.</li> <li>• In VSE, if .SI [Segment Include] is used, then NOSEGLIB must be specified.</li> </ul> |
| <b>WIDTH</b>    | <p>Horizontal width value. If the <i>segname</i> is found in the segment library, this parameter is ignored unless ABSOLUTE is specified. If the <i>segname</i> is not found in the segment library, this value is used as the width of the segment.</p> <p>The default width value is the current column line length minus the current left and right indention.</p>                                                                                                                                                                                                                                 |
| <b>DEPTH</b>    | <p>Vertical depth value. If the <i>segname</i> is found in the segment library, this parameter is ignored unless ABSOLUTE is specified. If the <i>segname</i> is not found in the segment library, this value is used as the depth of the segment.</p> <p>The default depth value is zero.</p>                                                                                                                                                                                                                                                                                                        |
| <b>ABSOLUTE</b> | <p>This parameter indicates that the width and depth values specified on the .SI control word are to be used instead of the size specified in the segment. If the depth or width is not specified, the default is used.</p>                                                                                                                                                                                                                                                                                                                                                                           |
| <b>SEPMASR</b>  | <p>Specifies that the page segment is being associated with output separation masters.</p> <p>The <i>token</i> is a 1- to 8-character name that identifies which separation masters this page segment should be associated with. The token name corresponds to the items selected for separation masters with the .SM control word. You cannot specify OLD COLOR as a token name.</p> <p>The special token value of ALL is used to indicate that this segment should be contained in all separation masters, including the default.</p>                                                               |

## .SI [Segment Include]

For more information on the SEPMASSTR parameter, see “.SM [Separation Master]” on page 353.

|                |                                                                                                                                                                                                                                                                                                                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>INLINE</b>  | This parameter specifies that the actual contents of the requested page segment, not just its name, should be included in the output data stream that SCRIPT/VS produces for page printers.                                                                                                                                       |
| <b>NOBREAK</b> | <p>Specifies that SCRIPT/VS should <i>not</i> cause a line break either before or after the page segment. The page segment is formatted as if it were a word of text, with the bottom of the page segment positioned on the current baseline.</p> <p>If not specified, a line break occurs before and after the page segment.</p> |
| <b>NOMAP</b>   | Specifies that the page segment <i>segname</i> is not to be loaded in printer storage. NOMAP is the default.                                                                                                                                                                                                                      |
| <b>MAP</b>     | Specifies that the page segment <i>segname</i> is to be loaded in printer storage.                                                                                                                                                                                                                                                |

## Notes

- .SI ensures that the page is started.
- .SI causes a break unless NOBREAK is specified.

## Remarks

1. The segment is centered or left-aligned or right-aligned according to the current text-formatting settings.
2. Page segments must be created with a 0°,90° (inline direction, baseline direction) orientation in order to include the segments in a document.
3. The MAP and NOMAP parameters are ignored when formatting for a device other than a 3820 Page Printer or a 3800 Printing Subsystem Model 3.
4. The MAP and NOMAP parameters are mutually exclusive. The last one specified is used.
5. The MAP and INLINE parameters are mutually exclusive. The last one specified is used.
6. The MAP, NOMAP, and INLINE parameters override the default for the physical device. Unless changed at your installation, the default is NOMAP.
7. Page segments specified as NOBREAK can also be specified as MAP.
8. If MAP is specified for the 3820 Page Printer, the 3800 Printing Subsystem Model 3, or the LaserPrinter 4028, the page segment is downloaded to printer storage at the start of printing the page, unless the page segment already resides in printer storage. If you map a page segment and include it multiple times on the same page, the page segment is downloaded to the printer once and retrieved from printer storage every time it is included on that page. If the page segment is used on subsequent pages but is unmapped, the segment is downloaded each time it is called for, even though it still might reside in printer storage.
9. If a page segment is used only once in a document, do not use the MAP parameter.
10. If too many page segments are mapped on a page or too many fonts are used on the same page, the page might be unprintable. A page is unprintable when Print Services Facility (PSF) cannot load into printer storage all mapped page segments and all fonts used on the page. To correct this problem, use fewer mapped page segments or fewer fonts on the page.
11. When NOBREAK is specified, the segment is treated as a word; therefore, word spaces eligible for justification appear before and after the segment.

If not explicitly set, the line spacing for the output line containing the segment is adjusted to allow for the segment depth.

12. Sufficient space is normally reserved for the segment, so that it does not overlay other text in the column. The space reserved for the segment includes the depth of the segment or the value specified on the DEPTH parameter plus extra leading calculated as follows: the current line space value minus the height of the em in the current font. If line spacing has been set to zero with the .LS [Line Spacing] control word, the segment may overlay preceding text. If a negative baseline shift has been set with the .SB [Shift Baseline] control word, the segment may overlay subsequent text.
13. If NOBREAK is specified, the segment must fit on one line. If not enough room is left on the current line to contain the segment, the entire segment is moved to the next line, regardless of the overdraw option in effect. If the segment is the only “word” on the line and its width exceeds the column line length, the space for the segment extends beyond the column line length.
14. If NOBREAK is specified, the bottom of the page segment is positioned on the current baseline with the rest of the text on the line. However, if the segment does not appear to be positioned correctly because of white space inside the segment, use the .SB control word to shift the segment up or down.
15. The WIDTH and DEPTH parameters are ignored if the named segment is found in the segment library and ABSOLUTE was not specified. In this case, the actual segment dimensions are used for formatting.

The values given with the WIDTH and DEPTH parameters are always used if the segment is not found; this might be because:

- The document is being formatted for a line device.
- The NOSEGLIB option of the SCRIPT command was specified.
- The name given with the .SI control word does not exist in the segment library.

16. The Composed Document Printing Facility (CDPF), which is used to print SCRIPT/VS output on the 4250 Page Printer, does not allow the use of a segment of the same name more than once on the same page. SCRIPT/VS, however, has no such restriction for any printer.
17. If the requested page segment is not found and the INLINE parameter was not specified, SCRIPT/VS still puts the name of the page segment into the output data stream that it produces for page printers. If the requested page segment is not found and the INLINE parameter was specified, SCRIPT/VS does not put the name of the page segment into the output data stream. In either of these cases, SCRIPT/VS issues a message.
18. The segment depth is calculated based on the horizontal and vertical dimensions of the first Image Block (BIM) or Graphics Object (BGR) structured field within the segment. If a segment contains multiple Image Blocks, the Image Blocks after the first one is not used in the depth calculation. Consequently, if text follows a segment that has multiple Image Blocks and is formatted for a page mode device, the text may not be placed correctly. The ABSOLUTE parameter can be used to increase the space reserved.
19. The SEPMASTR parameter is ignored for line devices.
20. The SEPMASTR parameter is also ignored if the SEPMASTR command option is not specified.

**Note:** Examples using the SEPMASTR parameter can be found in the *Document Composition Facility: Text Programmer's Guide*.

21. The SEPMASTR parameter specifies whether or not the segment should be included in an output separation master other than the default master. Segments are included only if the value on the SEPMASTR parameter corresponds to a value selected with the .SM control word.

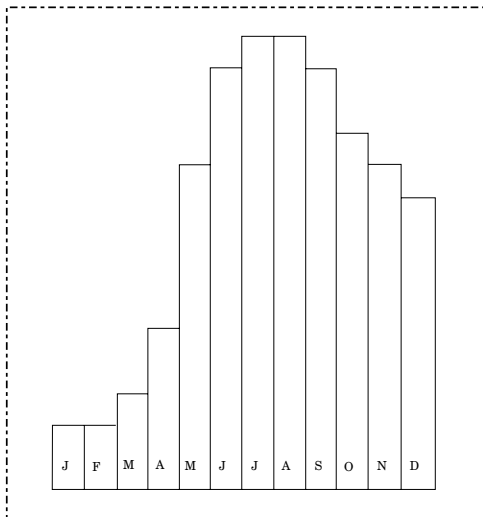
**Note:** Examples using the SEPMASTR parameter can be found in the *Document Composition Facility: Text Programmer's Guide*.

## .SI [Segment Include]

22. If you are printing on a 4224 Printer and the overlay you are including contains shading, or a page segment created for a 3820, the image expands in length and width by 66%.
23. Segments which contain objects which are positioned based on the page coordinate system instead of the current coordinate system are not supported.

## Examples

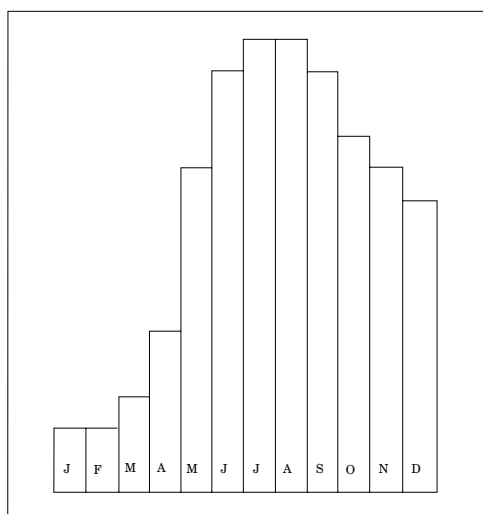
- Segments can be included as part of a figure. For example, suppose the system segment library contains a member named BARCHART, which has been prepared for a page printer:



This segment can be included as the body of a figure:

```
.kp on
.si barchart
Figure 12.
Monthly Incidence of
Reported Yeti Sightings
.sk 1
.kp off
```

When the document is printed, the segment is combined with the figure frame and figure caption:



**Figure 12. Monthly Incidence of Reported Yeti Sightings**

- The .SI control word may be used to reserve space for a segment when a document is formatted for a line device. For example, if the document containing the preceding example is occasionally formatted for a 1403 printer, which does not provide segments, the figure can be specified as

```
.kp on
.si barchart depth 3i
Figure 6.
Monthly Incidence of
Reported Yeti Sightings
.sk 1
.kp off
.li off
```

If the document is formatted for a page printer, the DEPTH parameter is ignored and the actual segment dimensions used. However, if the document is formatted for a line device, the .SI control word reserves three inches of space in the figure.

The &SW' and &SD' symbol attributes can be used to determine the width and depth of a segment. These symbol attributes can be useful when you are trying to dynamically place a segment on the page.

If a segment named seg1 exists, for example, the value of &SW' seg1 is the width of seg1 specified in unqualified horizontal space units. Similarly, &SD' seg1 returns the depth of seg1 in unqualified vertical space units.

If you are formatting for a printer with print resolution that is different from the segment resolution that was specified when the segment was created, the segment width and depth (image size) returned by the &SW' and &SD' symbol attributes might not be the same as the image size that is printed. The image that is printed could be larger or smaller than the width and depth that is returned by the &SW' and &SD' symbol attributes.

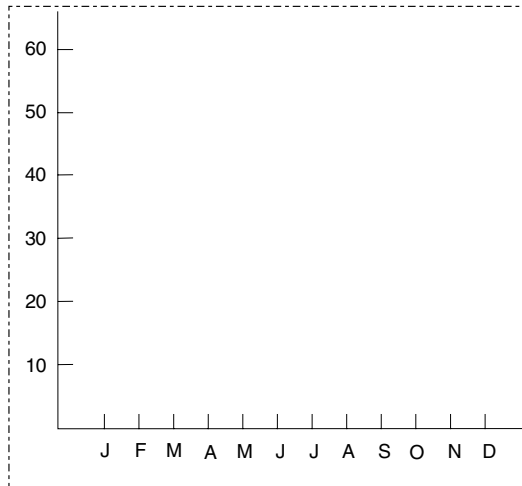
If you are formatting for a line printer or if you specified NOSEGLIB on the SCRIPT command, the value returned by both symbol attributes is 0.

To obtain the width or depth of a segment in device units, thereby avoiding rounding, use &DH'&SW'seg1 and &DV'&SD'seg1.

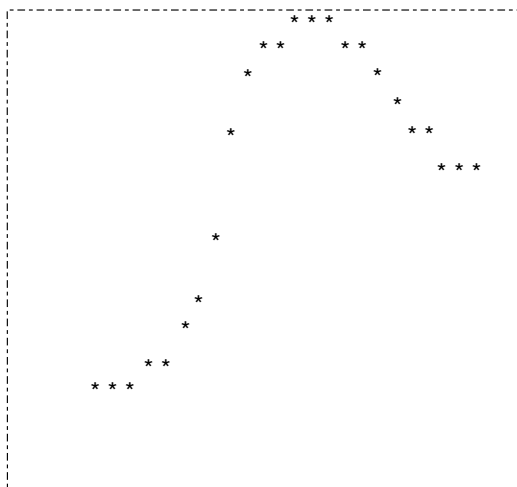
## .SI [Segment Include]

To obtain the resolution of a segment in measurement units, use &SH' and &SV' symbol attributes. For example, &SH' and &SV'seg1 could return measurement units of 240 or 140.

- Segments can overlay each other. For example, suppose the CMS file AXES PSEG3820 contains this image, which has been prepared for the 3820 printer:



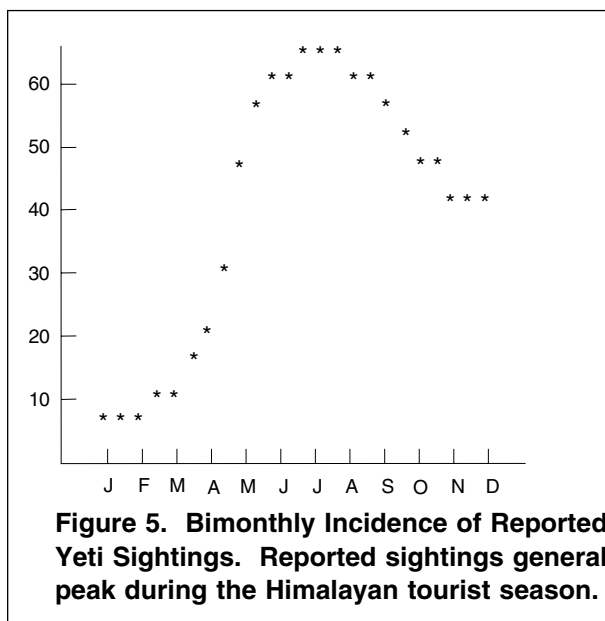
In addition, the CMS file PLOT1 PSEG3820 contains the following image, which has been prepared for the 3820 printer:



These segments can be combined to produce a single figure:

```
.bx thick left right
.si axes
.ls normal 0
.si plot1
.ls
.in +2
Bimonthly Incidence of
Reported Yeti Sightings:
Reported sightings generally peak
during the Himalayan tourist season.
.bx off
.sp
```

When the document is printed, the segments are combined with the figure frame and figure caption:




## **.SI [Segment Include]**

- When using NOBREAK, text does not flow around the segment. That is, if the segment depth is more than one line space and the segment fits on the current output line, there is blank space around the segment. For example,

```
first line of text
.br
this line has the
.si ibm3 inline nobreak
logo imbedded right on the line


looks like this
```

```
first line of text
this line has the

logo imbedded right on the line
```

The word spaces around the page segment can be removed in the following manner:

```
.dc cont +
first line of text
.br
this line has the+
.si ibm3 inline nobreak
.ctlogo imbedded right on the line

and then you get output that looks like this
```

```
first line of text
this line has the 
imbedded right on the line
```



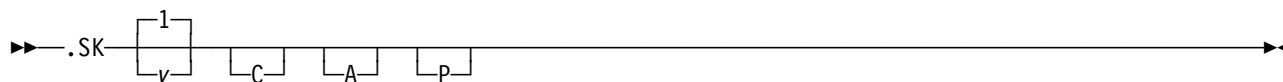
---

## .SK [Skip]

### Function

Use the .SK [Skip] control word to generate blank vertical space before the next text output line, except at the top or bottom of a column or page.

### Syntax



### Parameters

- v** Amount of space to be inserted in the output. If *v* is omitted, 1 is assumed. If the size in *v* is not qualified as any of the other space units (inches, picas, ciceros, or millimeters), it is a request to skip a number of lines. In this case the size of the request is multiplied by the appropriate factor if double-spacing or multiple-spacing is in effect, unless **A** is specified.
- C** Indicates conditional skips. These skips depend upon what follows them in the output column. If conditional skips are followed by a line of text, they appear in the column as requested. If they are followed by another skip or space request, the two skip or space requests are compared, and only the larger of the two remains in the column.
- A** Indicates absolute skips. Line spacing values specified with the BY parameter of the .LS [Line Spacing] control word and with the .DS [Double Space Mode] control word have no effect on the depth of the space skipped.
- The .DS control word doubles the line spacing set by the .LS control word. When double-spacing is in effect, each skip caused by a .SK control word is doubled (thus, .SK 2 yields four spaces). However, if **A** is specified, the space count is not doubled.
- If **A** is specified, the skip is *not* eligible for vertical justification.
- If the size of the skip is given in inches, picas, ciceros, or millimeters, it is taken as an absolute request. .SK .5i would, then, result in a skip of one-half inch, regardless of the line spacing set with .LS, and the actual requested depth is skipped to the closest approximation possible on the current logical device. If **A** is *not* specified, the skip is eligible for vertical justification.
- P** Indicates page skips. These skips generate skip space across the full width of the page, even when formatting in multiple columns. Because this type of skip causes a section break, it is not allowed in a keep.

### Notes

- .SK causes a break.
- .SK ensures that the page is started.
- .SK v P causes an unconditional section break before and after the skip is taken.

## **Remarks**

1. No blank space is generated if it would be printed at the top or bottom of a column of output. The top of a column can be at the top of the page or after a section break. This can result in columns being set short by the amount of discarded blank space. The discarded blank space is white space at the end of a column.
2. A skip at the bottom of a section is normally discarded. When multiple section areas are placed using the .AR PUT control word, all of the section areas are made equal in depth by padding the shorter areas with vertical white space. As a result, a skip at the bottom of the shorter areas is retained and padded if needed, whereas a skip at the bottom of the longest area is discarded.

As many sections as needed are created to place all the text in a section area. A skip falling at the top of any of those sections is discarded. A skip falling at the bottom of any of those sections is treated as described in the previous paragraph.

3. Page skips (the P parameter) are ignored only when they fall at the top of a page.
4. If the skip request is in lines (unqualified space units), the size of each line is as defined with the .LS [Line Spacing] control word.
5. If double-spacing is in effect, the number of skips generated is multiplied by the line spacing amount, unless absolute spacing is specified.
6. Conditional skip processing is not performed across the boundaries of areas, keeps (except vertical keeps of the form  $v+v$  or  $v$ ), floats, footnotes, running headings and running footings, nor is it performed if the skip is a page skip.

---

## **.SM [Separation Master]**

### **Function**

Use the .SM [Separation Master] control word to identify and group document items (text, rules, or graphics) that are to be contained in a specific output separation master.

The separation master function allows you to produce several different masters from one document. These separation masters can be used in an offset printing process to produce color documents or as multi-part forms.

### **Syntax**

►► .SM *n* token ◄◄

### **Parameters**

- n* Specifies the number of this separation master. The value *n* can be any integer from 0 to 16. The value zero represents the default master. The numbers correspond to the suboptions of the SEPMASTR command option, and they are used to select output masters that are produced during the processing of a specific document formatting run.
- token* Specifies names that identify the document items (text, rules, or graphics) that make up this separation master. The *token* names correspond to document elements that are specified with the SEPMASTR parameter on the .DF, .DR, .OI, .PO, .SD, and .SI control words. *Token* names can contain a maximum of 8 characters.

### **Remarks**

1. The .SM control word is ignored for line devices.
2. The .SM control word is ignored for all devices if the SEPMASTR command option is not specified.
3. Any item in the document that does not specify or inherit a SEPMASTR value from one of the following control words is placed in the default master: .DF, .DR, .OI, .PO, .SD, or .SI. You can add additional items to the default master by specifying .SM 0.
4. A .SM control word for a given number remains in effect until explicitly respecified.

For more information on using separation masters, refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.

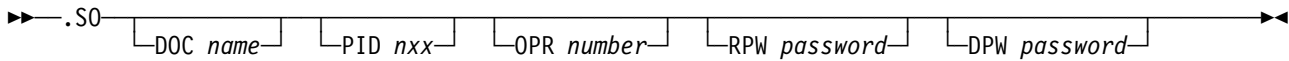
---

## .SO [STAIRS/VS Output]

### Function

Use the .SO [STAIRS/VS Output] control word to set the STAIRS paragraph code. The .SO control word takes effect only when output is being produced for STAIRS.

### Syntax



### Parameters

- DOC** Specifies a document name to be placed in the STAIRS/VS CTF blocks. The name can be up to 12 characters long.
- PID** Specifies the identification to be given to the next paragraph. Blocks are numbered in increments of 1 starting with the low-order position. The first position must be numeric (0–9). The second and third positions are alphanumeric in the ascending order: blank (X'40'), A–Z, 0–9. Lowercase alphabetic characters are folded to uppercase. A blank can be used in the second position only if the third position is blank.
- OPR** Specifies an operator number to be placed in the STAIRS/VS CTF blocks. The number can be any number up to 32,767.
- RPW** Specifies a read password to be placed in the STAIRS/VS CTF blocks. The password can be up to 5 characters long, and the default is blanks.
- DPW** Specifies a delete password to be placed in the STAIRS/VS CTF blocks. The password can be up to 5 characters long, and the default is the read password.

### Notes

- .SO causes a break.
- The DOC parameter starts a new page.

### Remarks

1. If the .SO [STAIRS/VS Output] control is omitted, blocks are numbered starting with zero followed by two blanks.
2. If you are using imbed files and you want the document name to remain constant, specify .SO {DOC name}.
3. If concatenation is off, each line has a separate identification code.
4. At least one pair of parameters must be specified.

## Examples

- Some valid uses of the .SO [STAIRS/VS Output] control word are

.so 34c

.so 0b

.so 20b

- Some examples of the numbering sequence are

|   |     |     |
|---|-----|-----|
| 0 | 0AY | 0A8 |
|---|-----|-----|

|    |     |     |
|----|-----|-----|
| 0A | 0AZ | 0A9 |
|----|-----|-----|

|     |     |    |
|-----|-----|----|
| 0AA | 0A0 | 0B |
|-----|-----|----|

|   |   |   |
|---|---|---|
| . | . | . |
|---|---|---|

|   |   |   |
|---|---|---|
| . | . | . |
|---|---|---|

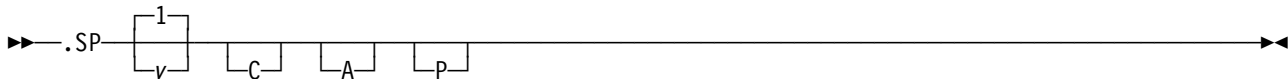
---

## .SP [Space]

### Function

Use the .SP [Space] control word to generate blank vertical space before the next text output line.

### Syntax



### Parameters

**v** Amount of space to be inserted in the output. If *v* is omitted, 1 is assumed. If the size in *v* is not qualified as any of the other space units (inches, picas, ciceros, or millimeters), it is a request to space a number of lines. In this case, the size of the request is multiplied by the appropriate factor if double spacing or multiple spacing is in effect, unless **A** is specified.

**C** Indicates conditional spaces. These spaces depend upon what follows them in the output column. If conditional spaces are followed by a line of text, they appear in the column as requested. If they are followed by another skip or space request, the two skip or space requests are compared, and only the larger of the two remains in the column.

**A** Indicates absolute spaces. Line spacing values specified with the BY parameter of the .LS [Line Spacing] control word and with the .DS [Double Space Mode] control word have no effect on the depth spaced.

The .DS control word doubles the line spacing set by the .LS control word. When double-spacing is in effect, each space caused by a .SP control word is doubled (thus, .SP 2 yields four spaces). However, if **A** is specified, the space count is not doubled.

If **A** is specified, the space is *not* eligible for vertical justification.

If the size of the space is given in inches, picas, ciceros, or millimeters, it is taken as an absolute request. .SP .5i would, then, result in a space of one-half inch, regardless of the line spacing set with .LS, and the actual requested depth is spaced to the closest approximation possible on the current logical device. If **A** is *not* specified, the space is eligible for vertical justification.

**P** Indicates page spaces. These spaces generate space across the full width of the page, even when formatting in multiple columns. Because this type of space causes a section break, it is not allowed in a keep.

### Notes

- .SP causes a break.
- .SP ensures that the page is started.
- .SP v P causes an unconditional section break before and after the space is taken.

## Remarks

1. If double-spacing is in effect, the number of spaces generated is multiplied by the line spacing amount, unless absolute spacing is specified.
2. If the space request is in lines (unqualified space units), the size of each line is as defined with the .LS [Line Spacing] control word.
3. If the space request exceeds the remaining column depth, the space is placed at the top of the next column. If the space request exceeds the maximum column depth, the excess space is discarded.
4. Spacing using the .SP control word for large amounts of space can produce undesirable results if column balancing is in effect, because the space is not split across columns.
5. Conditional space processing is not performed across the boundaries of areas, keeps (except vertical keeps of the form  $v+v$  or  $v$ ), floats, footnotes, running headings and running footings, nor is it performed if the space is a page space.

---

## **.SS [Single Space Mode]**

### **Function**

Use the .SS [Single Space Mode] control word to cancel a previous .DS [Double Space Mode] control word and to resume single-spacing of output.

### **Syntax**

►► .SS ◀◀

### **Remarks**

1. Output following the .SS [Single Space Mode] control word is single-spaced. Because this is the normal output format, .SS is needed only to cancel a previous .DS [Double Space Mode] control word or a setting of the BY parameter of the .LS [Line Spacing] control word.
2. If the previous output spacing was not single-spaced, the resumption of single-spacing is automatically deferred for one line if the next output is a text line. For example, if you entered the following:

```
.ds
These lines will be double-
spaced. They are the last
ones before we enter
the .SS [Single Space Mode]
control word and a bit
more text.
```

```
.ss
Now, we will begin single-
spacing, but we will have
to wait one line for the
deferred single-spacing
to begin.
```

the results look like this

```
These lines will be double-spaced.
```

```
They are the last ones before we enter
```

```
the .SS [Single Space Mode]
```

```
control word and a bit more text.
```

```
Now, we will begin single-spacing,
but we will have to wait one line
for the deferred single-spacing to begin.
```

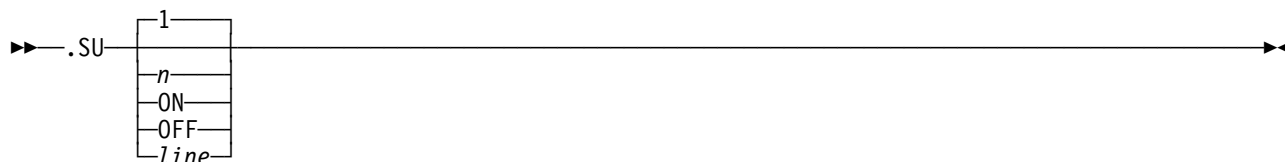


## .SU [Substitute Symbol]

### Function

Use the .SU [Substitute Symbol] control word to cause SCRIPT/VS to stop substitution of defined set symbols or to restore substitution.

### Syntax



### Parameters

- n* Specifies the number of lines to be scanned for set symbols to be substituted.
- ON** Turns on an open-ended substitution mode. ON is the initial setting.
- OFF** Turns off substitution mode if it was on, or if *n* was given and is not yet exhausted.
- line* A *line* containing symbols that you want SCRIPT/VS to substitute with values previously set. Symbols may be set using the .SE, .RV, .IM, or .AP control words, or by a macro call.

**Initial Setting:** ON

**Default:** 1

### Notes

- Symbol substitution is included in the page environment.
- This is a type 1 control word.
- The line form of .SU starts the page.

### Remarks

1. The .SU control word causes a specified number of the following input lines, control words as well as text, to be scanned for defined set symbols. If the parameter ON is in effect, every line up to a subsequent .SU OFF is processed for symbol substitution.  
  
Substitution is turned off with .SU OFF or with .SU *n*, after *n* lines have been read.  
  
If .SU *line* is used while .SU *n* is still in effect, substitution is turned off.
2. When an input line is substituted, each complex symbol can go through several stages of substitution until no further substitution can be done. Any symbol name for which no definition exists is left in the input line as text.
3. The substitution of set symbols can increase or decrease the length of a text line. If a line's length is reduced to zero by symbol substitution, it is processed by the .NL [Null Lines] control word.
4. The TWOPASS or FPASSES option of the SCRIPT command can result in symbols being defined during the first pass that can be substituted during the second, even though these symbols are defined physically later in the SCRIPT file than when they are referenced. If the length of the symbol value

## **.SU [Substitute Symbol]**

and the length of the symbol name are different, the formatting might come out slightly different in each pass.

### **Examples**

- Macro definitions almost always contain symbols, and these symbols are generally meant to be substituted anew each time the macro is executed. To this end, you should ensure that symbol substitution is turned off when you define a macro within a document and is turned on when you execute it. For example, if you enter

```
.se x = 1
.dm count /.se x = &x + 1/.ty &x
```

the number “1” is printed each time the macro is executed, and the value of &x is always “2,” because the symbols in the macro are substituted only once, when the macro is defined. To achieve the desired effect, enter instead

```
.se x = 1
.su off
.dm count /.se x = &x + 1/.ty &x
.su on
```

- If you use the .DM name ON form to define macros, you need not explicitly disable symbol substitution for the definition:

```
.se x = 1
.dm count on
.se x = &x + 1
.ty &x
.dm off
```

The symbol &x is saved as part of the macro definition and substituted whenever the macro is executed.

---

## **.SV [Spelling Verification]**

### **Function**

Use the .SV [Spelling Verification] control word to start and stop spelling verification. This control word must be enabled by the SPELLCHK option of the SCRIPT command. If the SPELLCHK option is not in effect, the .SV control word is ignored.

### **Syntax**



### **Parameters**

- ON** Specifies that spelling verification is to be started. Any addenda dictionary is searched before the user and main dictionaries. Stem processing is performed, but numbers are not checked.
- OFF** Stops spelling verification.
- NOADD** Turns on spelling verification and disables use of the addenda dictionary. Any user dictionaries and the main dictionary are searched. Stem processing is performed but words that contain numeric characters are not checked.
- The addenda dictionary is created using the .DU [Dictionary Update] control word.
- NOSTEM** Turns on verification if it was off and stops the spelling verification function from performing stem processing on words to be verified. The addenda dictionary, any user dictionaries, and the main dictionary are searched. Stem processing is described in more detail in the *Document Composition Facility: Text Programmer's Guide*.
- NUM** Turns on verification if it was off and indicates that spelling verification is to be started for words that contain numeric as well as alphabetic characters. This option allows text that contains numbers to be verified. If ON instead of NUM is specified, only words that contain alphabetic characters only are checked.

**Initial Setting:** ON

**Default:** ON

**Note:** Spelling verification is included in the active environment.

### **Remarks**

- Each time the .SV control word is used, all the settings that control spelling verification are reset. For example,
 

```
.sv noadd
```

 stops spelling checking against the addenda dictionary. If this is followed later in the document by
 

```
.sv num
```

 spelling verification now starts for numbers and is *resumed* from the addenda dictionary.
- When words are found in a document that cannot be verified against the active dictionaries, SCRIPT/VS synthesizes an .UW [Unverified Word] control word and executes it with all of the

## **.SV [Spelling Verification]**

unverified words from a single input line. You can write a .UW macro to perform specialized processing for unverified words, as described in “.UW [Unverified Word]” on page 422.

3. IBM-supplied dictionaries exist for all of the supported languages but the following dictionaries are empty: Danish, Finnish, Icelandic, Norwegian, Portuguese, and Swedish. However, you can create user dictionaries for spelling verification for these languages using the Dictionary Maintenance Program supplied with DCF.

**Note:** For more information about IBM-supplied dictionaries, refer to the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.

### **Performance Considerations:**

When spelling verification is performed, each occurrence of every word in the document being formatted is reduced to its root form and checked against the active dictionaries. This can result in a significant increase in the processing time required to format a document.

Often it is sufficient to perform spelling verification only twice: once, when the document is first created, to find entry errors, acronyms, and valid words that are not in the dictionaries, and again, just before the final formatting runs, to catch any errors made while updating or revising the document.

## **Examples**

Because spelling verification is initially on, all text in a document is checked when it is formatted with the SPELLCHK option of the SCRIPT command. The .SV control word can be used to exclude portions of the document that contain nontextual information. For example, if your document contains a sample basic program, enter the following to prevent spelling verification:

```
.fo off
.sv off
10 READ A(1)
20 IF A=20 THEN GOTO 30 ELSE STOP
30 LPRINT "TESTING"
40 END
.sv on
```

---

## .SX [Split Text]

### Function

The .SX [Split Text] control word is used to split a string of text between the left and right column margins, with a filler between the two.

### Syntax



### Parameters

- F** Allows the left part of a split line to be folded if it does not fit within the column. The fill string and the right part are never folded; they must fit within the column.
- C** Specifies that the middle part of the string, usually the fill string, is to be centered and not replicated.
- /** Any delimiter character. The first nonblank character is taken as the delimiter character.
- lpart* String to be placed against the current left margin.
- fill* String to be used to fill the space between *lpart* and *rpart*. If the fill string is null, blanks are used. The beginning position of the fill string is a multiple of its length from the left margin. Multiples of the fill string are inserted until they would overlap the right part. If the fill string is longer than the space between the strings, it is not used.
- If the **C** parameter was specified, the fill string is centered between the left end of the left part and the right end of the right part, rather than replicated.
- rpart* String to be placed against the current right margin.

### Notes

- .SX causes a break.
- .SX ensures that the page is started.

### Remarks

1. The delimiter character between the strings can be any unique character that does not occur within the strings themselves.
2. Any of the three parts of the line can be null.
3. Split text lines are printed in the font that is in effect when the .SX control word is encountered.
4. Tab characters in a fill string or right part are treated as word spaces.
5. The .PT [Put Table of Contents] control word writes .SX control words into the table of contents file to be processed when the table of contents is formatted. The delimiter used for these internally generated .SX control words is hexadecimal X'00'.

### Examples

- Split text with null fill string  
.sx /left part//right part/

## **.SX [Split Text]**

results in

left part                      right part

- A foldable split text, as used in tables of contents

.of 1

.sx f /An example of a folded split text line/ ./&/

results in

An example of a folded  
split text line      . . . . 364

If the F parameter had not been used, an error message would have been generated. If the EXTEND overdraw option of the .FO control word is in effect, the left part of the split text extends; otherwise, it is truncated at the column line length.

- Split text with null left and right parts:

.sx //-+//

results in

-+-+-+-----+

- Split text with null fill string showing one side of the split text in a different font

.df sxbold type(itc avant garde gothic semibold)

.dv lpart font sxbold /left part

.dv rpart /right part

.sx /&lpart.//&rpart./

results in

**left part**

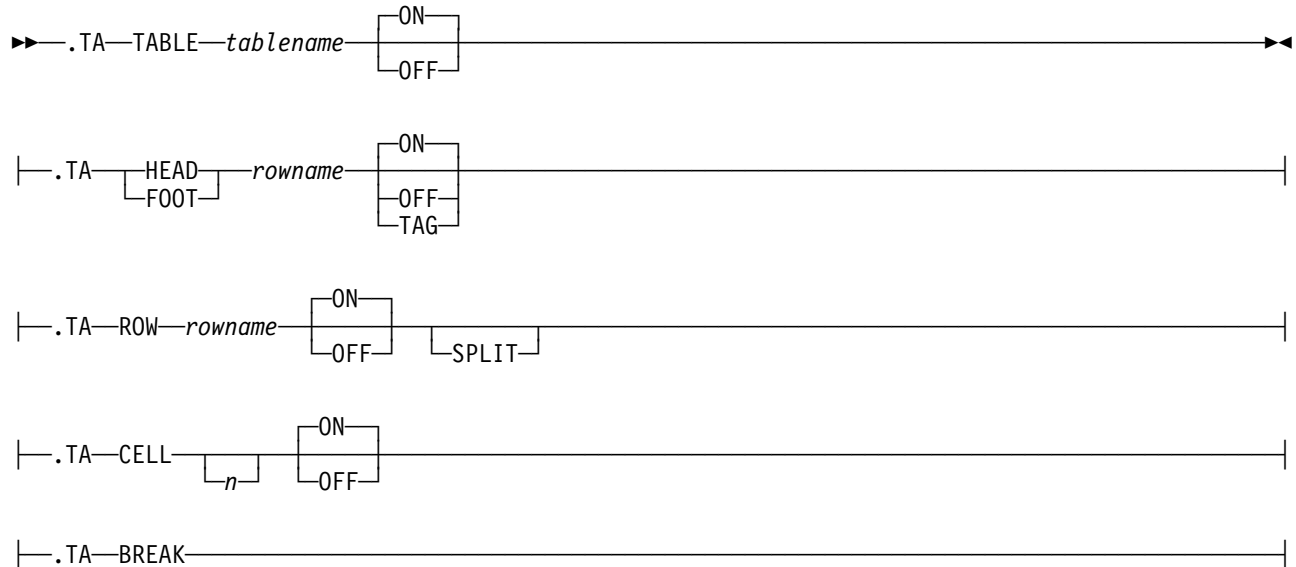
right part

## .TA [Table]

### Function

Use the .TA [Table] control word to start a table, row, cell, table heading, or table footing.

### Syntax



### Parameters

- TABLE** Specifies the start or end of a table. The *tablename* parameter must be specified when the ON parameter is used, but it is optional when the OFF parameter is used. The *tablename* parameter must have been defined with a previous .TD [Table Definition] control word.
- ON** Specifies the start of a table, header definition, footer definition, row, or cell.
- OFF** Specifies the end of a table, header definition, footer definition, row, or cell. If OFF is specified, any other optional parameters are ignored. When the OFF parameter is used, *tablename* and *rowname* are not required.
- HEAD** Specifies the start or end of a table-header definition. The *rowname* must be specified when the ON or TAG parameter is used, but it is optional when the OFF parameter is used. The *rowname* must have been defined with a previous .TD control word.
- FOOT** Specifies the start or end of a table-footer definition. The *rowname* must be specified when the ON or TAG parameter is used, but it is optional when the OFF parameter is used. The *rowname* must have been defined with a previous .TD control word.
- TAG** Specifies that the following text, control words, and tags are to be placed into the table-header or table-footer definition until a GML end tag that matches the start tag that activated the .TA HEAD *rowname* TAG or .TA FOOT *rowname* TAG control word is encountered.
- ROW** Specifies the start or end of a table row. The *rowname* must be specified when the ON parameter is used, but is optional when the OFF parameter is used. The *rowname* must have been defined with a previous .TD control word.
- SPLIT** Indicates that the table can be split prior to this row, if necessary. If the entire table does not fit within the current column, the table is split across pages or columns prior to rows that have specified the SPLIT parameter.

## **.TA [Table]**

- CELL** Specifies the start or end of a table cell. If *n* is given, cell *n* in the current row is started. If *n* is not given, the next sequential cell in the current row is started, starting with the lowest number cell in the row. If the requested cell has not been defined for the current row, an error occurs.
- BREAK** Ends the current row and causes the rows that have been formatted to be placed. BREAK also causes a column eject, and returns the storage that has been used for formatting that portion of the table. SCRIPT/VS then continues processing the table with the current header and footer. .TA BREAK does not end the table.

## **Notes**

- .TA causes a break.
- .TA ensures that the page is started.
- This control word saves the current environment.

## **Remarks**

1. The &\$ENVARRAY system symbol array returns "TA" in the correct position when a table, table header, or table footer is currently active.
2. Tables can occur anywhere boxes can occur. This includes named areas, keeps, floats, and footnotes.
3. When a .TA control word using the ROW parameter is encountered and a table is not currently active, an error occurs. A table must be active before a row is started.
4. When a .TA control word using the CELL parameter is encountered and a row, table header, or table footer is not currently active, an error occurs. A table and a row, table header, or table footer must be active before a cell can be started.
5. A table header or footer can contain only .TA control words that use the CELL parameter. The .TA ROW control word is not allowed within a table header or footer. This means only one row can be created in a table header or footer. If a .TA ROW or .TA TABLE control word is encountered when defining a table header or footer, a message is issued and the definition is ended before the .TA ROW or .TA TABLE control word can be executed.
6. A table header or table footer is allowed only before the first row in the table. If a table header or table footer is defined, it is placed at the top or bottom of each subsequent column of every page upon which the table appears.  
  
Table headers and table footers are reformatted between pages if the table spans more than one page.
7. When a table is started, the current environment is saved, and any offsets (.OF [Offset]) or undents (.UN [Undent]) are reset to zero. All other formatting characteristics are unchanged. When a table is ended, the environment saved at the start of the table is restored. Any formatting characteristics that were in effect at the start of the table are in effect when the table is ended. See "The SCRIPT/VS Formatting Environment" on page 451 for a list of the active environment values that are saved and restored around tables.
8. The .FO [Format Mode] and .NF [No Formatting] control words can be used to specify the horizontal alignment of text within the cell.  
  
The .IN [Indent] and .IR [Indent Right] control words can be used to create left and right indents for positioning text within the cell. The indentation is measured from the left or right edge of the cell.
9. The vertical rules surrounding the table, rows, and cells are left-aligned.



10. The left vertical rule of a cell occupies space within the cell. Ensure that you don't cause the contents of the cell to print on top of the left vertical rule for the cell. The way to accomplish this depends on the rotation of the cell:
  - 0°-rotated cells: use the .IN [Indent] control word to create a left indent.
  - 90°-rotated cells: use the .SP [Space] control word to create a space at the bottom of the contents of the cell.
  - 180°-rotated cells: use the .IR [Indent Right] control word to create a right indent.
  - 270°-rotated cells: use the .SP [Space] control word to create a space at the top of the contents of the cell.
11. The right vertical rule for the table occupies space only within the rightmost cells in the table.
12. If the table is being formatted for the 3800 Printing Subsystem Model 3, an error results if the composite rotation of the table or the contents of the cell is 180° or -180°
13. If a table is active when a .TA TABLE *tablename* ON control word is encountered, a warning message is issued, and the current table is ended before the new table is started.
 

If a row is active when a .TA ROW *rowname* ON is encountered, the current row is ended without a message and the new row is started. The same holds true for table header definitions, table footer definitions, and cells.
14. If the OFF parameter of the .TA control word is used to end a certain type of table object (table, header, footer, row, or cell) and a table object of that type is not active, the control word is ignored.
15. If the total depth of a table header and footer is greater than the body depth of the current page, the table header and table footer are ignored for that page.
16. Vertical justification is not performed on text inside of a table. The setting of the .FV [Format Vertically] control word has no affect on text within a table; it determines the vertical placement of the table as a whole within a column or page. Lead out points defined with the .LO [Lead-Out] control word are ignored within a table.
17. See Table 15 on page 442 for a list of the control words that are not allowed within a table header, table footer, row, or cell. If any of these control words are found within a table header or footer, a message is issued, the table header or footer is ended, and the rest of the definition is ignored. If any of these control words are found within a row or cell, a message is issued, the table is ended, and the disallowed control word is then executed.
18. If none of the rows within a table use the SPLIT parameter, the entire table must fit on a single page. If the table doesn't fit on the current page and the current page is not empty, a page eject is done prior to the table to place the table at the top of the next page.
 

If the table still does not fit on the empty page, a warning message is issued, and the table is split, fitting as many rows on the current page as possible. A table footer is not placed on the current page, and the last row on the current page is ended with that row's bottom horizontal rule, instead of the table's bottom horizontal rule. A table header is not placed on the next page, and the first row on the next page is started with that row's top horizontal rule, instead of the table's top horizontal rule.
19. If a table occurs in a non-vertical keep, float, area, footnote, running heading, or running footing, the SPLIT parameter on any row is ignored. Tables inside these items must fit on a single page.
 

If the table does not fit in a non-vertical keep or float, the non-vertical keep or float is ended, the table is placed in the body of the page, and an error message is issued. If the table does not fit in an area or a footnote, the entire table is discarded and a warning message is issued. If the table does not fit in a running heading or running footing, processing stops and a severe error message is issued.

If a table is to be placed within a vertical keep (*v*, *v* + *v*) and keeps associated with headings, the table is searched for a split point that satisfies the size of the keep. If such a split point is found, only the

## .TA [Table]

portion of the table up to that split point is retained with the keep. If no such split point is found, the whole table is placed into the keep.

20. Be careful when starting and ending fonts within rotated cells. If a font is started within a rotated cell and ended with a .PF [Previous Font] control word in another cell with a different rotation, you receive an error. The font restored with the .PF control word has the rotation value of the first cell, not the current cell.
21. Revision codes used within a table print to the left or right, outside the table. Revision codes for 90°, 180°, or 270° rotated cells are ignored. If you have revision codes active in two cells that are horizontally adjacent, the revision codes might overlap.
22. Left revision codes on tables are ignored for the 3800 Printing Subsystem Model 1.

23. Table headers and footers work in much the same way as running headings and footings. With table headers and footers, there is a definition phase and an execution phase. When the .TA HEAD *rowname* ON or .TA FOOT *rowname* ON control word is encountered, a table header or footer definition is started. All control words, text, and GML tags encountered prior to the .TA HEAD OFF or .TA FOOT OFF control word are placed into the table header or footer definition. As with running headings and footings, substitution and GML tag scanning is off during this definition phase.

When a table header or table footer is being defined, only the .TA HEAD OFF or .TA FOOT OFF control words are executed. Therefore, these control words must appear in column 1 of an input line, and they cannot occur after a control word separator.

The execution phase of a table header or footer occurs when the first row in a table is started and between pages when the table spans more than one page. When a table header or footer is executed, the formatting environment is set to the default, the same as with running headings or footings. Therefore, any indents or revision codes that were active when the table header or footer was defined are not active when the table header or footer is executed. You must specifically place any such control words inside the table header or footer definition in order for those control words to effect the execution of the table header or footer.

24. Vertical rules started inside of a cell with the .BX [Box] and .VR [Vertical Rule] control words are ended when the cell ends.
25. If a row has a bottom horizontal rule specified as NONE, no vertical space is added at the bottom of that row, prior to the next row.
26. Skips (.SK [Skip]) at the top and bottom of cells are discarded.
27. Tables are processed similarly to areas, that is, the entire table is formatted before any part of the table is placed on the page.
28. The output from SCRIPT/VS when formatting for a PostScript or 4028 physical device or an AFP logical device is a resolution-independent data stream. Because of the variations in printer resolutions, the same SCRIPT/VS file might produce different results for different devices. Therefore, when you format for one of these devices, there may be on pel rounding errors at the printer that are evident in rule widths and rule intersections. If your table cell is shaded, this could also cause the shading to appear to not fit within the cell. Moving the position of the table on the page slightly can improve the appearance.
29. Use .TA BREAK to avoid running out of storage when formatting large tables.
30. When formatting complex tables for the 2741 output device, it is sometimes possible that a horizontal and vertical rule intersection are created by overprinting a horizontal rule character (a hyphen) and a vertical rule character (a vertical bar), instead of placing a rule intersection character (a plus sign). The required backspaces are produced to cause the two characters to overprint producing the desired intersection.

31. Vertical white space is placed before a table to separate it from the data before the table. This white space is called leading and is determined in the following manner:

- If implicit linespacing is in effect, the leading is determined by taking the height of the linespacing as defined in the current font, multiplying it by linespacing factor (from .LS BY), and subtracting the em height of the current font.
- If explicit linespacing is in effect, (when .LS NORMAL has been used to indicate a specific value for the linespacing, and the linespacing is not determined by the font size) the leading is determined by taking the linespacing value as specified by .LS NORMAL, multiplying it by the factor as specified by .LS BY and subtracting the height of the rule that will appear as the top rule in the table (for 90-degree tables this is the right vertical table rule, for 270-degree tables it is the left vertical table rule, for 0-degree tables it is the top horizontal table rule, for 180-degree tables it is the bottom horizontal table rule).

Notice that when implicit linespacing is active, the leading is based on the size of the current font. When explicit linespacing is active, the leading is based on the current linespacing *and* the height of the rule at the top of the table. Therefore, with explicit linespacing, if the height of the rule at the top of the table is decreased, the leading will increase, and thus more vertical space is required for the table to fit on the page. Keep in mind that the height of the NONE rule is zero.

## Examples

- The following shows an example of a table that has a table header and footer:

```
.td table m1
.td row m1 arrange(1 2)
.td row m2
.ta table m1 on
.ta head m2 on
.in .li
.ta cell 1 on
This is a table header.
.ta head off
.ta foot m2 on
.in .li
.ta cell 1 on
This is a table footer.
.ta foot off
.ta row m1 on
.in .li
.ir .li
.ta cell 1 on
This is stuff in the first cell of this table.
In this row the second cell is empty.
.ta row m1 on
.ta cell 1 on
This is in the first cell.
.ta cell 2 on
This is stuff in the second cell. Because this cell has more text
in it, the depth of cell 1 has been increased to cause both
cells to end at the same place.
.ta table off
```

That input results in the following:

|                                                                                      |                                                                                                                                                               |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| This is a table header.                                                              |                                                                                                                                                               |
| This is stuff in the first cell of this table. In this row the second cell is empty. |                                                                                                                                                               |
| This is in the first cell.                                                           | This is stuff in the second cell. Because this cell has more text in it, the depth of cell 1 has been increased to cause both cells to end at the same place. |
| This is a table footer.                                                              |                                                                                                                                                               |

|                                                                                      |                                                                                                                                                               |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| This is a table header.                                                              |                                                                                                                                                               |
| This is stuff in the first cell of this table. In this row the second cell is empty. |                                                                                                                                                               |
| This is the first cell.                                                              | This is stuff in the second cell. Because this cell has more text in it, the depth of cell 1 has been increased to cause both cells to end at the same place. |
| This is a table footer.                                                              |                                                                                                                                                               |

- Revision codes work differently in tables than in normal body text.

In the following example the first cell has a revision code of 'A' and the second cell has a revision code of '¢'. Both revision codes are printed:

```
.td table t1
.td row t1 arrange (1 2)
.td cell 1 t1 center
.in .li
.ir .li
.rc 1 A
.rc 2 ¢
.ta table t1
.ta row t1 on
.ta cell 1 on
.rc 1 on
This is text in the first cell.
.rc 1 off
.ta cell 2 on
.rc 2 on
This is text in the second cell. Notice that the revision characters
overlap.
.rc 2 off
.ta table off
```

That input results in the following:

|                                 |                                                                               |
|---------------------------------|-------------------------------------------------------------------------------|
| This is text in the first cell. | This is text in the second cell. Notice that the revision characters overlap. |
|---------------------------------|-------------------------------------------------------------------------------|

**Note:** You can avoid the overlapping characters by using the FONT parameter of the .RC [Revision Code] control word to select a smaller font.

## .TA [Table]

- Cells in tables can be shaded. In the following example, the .TD table cell named sdcell1 is shaded with a STANDARD pattern and LIGHT shading and the .TD table cell named sdcell2 is shaded with a STANDARD pattern and 14 percent shading;

```
.sd sdcell1 standard shade light
.sd sdcell2 standard shade 14
.td table xmp5
.td row xmp5 arrange (1 2)
.td cell 1 xmp5 shade sdcell1
.td cell 2 xmp5 shade sdcell2
.ta table xmp5 on
.in .li
.ir .li
.ta row xmp5 on
.ta cell 1
The first cell in the table shaded
with a STANDARD pattern and LIGHT shading.
.ta cell 2
The second cell in the table shaded
with a STANDARD pattern and 14 percent shading.
.ta table off
```

The above input results in

|                                                                                  |                                                                                        |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| The first cell in the table shaded with a<br>STANDARD pattern and LIGHT shading. | The second cell in the table shaded with a<br>STANDARD pattern and 14 percent shading. |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|

- In the following example, revision codes in the rotated cells are ignored:

```
.td table a1
.td row a1 arrange (1 2)
.td cell 1 a1 center rotate 90 mindepth 1i
.rc 1 |
.rc 2 ¢
.ta table a1 on
.ta row a1 on
.ta cell 1 on
.rc 1 on
.in 0
.ir 0
Because this is a rotated cell, the revision code is ignored.
.rc 1 off
.ta cell 2 on
.rc 2 on
.in .li
.ir .li
This is stuff in the second cell. This text has a revision code.
.rc 2 off
.ta table off
```

That input results in the following:

|        |                                                                        |                                                                     |
|--------|------------------------------------------------------------------------|---------------------------------------------------------------------|
| ¢<br>¢ | Because this is<br>a rotated cell,<br>the revision<br>code is ignored. | This is stuff in the second cell. This text has a<br>revision code. |
|--------|------------------------------------------------------------------------|---------------------------------------------------------------------|

- A cell can be repeated within the same row. If the same cell is filled more than once, a horizontal rule is drawn, and the second iteration of the cell is added. In the following example, the first and second cells are each repeated 3 times. Notice that for the cells rotated 0°, the vertical alignment<sup>12</sup> applies to each iteration of the cell; for the cells rotated 90°, the vertical alignment applies to all the iterations collectively:

|                                                   |                                                    |                                                   |                                                   |                                              |
|---------------------------------------------------|----------------------------------------------------|---------------------------------------------------|---------------------------------------------------|----------------------------------------------|
| Third iteration of the 90° rotated cell number 1. | Second iteration of the 90° rotated cell number 1. | First iteration of the 90° rotated cell number 1. | First iteration of the 0° rotated cell number 2.  | This is the only iteration of cell number 3. |
|                                                   |                                                    |                                                   | Second iteration of the 0° rotated cell number 2. |                                              |
|                                                   |                                                    |                                                   | Third iteration of the 0° rotated cell number 2.  |                                              |

The input for the above table is:

```
.td table retable 6p6 width 33p
.td row reprow arrange(1 2 3) cwidths(1,5i * 1,75i)
.td cell 1 reprow rotate 90 center mindepth 3i
.td cell 2 reprow bottom mindepth 1i
.ta table retable on
.ta row reprow on
.in .1i
.ir .1i
.ta cell 1 on
First iteration of the 90° rotated cell number 1.
.ta cell 2 on
First iteration of the 0° rotated cell number 2.
.ta cell 1 on
Second iteration of the 90° rotated cell number 1.
.ta cell 2 on
Second iteration of the 0° rotated cell number 2.
.ta cell 1 on
Third iteration of the 90° rotated cell number 1.
.ta cell 2 on
Third iteration of the 0° rotated cell number 2.
.ta cell 3 on
This is the only iteration of cell number 3.
.ta table off
```

<sup>12</sup> Vertical alignment is specified with the TOP, CENTER, or BOTTOM parameter on the .TD CELL control word.

---

## .TB [Tab Setting]

### Function

Use the .TB [Tab Setting] control word to define how tab characters (hexadecimal X'05') are to be resolved. They can be changed to a number of blanks or to a string of another character.

The .TB control word is provided for compatibility with earlier releases of SCRIPT/VS. Most of the same functions are provided by the .TP [Tab Position] control word.

### Syntax



### Parameters

- SET** Specifies that all the old tab stops are to be cleared and a new set of tab stops is to be defined.
- ADD** Specifies that the tab displacements given are to be added to those currently defined.
- CLR or DEL** Specifies that the tab displacements given are to be removed from those currently defined.
- h*** Specifies the horizontal displacements of the tab stops. SCRIPT/VS displaces to the next stop by padding with blanks or other fill characters. The sequence for any single .TB control word must consist of increasing positive values separated by one or more blanks. However, a .TB ADD control word can insert new tab stops between existing ones.
- If no parameters are specified with .TB SET, the default tab settings are restored.
- If .TB SET 0 is specified, all tab settings are cleared.
- f/*** Specifies the fill character to be used in displacing through position *h*. If the fill character is to be the blank, it need not be specified.

**Default:** 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, and 80.

**Note:** The .TB control word causes a break.

- | The tab settings are included in the active environment.

### Remarks

1. The .TB and .TP [Tab Position] control words work differently. Refer to “.TP [Tab Position]” on page 400 to learn how the .TP [Tab Position] control word differs from the .TB control word. Do not use both control words in the same document.
2. The tab settings on any single .TB control word must increase. If the tab settings do not increase, an error message is issued.
3. Tab characters that are found beyond (to the right of) the last defined tab stop are converted to a single blank.
4. Using fill characters on the 3800 Model 3 causes printer performance problems. Use the .SX [Split Text] control word instead, if possible.



5. The fill character is formatted in the current font when the fill string is being formatted.
6. If the space to the next tab stop is less than the width of one fill character, the following tab stop is used.
7. On the 3800 Model 1, fill characters are supported only with monospaced fonts. If you use fill characters with proportionally spaced fonts, vertical misalignment might result.
8. Backspaces after a tab reduce the tab position for non-3800 Model 1 logical devices, but the distance to be tabbed is never reduced to less than one character space.
9. No more than 64 tab stops can be set at one time.
10. Tabs beyond the column margin set by .CL [Column Line Length] are ignored.

## Examples

- .tb 10 20 \*/30 40

Tab positions are interpreted as character positions 10, 20, 30, and 40. If a tab character is processed between positions 20 and 30 of a line, the positions from the current position up through and including position 30 are filled with asterisks (\*) instead of blanks. The next character goes in position 31. For example, using the system symbol &\$TAB to generate tab characters, the line

&\$TAB.text&\$TAB.text&\$TAB.text

results in

```
text      text*****text
```

- .tb

Tab positions revert to default values of 5, 10, 15, and so forth.

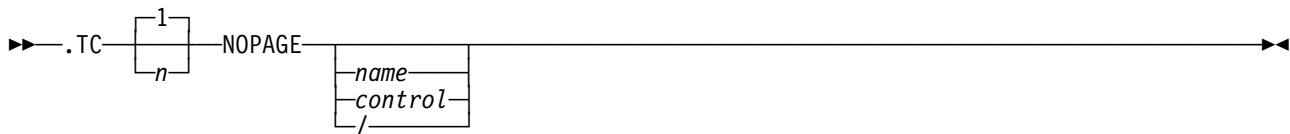
---

## .TC [Table of Contents]

### Function

The .TC [Table of Contents] control word causes the automatically generated table of contents to be imbedded and printed. Entries can be placed in the table of contents by head-level control words .H0 [Heading Level 0] and by the .PT [Put Table of Contents] control word.

### Syntax



### Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                          |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>n</i>       | The number of page numbers that are reserved for the table of contents. If <i>n</i> is omitted, 1 is assumed.                                                                                                                                                                                                                                                            |
| <b>NOPAGE</b>  | Specifies that SCRIPT/VS not perform a page eject before and after the table of contents. The table of contents is placed on the current page, and any text following is continued on the last page of the table of contents.                                                                                                                                            |
| <i>name</i>    | An optional line to be used as the title of the table of contents. If no <i>name</i> is given and the system symbol (&SYSCONTENT) is not set, the word CONTENTS is used, by default. A head-level 1 (.H1) is generated at the top of the table of contents, using either the <i>name</i> given or the word CONTENTS. The first word of the <i>name</i> cannot be NOPAGE. |
| <i>control</i> | A control word or macro to be processed at the top of the table of contents instead of the .H1 control word. If this parameter begins with a period, it is assumed to be a control word, not a name.                                                                                                                                                                     |
| <i>/</i>       | Signals SCRIPT/VS not to generate any head-level 1 for the table of contents, but a page eject is still performed, unless NOPAGE is also specified. Use “/” when you want no name on the table of contents, you have no control word to be executed, and you want the default name CONTENTS to be ignored.                                                               |

### Notes

- .TC ends a keep, float, footnote, named area, or table.
- .TC causes a break.
- .TC ensures that the page is started.

### Remarks

1. Although all of the parameters are optional, if they are used, they must appear in the order entered in the description above.
2. When the .TC control word is encountered, a head-level 1 is processed (unless overridden by the control parameter), but no entry is placed in the table of contents for the heading. Only on the last formatting pass are the table of contents entries (saved in the DSMUTTOC utility file) formatted and printed. The entries come from the head-level control words whose definitions call for table of contents entries (by default, the control words .H0 through .H3 cause these entries) and from any

explicit .PT control words in the source file that have been executed prior to the .TC control word, either on the current or on a previous pass.

When the NOPAGE parameter is specified, the head-level 1 used to process the name parameter does not cause a page eject. There is no space between the last line of text on the page and the head-level 1 that follows, because the default for a head-level one has no space or skip before the heading. To put space before the head-level 1, use .SP or .SK before the .TC, or redefine the space or skip value for head-level 1 headings using the SPBF or SKBF parameters on .DH.

3. Any head levels that are defined to cause a page eject do not generate a page eject while a NOPAGE table of contents is active. This includes headings generated using .PT, for example, a .PT .H1 heading.
4. When the NOPAGE parameter is specified, the pending floats are not dumped. However, the keeps and floats that fit on the current page are placed before starting the table of contents. To get a total float dump before the table of contents, used .FL DUMP.
5. The table of contents is formatted according to the line and page dimensions in effect at the time the .TC control word is encountered, not those in effect when the heading or .PT control word was processed. Each entry in the table of contents has the revision code and the page number that were in effect when the heading or .PT control word was processed.
6. When the table of contents is completely formatted, a conditional page eject is done (unless NOPAGE was specified), and the new page is numbered as though the table of contents occupied exactly  $n$  pages. If the table of contents takes more or fewer than  $n$  pages, there is either an overlap or a gap in page numbering.

When you specify  $n$ , you reserve  $n$  page numbers for the table of contents. For example, if you specify .TC 5 and the table of contents starts on page 2, then the first page after the table of contents is numbered 7. If you do not specify a value for  $n$  on .TC and the table of contents starts on page 2, the value of  $n$  defaults to one (1), and the first page after the table of contents is numbered 3, regardless of how many pages the table of contents actually occupies.

7. If the NOPAGE parameter is specified along with the  $n$  parameter, the pages that the table of contents start and end on must be included in the number of pages reserved for the table of contents ( $n$ ), even if text precedes or follows the table of contents on the same page.
8. If NOPAGE was specified, the table of contents could start at the bottom of a page, but there might not be enough room for the heading or the first line of text. When this happens, the text of the table of contents starts on the next page. The page number for the page following the table of contents is calculated while still on the bottom of the first page. Therefore, that page needs to be included in the number of pages reserved on .TC in order for the page following the table of contents to be numbered correctly.
9. If you need a page break either before or after the table of contents, but not in both places, use .PA either before or after the .TC NOPAGE.

If you add pages to the current page number, an error can result on the .PA control word if the page number exceeds 99,999,999.

10. When NOPAGE is not specified on .TC, the table of contents contains entries only from pages that precede the .TC control word. Entries created with .PT [Put Table of Contents] control words that refer to pages on which the table of contents is formatted are not included. However, if NOPAGE is specified, any .PT control words that are processed before the .TC NOPAGE is encountered appear in that table of contents. This includes .PT control words within items that are placed at the beginning of a page, such as running headings, running footings, and floats.

Page and body areas are normally placed when the page is ended. When a .TC NOPAGE is specified, however, any page and body areas that are placed (using the .AR [Area] control word) prior

## .TC [Table of Contents]

to the .TC NOPAGE are placed as part of NOPAGE processing. Any .PT control words in those areas are processed at that time and appear in the NOPAGE table of contents.

**Note:** .PT is a deferred control word, meaning that the control word is not processed until the text it is associated with is actually placed on the page. For example, if you define a float before the .TC NOPAGE, but it is placed on the page following the table of contents, the .PT control words within that float do not appear in the table of contents.

11. If the .TC control word is used at the beginning of a document and the TWOPASS or FPASSES(2) option is specified on the SCRIPT/VS command line, the resolution of page numbers in the table of contents and cross-references inside the book may be inaccurate. Using the FPASSES option on the SCRIPT/VS command line to specify more than two formatting passes should eliminate this problem.
12. It is recommended that you use the *n* parameter to specify the number of pages you want for your table of contents. If not, any references occurring *after* the table of contents may not be correct. This could occur when page numbering is not reset after the table of contents is formatted, or when a page eject is not performed after the the table of contents, and you are using the **NOPAGE** paramter. Even if the *n* parameter is used, using the **NOPAGE** parameter can cause some of your references following the table of contents to be incorrect. If this is the case, use the **W SYSVAR** to save all of your references during the last formatting pass. Then format your document again with on pass and the **R SYSVAR** to include the saved references.

## Examples

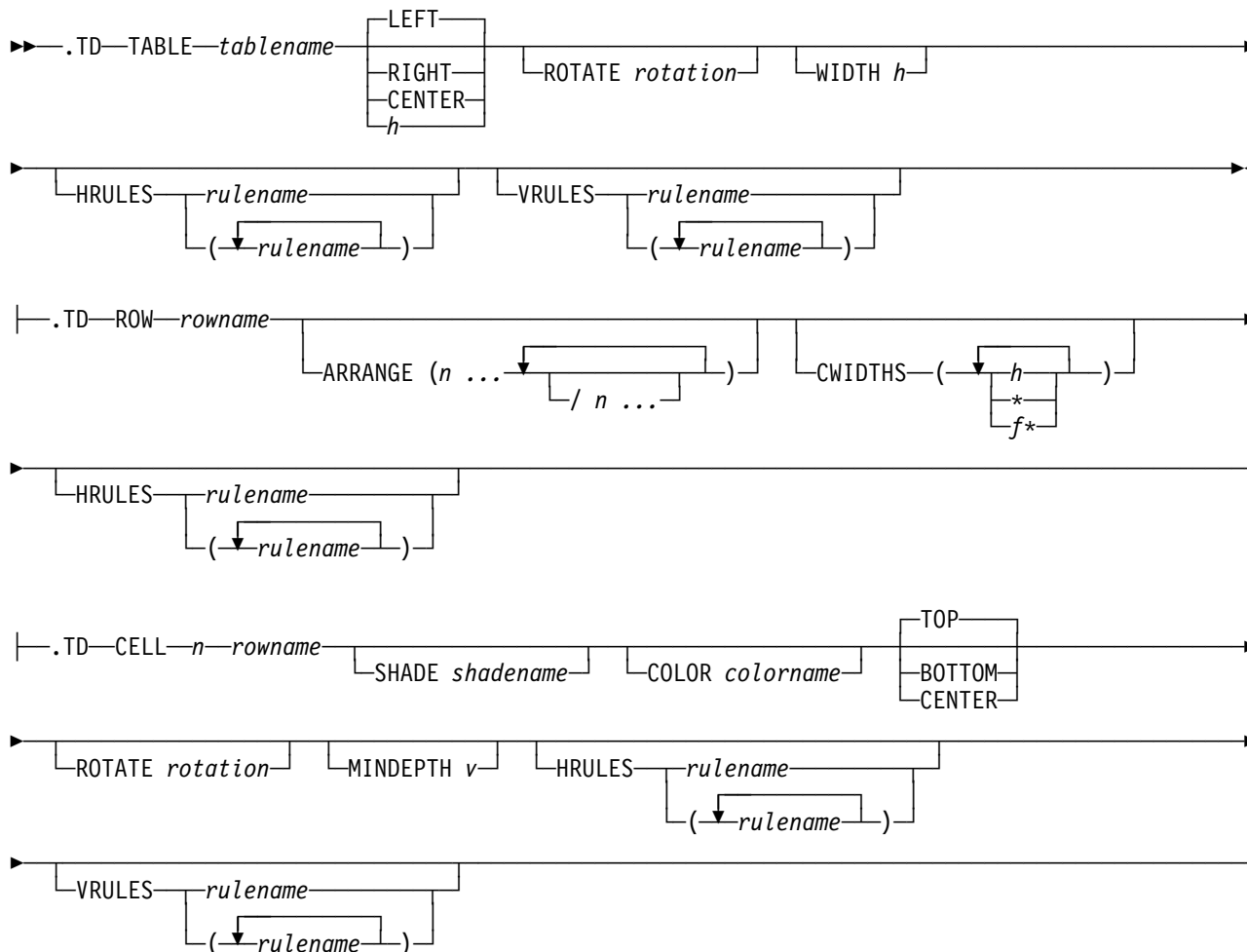
See the table of contents of this document for an example of an automatically generated table of contents.

## .TD [Table Definition]

### Function

Use the .TD [Table Definition] control word to create a table, row, or cell definition for use in building tables.

### Syntax



### Parameters

**TABLE** Specifies a table definition with a name of *tablename*.

*tablename* Assigns the name of the table being defined. After *tablename* has been defined, it can be used on the .TA control word with the TABLE parameter to start a table. The *tablename* can contain a maximum of 16 national characters and is not case sensitive.

“OFF” cannot be used as a tablename, as it could be confused with the OFF parameter on the .TA [Table] control word.

**LEFT** Specifies that a table using this table definition has its left edge aligned with the left margin. LEFT is the default.

## .TD [Table Definition]

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RIGHT</b>    | Specifies that a table using this table definition has its right edge aligned with the right margin.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>CENTER</b>   | <p>When used with the TABLE parameter, specifies that a table using this table definition is centered horizontally across the column line length.</p> <p>When used with the CELL parameter, indicates that this cell is to have its contents vertically centered in the cell.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b><i>h</i></b> | Specifies the amount of horizontal displacement a table using this table definition is offset from the left margin.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>ROTATE</b>   | <p>When used with the TABLE parameter, indicates the rotation the table has relative to the current area or page. If not specified, 0 degrees is assumed.</p> <p>When used with the CELL parameter, indicates the rotation the contents of the cell has relative to the table. If not specified, the contents of the cell are rotated 0° relative to the table.</p> <p>The rotation value is given in degrees, and must be one of these multiples of 90°:<br/>-270 -180 -90 0 90 180 270</p> <p><b>Note:</b> If a rotation value of -270°, -90°, 270°, or 90° is used, you must specify a MINDEPTH value.</p> <p>Rotation is done in a clockwise direction about the upper-left corner of the area. This parameter is ignored for all devices that do not support rotation. The 180° rotation is not valid for the 3800 Printing Subsystem Model 3.</p>                                                                                                                                                                                                                                                                                                                                                                      |
| <b>WIDTH</b>    | <p>Specifies the width of the table. Any valid horizontal space unit notation can be used.</p> <p>If the WIDTH parameter is not specified, the table uses the current column line length when the table is started.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>HRULES</b>   | <p>Specifies the horizontal rules to be used above and below the table, row, and cell. The rulename given must have been previously defined with the .DR [Define Rule] control word.</p> <p>If only one rulename is given, it is used for both the top and the bottom horizontal rules. If two rulenames are given, the first is used for the top horizontal rule, and the second is used for the bottom horizontal rule.</p> <p>If the HRULES parameter is not specified, the default rule is used for the horizontal rules.</p> <p>Specifying NONE as a <i>rulename</i> indicates that no rule is to be used in that position.</p> <p>For line devices, only one <i>rulename</i> is used in the table. This is the top horizontal rule specified for the table. If a <i>rulename</i> of NONE is specified as the top horizontal rule, the table contains no horizontal or vertical rules. A <i>rulename</i> of NONE is ignored if used to define any rule other than the top horizontal rule.</p> <p><b>Note:</b> For printers that support color, one color can be specified for the rules of the entire table. The color of the rules for the table is the color specified for the top horizontal rule of the table.</p> |
| <b>VRULES</b>   | <p>Specifies the vertical rules to be used to the left and right of the table or cell.</p> <p>This parameter is not allowed with the ROW parameter.</p> <p>The</p> <p>rulename given must have been previously defined with the .DR [Define Rule] control word.</p> <p>If only one <i>rulename</i> is given, it is used for both left and right vertical rules. If two <i>rulenames</i> are given, the first is used for the left vertical rule, and the second is used for the right vertical rule.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

If the VRULES parameter is not specified, the default rule is used for the vertical rules.

Specifying NONE as a *rulename* indicates that no rule is to be used in that position.

For line devices, only one *rulename* is used in the table. This is the top horizontal rule specified for the table. If a *rulename* of NONE is specified as the top horizontal rule, the table contains no horizontal or vertical rules. A *rulename* of NONE is ignored if used to define any rule other than the top horizontal rule.

**Note:** For printers that support color, one color can be specified for the rules of the entire table. The color of the rules for the table is the color specified for the top horizontal rule of the table. The color of any rule specified with VRULES is ignored.

## **ROW**

Specifies a row definition with a name of *rowname*.

### *rowname*

Assigns the name of the row being defined. After *rowname* is defined, it can be used on the .TA control word with the ROW, HEAD, or FOOT parameters to start a row, header definition, or footer definition. It can also be used on a subsequent .TD control word with the CELL parameter to define a particular cell within a given row. The *rowname* can contain a maximum of 16 characters and is not case sensitive.

“OFF” cannot be used as a *rowname*, because it could be confused with the OFF parameter on the .TA [Table] control word.

## **ARRANGE**

Defines the arrangement of the cells within the row by using an implicit grid pattern. The values specified define which grid elements are associated with which cells. The values correspond to cell numbers (which are used with the CELL parameter of the .TD control word) and must be integers greater than zero. If more than one set of integers is given between slashes, the number of integers between each set of slashes must be the same. The maximum number of ARRANGE values between the slashes is 64. Refer to the examples in the *Document Composition Facility: Text Programmer's Guide* for more information on using this parameter.

If the ARRANGE parameter is not specified, the default is *n* cells, where *n* is the number of specifications given on the CWIDTHS parameter. The cells are numbered left to right in sequential order, starting with the number 1. If the CWIDTHS parameter was not specified either, the default is one cell with a cell number of 1.

## **CWIDTHS**

Specifies the widths of the grid elements in the arrangement of the row. Any valid horizontal space-unit notation can be used. The maximum number of CWIDTHS values is 64.

If the ARRANGE parameter is used, the number of values specified on the CWIDTHS parameter must be the same as the number of values entered between the slashes on the ARRANGE parameter (the number of grid elements). The CWIDTHS values indicates the widths of the individual grid elements that were defined on the ARRANGE parameter. The width of a particular cell is computed by adding up the CWIDTHS values given for all the grid elements that comprise that particular cell (all the grid elements that have the same number as the cell).

The values entered for the CWIDTHS parameter can be absolute values (any valid horizontal space-unit notation) or proportional values indicated by an asterisk (\*). After all absolute (nonasterisk) CWIDTHS specifications are taken into account, the remaining horizontal space (the table width minus the absolute values) is divided proportionally among all the grid elements that have specified an asterisk (\*) for the width. A factor, *f*, given before the asterisk indicates that particular grid element is to be *f* times as wide as any grid element that has its width specified with just one asterisk.

If the CWIDTHS parameter is not specified, the default is *n* evenly-spaced grid elements, where *n* is the number of values given between slashes on the ARRANGE parameter. If

## .TD [Table Definition]

the ARRANGE parameter was not specified either, the default is one cell equal to the width of the table.

**CELL** Specifies a cell definition for cell *n* in the row named *rowname*.

*n* Gives the number of the cell being defined. The cell number *n* may be used with the .TA control word with the CELL parameter. The *n* is required and must be an integer greater than zero.

*rowname* Is the name of the row containing the cell being defined. The *rowname* is required and must have been defined with the ROW parameter on a previous .TD control word.

**SHADE** Specifies that the table cell is to be shaded by using the named shading definition.

The *shadename* specifies the name of the shading definition. The name can be a maximum of 16 national characters, and it is not case sensitive. The shading definition must have been defined with the .SD [Shading Definition] control word before specifying the .TD control word and its associated parameters.

**COLOR** Specifies that the table cell is to be colored by the named color definition. The COLOR parameter applies only to printers with color capability. *colorname* can be any color previously defined with the .CR [Color] control word.

**TOP** Indicates that this cell is to have its contents vertically aligned at the top of the cell. TOP is the default.

**BOTTOM** Indicates that this cell is to have its contents vertically aligned at the bottom of the cell.

**MINDEPTH** Specifies the minimum depth of the cell. If the contents of the cell are not sufficient to achieve the minimum depth specified, the cell is padded with white space to achieve the required minimum depth. If MINDEPTH is not specified for a cell, the depth of that cell is determined by the contents of the cell and the depth of the other cells in the row. If the text in the cell is rotated -270°, -90°, 270°, or 90° relative to the table, the MINDEPTH parameter is required on the same or on a previous .TD control word.

## Notes

- .TD causes a break.

## Remarks

1. The .TD control word is used to create three different types of definitions for use in creating tables: tables, rows, and cells. A single occurrence of the .TD control word can be used for only one of the three possible definitions.
2. When the ARRANGE parameter is used, a grid is produced by taking the first set of values between slashes on the ARRANGE specification as the first line in the grid, the second set of values as the second line in the grid, and so on. This grid is used to define the layout of the cells. Each cell is composed by combining adjacent grid elements with the same number. If all the grid elements with the same number are not adjacent or do not produce a rectangle, an error message results. Refer to "Creating Tables" in the *Document Composition Facility: SCRIPT/VS User's Guide* for information on how to use the ARRANGE and CWIDTHS parameters to define the layout of cells in a row.
3. If, on a given .TD ROW control word, more CWIDTHS values are given than horizontal grid elements defined with the ARRANGE parameter, a warning message is issued, and the extra CWIDTHS values are ignored. If fewer CWIDTHS values are given than grid elements defined with the ARRANGE parameter, the extra grid elements are given a value as if an asterisk had been specified. If a cell ends up with a width of zero, an error message is given when an attempt is made to place text into that cell, and processing stops.



4. The sum of the values given on a CWIDTHS parameter should be equal to the width of the table (as given on the WIDTH parameter in the .TD TABLE control word). If the CWIDTHS values are such that the row width is greater or less than the table width, and proportional cell widths are specified, a cell width of zero or less might result. In this case, a severe error message is issued, and processing stops.
5. The .TD control word is not allowed within a table. If found, an error message is issued, and the .TD control word is ignored.
6. If you want to specify more parameters for a particular definition than can easily fit on one input line, use more than one .TD control word. Each time .TD is processed, only those parameter values specified are changed.
7. If .TD TABLE tablename is specified with no other parameters, the table definition is set to these default values:

| Alignment | ROTATE | WIDTH              | HRULES       | VRULES       |
|-----------|--------|--------------------|--------------|--------------|
| LEFT      | 0      | Column line length | Default rule | Default rule |

8. If .TD ROW rowname is specified with no other parameters, the row definition is set to these default values:

| ARRANGE | CWIDTHS | HRULES       | VRULES       |
|---------|---------|--------------|--------------|
| None    | *       | Default rule | Default rule |

9. If .TD CELL *n* rowname is specified with no other parameters, the cell definition is set to these default values:

| Vertical Alignment | ROTATE | MINDEPTH | HRULES       | VRULES       |
|--------------------|--------|----------|--------------|--------------|
| TOP                | 0      | None     | Default rule | Default rule |

10. An unrecognized parameter on the .TD TABLE control word is interpreted as a horizontal displacement for the table, which might result in an unexpected error message.
11. If a particular parameter is specified more than once on a .TD control word, the last specification is used.
12. A rotation specification for a cell indicates the rotation of the *contents* of the cell. For example, when text is rotated 90° within the cell, the left margin of the text is at the top of the cell. The rules around a cell always have a 0° rotation relative to the table.
13. The WIDTH value specified must fit on the current page, or a warning message is issued.
14. For line mode, the top horizontal rule for the table is used for all rules in the table.
15. If the text in the cell is rotated -270°, -90°, 270°, or 90°, the MINDEPTH value is required. The width of such a cell (as computed from the CWIDTHS and ARRANGE values) becomes the vertical depth of the contents of the cell. The MINDEPTH value given for the cell is used as the column line length when formatting the contents of the cell.  
  
If the cell text exceeds the cell width, an error message is issued, and the text overlays the area outside of the cell.
16. The vertical alignment value for text within a cell is relative to the rotation of the contents of the cell, not the cell itself. For example, if a cell has text rotated 90° with BOTTOM vertical alignment, the text

## .TD [Table Definition]

in the cell is bottom aligned. Because the text is rotated 90°, the text is pushed down, so the last line of the text appears next to the left side of the cell.

17. The SHADE parameter is ignored if you are formatting for a line device or for a 4250 printer.
18. For more information about the SHADE parameter, see “.SD [Shading Definition]” on page 328.
19. Complicated cell arrangements specified on the ARRANGE parameter can result in .TD control word lines that are too long for SCRIPT/VS to process. The maximum number of characters in an input line is 256 after all symbols have been resolved. Any data past the 256 character mark is cut off and handled as the next input text line. For a .TD control word line with a very long ARRANGE specification this can cause part of that specification to be cut off resulting in an invalid ARRANGE specification. The portion of the control word line that was cut off is handled as the next input line.

## Examples

- The widths of the rows in a table do not have to be the same width as the table. In the following example, the width of the table is 4 inches, the width of the first row is 2 inches, the width of the second row is 3 inches, and the width of the third row is as wide as the table.

```
.td table xmp1 width 4i
.td row xmp1 cwidths 2i
.td row xmpla cwidths 3i
.td row xmp1b
.ta table xmp1 on
.in .1i
.ir .1i
.ta row xmp1 on
.ta cell 1
This row is 2 inches wide and consists of 1 cell.
.ta row xmpla on
.ta cell 1
This row is 3 inches wide and consists of 1 cell.
.ta row xmp1b on
.ta cell 1
CWIDTHS was not specified for this row, so it uses the
table width as its width.
.ta table off
```

The above input results in

|                                                                                     |
|-------------------------------------------------------------------------------------|
| This row is 2 inches wide<br>and consists of 1 cell.                                |
| This row is 3 inches wide and consists of 1<br>cell.                                |
| CWIDTHS was not specified for this row, so it uses the<br>table width as its width. |

- The width of the table is used to determine the actual width of cells that have been given a proportional width by use of the “\*” character on the CWIDTHS parameter. In the following example, the first cell is one inch and the last three cells split the remaining table width, with the fourth cell twice as wide as the second and third cells:

```
.td table xmp2 width 4i
.td row xmp2 cwidths (1i * * 2*)
.ta table xmp2 on
.in .1i
.ir .1i
.fo left
.ta row xmp2 on
.ta cell 1
This is the first cell in the row. It is 1 inch wide.
.ta cell 2
This is the second cell in the row. It is one fourth of 3 inches
wide (4 inches table width-1 inch cell 1 width).
.ta cell 3
This is the third cell in the row. It is one fourth of 3 inches
wide (4 inches table width-1 inch cell 1 width).
.ta cell 4
This is the fourth cell in the row. It is one half of 3 inches
wide (4 inches table width-1 inch cell 1 width).
This cell is also twice as wide as cell 3.
.ta table off
```

The above input results in

|                                                       |                                                                                                                   |                                                                                                                  |                                                                                                                                                            |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| This is the first cell in the row. It is 1 inch wide. | This is the second cell in the row. It is one fourth of 3 inches wide (4 inches table width—1 inch cell 1 width). | This is the third cell in the row. It is one fourth of 3 inches wide (4 inches table width—1 inch cell 1 width). | This is the fourth cell in the row. It is one half of 3 inches wide (4 inches table width—1 inch cell 1 width). This cell is also twice as wide as cell 3. |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

## .TD [Table Definition]

- The width of the table is used to determine the horizontal displacement of the table when the table is being right-aligned or center-aligned because of the specification of RIGHT or CENTER when using the TABLE parameter. The following table is right-aligned:

```
.td table xmp3 width 4i right
.td row xmp3 cwidths 3i
.td row xmp3a
.ta table xmp3 on
.in .1i
.ir .1i
.ta row xmp3 on
.ta cell 1
This is the first row in this table. The table is
4 inches wide, but the row is only 3 inches wide.
.ta row xmp3a on
.ta cell 1
This is the second row in this table. The table is
4 inches wide, and this row is 4 inches wide also.
.ta table off
```

The above input results in

|                                                                                                     |
|-----------------------------------------------------------------------------------------------------|
| This is the first row in this table. The table is 4 inches wide, but the row is only 3 inches wide. |
|-----------------------------------------------------------------------------------------------------|

|                                                                                                       |
|-------------------------------------------------------------------------------------------------------|
| This is the second row in this table. The table is 4 inches wide, and this row is 4 inches wide also. |
|-------------------------------------------------------------------------------------------------------|

Notice that the width of the table is right-aligned, not the width of the row. Because the width of the first row is one inch less than the width of the table, the row looks as if it is indented one inch from the right margin.

- For page-mode devices, where a table has a common boundary with a row, the horizontal rule specified for the table is always used. The first row in the table uses the top table rule instead of the top row rule. In the following example, the top rule for the first row is the top table rule, which is a .6mm weight rule. The row rules were defined to be .3mm thick:

```
.dr thick weight .6mm
.dr normal weight .3mm
.td table xmp4 hrules thick
.td row xmp4 hrules normal
.ta table xmp4 on
.in .1i
.ir .1i
.ta row xmp4 on
.ta cell 1
This is in the first row of the table.
The horizontal rule on top of this row is the thicker table horizontal
rule.
.ta row xmp4 on
.ta cell 1 on
This is the second row of the table.
The top horizontal rule for the row is used on top of this row.
.ta table off
```

The above input results in

|                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------|
| This is in the first row of the table. The horizontal rule on top of this row is the thicker table horizontal rule. |
|---------------------------------------------------------------------------------------------------------------------|

|                                                                                                      |
|------------------------------------------------------------------------------------------------------|
| This is the second row of the table. The top horizontal rule for the row is used on top of this row. |
|------------------------------------------------------------------------------------------------------|

- For page-mode devices, where a row or table has a common boundary with a cell, the horizontal rule specified for the row or table is always used. For example, the top rule for the cells that occur at the top of a table use the top table rule, not the top cell rule. In the following example, the top rule for the first and second cells and the bottom rule for the first and third cells are the top and bottom table rules, which are defined as .3mm and .9mm thick, respectively. The cell horizontal rules are defined to be .6mm thick:

```
.dr thick weight .6mm
.dr normal weight .3mm
.dr thicker weight .9mm
.td table xmp5 hrules (normal thicker)
.td row xmp5 arrange (1 2 / 1 3)
.td cell 1 xmp5 hrules thick
.td cell 2 xmp5 hrules thick
.td cell 3 xmp5 hrules thick
.ta table xmp5 on
.in .li
.ir .li
.ta row xmp5 on
.ta cell 1
First cell in the table
.ta cell 2
Second cell in the table
.ta cell 3
Third cell in the table
.ta table off
```

The above input results in

|                         |                          |
|-------------------------|--------------------------|
| First cell in the table | Second cell in the table |
|                         | Third cell in the table  |

## .TD [Table Definition]

- For page-mode devices that support shading, you can shade individual table cells. In the following example, the .TD table cell named sdcell1 is shaded with a STANDARD pattern and light shading, the .TD table cell named sdcell2 is shaded with a SCREEN pattern and 50 percent shading, and the .TD table cell named sdcell3 is shaded with a STANDARD pattern and 50 percent shading.

```
.sd sdcell1 standard shade light
.sd sdcell2 screen shade 50
.sd sdcell3 standard shade 50
.td table xmp5
.td row xmp5 arrange (1 2 / 1 3)
.td cell 1 xmp5 shade sdcell1
.td cell 2 xmp5 shade sdcell2
.td cell 3 xmp5 shade sdcell3
.ta table xmp5 on
.in .li
.ir .li
.ta row xmp5 on
.ta cell 1
First cell in the table
.ta cell 2
Second cell in the table
.ta cell 3
Third cell in the table
.ta table off
```

The above input results in

|                         |                          |
|-------------------------|--------------------------|
| First cell in the table | Second cell in the table |
|                         | Third cell in the table  |

For more information on shading, see “.SD [Shading Definition]” on page 328.

- The rule used for the boundary between two rows is the heavier of the two rules specified for that position. A rule with a specification of NONE has lowest priority. In the following example the horizontal rule between the first and second rows is .3mm, the thicker of *normal* and NONE. The horizontal rule between the second and third rows is .6mm, the thicker of *normal* and *thick*. The horizontal rule between the third and fourth rows is also .6mm, the thicker of *normal* and *thick*.

```
.dr thick weight .6mm
.dr normal weight .3mm
.td table xmp6
.td row xmp6 hrules (none normal)
.td row xmp6a hrules (thick normal)
.ta table xmp6 on
.in .1i
.ir .1i
.ta row xmp6 on
.ta cell 1
First occurrence of the XMP6 row
.ta row xmp6 on
.ta cell 1
Second occurrence of the XMP6 row
.ta row xmp6a on
.ta cell 1
First occurrence of the XMP6A row
.ta row xmp6a on
.ta cell 1
Second occurrence of the XMP6A row
.ta table off
```

The above input results in

|                                    |
|------------------------------------|
| First occurrence of the XMP6 row   |
| Second occurrence of the XMP6 row  |
| First occurrence of the XMP6A row  |
| Second occurrence of the XMP6A row |

## .TD [Table Definition]

- The rule used for the boundary between two cells (both vertical and horizontal rules) is the heavier of the two rules specified for that position. A rule with a specification of NONE has lowest priority. In the following example, there is no vertical rule between cell 1 and cell 2, because cell 1 specified NONE as its right rule and cell 2 specified NONE as its left rule. The vertical rule between cells 2 and 3 is .6mm, the thicker of *thick* and *normal*. The vertical rule between cells 3 and 4 is .3mm, the thicker of NONE and *normal*.

```
.dr thick weight .6mm
.dr normal weight .3mm
.td table xmp7 vrules thick
.td row xmp7 arrange(1 2 3 4)
.td cell 1 xmp7 vrules (normal none)
.td cell 2 xmp7 vrules (none thick)
.td cell 3 xmp7 vrules (normal none)
.td cell 4 xmp7 vrules (normal)
.ta table xmp7 on
.in .li
.ir .li
.ta row xmp7 on
.ta cell 1
cell 1
.ta cell 2
cell 2
.ta cell 3
cell 3
.ta cell 4
cell 4
.ta table off
```

The above input results in

|        |        |        |        |
|--------|--------|--------|--------|
| cell 1 | cell 2 | cell 3 | cell 4 |
|--------|--------|--------|--------|

Notice that the left vertical rule for the leftmost cell and the right vertical rule for the rightmost cell are overridden by the table vertical rules.



- For any continuous horizontal or vertical rule in the table, only one thickness of rule is used. The rulename used is the thickest rule specified for any portion of the rule. In the following example, the horizontal rule between cell 1 and cell 3 and between cell 2 and 4 is .6mm, as this is the heaviest rule specified when comparing the bottom rules specified for cells 1 and 2 and the top rules specified for cells 3 and 4.

```
.dr thick weight .6mm
.dr normal weight .3mm
.td table xmp8
.td row xmp8 arrange(1 2 / 3 4)
.td cell 1 xmp8 hrules (normal none) vrules (normal none)
.td cell 2 xmp8 hrules (none normal) vrules none
.td cell 3 xmp8 hrules (thick) vrules (normal)
.td cell 4 xmp8 hrules (normal none) vrules (thick)
.ta table xmp8 on
.in .li
.ir .li
.ta row xmp8 on
.ta cell 1
cell 1
.ta cell 2
cell 2
.ta cell 3
cell 3
.ta cell 4
cell 4
.ta table off
```

The above input results in

|        |        |
|--------|--------|
| cell 1 | cell 2 |
| cell 3 | cell 4 |

In the above example, the horizontal rule between cells 1 and 3 was specified as *thick* (NONE was specified as the bottom rule for cell 1 and *thick* was specified as the top rule for cell 3), and the horizontal rule between cells 2 and 4 was specified as *normal*. Because *thick* is thicker than *normal*, *thick* is used for the horizontal rule that separates cells 1 and 3 and cells 2 and 4.

## .TD [Table Definition]

- If a MINDEPTH value, for example, “2i,” is given for a 90° or 270° rotated cell, the contents of that cell is formatted with a width of 2 inches, even if the actual depth of the cell is made greater to accommodate other cells in the row. The following example demonstrates this using a 90° rotated cell with a MINDEPTH value of 2 inches:

```
.td table xmp9
.td row xmp9 arrange(1 2) cwidths (1i *)
.td cell 2 xmp9 rotate 90 mindepth 2i
.ta table xmp9 on
.in .1i
.ir .1i
.ta row xmp9 on
.ta cell 1
This cell has 0° rotation.
There is more and more and more
and more and more and more and more and more and more
and more and more and more and more and more and more
text in this cell.
.ta cell 2
This cell is rotated 90° and has a MINDEPTH value of 2 inches.
Notice that the depth of the cell has been increased beyond 2 inches
because of the depth of the first cell in this row being greater than
2 inches.
The contents of this cell is still formatted only 2 inches wide.
.hr left right
Notice that the length of the horizontal rule that was
specified as .HR LEFT RIGHT is only 2 inches long.
.ta table off
```

The above input results in

|                                                                                                                                                       |  |                                                                                                           |                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| This cell has 0° rotation. There is more and more and more and more and more and more and more and more and more and more and more text in this cell. |  | Notice that the length of the horizontal rule that was specified as .HR LEFT RIGHT is only 2 inches long. | This cell is rotated 90° and has a MINDEPTH value of 2 inches. Notice that the depth of the cell has been increased beyond 2 inches because of the depth of the first cell in this row being greater than 2 inches. The contents of this cell is still formatted only 2 inches wide. |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## .TE [Terminal Input]

### Function

Use the .TE [Terminal Input] control word when you want to enter text or control lines during the processing of the input file.

### Syntax



### Parameters

- n* Specifies the number of lines that are accepted from the terminal. If *n* is omitted, 1 is assumed.
- ON** Starts an open-ended terminal input mode.
- OFF** Turns off terminal input mode if it was ON, or if *n* was given and has not yet been exhausted.
- line* Identifies an input line to be processed. The *line* form is available with .TE because it is a type 1 control word, but it actually does not read anything from the terminal. The control word .te read this line causes the line “read this line” to be processed as an ordinary input line, but SCRIPT/VS obviously does not read it from the terminal, because it already has the line.

### Notes

- This is a type 1 control word.
- The line form of the .TE control word starts a page.
- The terminal input mode is included in the active environment.

### Remarks

1. When the .TE control word is encountered, your terminal keyboard is unlocked to accept input lines. The input lines can be text or control words and are processed as if they had been read from an imbedded file (see the .IM [Imbed] control word). The only exceptions to this are the .GO [Goto] and ... [Set Label] control words, which are not allowed during terminal input. If a numeric operand was specified, terminal input is ended after reading *n* lines. If no operand was specified, only one line is read from the terminal. If ON was specified, input is accepted from the terminal until ended with .TE OFF.

When terminal input is ended, processing reverts back to the line following the .TE control word in the file. If the TWOPASS or FPASSES option of the SCRIPT command is in effect, .TE control words in the input file are processed on each pass.

2. If you use the .TE control word while the formatted output is being displayed at your terminal, the input and output can be interspersed. This can be useful for testing or experimentation, but it is not usually appropriate for final output.

## **.TE [Terminal Input]**

3. The .RD [Read Terminal] control word merely unlocks the keyboard to allow you to type lines in the midst of the normal terminal output. It does not process what you type. The .TE control word, on the other hand, can be used to enter control words or to cause text input to be formatted and to appear in the output when the output is written to a device other than the terminal. The .TE control is, in effect, an imbed, where the “file” imbedded is your keyboard.
4. Use the .TY [Type on Terminal] control word immediately before the .TE control word to display prompting messages.
5. If .TE ON was specified, the number of lines to be read is open ended. It can be ended by .TE OFF, but because your keyboard is a simulated imbed file, the .EF, .QQ, or .QU control words also end it.
6. The .TE control word can be used to enter control words to specify a particular processing of the input file, such as revision codes or conditional sections.
7. Terminal input can be read from a disk file if the terminal input file name DSMTERMI has been associated with the file or data set name with the .DD [Define Data File-id] control word. See the discussion of .DD for more information.
8. In the ATMS-III environment, a null line is substituted for the .TE [Terminal Input] control word. In batch environments, the file DSMTERMI is read.

---

## **.TH [Then]**

### **Function**

The .TH [Then] control word can be used in conjunction with the .IF [If] control word to process SCRIPT/VS input lines conditionally. The target line is processed only if the most recently performed .IF [If], .AN [And], or .OR [Or] control word resulted in a true condition.

### **Syntax**

►► .TH *target* —————►◄

### **Parameters**

*target* Any valid SCRIPT/VS input line containing a control word, a macro, or text. If the most recently performed .IF [If], .AN [And], or .OR [Or] is true, the target line is processed next, with the first nonblank character after the .TH control word treated as the first position of the subject line. If the condition is not true, the target line is ignored, and processing continues with the input line that follows the .TH control line.

### **Remarks**

1. For readability, an optional EN can be added without an intervening blank to the .TH control word. This allows the control word to be written as .TH or .THEN. However, the short form (.TH) is recommended for performance reasons.
2. The .TH [Then] and .EL [Else] control words, in conjunction with .IF [If], .AN [And], and .OR [Or], allow you to construct complex logic statements.
3. The .TH and .EL control words do not cause a break or change the true/false condition; a target control word might, if it is processed. For example, the input lines
 

```
.if &a eq &b
.th .if &c eq &d
.el .ty Yes.
```

 changes the true/false condition if &a is equal to &b and &c is not equal to &d. Also, the following input lines
 

```
.if &a eq &b
.th .cl 20
```

 causes a break if &a is equal to &b.
4. Multiple .TH [Then] and .EL [Else] control words can follow an .IF [If], .AN [And], or .OR [Or] control word; only the .TH [Then] control words are executed if the .IF [If], .AN [And], or .OR [Or] resulted in a true comparison; only the .EL [Else] control words are executed if the .IF [If], .AN [And], or .OR [Or] resulted in a false comparison.
5. If there is no most recently performed comparison, the target is processed.

## Examples

- The following input lines

```
.if &a eq &b .ty Yes,  
.if &a eq &b .ty still.
```

are equivalent to the following lines:

```
.if &a eq &b  
.th .ty Yes,  
.th .ty still.
```

- You can use the .TH and the .IF control words to adapt a document for a specific audience. For example, if you specify

```
.if &SYSVARA = 3  
.th .cs 1 ignore  
.th .cs 2 ignore
```

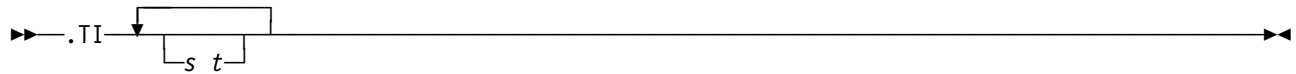
then if SYSVARA is equal to 3, the two .TH control words are processed, causing conditional sections 1 and 2 to be ignored.

## .TI [Translate Input]

### Function

Use the .TI [Translate Input] control word to translate the input text from one input representation to another. This control word *should be used with caution*, because this translation occurs before any other processing is done.

### Syntax



### Parameters

- s* Source character to be translated. It can be a single character or a 2-character hexadecimal code.
- t* Desired output representation of the source character. It can be a single character or a 2-character hexadecimal code.

**Initial Setting:** No translate is in effect.

**Default:** Restores the initial setting.

### Remarks

1. Multiple pairs of translate characters can be specified with a single .TI control word.
2. The text associated with running headings and running footings is translated using the initial translate tables as modified by any .TI control words that you used in the running heading or footing.
3. Translate-character specifications remain in effect until explicitly respecified.
4. Input translations specified with .TI are performed on the input lines *prior* to any other SCRIPT/VS processing. Output translations specified with .TR are performed on the output lines *after* SCRIPT/VS has completed all processing.
5. The translation defined by .TI is performed both on control word lines and on text lines. You should be aware that the following character codes have special meaning when they appear in text lines, and that the .TI translation is performed *before* these characters are examined:

- 05** Horizontal tab
- 16** Backspace
- 40** Blank
- 41** Required blank

**Note:** The character code for required blank can be changed with the .DC [Define Character] RB control word.

When these characters are found in text lines, they are removed and replaced with other characters or controls, depending upon the physical output device type. All other characters found in text lines are treated as data and are placed in the formatted output after processing, according to the .TR [Translate Character] and, if applicable, .TU [Translate Uppercase] control words.

6. Problems might arise during processing if the following characters are used with the .TI control word: =, <, and >.

## Examples

- If your terminal does not have a tab key, you can translate some seldom-used key, such as the not-sign, to perform the tab function in your document. For example, if you specify

```
.ti ~ 05
```

this translates the not-sign to hexadecimal X'05', the tab character code. While translation is in effect, any not-sign on an input line works as a tab.

- Be careful which characters you use as source characters for translating. Input translation is done before control word processing, so if you specify

```
.ti . 05
```

then all periods are translated to hexadecimal X'05', the tab character, and no control words are recognized.

- To reset an input translation pair back to their original value, you must use the hexadecimal value of the character as the source character. If you had specified

```
.ti ~ 05
```

and later you wanted the not-sign to produce a not-sign and not a tab, you would have to specify

```
.ti 5F 5F
```

If you had just specified

```
.ti ~ ~
```

the not-signs would be translated to hexadecimal X'05' before the .TI control word is processed and this would result in

```
.ti 05 05
```

therefore, the not-signs would continue to be translated to hexadecimal X'05' after the .TI control word was processed.

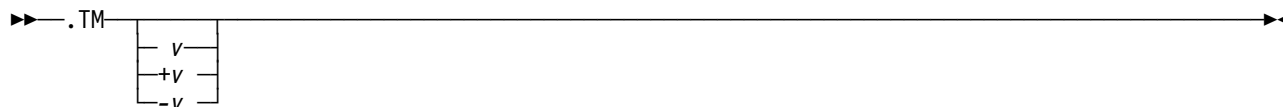


## .TM [Top Margin]

### Function

The .TM [Top Margin] control word specifies the amount of vertical space to be left at the top of subsequent output pages.

### Syntax



### Parameters

- $v$  Specifies the amount of vertical white space to be left at the top of output pages.
- $+v$  or  $-v$  increases or decreases the existing top margin by the amount given.

**Initial Setting:** Dependent upon the logical device specified.

**Default:** Restores the initial setting.

### Notes

- .TM ends a keep, float, footnote, named area, or table.
- .TM takes effect on the next page.
- The size of the top margin is included in the page environment.

### Remarks

1. The running heading is placed directly below the top margin, as shown in Figure 6 on page 438.
2. The value given should not be so large that the top margin and bottom margin together fill the entire page.
3. The size of the top margin is not affected by line spacing.
4. For the 1403, SCRIPT/VS assumes that the channel one punch of the carriage tape is located one-half inch below the top edge of the page. The default top margin for 1403 devices is one-half inch, and SCRIPT/VS issues a Skip Immediate to Channel One carriage control to advance to each new page.

If you set the top margin to less than one-half inch, or instruct SCRIPT/VS to print above channel one, a blank page precedes the first page of output, and SCRIPT/VS spaces from each page to the next. If you also change the page length, the top of the logical page may not coincide with the top of the physical page. If SCRIPT/VS is instructed to print below channel one, regardless of the top margin setting, SCRIPT/VS issues a Skip Immediate to Channel One carriage control before printing the first line.

5. The default values for each logical device are listed in the logical device tables on pages 27 through 28.

---

## .TP [Tab Position]

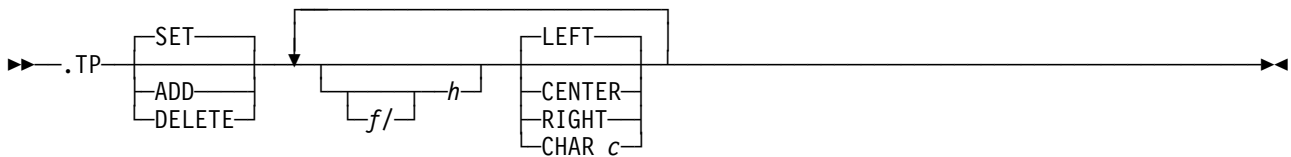
### Function

Use the .TP [Tab Position] control word to define how tab characters (hexadecimal X'05') in text are to be resolved. When tab positions have been defined with .TP, the text following each tab character in an input line is aligned at the corresponding tab position. That is, the text following the first tab character is aligned at the first tab position, the text following the second tab character is aligned at the second tab position, and so on.

Text can be left-, center-, or right-aligned at the tab position, or it can be aligned with the first occurrence of a specific character at the tab position.

The .TP control word completely replaces any previous .TB [Tab Setting] control word.

### Syntax



### Parameters

**SET** Specifies that the tab positions given completely replace the old tab positions.

If no parameters are specified, all tab settings are cleared.

**ADD** Specifies that the tab positions given are to be added to any positions defined with a previous .TP control word.

**DELETE** Specifies that the tab positions given are to be removed from those defined with a previous .TP control word.

**f/** Specifies a fill character to be used in filling the space between any text already on the output line and this tab position. *f* can be given as a single character or as a 2-character hexadecimal code.

If no fill character is given, horizontal white space is used.

**h** Specifies the horizontal position of a tab. When the corresponding tab character is found in a text line, SCRIPT/VS aligns the text following the tab at this position. That is, the text following the first tab character is aligned at the first tab position, the text following the second tab character is aligned at the second tab position, and so on.

**LEFT** Specifies that text following a tab character is to be left-aligned at this tab position. This is the default.

**CENTER** Specifies that the text following a tab character is to be centered about this tab position.

**RIGHT** Specifies that the text following a tab character is to be right-aligned at this tab position.

**CHAR** Specifies that the text following a tab character is to be aligned with the first occurrence of the character *c* left-aligned at this tab position. *c* can be given as a single character or as a 2-character hexadecimal code.

**Initial Setting:** None

**Default:** Clears the tab table.

## Notes

- .TP causes a break.
- The tab settings are included in the active environment.

## Remarks

1. Tab characters (hexadecimal X'05') can be created in several different ways:

- Some terminals, such as the 2741 terminal, have a special key that generates a tab character when pressed.
- Some text editors, such as the CMS editor, allow you to assign the tab function to any keyable character. When the specified character is entered, the editor changes it to a tab character.

Both of these techniques have the disadvantage of putting nondisplayable data into the input file. When such a file is examined with a different terminal or editor, the tab characters may be invisible. SCRIPT/VS provides several additional techniques for creating tabs without placing tab characters in the input file:

- The value of the SCRIPT/VS system symbol &\$TAB is a tab character (hexadecimal X'05'). This symbol can be used whenever a tab is needed.
- The .TI [Translate Input] control word can be used to translate any keyable character to a tab character on input.

2. If an input line contains more tab characters than the number of defined tab positions, the extra tab characters are converted into word spaces.

3. The .TP and .TB [Tab Setting] control words work differently. Refer to “.TB [Tab Setting]” on page 374 to learn how the .TB [Tab Setting] control word differs from the .TP control word. Do not use both control words in the same document.

4. Tab stops specified with the .TP control word need not be specified in ascending order.

5. When a fill character has been specified for a tab position, the space between the preceding text and the tabbed text is filled with as many fill characters as will fit. These fill strings are formatted in the font that is current when the corresponding tab character is processed.

6. Using fill characters on the 3800 Model 3 causes printer performance problems. Use the .SX [Split Text] control word instead, if possible.

7. Text aligned at a tab position defined with .TP can overlay other text on the line if there is not enough room for it.

8. The .TP control word completely replaces any previous .TB [Tab Setting] control word: if the tab table currently in effect was defined with .TB, then .TP ADD is processed as .TP SET, and .TP DELETE clears the tab table.

9. No more than 64 tab positions can be set at one time.

10. If the space to the next tab stop is less than the width of one fill character, the following tab stop is then used.

## Examples

- When formatting tabular material, text can be overlaid if some text is wider than the distance between consecutive tabs. For example, the following table

```
.ti ~ 05
.tp 14 24
.fo off
Archeologia~Italian~ITDRM
Canadienne~French~FNDRM
Deutschsprachig~German~GEDRM
Elliptique~French~FNDRM
Magnifique~French~FNDRM
Polverizzo~Italian~ITDRM
Rechtsprechung~German~GEDRM
Studenteschi~Italian~ITDRM
```

is formatted as

|                 |         |       |
|-----------------|---------|-------|
| Archeologia     | Italian | ITDRM |
| Canadienne      | French  | FNDRM |
| Deutschsprachig | German  | GEDRM |
| Elliptique      | French  | FNDRM |
| Magnifique      | French  | FNDRM |
| Polverizzo      | Italian | ITDRM |
| Rechtsprechung  | German  | GEDRM |
| Studenteschi    | Italian | ITDRM |

| —————>—————>

Because the German word “Deutschsprachig” is wider than the space allowed for the first column, the text of the second column has overlaid part of the word. Compare this effect to the results obtained when tab positions are defined with the .TB [Tab Setting] control word.

- You can define tab positions at which text is centered or right-aligned. For example:

```
.ti ~ 05
.tp .5i right 1.5i center
~i)~Algol
~ii)~Altair
~iii)~Betelgeuse
~iv)~Deneb
~v)~Regulus
~vi)~Vega
```

The text following the first tab character is right-aligned at the first tab position, one-half inch from the margin. The text following the second tab character is centered about the second tab position, one and one-half inches from the margin:

|      |            |
|------|------------|
| i)   | Algol      |
| ii)  | Altair     |
| iii) | Betelgeuse |
| iv)  | Deneb      |
| v)   | Regulus    |
| vi)  | Vega       |

„————>—————>

- You can also indicate that the text following a tab character be aligned with the first occurrence of a specific character at the tab position. For example:

```
.ti ~ 05
| .tp .5i left -/2i char .
  ~Expensive~$1234
  ~Cheap~$.005
  ~Reasonable~$1.50
```

The text following the first tab character is left-aligned at the first tab position. The text following the second tab character is aligned with the first period (&period) at the second tab position, and the space between the two pieces of text is filled with hyphens (-):

```
Expensive -----$1234
Cheap -----$.005
Reasonable -----$1.50
```

```
|  ———> —————>
```

If, as in the first line in this example, the alignment character does not appear in the text, the text is right-aligned at the tab position.

- When tabs are added with .TP, they always follow any previously defined tab positions. For example,

```
.ti ~ 05
.tp 10 20 30
~1~2~3
```

is formatted as

```
          1          2          3
|—————> —————> —————>
```

You can add tabs with the ADD parameter of .TP:

```
.ti ~ 05
.tp add 15 25
~1~2~3~4~5
```

This is formatted as

```
          1      4      2      5      3
|—————> ———> ———> ———> ———>
```

The text following the fourth tab in the input line was positioned according to the fourth tab defined; the text following the fifth tab was positioned according to the last tab defined. The same results can be achieved by specifying:

```
.tp set 10 20 30 15 25
~1~2~3~4~5
```

- If the .TP control word is entered with no parameters, the tab table is cleared. For example,

```
.ti ~ 05
.tp
~x~x~x~x
```

is formatted as

```
x x x x
```

Because the tab table contains no entries, all tabs in the input line are effectively beyond the last defined tab and are converted into blanks.

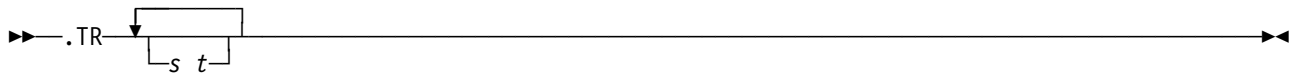
---

## .TR [Translate Character]

### Function

The .TR [Translate Character] control word allows you to specify the output representation of each character in the source text.

### Syntax



### Parameters

*s* Source character to be translated. It can be a single character or a 2-character hexadecimal code.

*t* Desired output representation of the source character.

**Default:** Restores the initial output translate table.

### Remarks

1. The .TR control word is primarily useful when the final output device uses a different character set than was used to create the source SCRIPT file. When you use a .TR in a document, it is usually for a specific font. The .TR can cause unpredictable results when you are printing on different printers.
2. The .TR control word should not be used for drawing boxes. You should use the .BX control word instead.
3. More than one pair of source and intended output codes can be specified with a single .TR control word.
4. The text associated with running headings and running footings is translated using the initial translate tables as modified by any .TR control words that you used in the running heading or footing.
5. Because control words are processed only internally, they are never translated by the .TR control word. However, text data associated with a control word (as in running headings and footings and typed messages) can be translated.
6. Translate-character specifications remain in effect until explicitly respecified or until the translation table is reinitialized with a .TR control word.
7. By using the .IF, .CS, or .TE control words, you can specify different output character sets for different runs with different output devices.
8. Output translations specified with the .TR control word are performed on the output lines during formatting just before the characters' widths are measured for justification. Input translations specified with the .TI control word are performed on the input lines *prior* to any other SCRIPT/VS processing.
9. Unprintable characters are handled differently by different output devices. Therefore, you should ensure that the output character codes you specify with the .TR control word can be printed by the destination device using the current font.
10. On page printers, the blank (hexadecimal X'40') is used as a variable space character. Therefore, you should not specify the output representation of any character as hexadecimal X'40'.
11. The .TR control word should not be used for pairs of characters of different sizes. The difference in size changes line lengths and heights in an unpredictable manner. Use the .TI [Translate Input] control word for this type of translation.

## Examples

The following .TR control word:

```
.tr 0 b0 1 b1 2 b2 3 b3 4 b4 5 b5 6 b6 7 b7 8 b8 9 b9
```

causes the characters 0 through 9 to print as their corresponding superscript characters if they are available in the current font. For example, the formula:

X2+Y2=Z3 prints as:  $X^2+Y^2=Z^3$

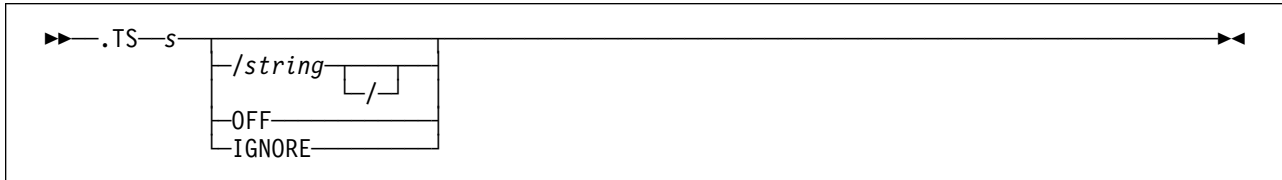
---

## .TS [Translate String]

### Function

Use the .TS [Translate String] control word to translate an input character to a string. This control word *should be used with caution*, because this translation occurs after input translation but before any other processing is done.

### Syntax



### Parameters

- |               |                                                                                                                                                                                                                                        |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>s</b>      | Source character to be translated. It can be a single character or a 2-character hexadecimal code.                                                                                                                                     |
| <b>/</b>      | Any delimiter character.                                                                                                                                                                                                               |
| <b>string</b> | Desired output representation of the source character. It can be any string from 0 to 255 characters long.                                                                                                                             |
| <b>OFF</b>    | Specifies that <i>string</i> translation is to be stopped for the specified character.                                                                                                                                                 |
| <b>IGNORE</b> | Specifies that the given source character is to be retained in the text and included in the output, but not measured during formatting. That is, a character specified with the IGNORE parameter is treated as a zero-width character. |

### Remarks

1. A character can be set to a null string by either specifying adjacent delimiters or omitting the string specification altogether. This effectively deletes the character from the output.
2. Unlike other delimited strings (for example, on the .SX [Split Text] control word), the .TS string is not scanned for internal delimiters. Thus the end delimiter is required only if you want trailing blanks.
3. String translation is performed only when symbol substitution is in effect.
4. Unlike the .TI [Translate Input] and .TR [Translate Character] control words, you must explicitly stop each string translation by specifying the OFF option. For example,

```
.ts % /<percent sign>
```

causes all occurrences of % in the input to be replaced by the string <percent sign> until you enter

```
.ts 6c off
```

Note that it can be difficult to use the .TS control word unless you turn substitution off or you specify the value in hexadecimal. If .TS % OFF had been specified, the percent sign (%) would get translated to <percent sign> before the .TS function was performed.

Be careful if you have translated one of the characters used as a hexadecimal digit.

5. String translations specified with the .TS control word are stored in the symbol table. Because string translation occurs at the same time as symbol substitution, the translated string is not subject to further symbol substitution or GML scanning.



6. The IGNORE parameter is intended to be used for unprintable characters, such as inline control characters. This parameter allows the formatter to ignore the width of unprintable characters when calculating the length of the output line. If you are using translated characters that are printable but have been ignored because the IGNORE parameter was used, an ignored character has a zero width only in relation to text that is formatted when the character is formatted. Vertical rules, tabs, or text from other columns or areas that occur on the same output line and to the right of an ignored character does not behave as if that character has a zero width.
7. Do not attempt to translate the ampersand (&) and the GML tag delimiter (:) or their hexadecimal equivalents. These are special characters required for SCRIPT/VS processing, and an error message is issued if an attempt is made to translate them into a string.
8. If you use .TS to translate a character to a string that contains the same character, do not enter the string using the control word separator. This causes a recursive redefinition of the .TS character. For example:

```
. 'se text ' .ct ---;.bf =;.ct nn_or_nnn_or_console;.pf;.ct ---'
.ts 6d /us_/
&text
```

results in:

```
---nnususus_orususus_nnnususus_orususus_console---
```

---

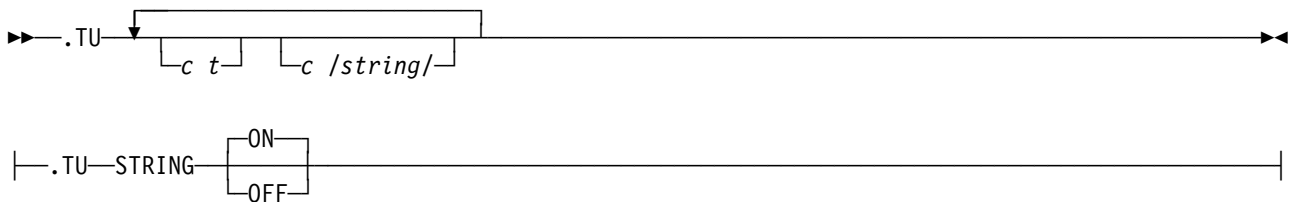
## .TU [Translate Uppercase]

### Function

The .TU [Translate Uppercase] control word allows you to specify the output representation of characters when uppercase output has been requested with the .BF, .UC, or .UP control words, with the &U' attribute, or with the UPCASE option on the SCRIPT command.

The .TU [Translate Uppercase] control word allows you to specify character-to-character translation, which allows translation of a single lowercase character to a single uppercase character, or character-to-string translation which allows translation of a single lowercase character to a string of uppercase characters.

### Syntax



### Parameters

- c** Source character to be translated to an uppercase single character or character string. This character can be a single character or a 2-character hexadecimal code.
- t** Desired single character uppercase output representation of the source character. This character can be a single character or a 2-character hexadecimal code.
- /** Any delimiter character. Both the beginning and ending delimiters are required on the specified string. Both delimiters must be the same character. If both delimiters are not specified, an error message is issued.
- string** Desired uppercase output representation of the source character. The maximum length of the string is 4 characters. The *string* can be either characters or 2-character hexadecimal codes if the &X' attribute is used. You must put a period after the hexadecimal code.
- Note:** If you specify more than 4 characters or 4 hexadecimal codes, a warning message is issued and the string is truncated to 4.
- STRING** Required parameter to change from character-to-character translation to character-to-string translation. The minimum abbreviation is ST.
- ON** Specifies to do character-to-string translation with the source characters and strings specified with the .TU control word. The default is ON if STRING is specified.
- OFF** Specifies that character-to-string translation is stopped and character-to-character translation continues.

**Initial Setting:** No translate is in effect.

**Default:** Restores initial setting.

## Remarks

1. The .TU [Translate Uppercase] control word is used primarily when the default uppercase translate table designed for English is not appropriate. Character-to-string translation is useful where a one-to-many character translation is needed. For example, the sharp s small ( $\beta$ ) character can be translated to an uppercase “SS.” Character-to-string translation is controlled with the .TU STRING ON/OFF parameter.
2. Translate specifications remain in effect until explicitly respecified.
3. The default uppercase translation has the same effect as the following .TR [Translate Character] control word:  

```
.tr a A b B ...z Z
```
4. More than one pair of source and intended output codes can be specified with a single .TU control word:  

```
.tu a A c /CC/ g G 1 /LLL/
```

**Note:** Both character-to-character and character-to-string translations can be specified with a single .TU control word.
5. The delimiters on the specified string on .TU define character-to-string translation to SCRIPT/VS. The STRING ON and OFF parameters activate and deactivate the actual character-to-string translations. The character string can contain blanks, but you cannot have a blank as the first character after the delimiter.
6. Hyphenation and spelling verification are performed **after** uppercase translation is done. You can put entries into a dictionary to show desired hyphenation points in words containing the translated string.
7. If a character is specified without specifying an output representation, an error message is issued and the .TU control word is ignored.

## Examples

These examples show the formatting results using character-to-string translation with .TU. Entering

```
.tu a /&x'c2./
.tu string on
.up on
the following should uppercase to B's:
.br
aaaaaaaaaaaaa
```

results in

```
THE FOLLOWING SHOULD UPPERCASE TO B'S:
BBBBBBBBBBBBB
```

## **.TU [Translate Uppercase]**

Entering

```
.fo off
.up on
uppercase this
.tu a B
uppercase this
.tu a /CC/
.tu string on
uppercase this
.tu string off
uppercase this
.up off
```

results in

```
UPPERCASE THIS
UPPERCBSE THIS
UPPERCCCSE THIS
UPPERCBSE THIS
```

Notice that the delimiters on the .TU control word indicated to SCRIPT/VS that character-to-string translation was to be done. Turning off character-to-string translation returns translation to the specified single character translation and not to the default “a” to “A” translation. If single character translation had not been specified with .TU, translation would have defaulted to “a” to “A” when “.tu string off” was specified.

---

## **.TY [Type on Terminal]**

### **Function**

The .TY [Type on Terminal] control word causes one line of information to be displayed at your terminal or written into the file DSMTERMO, no matter where the SCRIPT/VS formatted output is going.

### **Syntax**

►► .TY—*text*—————►◄

### **Parameters**

*text*    The line to be typed. It is used only for this message. It does not become part of your document unless the document output is also being typed at your terminal.

### **Remarks**

1. You can use the .TY control word to issue a prompting message before a .TE [Terminal Input] or .RV [Read Variable] control word.
2. The .TY control word causes the file DSMTERMO to be created in the DLF environment.
3. When the .TY control word is processed, the text line given is typed at the terminal. This line is *not* part of the document. SCRIPT/VS does not process the line for output; the line is not justified, or formatted in any way. However, the line is scanned for control word separators, and symbols are substituted. The text to be typed is translated according to the .TR [Translate Character] translations currently in effect.
4. The information line printed is not counted as part of the normal output. Thus, if the formatted output is being typed on the terminal, the paper positioning can become incorrect and require manual adjustment. In general, the .TY control word should be used for document-driven messages when the formatted output is going to a printer or to a disk file.
5. Contrast this control word with the .MG [Message] control word, which allows you to issue a true SCRIPT/VS message. A true message can have any of several degrees of severity, it can stop SCRIPT/VS processing, and its destination and final form are controlled by the MESSAGE command option. The .TY control word merely types out a line without any of the function of a true message.
6. The .TY control word is ignored in the ATMS-III environment.

### **Examples**

Typing in information at the terminal

```
.ty Do you want 2 column output?
.rv answer
.if x&U'&answer ne xYES .go by2col
.cd 2 0 46
.cl 43
...by2col
```

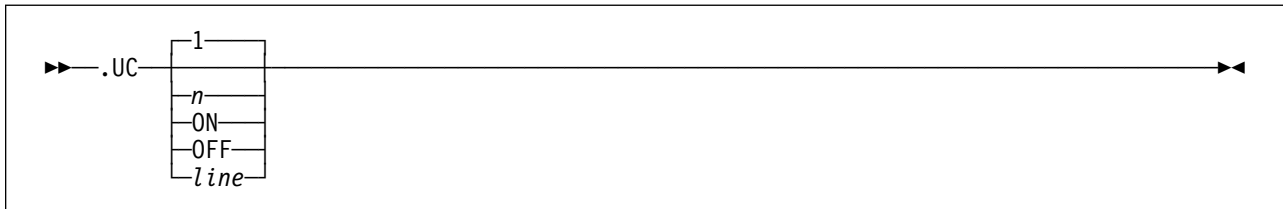
---

## .UC [Underscore and Capitalize]

### Function

The .UC [Underscore and Capitalize] control word automatically underscores and capitalizes one or more input lines.

### Syntax



### Parameters

- n* Specifies the number of input lines to be underscored and capitalized. If *n* is omitted, 1 is assumed. If .UC *n* is specified when .UC ON is in effect, underscoring and capitalization are turned off when *n* lines have been underscored and capitalized, or when a .UC OFF is encountered, whichever occurs first.
- ON** Specifies that subsequent text lines are to be underscored and capitalized.
- OFF** Terminates underscore and capitalization mode if it was ON, or if *n* has been specified and has not been exhausted.
- line* A single text line to be capitalized and underscored.

**Initial Setting:** OFF

**Default:** 1

### Notes

- This is a type 1 control word.
- The line form of the .UC control word starts the page.
- Capitalization and underscoring are included in the active environment.

### Remarks

1. Use the .UC control word whenever you have a line of data that is to be formatted in capital letters and underscored. This control word provides the combined function of .US [Underscore] and .UP [Uppercase].
2. The use of “.UC *line*” while a “.UC *n*” is still in effect resets to zero the number of lines to be underscored and capitalized.
3. The .UC control word does not cause an automatic break; single words in a sentence can be underscored and capitalized.
4. By default, capitalization is performed by translating a–z to A–Z. The .TU [Translate Uppercase] control word can be used to extend capitalization for languages other than English.
5. If a .UC control word is active when the table of contents is formatted, the lines in the table of contents are underscored and capitalized accordingly.

6. The .UD [Underscore Definition] control word controls whether blank characters (X'40', X'41') and spaces generated by the .IS control word are underscored by the .UC control word. It also allows you to specify, for page printers, what *named* rule is used for underscoring and where the rule is placed relative to the baseline.
7. Underscores are not affected by the .SB control word. Use the .UD [Underscore Definition] control word to specify the vertical position of underscores relative to the baseline.

## Examples

Underscoring and capitalizing a single word

This sentence has

.uc one

word processed by .UC.

results in

This sentence has ONE word processed by .UC.

---

## .UD [Underscore Definition]

### Function

Use the .UD [Underscore Definition] control word to determine how automatic underscoring with the .US [Underscore] and .UC [Underscore and Capitalize] control words, or with fonts defined with the US or UC parameters of the .DF [Define Font] control word, or with headings defined with the US or UC parameters of the .DH [Define Head Level] control word, should be performed. You can indicate whether blanks are to be underscored and, on page devices, which *named* rule is to be used for underscoring, underscored fonts, and underscored headings and where it is to be located with respect to the baseline.

### Syntax



### Parameters

**ON** Specifies that blanks are to be underscored.

**OFF** Specifies that blanks are not to be underscored.

*rulename* Identifies a *named* rule to be used in drawing underscores on page printers. The *rulename* parameter is ignored for line devices.

**Note:** If a color is desired for the underscore, specify a *rulename* that was defined with that color on the .DR [Define Rule] control word.

*v* Indicates where the underscore should be placed, with respect to the baseline, for page printers. If *v* is positive, the rule is drawn above the baseline. If *-v* is specified, the rule is drawn below the baseline. *v* is ignored for line devices.

**Initial Setting:** ON

**Default:** Restores the initial setting. For page printers, the default underscore is .2mm thick and is placed .4mm below the baseline.

**Note:** The underscore definition mode is included in the active environment.

### Remarks

1. Word spaces are normally underscored, unless the OFF parameter has been specified. Tab expansions and horizontal space inserted with the TO parameter of the .IS [Inline Space] control word are never underscored. Blank characters, required blanks, and spaces generated by the .IS control word are underscored if ON is specified.
2. On line devices, underscoring is achieved by overstriking text with the underscore character (\_).  
On page printers, underscoring is achieved by drawing a rule beneath text. The default underscore is 0.2 millimeters thick and is placed 0.4 millimeters below the baseline.
3. If the underscore rule is positioned above the normal baseline on page printers, it may overlay text.
4. The 3800 Printing Subsystem Model 3 fonts, the 3820 Page Printer fonts, and the PostScript fonts include underscoring information in the font objects. This built-in underscore definition is used for these fonts unless you explicitly specify the .UD control word with a *rulename* or position.



Each time a new font is started or restarted, the underscore definition is changed to use the underscore definition in the new font. However, this definition does not take effect until a new underscore rule is started for the next output line if underscoring of blanks is on, or for the next word if underscoring blanks is off.

5. The vertical position of underscores is not affected by the .SB [Shift Baseline] control word.

## Examples

- By default, when you underscore text by entering

```
.us on
```

all characters, including word spaces, are underscored. But if you have entered

```
.ud off
```

word spaces are not underscored. Nonblank characters are always underscored.

- You can explicitly position the underscore rule on page printers. For example, if you enter

```
.dr thick weight .6mm
```

```
.ud thick -p2
```

a rule is drawn two pica points below the baseline of underscored text:

a rule is drawn two pica points below the baseline of underscored text:

If you enter

```
.dr thin weight .4mm
```

```
.ud thin p3
```

a rule is drawn three pica points above the baseline, right through the middle of the text:

~~a rule is drawn three pica points above the baseline, right through the middle of the underscored text:~~

- When the underscore definition is changed because a .UD control word or a new 3800 Printing Subsystem Model 3 font has been specified, the new definition takes effect with the next underscore rule. For example, the following lines

```
.df hi1 type(bold)
```

```
.us on
```

This text is of great interest to us and is

```
.bf hi1
```

underscored

```
.pf
```

with the same underscore definition.

```
.ud off
```

This text, however, is also of great interest to us and is

```
.bf hi1
```

underscored

```
.pf
```

but with different underscore rules.

result in

This text is of great interest to us and is **underscored** with the same underscore definition.  
This text, however, is also of great interest to us and is **underscored** but with different  
underscore rules.

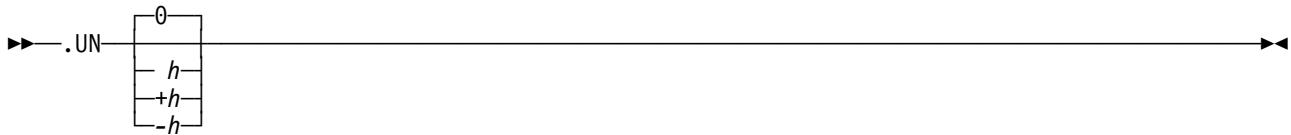
---

## .UN [Undent]

### Function

Use the .UN [Undent] control word to cause the next line to be shifted. The current indentation is changed only for the next line, then it is restored to its previous value for subsequent lines.

### Syntax



### Parameters

$h$  Specifies the amount of horizontal space by which the indentation is to be reduced only for the next output line. If  $-h$  is specified, the .UN control word is effectively the same as the .IL [Indent Line] control word. If  $h$  is omitted, 0 is assumed, and the indentation is not changed.

**Initial Setting:** 0

**Default:** 0

### Notes

- .UN causes a break.
- The undent value is included in the active environment.

### Remarks

1. The .UN control word provides function similar to that provided by the .OF [Offset Mode] control word. The choice between using .UN and .OF is usually a matter of personal preference. They can also be used at the same time to control margins that shift both right and left.
2. The value specified in a .UN control word is subtracted from the current indentation (indent value plus offset value) to determine where to format the next line. If the value specified with the .UN control word exceeds the current indentation amount, an error message results.
3. If successive .UN or .IL control words with positive or negative specification for  $h$  are encountered without intervening text lines, the .UN value is reset to the latest specified value each time.
4. The .UN control word is triggered by the next text, skip, or space line.
5. The value of  $h$  represents the amount of blank space left before text. Thus, .UN .5i indents only the next line one-half inch less than the currently set margin, and the text begins *after* this blank margin area.

### Examples

This example uses the .UN control word to change the indentation specified in a previous .IN control word:

`.in 3p`

`.un 3p`

If an indentation of 3 picas is in effect (as in these lines), the next line is undented to the left margin; all following lines have the normal indentation of 3 picas from the left margin.

results in

If an indentation of 3 picas is in effect (as in these lines), the next line is undented to the left margin;  
all following lines have the normal indentation of 3 picas from the left margin.

---

## .UP [Uppercase]

### Function

The .UP [Uppercase] control word automatically capitalizes one or more input lines.

### Syntax



### Parameters

- n* Specifies the number of input lines to be capitalized. If *n* is omitted, 1 is assumed. If .UP *n* is specified when .UP ON is in effect, capitalization is turned off when *n* lines have been capitalized, or when a .UP OFF is encountered, whichever occurs first.
- ON** Specifies that subsequent text lines are to be capitalized.
- OFF** Terminates capitalization mode if it was ON or if *n* has been specified and has not been exhausted.
- line* The line to be capitalized.

**Initial Setting:** OFF

**Default:** 1

### Notes

- This is a type 1 control word.
- The line form of the .UP control word starts the page.
- Capitalization is included in the active environment.

### Remarks

1. Use the .UP control word whenever you have a line of data that is to be formatted in capital letters. If your entire document is to be in capital letters, use the UPCASE option of the SCRIPT/VS command line.
2. The use of .UP *line* while a .UP *n* is still in effect resets to zero the number of lines to be capitalized.
3. The .UP control word does not cause an automatic break. Single words in a sentence can be capitalized.
4. Another method of capitalizing a single word is to use the uppercase attribute symbol &U' that is recognized by the symbol processor.
5. By default, capitalization is performed by translating a–z to A–Z. This translation can be extended with the .TU [Translate Uppercase] control word for languages other than English.
6. If a .UP control word is active when the table of contents is formatted, the lines in the table of contents are capitalized accordingly.

## **Examples**

- Capitalizing a single word

This sentence has  
.up one  
capitalized word.

results in

This sentence has ONE capitalized word.

- Capitalizing a single word using the symbol processor's uppercase attribute

This sentence  
has &U'one capitalized word.

results in

This sentence has ONE capitalized word.

---

## .US [Underscore]

### Function

The .US [Underscore] control word automatically underscores one or more input lines.

### Syntax



### Parameters

***n*** Specifies the number of input lines to be underscored. If *n* is omitted, 1 is assumed. If .US *n* is specified when .US ON is in effect, underscoring is turned off when *n* lines have been underscored, or when a .US OFF is encountered, whichever occurs first.

**ON** Specifies that subsequent text lines are to be underscored.

**OFF** Terminates underscoring if it was ON, or if *n* has been specified and has not been exhausted.

***line*** A single text line to be underscored.

**Initial Setting:** OFF

**Default:** 1

### Notes

- This is a type 1 control word.
- The line form of .US starts the page.
- Underscoring is included in the active environment.

### Remarks

1. The .US control word does not cause an automatic break; single words in a sentence can be underscored.
2. The use of .US line while a .US *n* is still in effect resets to zero the number of lines to be underscored.
3. The .UD [Underscore Definition] control word controls whether blank characters (X'40', X'41') and spaces generated by the .IS control word are underscored by the .US control word. It also allows you to specify, for page printers, what *named* rule is used for underscoring and where the rule is placed relative to the baseline.
4. If a .US control word is active when the table of contents is formatted, the lines in the table of contents are underscored accordingly.
5. Underscores are not affected by the .SB [Shift Baseline] control word. Use the .UD [Underscore Definition] control word to specify the vertical position of underscores relative to the baseline.

## Examples

Underscoring a single word

This sentence has  
.us one  
underscored word.

results in

This sentence has one underscored word.

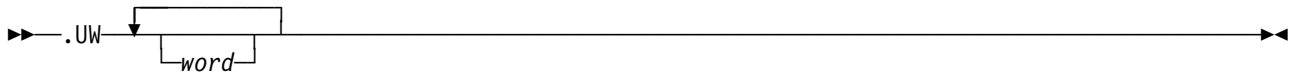
---

## **.UW [Unverified Word]**

### **Function**

The .UW [Unverified Word] control word is generated by SCRIPT/VS and processed whenever unverified and potentially misspelled words are found when the SPELLCHK option of the SCRIPT command is specified and spelling verification has not been turned off with .SV OFF.

### **Syntax**



### **Parameters**

*word*     An unverified word found in an input line.

### **Remarks**

1. Whenever misspelled words are found in an input line, the .UW control word is executed with the misspelled words as parameters. This control word issues an error message to tell you that those words were not verified.
2. If you want to perform some function other than a message when misspelled words are encountered, you can define a .UW macro. When macro substitution is on, your .UW macro is executed whenever misspelled words are found.
3. After the .UW control word or macro is processed, the misspelled words are still on the line and are processed as part of that text input line. In other words, you cannot use the .UW macro to correct or remove such words from a line.

### **Examples**

When unverified words are found, you might want to add them to an addenda dictionary, using the .DU [Dictionary Update] control word, so that only the first occurrence is detected, or write the words to a file using the .WF [Write to File] control word so that you can review the file created for possible permanent addition to your dictionary. The following .UW macro does both:

```
.dm uw on
.mg //Unverified Words: &*
.du add &*
.wf .du add &*
.dm off
.ms on
```

After you have formatted a document containing this macro with the SPELLCHK option of the SCRIPT command, the DSMUTWTF file contains a list of all unverified words, prefixed with ".du add." This file must be edited to remove any truly misspelled words and can then be renamed and imbedded the next time the document is formatted, to create an addenda dictionary.

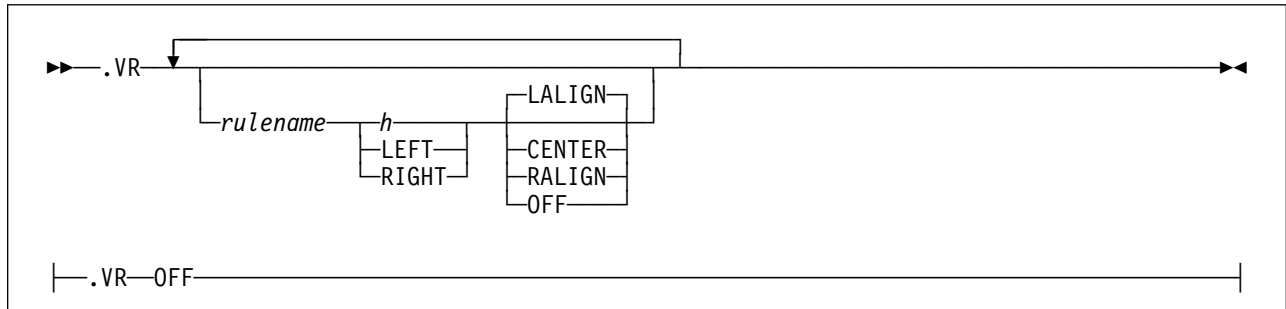


## **.VR [Vertical Rule]**

## Function

Use the `.VR` [Vertical Rule] control word to start or end vertical rules in the current column.

## Syntax



## Parameters

*rulename* Any *named* rule defined with the .DR [Define Rule] control word. If no name is given, the vertical rule is drawn with the same rule as was used for the previous vertical rule on this .VR control word. If you specify *rulename* you must also specify at least one set of vertical displacements.

*h* The nominal horizontal position of the vertical rule, specified as any valid space unit.

**LEFT** Indicates that the nominal position of the vertical rule is flush left.

**RIGHT** Indicates that the nominal position of the rule is flush right.

**LALIGN** Specifies that the *left edge* of the vertical rule is to be aligned at the nominal horizontal position. LALIGN is ignored for line devices. This is the default for page printers.

**CENTER** Specifies that the *center* of the vertical rule is to be aligned at the nominal horizontal position. CENTER is ignored for line devices.

**RALIGN** Specifies that the *right edge* of the vertical rule is to be aligned at the nominal horizontal position. RALIGN is ignored for line devices.

|            |                                                                                                                      |
|------------|----------------------------------------------------------------------------------------------------------------------|
| <b>OFF</b> | Specifies that a rule at the designated horizontal position is to end. If OFF is specified, any rulename is ignored. |
|------------|----------------------------------------------------------------------------------------------------------------------|

If OFF is the only parameter specified, all vertical are ended.

## Notes

- .VR causes a break.
- .VR ensures that the page is started.

## Remarks

1. Several separate vertical rules can be drawn with a single .VR control word. For each rule, you can specify three things:
  - The *name* of the rule to be used in drawing the vertical rule
  - The nominal horizontal position of the rule

## .VR [Vertical Rule]

- The alignment of the rule, that is, whether the left edge, the right edge, or the center of the vertical rule is aligned with the nominal position.
2. Vertical rules cannot overlap. A .VR control word can start some new vertical rules, but all vertical rules that were previously started continue, if possible. If any of the new vertical rules overlaps an old one, then the old one is ended.
  3. If a requested alignment cannot be performed, the rule is drawn as close to the requested position as possible. For example, if you requested that a very thick rule be centered on a position that was closer to the left margin than half the thickness of the rule, this request could not be satisfied without placing part of the rule to the left of the left margin. In this case, the left edge of the rule would be positioned flush left.

Other examples of impossible alignment requests are:

- A request to right align a rule at the left margin
  - A request to left align a rule at the right margin.
4. All horizontal positions on a single .VR control word, whether stated or implied, must be sequentially increasing and cannot overlap. In other words, you must specify all rules from left to right.
  5. For the 3800 Printing Subsystem Model 1 and the 3800 Printing Subsystem Model 3 (in compatibility mode), several restrictions are imposed on the use of the .VR control word:
    - Only monospace fonts can be used with vertical rules, and all fonts used must be of the same pitch. All of the 3800 fonts distributed with SCRIPT/VS, listed in the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*, are monospace with the exception of GP12.
    - When a vertical rule is overlaid on a text character, the rule replaces the character.
    - When a vertical rule is overlaid on horizontal white space in another column, the text might be misaligned.
  6. If rulename is not specified, the rule is drawn with a default rule appropriate for the logical device:
    - For line devices, the rule is constructed using the box character set of the current font.
    - For page printers, the rule is drawn with the default rule, as described under “.DR [Define Rule]” on page 151.
  7. Whether they are created with .VR or with the .BX [Box] control word, Vertical rules, that cross columns or pages extend to the bottom of the section or page.
  8. If a section area contains an unended vertical rule, the rule extends to the bottom of the whole section regardless of any depth specification in the specific section area. For page and body areas, the rule extends only to the bottom of the area.
  9. The output from DCF when formatting for a PostScript or 4028 physical device is a resolution-independent data stream. Therefore, when you format for one of these devices, there may be one pel rounding errors at the printer that are evident in rule widths and rule intersections. Because of the variations in printer resolutions, the same SCRIPT/VS output file might produce different results on different devices.

## Examples

- The .VR control word can be used to draw a single vertical rule anywhere in the column:

```
.vr 1.75i  
.sp 3  
.vr off
```



- Several vertical rules can be drawn with a single .VR control word, and new vertical rules can be started while previous vertical rules are still going.

```
.dr thin weight .4mm  
.dr thick weight .6mm  
.vr thin 35mm thin 55mm  
.sp 2  
.vr thick 45mm  
.sp 2  
.vr off
```

results in

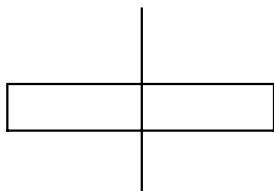


On line devices, the rulename used for new vertical rules is ignored if previous vertical rules are already going. All concurrent vertical rules must be drawn with a single font and a single box character set.

- Vertical rules can intersect with boxes drawn with the .BX [Box] control word. For example,

```
.vr 15m  
.sp 2  
.bx 5m 25m  
.sp  
.bx off  
.sp 2  
.vr off
```

results in



---

## .VT [Variable Text]

### Function

The .VT [Variable Text] control word is used to indicate the place in the formatted document where variable text is to be inserted by a user's postprocessor at a later time and the amount of physical space to save for the text. Refer to the *Document Composition Facility PostProcessor Examples* for information about using postprocessors.

### Syntax

►► .VT—*fieldname*—LENGTH *n*—  
└───┬───┘  
    WIDTH *h*

### Parameters

|                  |                                                                                                                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>fieldname</i> | The name of the field to contain the variable text. The <i>fieldname</i> is used by a user's postprocessor to associate the text to be inserted with the correct place in the AFPDS document. The name can be up to 8 characters long and is folded to uppercase by SCRIPT/VS. |
| <b>LENGTH</b>    | The number of characters to be reserved in the AFPDS output for overlaying with variable text. The maximum number of characters allowed is 230.                                                                                                                                |
| <b>WIDTH</b>     | The amount of horizontal white space to be reserved in the document for the variable text. Any valid horizontal space units are allowed. If WIDTH is not specified, the amount of space required by <i>n</i> figure spaces in the default font is used.                        |

### Remarks

1. The variable text postprocessing is intended only for page printers. If a .VT control word is encountered in a document being formatted for a line device or a PostScript device, the .VT is treated as an .IS [Inline Space] control word for the amount of the WIDTH parameter. However, a word space (subject to justification) is added before and after the space, so that the white space is actually longer than that specified on the WIDTH parameter.
2. If you want to have the variable text in a different font, a .BF control word can be used before the .VT. Remember to consider the size of the font when specifying the WIDTH parameter.
3. The FILE option should be specified on the SCRIPT command when formatting the source file containing the .VT control words, so the output is available for the postprocessor.
4. The space for the variable text must fit on one line. If not enough room is left on the current line to contain the space for the variable text, the entire space is moved to the next line, regardless of the overdraw option in effect. If the space for the variable text is the only "word" on the line and if its width exceeds the column line length, the space for the variable text extends beyond the column line length.
5. If the WIDTH value specified is greater than the column line length, an error message is issued and the .VT is ignored.
6. It is the user's responsibility to ensure that the WIDTH of the space reserved is sufficient to contain the number of characters specified with the LENGTH parameter. If it is not sufficient, printing might be overlapped or go off the page when the variable text is inserted.

7. If the document is printed without the postprocessor, blank space is left where the variable text would be inserted.

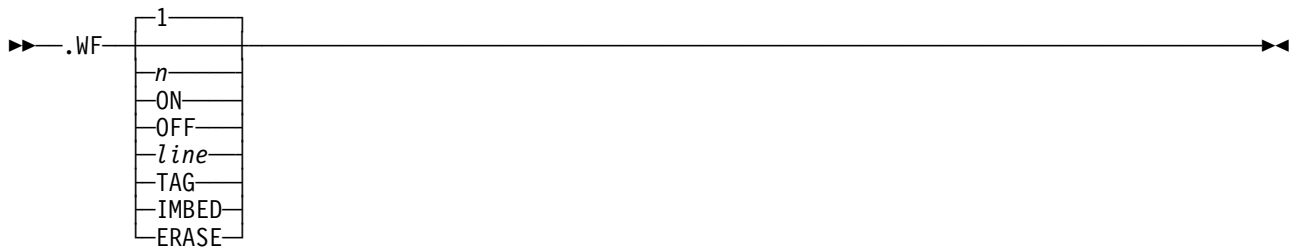
---

## .WF [Write to File]

### Function

Use the .WF [Write to File] control word to cause lines of text or control words to be written to the output file DSMUTWTF.

### Syntax



### Parameters

- n* Specifies a number of input lines to be written into the DSMUTWTF file. If *n* is omitted, 1 is assumed.
- ON** Specifies that the following text and control words are to be written into the DSMUTWTF file until .WF OFF is encountered.
- OFF** Stops writing text and control words to the DSMUTWTF file, if ON was specified or if *n* has been specified and has not yet been exhausted. If you specify .WF OFF, the following rules apply:
- It must be on a line by itself.
  - There must be a blank between the control word and the parameter.
  - A control word modifier cannot be used on this line.
- line* A line of text or control words to be written to the file.
- TAG** Specifies that the following text and control words are to be written into the DSMUTWTF file until the GML end tag that matches the start tag that activated the .WF TAG is encountered.
- IMBED** Causes the DSMUTWTF file to be imbedded.
- ERASE** Causes the DSMUTWTF file to be erased.

### Remarks

1. All the text and control words between the .WF ON and .WF OFF control words are written into the DSMUTWTF file without any processing other than input translation, symbol substitution, and tag processing. .WF control words are not written to the file. Any .WF other than .WF OFF is ignored when .WF is writing lines to the file.
2. If the DSMUTWTF file contains GML tags, you should precede the .WF ON with .GS TAG OFF and follow the .WF OFF with .GS TAG ON. If you are using the TAG form of .WF, you do not have to turn off GML tag processing because it is done automatically.
3. If symbol substitution is ON, the lines that are written to the file are substituted. If substitution is ON when the file is imbedded, the lines are substituted again if any unresolved symbols remained after the first substitution.

4. The file-id DSMUTWTF can be associated with different file or data set names using the .DD [Define Date File-id] control word. See the descriptions of the .DD control word for more information. Different groups of information can be written to different actual files when .DD is used.
5. The .WF control word cannot write into a file that is currently in use for .AP [Append] or .IM [Imbed]. If an imbedded or appended file is ended with the .EF [End of File] control word, it is still “in use” unless the CLOSE parameter of the .EF [End of File] control word was specified.
6. The data written to a file with the .WF [Write to File] control word is added after any existing data in the file, unless DSMUTWTF has been associated with a member of a partitioned data set. In this case, the data replaces any existing member.
7. In MVS, if DSMUTWTF is reassocated using the DD parameter of the .DD [Define Date File-id] control word, .WF ERASE or IMBED behave differently, depending on the DISP option given for the associated file. If the disposition is other than MOD, the contents of the file are erased. If the disposition is MOD, the file contains any data it held before the .WF.

## Examples

- You can insert one input line into the DSMUTWTF file with

```
.wf contents of the input line
```

- You can insert a specific number of input lines into the file with

```
.wf 5
.in 3m
.ce 3
These are the
lines to go
into DSMUTWTF.
```

Input lines that are written to the file are processed for symbol substitution and GML tag processing unless these functions have been specifically inhibited.

- You can also insert a number of input lines into the file with

```
.wf on
.
.
.
Many input lines
.
.
.
.wf off
```

**Note:** The .WF OFF control word must appear on an input line by itself exactly as it is shown here.

- If you want to use .WF ON in a GML start tag APF, you can use the .WF TAG form as shown in the following example:

```
.aa tag tag etag
.gs tag on
.ms on
.dm tag on
.wf tag
.dm off
.dm etag /*
```

Then when you specify :TAG., all lines are written to DSMUTWTF until the end tag (:ETAG.) is encountered.

## **.WF [Write to File]**

The write-to-file request is automatically ended before the end APF is invoked. GML scanning is off during this form of write-to-file processing until the end tag is found.

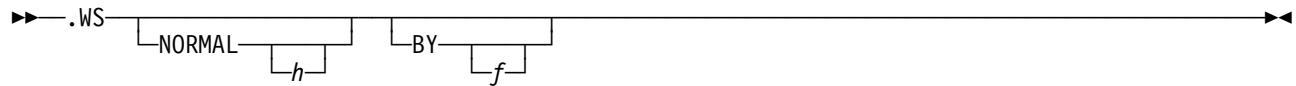


## .WS [Word Space]

### Function

Use the .WS [Word Space] control word to control the width of word spaces.

### Syntax



### Parameters

**NORMAL** Establishes the width of word spaces until changed by another .WS control word.

*h* Gives the width of the word spaces and overrides the default width, which is equal to the word space width of the current font.

If *h* is omitted, the width of subsequent word spaces is determined by the default word-space value of the current font.

**BY** Increases or decreases the width of word spaces.

*f* Is a factor by which the normal width of word spaces is multiplied to obtain the actual width. (The normal width is the default word-space width of the current font, unless it is overridden with .WS NORMAL.)

If *f* is omitted, “1.0” is used. The maximum value of *f* is 100.0.

**Initial Setting:** Dependent upon the default word-space value of the current font.

**Default:** Restores the initial setting.

**Note:** The word-space values are included in the active environment.

### Remarks

1. The width of blanks between words is determined by the .WS control word. When consecutive blanks are found between words, the first blank is considered to be a word space; the rest are considered to be extra spaces. The width of the first blank is determined by .WS. The width of each blank after the first is determined by the .ES [Extra Space] control word.
2. Each font is designed with a default word space value appropriate to the size of the characters, and SCRIPT/VS uses this as the initial width of blanks. The BY parameter of .WS can be used to increase or decrease the width of word spaces proportionally for all fonts.
3. The NORMAL parameter can be used to override the default width that comes from the current font. This establishes fixed word spaces, regardless of any font size change.
4. The actual width of word spaces is always determined by multiplying the normal width by the factor. When neither of these has been changed with .WS, the result is 1.0 times the word-space width of the current font.
5. The width of word spaces determined by the .WS control word can be changed as follows:
  - Whether hyphenation is on or off, word spaces can be compressed or expanded, within the range specified with the .HY [Hyphenate] control word, to avoid breaking a word.
  - When justification has been turned on with the .FO [Format Mode] control word, word spaces can be expanded to align the last word with the right margin.

6. Dimensions are always rounded to the nearest integral device unit. Therefore, the width of word spaces can be set or rounded to zero. For example,  
`.ws by .33`  
decreases word spaces to one-third their former width when printed on a page printer, but sets word spaces to zero on a line device.
7. For PostScript devices, the Adobe Font Metrics information does not contain em-space, figure space, or interword space values. These values are calculated for proportional fonts as follows:
  - a. Calculate the em-space for the font.
    - 1) If the AFM file contains the *emdash* character, use its width for the width of the em-space.
    - 2) If the AFM file doesn't contain the *emdash* character, use two times the width of the *endash* character if there is one.
    - 3) If neither the *emdash* nor the *endash* character is found in the AFM file, use four times the width of the *space* character if there is one.
    - 4) If none of the above applies, use the width of the point size.
  - b. Set the interword space to one-third of the calculated em-space.
  - c. Set the figure space to one-half of the calculated em-space.

## Examples

- The .WS control word can be used to increase or decrease the width of word spaces. For example,

```
.fo off
.ws by .8
The Walrus and The Carpenter
.ws by 1.0
The Walrus and The Carpenter
.ws by 1.2
The Walrus and The Carpenter
```

are formatted like this for the 3800 Printing Subsystem Model 3 and the 3820 Page Printer:

```
The Walrus and The Carpenter
The Walrus and The Carpenter
The Walrus and The Carpenter
```

On monospaced devices, such as the 1403, all of these word spaces are rounded to the same value:

```
The Walrus and The Carpenter
The Walrus and The Carpenter
The Walrus and The Carpenter
```

- Word spaces can be set to a fixed width. For example,

```
.ws normal .5i
And the Oysters.
```

results in

```
And      the      Oysters.
```

- In the preceding example, the width of extra spaces was fixed at one-half inch because the normal width was set to .5i, and the BY factor was 1.0 by default. You would obtain the same result with

```
.ws norm 0.25i by 2.0
```

Here, the fixed normal width (.25 inch) is multiplied by 2.0 to give a fixed nominal width of one-half inch.

- To restore the width of word spaces to the default for the font, enter  
.WS

---

## .WZ [Widow Zone]

### Function

The .WZ [Widow Zone] control word controls the handling of single-line widows in the body text.

### Syntax



### Parameters

**ON** Specifies that widows are to be suppressed. This is the default and the initial setting.

**OFF** Specifies that widows are not to be suppressed.

**Initial Setting:** ON

**Default:** ON

### Notes

- .WZ causes a break.

### Remarks

1. Widow-zone processing is not performed if the depth of the body of the page is less than eight lines.
2. All controls that cause a break effectively end any widow zone, and the next line of text starts a new widow zone. This can result in what appears to be a single-line widow but which is actually a single-line paragraph.
3. Whenever a line is encountered whose depth is more than one-third the depth of the page body, the current widow is ended after the line is placed into it.
4. Widow-zone processing is performed only in the body of a document.
5. If you have less than 8 lines of output in multiple columns, turn off widow zone processing (.wz off). Otherwise, the columns are not balanced correctly.

---

## Chapter 5. A Summary of SCRIPT/VS

|                                                |     |
|------------------------------------------------|-----|
| <b>Chapter 5. A Summary of SCRIPT/VS</b> ..... | 437 |
| The SCRIPT/VS Formatting Environment .....     | 451 |



---

## Chapter 5. A Summary of SCRIPT/VS

The following table lists the figures and tables in this chapter with a brief description.

| Figure or Table                                    | Description                                                                                                                                                             |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Table 11 on page 438                               | List of the utility files that SCRIPT/VS creates or uses                                                                                                                |
| Figure 6 on page 438                               | Picture of the SCRIPT/VS page layout.                                                                                                                                   |
| Table 12 on page 439                               | Summary of the options of the SCRIPT command                                                                                                                            |
| Table 13 on page 441                               | List of the control words that always result in a break between input lines of text                                                                                     |
| Table 14 on page 441                               | List of the control words that always take effect on the next output page                                                                                               |
| Table 15 on page 442                               | List of the control words that are not allowed within a keep, float, footnote, named area, or table                                                                     |
| Table 16 on page 442                               | List of control words that start a page                                                                                                                                 |
| Tables 18 through 24 on pages 443 through 449.     | Summary of SCRIPT/VS system symbols                                                                                                                                     |
| Table 24 on page 450                               | List of the attributes of a symbol                                                                                                                                      |
| "The SCRIPT/VS Formatting Environment" on page 451 | List of parameters contained within the formatting environment                                                                                                          |
| Table 27 on page 452                               | List of parameters contained within the formatting environment that are carried over from the active environment to the running heading and running footing environment |
| Table 28 on page 453                               | List of output file formats                                                                                                                                             |
| Figure 7 on page 455                               | IBM 1403 Printer's TN print train character set                                                                                                                         |
| Table 29 on page 455                               | List of the fonts provided with SCRIPT/VS for use with the IBM 3800 Printing Subsystem                                                                                  |
| Table 31 on page 456                               | List of the fonts provided with the IBM 3800 Printing Subsystem                                                                                                         |
| Table 32 on page 457                               | List of the fonts provided with the IBM 3800 Printing Subsystem that are available for the 6670 for compatibility reasons                                               |

*Table 10. Listings and Summaries for SCRIPT/VS*

| SCRIPT/VS Utility Files |                        |              |                |
|-------------------------|------------------------|--------------|----------------|
| File-id                 | Description            | Control Word | Option         |
| DSMTERMI                | Terminal input file    | .TE, .RV     |                |
| DSMTERMO                | Terminal output file   | .TY          | TERM           |
| DSMUTCTF                | STAIRS/VS CTF file     | .SO          | CTF            |
| DSMUTMSG                | Error messages file    | .MG          | MESSAGE(DELAY) |
| DSMUTTOC                | Table of Contents file | .TC, .PT     |                |
| DSMUTWTF                | Write to File file     | .WF          |                |

Table 11. File Identifiers of SCRIPT/VS Utility Files. SCRIPT/VS uses or creates these files as a result of the control words or options indicated. Any of these files can be redefined with the .DD [Define Date File-id] control word, subject to certain restrictions, depending on whether the NODDUT or DDUT SCRIPT command option is in effect. Messages produced with the .MG [Message] control word are written to DSMTERMO if the MESSAGE (DELAY) option of the SCRIPT command is not specified.

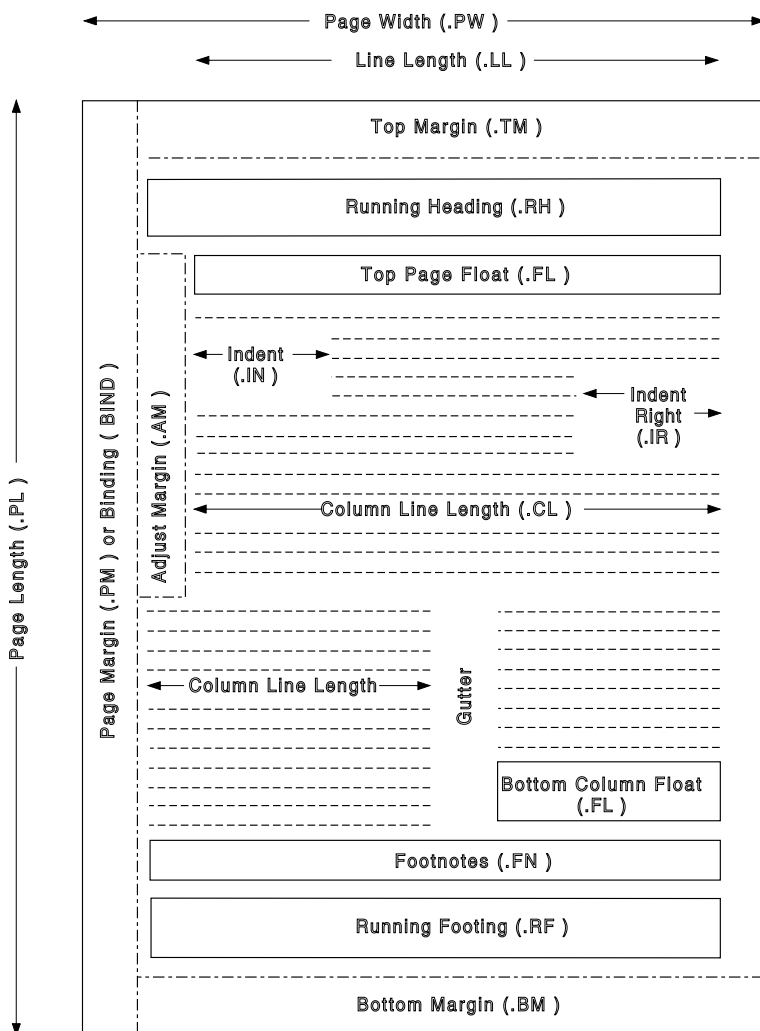


Figure 6. SCRIPT/VS Terms for Parts of the Page



| Option          | Parameters                   | Description                                                                                                                                                                                                                  |
|-----------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>BIND</b>     | (bind) (obind ebind)         | Shifts the page to the right.<br><b>SCRIPT/VS System Symbol:</b> &\$BO, &\$BE                                                                                                                                                |
| <b>CHARS</b>    | (font1 ... font4)            | Specifies up to 32 fonts. <i>font1</i> is used as the default font for the device. For the 3800-1, only the first four fonts specified with CHARS can be used in the document.<br><b>SCRIPT/VS System Symbol:</b> &\$CHAR( ) |
| <b>CONTINUE</b> |                              | Continues processing after a nonsevere error occurs. A nonsevere error means a return code less than or equal to 8.                                                                                                          |
| <b>CTF</b>      |                              | Prepares output in STAIRS/VS Condensed Text Format.                                                                                                                                                                          |
| <b>DDUT</b>     |                              | Allows the redefinition of a SCRIPT/VS utility file to a nonutility file.<br><b>SCRIPT/VS System Symbol:</b> &\$DDUT                                                                                                         |
| <b>DEST</b>     | (station-id)                 | Specifies a remote output station. (Valid only in TSO.)                                                                                                                                                                      |
| <b>DEVICE</b>   | (devtype)                    | Specifies a logical output device. See "SCRIPT/VS Logical Devices" on page 27 for a table of logical devices.                                                                                                                |
| <b>FILE</b>     | [(fileid)]                   | Specifies a disk file for output. (Not valid in ATMS-III.)                                                                                                                                                                   |
| <b>FONTLIB</b>  | font library name            | Specifies a font library.                                                                                                                                                                                                    |
| <b>FPASSES</b>  | (n)                          | Specifies the number of formatting passes to be applied to the document.<br><b>SCRIPT/VS System Symbol:</b> &\$PSNO, &\$PASS                                                                                                 |
| <b>INDEX</b>    |                              | Enables the .PI [Put Index] control word.<br><b>SCRIPT/VS System Symbol:</b> &\$INDEX                                                                                                                                        |
| <b>LIB</b>      | (libname ...) (opnum ...)    | Specifies symbol and macro libraries. (Only one in TSO, although multiple libraries can be concatenated by preallocating a ddname of SCRPTLIB; up to eight in CMS and ATMS-III.) Specify operator number in ATMS-III.        |
| <b>MESSAGE</b>  | [(DELAY]<br>[ID]<br>[TRACE]) | Controls message printing.                                                                                                                                                                                                   |
| <b>NOCONT</b>   |                              | Terminates processing after any error occurs with a return code less than or equal to 8.                                                                                                                                     |
| <b>NODDUT</b>   |                              | Disallows the redefinition of a SCRIPT/VS utility file with the .DD control word.<br><b>SCRIPT/VS System Symbol:</b> &\$DDUT                                                                                                 |
| <b>NOPROF</b>   |                              | Suppresses the profile processing.                                                                                                                                                                                           |
| <b>NOSEGLIB</b> |                              | Suppresses the segment library processing.                                                                                                                                                                                   |
| <b>NOSORCDD</b> |                              | Specifies that the primary input file is a data set name in TSO.                                                                                                                                                             |
| <b>NOSPIE</b>   |                              | Prevents entering SPIE exit routines. (Valid only in CMS and TSO.)                                                                                                                                                           |
| <b>NOWAIT</b>   |                              | Prevents prompting for paper adjustment. (Valid only for typewriter terminals in CMS and TSO.)                                                                                                                               |
| <b>NUMBER</b>   |                              | Prints file name and line number. Ignored for page printers and PostScript devices. (Suboptions are valid only in TSO.)                                                                                                      |
| <b>OPTIONS</b>  | [(fileid)]                   | Specifies a file that contains SCRIPT options. (Valid only in CMS and ATMS-III.)                                                                                                                                             |

| Option              | Parameters                                                                   | Description                                                                                                                                                      |
|---------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PAGE</b>         | [[ <u>PROMPT</u> ] [[FROM] p<br>[TO] q] [[FROM] p FOR n]<br>[[FROM] p ONLY]] | Prints pages selectively. (PROMPT is valid only in CMS and TSO.)                                                                                                 |
| <b>PRINT</b>        | [(copies,class, fcb,ucs)]                                                    | Sends output directly to the system SPOOL. Ignored for 4250 printers and PostScript devices. (Suboptions are valid only in TSO.)                                 |
| <b>PROFILE</b>      | [(fileid)]                                                                   | Specifies a profile. (A file to be imbedded before the primary input file is processed.)                                                                         |
| <b>PSOUT</b>        | ( <u>ASCII</u> )(EBCDIC)                                                     | Specifies whether SCRIPT/VS should generate PostScript output in ASCII or EBCDIC. Only ASCII PostScript is printable.<br><b>SCRIPT/VS System Symbol:</b> &\$ASCI |
| <b>QUIET</b>        |                                                                              | Suppresses formatter identifier message.                                                                                                                         |
| <b>SEARCH</b>       | (libname) (opnum ...)                                                        | Specifies a library in TSO. Specifies the default filetype for imbed and append files in CMS. Specifies operator number in ATMS-III.                             |
| <b>SEGLIB</b>       | [(seglib)]                                                                   | Specifies the page segment library for AFP and 4250 devices.                                                                                                     |
| <b>SEPMASR</b>      | [(0 ... 16)]                                                                 | Specifies which separation masters should be produced. The default is all masters if no suboptions are given.<br><b>SCRIPT/VS System Symbol:</b> &\$SEPM         |
| <b>SORCDD</b>       |                                                                              | Specifies that the primary input file is a ddname in TSO.                                                                                                        |
| <b>SPELLCHK</b>     |                                                                              | Enables the .SV [Spelling Verification] control word.                                                                                                            |
| <b>STOP</b>         |                                                                              | Prints separate pages at the terminal. (Valid only for typewriter terminals in CMS and TSO.)                                                                     |
| <b>SYOFF</b>        |                                                                              | Disables the .SY [System Command] control word.                                                                                                                  |
| <b>SYON</b>         |                                                                              | Enables the .SY [System Command] control word.                                                                                                                   |
| <b>SYSVAR</b>       | (x value ...)                                                                | Sets symbol values for &SYSVARx to pass information to the profile.                                                                                              |
| <b>TERM</b>         |                                                                              | Displays the output at a user's terminal.                                                                                                                        |
| <b>TLIB</b>         | (libname ... )                                                               | Specifies spelling verification and hyphenation libraries. (Valid only in CMS.)                                                                                  |
| <b>TWOPASS</b>      |                                                                              | Prepares with two formatting passes and produces output on the second pass.<br><b>SCRIPT/VS System Symbol:</b> &\$TWO, &\$PASS, &\$PSNO                          |
| <b>UNFORMAT</b>     |                                                                              | Prints all input lines without formatting. Ignored for page printers and PostScript devices.<br><b>SCRIPT/VS System Symbol:</b> &\$UNF                           |
| <b>UPCASE</b>       |                                                                              | Folds lowercase letters to uppercase before processing.                                                                                                          |
| <b>@user-option</b> | [(sub-options...)]                                                           | User-defined options, which must begin with the character "@". (Valid only in CMS.)                                                                              |

Table 12. Summary of SCRIPT Options

|                              |                          |
|------------------------------|--------------------------|
| .AM [Adjust Margin]          | .LB [Leading Blank]      |
| .AR [Area]                   | .LL [Line Length]        |
| .BL [Blank]                  | .LO [Lead-Out]           |
| .BR [Break]                  | .LS [Line Spacing] *     |
| .BX [Box]                    | .LT [Leading Tab]        |
| .CB [Column Begin]           | .MC [Multicolumn Mode]   |
| .CC [Conditional Column]     | .NF [No Format]          |
| .CD [Column Definition]      | .OF [Offset]             |
| .CE [Center]                 | .OI [Overlay Include] *  |
| .CG [Copy Group]             | .PA [Page Eject]         |
| .CL [Column Line Length]     | .PM [Page Margins]       |
| .CP [Conditional Page Eject] | .PO [PostScript] *       |
| .FO [Format Mode]            | .RD [Read Terminal]      |
| .HR [Horizontal Rule]        | .RF [Running Footing]    |
| .H1 [Head Level 1]           | .RH [Running Heading]    |
| .H2 [Head Level 2]           | .RI [Right Adjust]       |
| .H3 [Head Level 3]           | .SC [Single Column Mode] |
| .H4 [Head Level 4]           | .SK [Skip]               |
| .H5 [Head Level 5]           | .SO [STAIRS/VS Output]   |
| .H6 [Head Level 6]           | .SI [Segment Include] *  |
| .IE [Index Entry]            | .SP [Space]              |
| .IL [Indent Line]            | .SX [Split Text]         |
| .IN [Indent] *               | .TC [Table of Contents]  |
| .IR [Indent Right] *         | .TA [Table]              |
| .IX [Index]                  | .UN [Undent]             |
| .JU [Justify Mode]           | .VR [Vertical Rule]      |
|                              | .WZ [Widow Zone]         |

**Note:**

.RE [Restore Environment] may cause a break also.

\* This control word does not cause a break if NOBREAK is specified.

*Table 13. Control Words That Cause a Break. When concatenation is on (see the .FO [Format Mode] control word), words from input lines are rearranged on output lines to make each column line as full as possible. This process is inhibited for the current line if SCRIPT/VS encounters any of these control words.*

|                       |                           |
|-----------------------|---------------------------|
| .BM [Bottom Margin]   | .PN [Page Numbering Mode] |
| .FN [Footnote] leader | .PW [Page Width]          |
| .H1 [Head Level 1]    | .RF [Running Footing]     |
| .LL [Line Length]     | .RH [Running Heading]     |
| .PL [Page Length]     | .TM [Top Margin]          |
| .PM [Page Margins]    |                           |

*Table 14. Control Words That Take Effect On the Next Page. If no data has yet been placed on the first page of the document, or if the previous page was ended with a .PA NOSTART control word, the first, or next, page has not yet been started, and these control words can take effect on this page.*

|                              |                           |
|------------------------------|---------------------------|
| .AM [Adjust Margin]          | .KP [Keep]                |
| .AR [Area]                   | .LL [Line Length]         |
| .BM [Bottom Margin]          | .MC [Multicolumn Mode]    |
| .CB [Column Begin]           | .PA [Page Eject]          |
| .CC [Conditional Column]     | .PL [Page Length]         |
| .CD [Column Definition]      | .PM [Page Margins]        |
| .CG [Copy Group]             | .PN [Page Numbering Mode] |
| .CP [Conditional Page Eject] | .PW [Page Width]          |
| .DA [Define Area]            | .RF [Running Footing]     |
| .EF [End of File]            | .RH [Running Heading]     |
| .FL [Float]                  | .RN [Reference Numbers]   |
| .FN [Footnote]               | .SC [Single Column Mode]  |
| .FV [Format Vertically]      | .TC [Table of Contents]   |
| .GR [Group]                  | .TM [Top Margin]          |
| .IX [Index]                  |                           |

**Note:** .RF and .RH are disallowed in keeps, floats, footnotes, *named* areas, and tables. The .Hn (head level) control words extend vertical keeps (.KP v and .KP v+v), and they are disallowed in an area or float if they cause a page eject (.PA) or section break (SECT). No message is issued when a v or v + v type keep is ended. If one of the above control words is found within a footnote leader, the control word is ignored.

*Table 15. Control Words That End a Keep, Float, Footnote, Named Area, or Table. If any of these control words is found within a keep, float, footnote, named area, or table, a message is issued and the keep, float, footnote, named area, or table is terminated before the control word is processed.*

|                              |                            |
|------------------------------|----------------------------|
| .AR [Area]                   | .IS [Inline Space]         |
| .BL [Blank]                  | .IX [Index]                |
| .BX [Box]                    | .KP [Keep]                 |
| .CB [Column Begin]           | .NV [Navigate]             |
| .CC [Conditional Column]     | .OI [Overlay Include]      |
| .CP [Conditional Page Eject] | .PO [PostScript]           |
| .CT [Continued Text]         | .RD [Read Terminal]        |
| .DO [Define Object]          | .SI [Segment Include]      |
| .FL [Float]                  | .SK [Skip]                 |
| .FN [Footnote]               | .SO [STAIRS/VS Output] DOC |
| .HR [Horizontal Rule]        | .SP [Space]                |
| .HW [Hyphenate Word]         | .SX [Split Text]           |
| .H0 [Head Level 0]           | .TA [Table]                |
| .IE [Index Entry]            | .TC [Table of Contents]    |
| .IO [Include Object]         | .VR [Vertical Rule]        |
|                              | .VT [Variable Text]        |

*Table 16. Control Words That Ensure the Page Is Started. If the page is not already started, either because no text has yet been formatted for page one, or because the previous page was ended with .PA NOSTART, any of these control words cause the page to be started.*

| Date and Time <sup>1</sup>                                                                                                                                      |                             |                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|---------------------|
| Symbol                                                                                                                                                          | Description                 | Value               |
| &\$YEAR                                                                                                                                                         | Current year (4-digit year) | 1990–2090           |
| &SYSYEAR                                                                                                                                                        | Year of the century         | 00–99               |
| &SYSMONTH                                                                                                                                                       | Month of the year           | 01–12               |
| &SYSDAYOFM                                                                                                                                                      | Day of the month            | 01–31               |
| &SYSDAYOFW                                                                                                                                                      | Day of the week             | 1–7 (“1” is Sunday) |
| &SYSDAYOFY                                                                                                                                                      | Day of the year             | 001–366             |
| &SYSHOUR                                                                                                                                                        | Hour of the day             | 00–23               |
| &SYSMINUTE                                                                                                                                                      | Minute of the hour          | 00–59               |
| &SYSSECOND                                                                                                                                                      | Second of the minute        | 00–59               |
| <b>Note:</b><br><sup>1</sup> These symbols can contain leading zeros. They can be eliminated with a .SE [Set Symbol] control word:<br>“SYSHOUR = &SYSHOUR + 0.” |                             |                     |

Table 17. SCRIPT/VS System Symbol Names — Date and Time

| Output Device Characteristics                                                                                                                                                                                                                                                                                                               |                                    |                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------|
| Symbol                                                                                                                                                                                                                                                                                                                                      | Description                        | Value                                                                                                   |
| &\$LDEV                                                                                                                                                                                                                                                                                                                                     | Logical output device <sup>1</sup> | 1–8 characters                                                                                          |
| &\$OUT                                                                                                                                                                                                                                                                                                                                      | Output destination                 | TERM, PRINT, FILE                                                                                       |
| &\$PDEV                                                                                                                                                                                                                                                                                                                                     | Physical output device             | 1403, 2741, 3270, 3800, 38PP, 3820, <sup>2</sup> 4250, 4224, <sup>3</sup> 4028, <sup>4</sup> STAIRS, PS |
| <b>Note:</b><br><sup>1</sup> Set by the DEVICE option of the SCRIPT command.<br><sup>2</sup> DEVICE (PG1xxxx) and DEVICE (PG2xxxx) results in a physical output device of 3820.<br><sup>3</sup> DEVICE (PG3xxxx) results in a physical output device of 4224.<br><sup>4</sup> DEVICE (PG4xxxx) results in a physical output device of 4028. |                                    |                                                                                                         |

Table 18. SCRIPT/VS System Symbol Names — Output Device Characteristics

| SCRIPT Command Options                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                        |                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Symbol                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Description                                                                                            | Value                                                                       |
| &\$ASCI                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | ASCII PostScript output requested <sup>4</sup>                                                         | 0, 1                                                                        |
| &\$BE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Even Bind <sup>1 2</sup>                                                                               | 0–nnn                                                                       |
| &\$BO                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Odd Bind <sup>1 2</sup>                                                                                | 0–nnn                                                                       |
| &\$CHAR(n)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Fonts <sup>3</sup>                                                                                     | 1–8 characters                                                              |
| &\$DCF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Current SCRIPT/VS release level                                                                        | n.n.n                                                                       |
| &\$DDUT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Utility file redefinition <sup>4</sup>                                                                 | 0, 1                                                                        |
| &\$EQ                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | SMFF release level if SMFF is installed; Null if not.                                                  | n.n.n                                                                       |
| &\$INDX                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Indexing <sup>4</sup>                                                                                  | 0, 1                                                                        |
| &\$LIB                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Macro library available <sup>4</sup>                                                                   | 0, 1                                                                        |
| &\$PARM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Command options <sup>5</sup>                                                                           | 8–256 characters                                                            |
| &\$PASS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Current pass number                                                                                    | 1–255                                                                       |
| &\$PRT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Current page print switch                                                                              | ON, OFF                                                                     |
| &\$PSNO                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Number of formatting passes requested with FPASSES(n) in the PARM                                      | 1–255                                                                       |
| &\$SEPM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Returns a string representing the masters that were requested with the SEPMASTR SCRIPT command option. | A string of 17 0's and 1*csq.s, where 1 indicates the master was requested. |
| &\$SYS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Environment                                                                                            | CMS, TSO, VS2, VSE, CICS, VM/XA                                             |
| &\$TWO                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | TWOPASS specified                                                                                      | 1 if TWOPASS or FPASSES(2) was requested; otherwise, 0                      |
| &\$UNF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Unformatted output <sup>4</sup>                                                                        | 0, 1                                                                        |
| &\$YEAR                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Current year (4-digit form)                                                                            | 1990-2090                                                                   |
| <b>Note:</b> <ol style="list-style-type: none"> <li><sup>1</sup> Set by the BIND option of the SCRIPT command and the .PM [Page Margins] and .AM [Adjust Margin] control words.</li> <li><sup>2</sup> The system symbol values are represented in figure spaces in the default font, regardless of the space units used in setting them. The maximum value depends on the logical output device. The values can be converted to device units by appending &amp;DH' to the front of the symbol names.</li> <li><sup>3</sup> Set by the CHARS option of the SCRIPT command. This is a symbol array; element 0 contains the number of fonts specified, and elements 1, 2, ... contain the names of the fonts specified.</li> <li><sup>4</sup> “1” indicates the command option was specified; “0” indicates it was not specified.</li> <li><sup>5</sup> This is the SCRIPT command options list. In CMS, the command options list is tokenized (divided into fields, each having 8 characters separated by blanks and parentheses) and truncated at 32 tokens (256 characters). In TSO, only the first 256 characters are available.</li> </ol> |                                                                                                        |                                                                             |

Table 19. SCRIPT/VS System Symbol Names — SCRIPT Command Options

| Page Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                       |       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|-------|
| Symbol                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Description                           | Value |
| &\$AM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Adjust Margin [.AM] <sup>2</sup>      | 0–nnn |
| &\$BE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Bind Even [BIND] <sup>2</sup>         | 0–nnn |
| &\$BM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Bottom Margin [.BM] <sup>1</sup>      | 0–nnn |
| &\$BO                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Bind Odd [BIND] <sup>2</sup>          | 0–nnn |
| &\$CL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Column Line Length [.CL] <sup>2</sup> | 1–nnn |
| &\$IN                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Left Indention [.IN] <sup>2</sup>     | 0–nnn |
| &\$IR                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Right Indention [.IR] <sup>2</sup>    | 0–nnn |
| &\$LC                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Internal Line Counter <sup>3</sup>    | 0–nnn |
| &\$LL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Line Length [.LL] <sup>2</sup>        | 1–nnn |
| &\$OF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Offset [.OF] <sup>2</sup>             | 0–nnn |
| &\$PL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Page Length [.PL] <sup>1</sup>        | 1–nnn |
| &\$PW                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Page Width [.PW] <sup>2</sup>         | 1–nnn |
| &\$RH                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Running Heading [.RH] <sup>1</sup>    | 0–nnn |
| &\$RF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Running Footing [.RF] <sup>1</sup>    | 0–nnn |
| &\$TM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Top Margin [.TM] <sup>1</sup>         | 0–nnn |
| <p><b>Note:</b> .</p> <p><sup>1</sup> These values are represented in line spaces, regardless of the space units used in setting them. The maximum value depends upon the logical output device. The values can be converted to device units by appending &amp;DV' to the front of the symbol names.</p> <p><sup>2</sup> The values of these symbols are represented in figure spaces in the default font regardless of the space units used in setting them. The maximum value depends upon the logical output device. The values can be converted to device units by appending &amp;DH' to the front of the symbol names.</p> <p><sup>3</sup> The value of the symbol &amp;\$LC is the number of lines remaining in the current column, excluding unplaced keeps, floats, footnotes, tables, widow zones, and partial lines. &amp;\$LC was originally intended for line mode. If used in page mode, it returns the value in linespaces in the default font. For page mode, you should use &amp;dv'&amp;\$LC.</p> |                                       |       |

Table 20. SCRIPT/VS System Symbol Names — Page Characteristics

| SCRIPT/VS Formatter Parameters |                                                |                                                                                  |
|--------------------------------|------------------------------------------------|----------------------------------------------------------------------------------|
| Symbol                         | Description                                    | Value                                                                            |
| &\$BS                          | Backspace character                            | Hexadecimal 16                                                                   |
| &\$CONT                        | Continuation character <sup>1</sup>            | 1 character                                                                      |
| &\$CW                          | Control word separator <sup>1</sup>            | (Default: “;”)                                                                   |
| &\$C256 <sup>5</sup>           | Identity vector                                | 256 characters                                                                   |
| &\$EGML                        | GML end-tag delimiter <sup>1</sup>             | (Default: “:.”)                                                                  |
| I &\$ENG                       | English algorithmic hyphenator <sup>6 7</sup>  | 0, 1                                                                             |
| &\$ENV                         | Formatting environment                         | BODY, FL, FN, KP, RF, RH, IBP, AR, FNL, TA                                       |
| &\$ENVARRAY(n)                 | Formatting environment <sup>2</sup>            | BODY, FL, FN, KP, RF, RH, IBP, AR, FNL, TA                                       |
| &\$FNAM                        | Current input file name                        | 8 characters                                                                     |
| I &\$GERM                      | German algorithmic hyphenator <sup>6</sup>     | 0, 1                                                                             |
| &\$GML                         | GML tag delimiter <sup>1</sup>                 | (Default: “:.”)                                                                  |
| &\$IXLG                        | Index sort-sequence language                   | DAN, DUTH, EAM, ECAN, EUK, FCAN, FIN, FNAT, GERM, ICE, ITAL, NOR, POR, SPAN, SWE |
| &\$KP                          | Keep in process                                | ON, OFF                                                                          |
| &\$LNUM                        | Last line number read                          | 0–nnn                                                                            |
| &\$LST                         | Line started                                   | 0, 1                                                                             |
| &\$PN                          | Page number <sup>3</sup>                       | 1–nnn                                                                            |
| &\$PS                          | Page number symbol <sup>1</sup>                | (Default: “&”)                                                                   |
| &\$RB                          | Required blank <sup>1</sup>                    | (Default: hexadecimal 41)                                                        |
| &\$RET                         | Return code from .SY <sup>4</sup>              | 0–nnn                                                                            |
| &\$SU                          | Symbol substitution enabled                    | ON, OFF                                                                          |
| &\$SVLG                        | Spelling verification and hyphenation language | DAN, DUTH, EAM, ECAN, EUK, FCAN, FIN, FNAT, GERM, ICE, ITAL, NOR, POR, SPAN, SWE |
| &\$TAB                         | Tab character                                  | Hexadecimal 05                                                                   |
| &\$TAG                         | Name of last GML tag found                     | Any valid tag name                                                               |
| &\$TAGD                        | GML delimiter of last tag                      | &\$GML, &\$EGML                                                                  |



| SCRIPT/VS Formatter Parameters                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                          |         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|---------|
| Symbol                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Description              | Value   |
| &\$VR                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Vertical rules in effect | ON, OFF |
| <b>Note:</b> <ol style="list-style-type: none"> <li>Set by the .DC [Define Character] control word.</li> <li>This symbol is an array that shows the different levels of the formatting environments.</li> <li>&amp;\$PN contains the numeric portion of the current page number. The page number as substituted can be obtained with the control word “.se x = &amp;.”</li> <li>In CMS, any possible return code value.<br/>In TSO, “0” to indicate that the command was stacked for execution after SCRIPT/VS terminates.<br/>In ATMS-III, “0” to indicate that the control word was ignored.<br/>In DLF, “-3” to indicate that .SY is not supported.</li> <li>If .IT SNAP is executed in VM/XA to a terminal, &amp;\$C256 breaks to a new line at the X'15' instead of printing the character as it does in other environments. Refer to the <i>VM/XA SP CMS Application Program Development Guide</i>, SC23-0355, for more information.</li> <li>A value of “1” indicates the alternate algorithmic hyphenator for that language is used. A value of “0” indicates the default algorithmic hyphenator for that language is used.</li> <li>In this context “English” includes English American, English Canadian, and English United Kingdom.</li> </ol> |                          |         |

Table 21. SCRIPT/VS System Symbol Names — SCRIPT/VS Formatter Parameters

| Font Information                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                  |                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|-----------------------------------------------------------------------|
| Symbol                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Description                                                      | Value                                                                 |
| &\$CODEPAGE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Current code page <sup>1</sup>                                   | 1–8 characters                                                        |
| &\$DFNAME                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Font name specified with .DF for the current font <sup>1 2</sup> | Font name                                                             |
| &\$DFKEYS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | .DF keywords for current font                                    | US, UP, STOP, OS CHAR c, OS RPT n, BOX name, FONT char, COLOR colname |
| &\$FIGSP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Figure space value of current font <sup>1 4</sup>                | In device units                                                       |
| &\$MAXA                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Maximum ascender value of current font <sup>1 3</sup>            | In device units                                                       |
| &\$MAXD                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Maximum descender value of current font <sup>1 3</sup>           | In device units                                                       |
| &\$TYPEFACE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Current typeface <sup>1</sup>                                    | 1–32 characters                                                       |
| &\$TYPEITAL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Current font is italic <sup>1</sup>                              | ITALIC or null                                                        |
| &\$TYPEOUTL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Current font is outline <sup>1</sup>                             | OUTLINE or null                                                       |
| &\$TYPEOVER                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Current font is overstruck <sup>1</sup>                          | OVERSTRUCK or null                                                    |
| &\$TYPEROT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Rotation of current font <sup>1</sup>                            | 0R, 90R, 180R, 270R                                                   |
| &\$TYPESIZE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Point size of current font <sup>1</sup>                          | 1–3276                                                                |
| &\$TYPEUNDER                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Current font is underscored <sup>1</sup>                         | UNDERSCORE or null                                                    |
| &\$TYPEWGT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Weight keyword of current font <sup>1</sup>                      | ULTRALIGHT to ULTRABOLD (9 weights) <sup>5</sup>                      |
| &\$TYPEWID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Width keyword of current font <sup>1</sup>                       | UNTRACONDENSED to ULTRAEXPANDED (9 widths) <sup>6</sup>               |
| &\$WDSP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Word space value of current font <sup>1 4</sup>                  | In device units                                                       |
| <b>Note:</b><br><sup>1</sup> Applies to page printers and PostScript devices; null for line printers.<br><sup>2</sup> As it was defined with .DF [Define Font] (see page “.DF [Define Font]” on page 130 form=pageonly).<br><sup>3</sup> So it is used correctly, DV is appended to the value.<br><sup>4</sup> So it is used correctly, DH is appended to the value.<br><sup>5</sup> MEDIUM is the default weight, see “.DF [Define Font]” on page 130 for a complete list.<br><sup>6</sup> NORMAL is the default width, see “.DF [Define Font]” on page 130 for a complete list. |                                                                  |                                                                       |

Table 22. SCRIPT/VS System Symbol Names — Font Information

| Table of Contents and Index                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                       |                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|-------------------------------------------|
| Symbol                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Description                                           | Value                                     |
| &SYSCONTENT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Name for first page of table of contents <sup>1</sup> | (Default: CONTENTS)                       |
| &SYSINDEX                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Name for first page of index <sup>2</sup>             | (Default: INDEX)                          |
| &SYSSEE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Phrase for <i>See</i> reference in index              | (Default: See)                            |
| &SYSSEEALSO                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Phrase for <i>See also</i> reference in index         | (Default: See also)                       |
| &SYSPRS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Index page range separator                            | (Default: hyphen {-})                     |
| &SYSPLS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Index page list separator                             | (Default: comma followed by a blank {, }) |
| <b>Note:</b><br><sup>1</sup> The character string to use as the heading for the table of contents. If a string is specified on the .TC [Table of Contents] control word, it is used. If a string is not specified on the .TC control word and the &SYSCONTENT symbol is not null, the &SYSCONTENT symbol is used. If a string is not specified on the .TC control word and the &SYSCONTENT symbol is null, the default phrase “CONTENTS” is used.<br><sup>2</sup> The character string to use as the heading for the index. If a string is specified on the .IX [Index] control word, it is used. If a string is not specified on the .IX control word and the &SYSINDEX symbol is not null, the &SYSINDEX symbol is used. If a string is not specified on the .IX control word and the &SYSINDEX symbol is null, the default phrase “INDEX” is used. |                                                       |                                           |

Table 23. SCRIPT/VS System Symbol Names — Table of Contents and Index

| Space Unit Symbol Attributes                                                                                                                                                                                                            |                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Attribute                                                                                                                                                                                                                               | Function                                                                                                                          |
| &a'                                                                                                                                                                                                                                     | Converts a numeric character string <sup>1</sup> to a “base-26” lowercase alphabetic “number.”                                    |
| &A'                                                                                                                                                                                                                                     | Converts a numeric character string <sup>1</sup> to a “base-26” uppercase alphabetic “number.”                                    |
| &AD'                                                                                                                                                                                                                                    | Returns the depth in line spaces of the material now in the specified <i>named</i> area <sup>2</sup> .                            |
| &DH'                                                                                                                                                                                                                                    | Converts a space designation <sup>1</sup> to an equivalent number of horizontal device units.                                     |
| &DV'                                                                                                                                                                                                                                    | Converts a space designation <sup>1</sup> to an equivalent number of vertical device units.                                       |
| &E'                                                                                                                                                                                                                                     | Verifies the existence of a symbol; the value is 1 if the symbol has been set; 0 if not.                                          |
| &L'                                                                                                                                                                                                                                     | Returns the number of characters in a character string. <sup>1</sup>                                                              |
| &r'                                                                                                                                                                                                                                     | Converts a numeric character string <sup>1</sup> to a lowercase roman numeral.                                                    |
| &R'                                                                                                                                                                                                                                     | Converts a numeric character string <sup>1</sup> to an uppercase roman numeral.                                                   |
| &SD'                                                                                                                                                                                                                                    | Returns the depth in line spaces of a named segment to be included with the .SI control word <sup>2</sup> .                       |
| &SH'                                                                                                                                                                                                                                    | Returns the horizontal measurement units of a named segment to be included with the .SI control word.                             |
| &SV'                                                                                                                                                                                                                                    | Returns the vertical measurement units of a named segment to be included with the .SI control word.                               |
| &SW'                                                                                                                                                                                                                                    | Returns the width in figure spaces in the default font of a named segment to be included with the .SI control word <sup>2</sup> . |
| &T'                                                                                                                                                                                                                                     | Yields the type of the current value of a symbol. The type is either “N” for numeric or “C” for character.                        |
| &U'                                                                                                                                                                                                                                     | Converts a lowercase character string to uppercase.                                                                               |
| &V'                                                                                                                                                                                                                                     | Yields the current value of a symbol.                                                                                             |
| &W'                                                                                                                                                                                                                                     | Yields the measured length of a character string <sup>1</sup> in figure spaces of the default font <sup>2</sup> .                 |
| &X'                                                                                                                                                                                                                                     | Converts a hexadecimal notation to a character string <sup>1</sup> .                                                              |
| <b>Note:</b><br><sup>1</sup> The character string or space designation may be the value of a symbol.<br><sup>2</sup> If you want the length of a character string in device units, use &DV' or &DH' in combination with this attribute. |                                                                                                                                   |

Table 24. Attributes of a Symbol's Value

## The SCRIPT/VS Formatting Environment

The .SA [Save Environment] and .RE [Restore Environment] control words save and restore a copy of the active and page environments along with the .TI [Translate Input] and .TR [Translate Character] translate tables. Only a copy of the active environment is saved at the beginning of a keep (except inline keep), float, running heading, running footing, footnote leader, footnote, table, and named area. These values are restored when the processing of a keep, float, and so on is completed.

The following table lists SCRIPT defaults:

| Active Environment                                                                                                                                                                                                                                                                                                                                            |                       |                 |                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-----------------|-----------------|
| Parameter                                                                                                                                                                                                                                                                                                                                                     | Control Word          | Initial Setting | Symbol          |
| Adjust margin                                                                                                                                                                                                                                                                                                                                                 | .AM                   | 0               | &\$AM           |
| Baseline                                                                                                                                                                                                                                                                                                                                                      | .SB                   | 0               |                 |
| Capitalization                                                                                                                                                                                                                                                                                                                                                | .UC, .UP              | OFF             |                 |
| Column balancing                                                                                                                                                                                                                                                                                                                                              | .BC                   | ON              |                 |
| Continuation character                                                                                                                                                                                                                                                                                                                                        | .DC CONT              | (null)          | &\$CONT         |
| Control word separator                                                                                                                                                                                                                                                                                                                                        | .DC CW                | “,”(X'5E')      | &\$CW           |
| Current font                                                                                                                                                                                                                                                                                                                                                  | .BF, .DF, .PF         | (1)             |                 |
| Column definition                                                                                                                                                                                                                                                                                                                                             | .CD                   | Single column   |                 |
| Centering <sup>(2)</sup>                                                                                                                                                                                                                                                                                                                                      | .CE                   | OFF             |                 |
| Column line length                                                                                                                                                                                                                                                                                                                                            | .CL                   | Line length     | &\$CL           |
| Conditional sections                                                                                                                                                                                                                                                                                                                                          | .CS                   | INCLUDE         |                 |
| Font save stack                                                                                                                                                                                                                                                                                                                                               | .BF, .PF              | empty           |                 |
| Line spacing                                                                                                                                                                                                                                                                                                                                                  | .LS                   | (1)             |                 |
| Formatting mode                                                                                                                                                                                                                                                                                                                                               | .FO, .NF, .FV         | .FO ON EXTEND   |                 |
| GML tag delimiters                                                                                                                                                                                                                                                                                                                                            | .DC GML               | “.”(X'7A')      | &\$GML, &\$EGML |
|                                                                                                                                                                                                                                                                                                                                                               |                       | “..”(X'7A')     |                 |
| Horizontal space                                                                                                                                                                                                                                                                                                                                              | .ES, .IC, .WS         | (1)             |                 |
| Hyphenation settings                                                                                                                                                                                                                                                                                                                                          | .HY ON, ALG, ALT, OFF | OFF             |                 |
| Indentation                                                                                                                                                                                                                                                                                                                                                   | (3)                   | 0               | &\$IN           |
| Markup content separator                                                                                                                                                                                                                                                                                                                                      | .DC MCS               | .               | &\$MCS          |
| Page number symbol                                                                                                                                                                                                                                                                                                                                            | .DC PS                | “&”(X'50')      | &\$PS           |
| Revision code, adjust                                                                                                                                                                                                                                                                                                                                         | .RC                   | OFF, 2          |                 |
| Right adjustment                                                                                                                                                                                                                                                                                                                                              | .RI                   | OFF             |                 |
| Right indentation                                                                                                                                                                                                                                                                                                                                             | .IR                   | 0               | &\$IR           |
| Spelling verification                                                                                                                                                                                                                                                                                                                                         | .SV                   | ON              |                 |
| Tab setting                                                                                                                                                                                                                                                                                                                                                   | .TB, .TP              | 5 10 15 ... 80  |                 |
| Terminal input <sup>(2)</sup>                                                                                                                                                                                                                                                                                                                                 | .TE                   | OFF             |                 |
| Underscoring                                                                                                                                                                                                                                                                                                                                                  | .UC, .US, .UD         | OFF             |                 |
|                                                                                                                                                                                                                                                                                                                                                               |                       | ON              |                 |
| <b>Note:</b><br>1 These parameters' initial settings are based on the logical output device.<br>2 The number of lines remaining, or ON or OFF, is saved.<br>3 The composite current indentation is determined from the .IN, .IL, .UN, and .OF control word values. These values are individually saved.<br>These defaults might be overridden by the profile. |                       |                 |                 |

Table 25. Active Formatting Environment

| Page Environment                                                                                                                                                                  |                                            |                                             |              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------|--------------|
| Parameter                                                                                                                                                                         | Control Word                               | Initial Setting                             | Symbol       |
| Bottom margin                                                                                                                                                                     | .BM                                        | (1)                                         | &\$BM        |
| GML processing                                                                                                                                                                    | .GS TAG                                    | OFF                                         |              |
| Hyphenation                                                                                                                                                                       | .HY                                        | OFF                                         | .            |
| Hyphenation limits                                                                                                                                                                | .HY MINPT,<br>MAXPT,<br>MINWORD,<br>LADDER | MINPT 4<br>MAXPT 4<br>MINWORD 6<br>LADDER 2 |              |
| Line length                                                                                                                                                                       | .LL                                        |                                             | &\$LL        |
| Page length                                                                                                                                                                       | .PL                                        | (1)                                         | &\$PL        |
| Page prefix and suffix                                                                                                                                                            | .PN PREF, .PN SUFFIX                       |                                             |              |
| Page margins                                                                                                                                                                      | .PM                                        | (1)                                         | &\$BO, &\$BE |
| Page numbering mode                                                                                                                                                               | .PN                                        | Arabic                                      |              |
| Page width                                                                                                                                                                        | .PW                                        | (1)                                         | &\$PW        |
| Symbol substitution(2)                                                                                                                                                            | .SU                                        | ON                                          | &\$SU        |
| Top margin                                                                                                                                                                        | .TM                                        | (1)                                         | &\$TM        |
| <b>Note:</b><br>1 These parameters' initial settings are based on the logical output device.<br>2 The number of lines remaining that are eligible for substitution, or ON or OFF. |                                            |                                             |              |

Table 26. Page Environment

| Active Environment       |                  |               |
|--------------------------|------------------|---------------|
| Description              | Control Word     | System Symbol |
| Page number symbol       |                  |               |
| .DC PS                   | &\$PS            |               |
| Control word separator   |                  |               |
| .DC CW                   | &\$CW            |               |
| Continuation character   |                  |               |
| .DC CONT                 | &\$CONT          |               |
| GML delimiter            |                  |               |
| .DC GML                  | &\$GML           |               |
| GML first end delimiter  |                  |               |
| .DC GML                  | &\$EGML          |               |
| GML second end delimiter |                  |               |
| .DC GML                  | &\$EGML          |               |
| Markup content separator |                  |               |
| .DC MCS                  |                  |               |
| GML prefix               |                  |               |
| .GS PREFIX               | First parameter  |               |
| GML end prefix           |                  |               |
| .GS PREFIX               | Second parameter |               |

Table 27. Fields of the Active Environment. These fields of the active environment are carried over from the current environment to the running heading and footing environment. Changes made to these fields in the running heading and footing are not carried back to the normal text environment.

| Output File Formats               |                                                                                                |                                               |                                          |                        |
|-----------------------------------|------------------------------------------------------------------------------------------------|-----------------------------------------------|------------------------------------------|------------------------|
| Device                            | CMS <sup>3</sup>                                                                               | MVS/TSO <sup>4 7</sup>                        | MVS/DLF <sup>4 8</sup>                   | VSE/DLF <sup>6 8</sup> |
| 3800-3 <sup>1 2</sup>             |                                                                                                |                                               |                                          |                        |
| Format: V<br>Record Size:<br>8201 | Format: VM, VBM<br>Record Size:<br>8205                                                        | Format: VM, VBM<br>Record Size:<br>8205       | Format: V, VB<br>Record Size: 8201       |                        |
| 3820 <sup>1 2</sup>               |                                                                                                |                                               |                                          |                        |
| Format: V<br>Record Size:<br>8201 | Format: VM, VBM<br>Record Size:<br>8205                                                        | Format: VM, VBM<br>Record Size:<br>8205       | Format: V, VB<br>Record Size: 8201       |                        |
| 4028 <sup>1 2</sup>               |                                                                                                |                                               |                                          |                        |
| Format: V<br>Record Size:<br>8201 | Format: VM, VBM<br>Record Size:<br>8205                                                        | Format: VM, VBM<br>Record Size:<br>8205       | Format: V, VB<br>Record Size: 8201       |                        |
| 4224 <sup>1 2</sup>               |                                                                                                |                                               |                                          |                        |
| Format: V<br>Record Size:<br>8201 | Format: VM, VBM<br>Record Size:<br>8205                                                        | Format: VM, VBM<br>Record Size:<br>8205       | Format: V, VB<br>Record Size: 8201       |                        |
| 4250 <sup>1</sup>                 |                                                                                                |                                               |                                          |                        |
| Format: V<br>Record Size:<br>2048 | Format: V, VB<br>Record Size:<br>2052                                                          | Format: V, VB<br>Record Size:<br>2052         | Format: V<br>Record Size: 2048           |                        |
| Line Printer <sup>2</sup>         |                                                                                                |                                               |                                          |                        |
| Format: V<br>Record Size:<br>256  | Format: VM, VBM<br>Record Size: 260<br><br>Format: FM, FBM<br>Record Size: 256<br><sup>5</sup> | Format: V, VB,<br>VM, VBM<br>Record Size: 256 | Format: F, FB, V, VB<br>Record Size: 256 |                        |
| PostScript <sup>1</sup>           |                                                                                                |                                               |                                          |                        |
| Format: V<br>Record Size:<br>256  | Format: V, VB<br>Record Size: 260                                                              | Format: V, VB<br>Record Size: 260             | Format: V<br>Record Size: 256            |                        |

| Output File Formats                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                   |                                   |                                          |                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|-----------------------------------|------------------------------------------|------------------------|
| Device                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | CMS <sup>3</sup>                  | MVS/TSO <sup>4 7</sup>            | MVS/DLF <sup>4 8</sup>                   | VSE/DLF <sup>6 8</sup> |
| Terminal                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                   |                                   |                                          |                        |
| Format: V<br>Record Size:<br>255 <sup>9</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Format: V, VB<br>Record Size: 259 | Format: V, VB<br>Record Size: 256 | Format: F, FB, V, VB<br>Record Size: 256 |                        |
| Format: F, FB<br>Record Size:<br>255 <sup>5</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                   |                                   |                                          |                        |
| <b>Note:</b> <ol style="list-style-type: none"> <li>Record sizes for these devices should not be changed. Changing these values can cause problems when the file is printed.</li> <li>These files <i>always</i> contain machine carriage control characters.</li> <li>CMS values are determined by SCRIPT/VS and cannot be altered by the user. The record sizes listed are maximums. The actual record length assigned to the file depends on the longest record written.</li> <li>SCRIPT/VS uses the QSAM access method to process the output file. Any block size (BLKSIZE) can be used that is consistent with access method requirements and the amount of storage available for buffers. Note that the record sizes shown in this figure include the record descriptor word.</li> <li>Values given here are the maximum logical record lengths (LRECL) written by SCRIPT/VS and are restricted by internal buffer lengths. If fixed-format records are specified and a record size larger than these values is used, errors or incorrect output results.</li> <li>In VSE, logical record length and block size values are established by the DLF ENVIRONMENT command. For variable-length records, this size does not include the record descriptor word. A block size specified on the ENVIRONMENT command does not include the block descriptor word.<br/><br/>For example, to write 256 byte variable-length records, blocked with a minimum of 10 records per block, the DLF ENVIRONMENT command should say "FORMAT(VB) RECORDSIZE(256) BLOCKSIZE(2600)." The longest physical record or block written to the file would contain 10 or more logical records and could be as long as 2604 bytes in length. The longest logical record could contain 256 bytes of text—including the machine carriage control character, if used—and a record descriptor word for a total length of 260 bytes. Where blocked records are supported, the block size chosen must be consistent with the VSE sequential access method requirements and storage limitations.</li> <li>With variable-length records, if the record size (LRECL) used is insufficient, ABEND 002-18 occurs.</li> <li>In the DLF environments, if the record size is too small, output records are truncated. Because the default record size is 256, the default record size and block size <i>must</i> be overridden when you format documents for the 4250 or 3800-3 printers.</li> <li>If formatting for a 2741 terminal, you could make the record size considerably larger than 255.</li> </ol> |                                   |                                   |                                          |                        |

*Table 28. Output File Formats. These are the output file record formats and record lengths supported by SCRIPT/VS for the various printer types and system environments.*



| Text Fonts                                                                 | Highlight Fonts                                            | Special Fonts                            |
|----------------------------------------------------------------------------|------------------------------------------------------------|------------------------------------------|
| GT10 Gothic (10-pitch)<br>GT12 Gothic (12-pitch)<br>GT15 Gothic (15-pitch) | GB10 Gothic Bold<br>GB12 Gothic Bold<br>GI12 Gothic Italic | GR10 Gothic Reverse<br>GP12 Proportional |
| ST10 Serif (10-pitch)<br>ST12 Serif (12-pitch)<br>ST15 Serif (15-pitch)    | SI10 Serif Italic<br>SI12 Serif Italic<br>SB12 Serif Bold  | RT10 Roman Text<br>SO12 Serif Overstruck |

Table 29. Complete 3800 Printing Subsystem Model 1 and Model 3 Operating in Compatibility Mode (only in MVS) Fonts Provided with SCRIPT/VS. Each font is a complete set of special, uppercase and lowercase characters. Any two of these fonts can be specified with the CHARS option of the SCRIPT command.

| Translate Tables      |              |                 |        |
|-----------------------|--------------|-----------------|--------|
| Parameter             | Control Word | Initial Setting | Symbol |
| Input translation     | .TI          | Identity        |        |
| Output translation    | .TR          | Identity        |        |
| Uppercase translation | .TU          | a-z→A-Z         |        |

Table 30. Translate Tables

| 0 1 2 3 4 5 6 7 8 9 A B C D E F |     |   |   |   |   |   |   |   |   |   |   |   |   |   |    |    |
|---------------------------------|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|----|----|
| 00                              |     |   |   |   |   |   |   |   |   |   |   |   |   |   | 0F |    |
| 10                              |     |   |   |   |   |   |   |   |   |   |   |   |   |   | 1F |    |
| 20                              |     |   |   |   |   |   |   |   |   |   |   |   |   |   | 2F |    |
| 30                              |     |   |   |   |   |   |   |   |   |   |   |   |   |   | 3F |    |
| 40                              |     |   |   |   |   |   |   |   |   |   |   |   |   |   | 4F |    |
| 50                              | £   |   |   |   |   |   |   |   |   |   |   |   |   |   | 5F |    |
| 60                              | - / |   |   |   |   |   |   |   |   |   |   |   |   |   | 6F |    |
| 70                              |     |   |   |   |   |   |   |   |   |   |   |   |   |   | 7F |    |
| 80                              |     | a | b | c | d | e | f | g | h | i | { | ≤ |   |   | 8F |    |
| 90                              |     | j | k | l | m | n | o | p | q | r | } |   | ± | • | 9F |    |
| A0                              |     |   | s | t | u | v | w | x | y | z | └ | ┐ |   | ≥ | •  | AF |
| B0                              | °   | ı | z | ı | ı | ı | ı | ı | ı | ı | ┘ | └ |   | ı | -  | BF |
| C0                              |     | A | B | C | D | E | F | G | H | I | └ | T |   |   |    | CF |
| D0                              |     | J | K | L | M | N | O | P | Q | R |   | S |   |   |    | DF |
| E0                              | \   |   | S | T | U | V | W | X | Y | Z | └ | └ |   |   |    | EF |
| F0                              | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |   |   |   |   |    | FF |
| 0 1 2 3 4 5 6 7 8 9 A B C D E F |     |   |   |   |   |   |   |   |   |   |   |   |   |   |    |    |

Figure 7. TN Translate Table for the 1403 Printer

|                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Additional Fonts</b>                                                                                                                                                                                                                  |
| 10-pitch Fonts                                                                                                                                                                                                                           |
| GS10 GF10 GU10 TU10                                                                                                                                                                                                                      |
| Gothic Gothic Folded Gothic Underscored Text Underscored <sup>1</sup>                                                                                                                                                                    |
| 12-pitch Fonts                                                                                                                                                                                                                           |
| GS12 GF12 GU12                                                                                                                                                                                                                           |
| Gothic Gothic Folded Gothic Underscored                                                                                                                                                                                                  |
| 15-pitch Fonts                                                                                                                                                                                                                           |
| GS15 GSC GF15 GFC GU15 GUC DUMP                                                                                                                                                                                                          |
| Gothic Gothic Condensed Gothic Folded Gothic Folded Condensed Gothic Underscored Gothic Underscored<br>Condensed Condensed DUMP <sup>2</sup>                                                                                             |
| Format Fonts                                                                                                                                                                                                                             |
| FM10 FM12 FM15                                                                                                                                                                                                                           |
| Format 10-pitch Format 12-pitch Format 15-pitch                                                                                                                                                                                          |
| 10-pitch Katakana Fonts                                                                                                                                                                                                                  |
| 2773 2774 KN1                                                                                                                                                                                                                            |
| Gothic and Katakana Gothic and Katakana <sup>2</sup> Gothic and Katakana <sup>2</sup>                                                                                                                                                    |
| 3211 Print Trains                                                                                                                                                                                                                        |
| A11 G11 H11 P11 T11                                                                                                                                                                                                                      |
| Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch Text 10-pitch <sup>1</sup>                                                                                                                                               |
| 1403 Print Trains                                                                                                                                                                                                                        |
| AN GN HN PCAN PCHN PN QN QNC RN XN YN SN TN                                                                                                                                                                                              |
| Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch<br>Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch Gothic 10-pitch Text 10-pitch <sup>1</sup> Text 10-pitch <sup>1</sup> |
| 0-pitch OCR Fonts                                                                                                                                                                                                                        |
| AOA AOD AON OAA ODA ONA BOA BON OAB ONB                                                                                                                                                                                                  |
| Gothic and OCR-A Gothic and OCR-A Gothic and OCR-A Gothic and OCR-A Gothic and OCR-A Gothic and OCR-A<br>OCR-A Gothic and OCR-B Gothic and OCR-B OCR-B Gothic and OCR-B                                                                  |
| <b>Note:</b>                                                                                                                                                                                                                             |
| <b>1</b> This is an uppercase and lowercase font which closely resembles the ST10 SCRIPT/VS font. It counts as two fonts when combined with other fonts in the CHARS option of the SCRIPT command.                                       |
| <b>2</b> This font contains more than 64 characters. It counts as two fonts when combined with other fonts in the CHARS option of the SCRIPT command.                                                                                    |

*Table 31. Fonts Supplied with the 3800 Printing Subsystem Model 1 and Model 3 Operating in Compatibility Mode (MVS only). These are all uppercase-only fonts, unless otherwise marked. Any four fonts (except those otherwise marked) of identical pitch (measured in characters per inch) can be specified with the CHARS option of the SCRIPT command.*

|                               |                                                      |                                                                                                                                                                |
|-------------------------------|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>10-pitch Fonts</b>         | CR10<br>CE10<br>CO10<br>PR10<br>OR10<br>OB10         | Courier-10 Regular<br>Courier-10 Extended<br>Courier-10 Overstruck<br>Prestige Pica Regular<br>Orator Regular<br>Orator Bold                                   |
| <b>12-pitch Fonts</b>         | CE12<br>PR12<br>PI12<br>PB12<br>LR12<br>LB12<br>SR12 | Courier-12 Extended<br>Prestige Elite Regular<br>Prestige Elite Italic<br>Prestige Elite Bold<br>Letter Gothic Regular<br>Letter Gothic Bold<br>Script Regular |
| <b>10-pitch Special Fonts</b> | DOTR<br>BRTR<br>BITR<br>ESTR<br>EBTR<br>EITR         | Document<br>Boldface Regular<br>Boldface Italic<br>Essay Standard<br>Essay Bold<br>Essay Italic                                                                |

*Table 32. Additional Fonts Supplied with the 3800 Printing Subsystem Model 3.. These additional fonts are supplied with the 3800 Printing Subsystem Model 3 operating in compatibility mode (MVS only). These fonts are available for the 6670 for compatibility reasons.*



This part contains the following appendixes:

- Appendix A, “Unsupported Control Words” on page 461
- Appendix B, “Related Publications and Products” on page 479



## Appendix A. Unsupported Control Words

This appendix contains detailed descriptions of control words that were made obsolete in earlier releases of the SCRIPT/VS language. **Central Services support is not available for these control words or their documentation.**

The following table lists the unsupported SCRIPT/VS control words. SCRIPT/VS continues to recognize these control words, but it is recommended that you use the new control words for these functions.

| Table 33. Unsupported Control Words |                                                                                    |
|-------------------------------------|------------------------------------------------------------------------------------|
| Unsupported Control Word            | Control Word Providing Comparable Function                                         |
| .BT                                 | .RF [Running Footing]                                                              |
| .CO                                 | .FO [Format Mode]                                                                  |
| .CW                                 | .DC [Define Character]                                                             |
| .DI                                 | .DA [Define Area]                                                                  |
| .EB                                 | .RF [Running Footing]                                                              |
| .EP                                 | .PA [Page Eject]                                                                   |
| .ET                                 | .RH [Running Heading]                                                              |
| .EZ                                 | .DH [Define Head Level]                                                            |
| .FI                                 | .FO [Format Mode]                                                                  |
| .FM                                 | .RF [Running Footing]                                                              |
| .FS                                 | .RF [Running Footing]                                                              |
| .FT                                 | .RF [Running Footing]                                                              |
| .HE                                 | .RH [Running Heading]                                                              |
| .HM                                 | .RH [Running Heading]                                                              |
| .HN                                 | .RH [Running Heading]                                                              |
| .HS                                 | .RH [Running Heading]                                                              |
| .NB                                 | .BC [Balance Columns]                                                              |
| .NC                                 | .FO [Format Mode]                                                                  |
| .NJ                                 | .FO [Format Mode]                                                                  |
| .OB                                 | .RF [Running Footing]                                                              |
| .OP                                 | .PA [Page Eject]                                                                   |
| .OT                                 | .RH [Running Heading]                                                              |
| .PP                                 | .SK [Skip]                                                                         |
| .PS                                 | .DC [Define Character]                                                             |
| .RT                                 | .RH [Running Heading]                                                              |
| .SL                                 | .LS [Line Spacing]                                                                 |
| .SY                                 | Refer to the <i>Document Composition Facility: Diagnosis Guide and Reference</i> . |
| .TT                                 | .RH [Running Heading]                                                              |

---

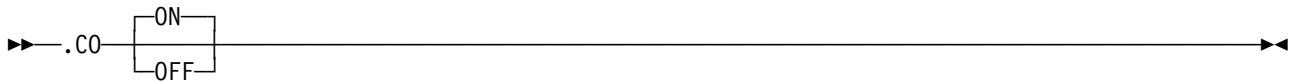
## **.CO [Concatenate Mode]**

### **Purpose**

Use the .CO [Concatenate Mode] control word to cancel or restore concatenation of input lines and truncation at the current column length.

This control word is provided for compatibility with earlier releases of SCRIPT/VS. The same function is provided by the .FO [Format Mode] control word.

### **Format**



### **Parameters**

**ON** Restores concatenation of input lines. ON is the initial setting, as well as the default value.

**OFF** Cancels concatenation of input lines. If justification is still in effect, .CO OFF results in each line being padded with blanks to the column length.

**Initial Setting:** ON

**Default:** ON

### **Restrictions**

1. When SCRIPT/VS is concatenating text, output lines are formed by shifting words to or from the next input line. The resulting line is as close to the specified column width as possible without exceeding it or splitting a word; if justification is OFF, output resembles normal typist output. Concatenation is the normal mode of operation for the SCRIPT command.

When SCRIPT/VS is not concatenating text, there is a one-to-one correspondence between the words on the input and output lines. If SCRIPT/VS is still justifying text, each line that is less than the column length is padded with blank space to fill the column.

2. Concatenation is one component of format mode, as controlled by the .FO [Format Mode] control word. The .CO control word is provided for occasions when you must control concatenation separately, but all ordinary formatting combinations are controlled by the .FO control word, and you should use it instead of .CO whenever possible.
3. This control word causes a break.



---

## **.CW [Control Word Separator]**

### **Purpose**

The .CW [Control Word Separator] control word allows you to change the symbol used to separate multiple control words on a single line. The initial value for the control word separator symbol is the semicolon ( ; ) character.

This control word is provided for compatibility with earlier releases of SCRIPT/VS. The same function is provided by the .DC [Define Character] control word.

### **Format**

►► .CW c ►

### **Parameters**

- c* Specifies the character to be used as the control word separator character. Any character can be used. If the character *c* is omitted, no character is assigned as the control word separator, and therefore you cannot have more than one control word on a line.

**Initial Setting:** Semicolon ( ; ).

**Default:** None.

### **Restrictions**

1. All control word lines are scanned for control word separators before they are processed, unless they are specified with the control word modifier. The control word modifier allows the line that accompanies a control word to be treated as text, which can contain control word separators as ordinary text characters.

The control word modifier is a single quotation mark immediately after the period. The control word

.ce one; two

is scanned before being processed into the two lines (.ce one and two). But the line

.'ce one; two

uses the control word modifier to allow the string (one; two) to be centered.

The .CW control word should always be specified with the control word modifier. The .CW control word line, like all unmodified control word lines, is scanned for control word separators before being processed. If you want to ensure that the control word separator is set to a semicolon by issuing .cw ;, the opposite occurs if the semicolon happened to be the current separator; the line is separated before being processed into the line .cw, followed by no more on that line. When the control word is processed, it undefines the semicolon as the separator.

If you always use the control word modifier with .CW, no separator scanning is done, and the character is preserved as the parameter on the control word:

.'cw ;

ensures that the separator is set to ; .

2. The control word separator is treated as a text character when it appears in a text line, except when immediately followed by a period and a two-character control word ID. Therefore,

## **.CW [Control Word Separator]**

Do ; .us on;not ;.us off; stop!

is formatted as

Do not stop!

On the other hand,

Centering is off; .ce turns it on

is formatted as text, because the control word separator is not immediately followed by a period.

3. When the .CW control word is processed, the initial value for the control word separator ( ; ) is reset. It might be necessary to change the control word separator character if it is inconvenient to type the initial-value character, or if the initial-value character is used as part of a control word operand, such as part of a symbol specification.
4. If a symbol value begins with the control word separator, the rest of the symbol value is treated as though it occupies the first position of the line.
5. Control word separators are recognized on a .CM [Comment] line, but not on a .\* line.
6. The following control words must begin in column 1 and can not be placed after a control word separator:

```
.cs n off  
.di off  
.wf off  
.li off  
...label
```

When SCRIPT/VS is ignoring a conditional section, preparing a delay imbed, writing to a file, reading literal lines, or searching for a label, no control word processing is done. Each input record is checked in column 1 for the presence of the control word that ends the special processing mode.

7. Control words that accept text data, for example, .US or .CE, should not contain the current control word separator as text, unless the control word modifier is used to prevent scanning for the separator.

## **Examples**

- Simple change:

```
.'cw ,  
.sp 2,.of 5, This section...
```

The above line is equivalent to the lines:

```
.sp2  
.of 5  
This section...
```

- Temporary cancellation to get the separator character into a symbol value:

```
. 'CW
.se 2col = ' ;.cd 2 0 46;.cl 43;'
.se 1col = ' ;.cd 1;.cl 89;'
. 'CW ;
```

In the sequence above, the control word separator is temporarily canceled so that the regular separator ( ; ) can be used as part of the .SE [Set Symbol] control word line. Because the symbols &2col and &1col contain the appropriate control words, they can be used instead of the actual control words. Because the control words are in a symbol that begins with the control word separator, they can be recognized as control words even if the symbol is encountered in the middle of a line. Because the symbols end with control word separators, the effective next line can be concatenated to the symbol name. With the symbols &2col and &1col set as shown, the line:

This is a line.&2col.Start 2 cols.

Has the same effect as the sequence:

```
This is a line.
.cd 2 0 46
.cl 43
Start 2 cols.
```

---

## .DI [Delay Imbed]

### Purpose

Use the .DI [Delay Imbed] control word to defer the inclusion of a portion of a SCRIPT file until the next page eject occurs.

### Format



### Parameters

- n* Specifies the number of lines to be delayed. If *n* is omitted, 1 is assumed.
- ON** Starts an open-ended delayed imbed. Subsequent lines, are included in the delay imbed file until a .DI OFF is encountered.
- OFF** Ends a delayed imbed, whether it was started with .DI ON or with a specified *n* that has not been exhausted.
- line* Is an input line that is to be delayed.

**Default:** 1

### Restrictions

1. The specified lines of the current file are saved in a temporary file. When the top of the next output page is reached, this temporary file is imbedded and processed by SCRIPT/VS. After the saved lines are included, normal processing resumes.  
However, any text that was formatted prior to the page eject, such as a widow zone is not reformatted. It should be noted that DCF might format up to three lines before it determines that a page eject must be performed.
2. This control word does not cause a break. However, a control word that causes a break can be included as the last line of the delayed imbed to ensure that the text of the delayed imbed is not formatted with the text that follows the imbed. The results are not readily predictable if a control word that causes a break is not included.
3. An automatic page eject is not performed at the end of the included lines. If you want SCRIPT/VS to resume normal processing on a new page, you should end the delayed section with a .PA control word.
4. The .DI OFF control word must begin in column 1, not after a control word separator. When SCRIPT/VS is processing a delay imbed, it processes input lines to look for .DI OFF on a line by itself.
5. No .DI control word is placed in the delay imbed file.

### Examples

- To delay the inclusion of one line, enter,  
.di .pa

The single line .pa is written into the delay imbed file. At the end of the current page, a blank page, is generated excluding the top and bottom titles. Output resumes on the page after the blank page.

- To include a figure at the top of the next page:

```
.di 3  
.sp .5i  
.im figure5  
.sp 5
```

The current page is processed as if the .DI control word and the three following lines did not exist. At the top of the next page, the three lines are processed. This results in a one-half inch space, followed by an imbedded file, named figure5, and then five lines of spacing.

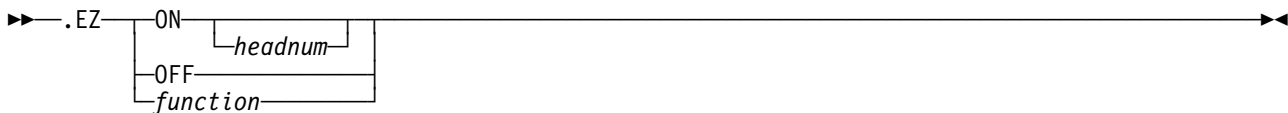
---

## .EZ [Easyscript]

### Purpose

EasyScript is an early implementation of GML that existed in SCRIPT/370. The .EZ [EasySCRIPT] control word provides automatic formatting functions used by EasyScript. These functions are available through a set of EasyScript tags or through the .EZ control word. The EasyScript tags are symbols that substitute for the appropriate .EZ control word. They are not true GML tags; they are delimited with the ampersand (&), rather than the GML delimiter (:). EasyScript tags are included in SCRIPT/VS to allow documents already marked up with them to be processed by SCRIPT/VS.

### Format



### Parameters

- ON** Initializes the EasyScript tags that provide the EasyScript numbering, paragraphing, and heading functions. The names of the tags are the same as the parameters of the .EZ control word that provide the associated function. ON also switches the head level definitions from the standard set to another set used only while EasyScript is in effect. The .DH [Define Head Level] control word operates on the set of head levels currently in effect (standard or EasyScript)
- headnum* Is the decimal number of the last heading that was used. EasyScript uses this number to set the counter for numbered headings. If counters are not specified, 0.0.0.0 is assumed. If you specify &xref, EasyScript resumes numbering where it left off when .EZ OFF was last processed. (&xref is the symbol EasyScript uses to keep track of the current heading number.)
- OFF** Cancels the EasyScript tags, so that they are not recognized by SCRIPT/VS. OFF also switches the head-level definitions back to the standard set.
- function* Is the name of the EasyScript *function* to be started. The line of text data that follows the *function* name is processed by the built-in function requested. The names of the functions are case sensitive. For example, there are two different bulleted list functions: the B function, in uppercase, starts a regular bulleted item, and the b function, in lowercase, starts a nested item.
- line* Is an input *line* of data. It must be separated from the function name by at least one blank.

### Restrictions

1. EasyScript functions provide a formatting method for text and documents, particularly those that require decimal numbering. EasyScript provides automatic numbering for heading levels and lists, if requested.
2. The names of the EasyScript functions are the same as the names of the tags set up by .EZ ON. For example, the N3 function identifies a numbered list item at level 3. This function can be started with the control word.

.ez N3 text of the numbered item  
or with the tag

&N3.text of the numbered item

but the latter is enabled only after .EZ ON has been processed.

3. The symbol &xref contains the entire number of the current heading level, when EasyScript's automatic numbering scheme is used. The symbols &xref1, &xref2, &xref3, and &xref4 contain the components of this number. For example, if &xref has the value of 1.0, then &xref1 has the value of 1 and &xref2 has the value of 0.
4. The EasyScript functions are summarized below in the following table. Note the differences in the uppercase and lowercase versions of a function name:

| <i>Table 34. Easyscript Functions</i> |                                                                |
|---------------------------------------|----------------------------------------------------------------|
| Hx                                    | Begins a decimal numbered heading of level x (1 through 4).    |
| hx                                    | Begins an unnumbered heading of level x.                       |
| P                                     | Begins a major paragraph by resetting the current indentation. |
| p                                     | Begins a minor paragraph at the current indentation.           |
| Nx                                    | Begins a numbered item of level x (1 through 4).               |
| nx                                    | Begins an unnumbered item of level x (1 through 4).            |
| B                                     | Begins a bulleted item.                                        |
| b                                     | Begins a nested item.                                          |
| toc                                   | Generates a table of contents.                                 |

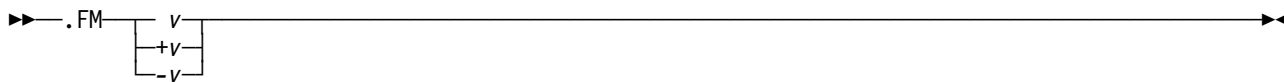
---

## .FM [Footing Margin]

### Purpose

Use the .FM [Footing Margin] control word to specify how much space to skip between the last line of text, on a full page, and the bottom titles, overriding the initial setting established for the device.

### Format



### Parameters

- v Specifies the amount of space to be skipped between the last line of text and the footings (bottom titles). If +v or -v is specified, the current value of the footing margin is either incremented or decremented. If no .FM control word is used in the file, or if the .FM control word is used without a parameter value, the initial value is used. The minimum value that can be specified for the footing margin is 0. If a negative result is calculated for the footing margin, the value is set to zero, and a message is issued. The maximum value that can be used for the footing margin is equal to the bottom margin (.BM) minus the footing space (.FS).

**Initial Setting:** Depends on the logical device for which the document is formatted.

**Default:** Restores the initial setting.

### Restrictions

1. The bottom titles are placed a specified amount of space below the last line of text. The location of the last line of text is explicitly defined by the .BM [Bottom Margin] control word, whether that line is actually filled or not.
2. This control word does not cause a break.
3. The .FM control word takes effect on the page after it is encountered. For example, if you enter,  
.fm .5i

The result is one-half inch of space left between the last line of text and the running bottom titles, if bottom titles are defined.

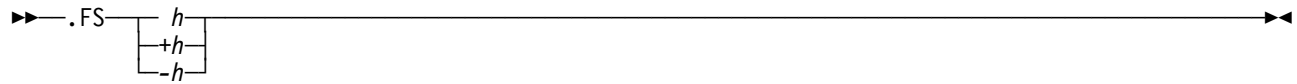


## .FS [Footing Space]

### Purpose

The .FS [Footing Space] control word allocates space from the bottom margin area for running bottom titles.

### Format



### Parameters

*n* Is the number of bottom title lines that you want to appear on this and all subsequent output pages. *n* must be an integer from 0 to 6. If no number is given, 1 line is assumed. This control word does not accept space units. The number must be less than the bottom margin (.BM) minus the footing margin (.FM). If you specify *+n* or *-n*, the current value of the footing space is either incremented or decremented. If the net result is a negative number, zero is assumed and a message is issued.

**Initial Setting:** 1

**Default:** 1

### Restrictions

1. The .FS [Footing Space] control word allocates space from the bottom margin for bottom titles. You need to use only the .FS control word if you want more than one bottom title in your document. If the bottom margin is not large enough to accommodate the footing space plus the footing margin, an error message is generated.
2. This control word does not cause a break, and takes effect on the page after it is encountered.
3. The running bottom title control words cause a title line to be saved in a storage area. The first bottom title used at the bottom of output pages is the default. To get more than one title at the bottom of your formatted output pages, you must do two things: define the titles using the .RT [Running Title] control word, and allocate space for the titles by using the .FS control word.
4. If you do not want bottom titles, define the footing space as 0 (.FS 0). This is more efficient than setting the bottom titles to null (.RT B ///), because SCRIPT/VS does not need to process titles to determine that none are specified.

### Examples

If you want three running bottom titles in your document, you could use the following sequence:

```
.rt b 3 /Chapter 4//&/
.r t b 2 ///
.r t b 1 $$&SYSMONTH./&SYSYEAR.$$
```

Only bottom title 1, the one nearest the bottom of the page, is used on formatted output pages because the default footing space of 1 is still in effect. Now that the three title lines have been saved, the following control word causes SCRIPT/VS to print all three:

```
.fs 3
```

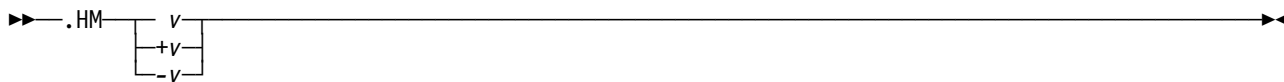
---

## .HM [Heading Margin]

### Purpose

The .HM [Heading Margin] control word specifies the amount of space to be skipped between the running top titles and the first line of the text area, overriding the initial value established for the device.

### Format



### Parameters

- v* Specifies the amount of space to be skipped at the top title lines. If +*v* or -*v* is specified, the current value of the heading margin is either incremented or decremented. If the calculated value of the heading margin is negative, the value is set to zero and a message is issued. The maximum value that can be set for the heading margin is equal to the top margin (.TM) minus the heading space (.HS). If *v* is not specified, the default value for the logical device is restored.

**Initial Setting:** Depends on the logical device for which the document is formatted.

**Default:** Restores the initial setting.

### Restrictions

1. The last running top title line is placed a specified amount of space above the first line of text. If no .HM [Heading Margin] control word is included in the file the default value is used, as determined for the logical output device.
2. This control word does not cause a break, and it takes effect on the page after it is encountered. For example, if you enter,

.hm 3

Three lines are left between the running title lines and the first line of text. If a top margin of 6 lines is in effect, the last top title is printed two lines from the top of the page, followed by three more blank lines (the heading margin), and then the text.

## .HS [Heading Space]

### Purpose

The .HS [Heading Space] control word allocates space from the top margin area for running top titles.

### Format



### Parameters

*n* Is the number of top title lines that you specify on each subsequent output page. The *n* must be an integer from 0 to 6. If no number is given, 1 is assumed. The .HS control word does not accept space units. The size of the top margin (.TM) minus the heading margin (.HM) must be large enough to accommodate the heading space specified. If *+n* or *-n* is specified, the current value for the heading space is either incremented or decremented. If the net result is less than zero, the heading space is set to zero, and an error message is issued.

**Initial Setting:** 1

**Default:** 1

### Restrictions

1. The .HS [Heading Space] control word allocates space from the top margin for running top titles. Use this control word only if the default value of one top title is not adequate for your document. If the top margin is not large enough to accommodate the heading space plus the heading margin, an error message is generated.
2. This control word does not cause a break, and it takes effect on the page after it is encountered.
3. The .RT [Running Title] control word causes a title line to be saved in a storage area for future use. Only the first top title is used by default at the top of output pages. To get more than one title at the top of your formatted output pages you must do two things: define the titles using .RT, and then allocate space for the titles by using the .HS control word.
4. If you do not want top titles, define the heading space as .HS 0. This is more efficient than setting the top titles to null .RT T ///, because SCRIPT/VS does not need to process titles to determine that none are required.

### Examples

If you want three running top titles in your document, you could use the following sequence:

```
.rt t 1 $$&SYSMONTH./&SYSYEAR.$$
.r t 2 ///
.r t 3 /CHAPTER 4//&/
```

only top title 1 is used on formatted output pages, because the default heading space of 1 is still in effect.

Now that the three title lines have been saved, the following causes SCRIPT/VS to print all three:

```
.tm 8
.hs 3
```

---

## **.PP [Paragraph Start]**

### **Purpose**

Use the .PP [Paragraph Start] control word to start a new paragraph.

### **Format**

►► .PP line ►◄

### **Parameters**

*line* Is the text that begins a new paragraph. If *line* is omitted, the text from the next input line after the .PP control word begins the new paragraph.

### **Restrictions**

When the .PP control word is encountered, a break occurs, a skip is generated, and the next line of text is indented three character spaces to the right of the current margin. The .PP control word is equivalent to the control words:

```
.sk  
.il +3
```

If these values are not satisfactory for your paragraph formatting, you can redefine the .PP control word as a SCRIPT/VS macro. For example,

The input lines:

```
.pp This line begins with a .PP control word.  
Here is more text to show the formatting.
```

Is formatted as:

This line begins with a .PP control word. Here is more text to show the formatting.

---

## **.PS [Page Number Symbol]**

### **Purpose**

The .PS [Page Number Symbol] control word allows you to change the page number symbol used in running top and running bottom titles and running headings and running footings. The default page number symbol is the ampersand (&) character.

This control word is provided for compatibility with earlier releases of SCRIPT/VS. The same function is provided by the .DC [Define Character] control word.

### **Format**

►► .PS c ►►

### **Parameters**

*c* Specifies the character to be used as the page number symbol. It can be any character other than a blank. If *c* is omitted, no character is assigned as the page number symbol.

**Initial Setting:** Ampersand (&)

**Default:** None.

### **Restrictions**

1. Every occurrence of the page number symbol is replaced with the current page number in running titles, running headings, and running footings, unless .PN OFF or .PN OFFNO is in effect.
2. The .PS control word allows you to change the page number symbol currently in effect. The initial page number symbol is the ampersand (&) character. It might be necessary to change the page number symbol if the & character is not a valid character on your terminal keyboard or the & character is required as a regular character in your title text.
3. This control word affects all running top and running bottom titles and all running headings and running footings, including those that were previously defined. Therefore, if a title is set by the control word:

.rt t ///page &/

and later the control word:

.ps ?

is encountered, the top-title must be reset to:

.rt t ///page ?/

otherwise, the current page number is not substituted into the title.

4. Do not confuse the page number symbol with the ampersand used on the right-hand side of a .SE [Set Symbol] control word. A single ampersand in a .SE control word means that the symbol is to be set to the current page number.

.se currpage = &

sets the symbol *currpage* to the current page number, regardless of what character, if any, is defined as the page number symbol.

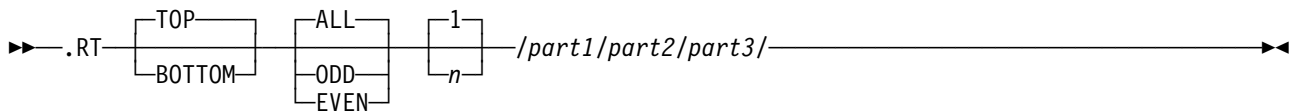
---

## .RT [Running Title]

### Purpose

The .RT [Running Title] control word saves a specified title line in a storage buffer. This title can be used at the top or bottom of the next page and each subsequent output page.

### Format



### Parameters

|               |                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TOP</b>    | Specifies that this control word refers to top titles. The TOP parameter can be abbreviated as T. This is the default.                                                                                                                                                                                                                                            |
| <b>BOTTOM</b> | Specifies that this control word refers to bottom titles. The BOTTOM parameter can be abbreviated as B.                                                                                                                                                                                                                                                           |
| <b>ODD</b>    | Specifies that the title that is defined is to be printed only on odd numbered pages. The ODD parameter can be abbreviated as O.                                                                                                                                                                                                                                  |
| <b>EVEN</b>   | Specifies that the title that is defined is to be printed only on even-numbered pages. If neither ODD nor EVEN is specified, the title that is defined prints on both even-numbered and odd-numbered pages. The EVEN parameter can be abbreviated as E.                                                                                                           |
| <b>ALL</b>    | Specifies that the title is to be printed on both odd-numbered and even-numbered pages. ALL is the default.                                                                                                                                                                                                                                                       |
| <i>n</i>      | Is the number of the title line to be set. The number can be from 1 to 6, and if <i>n</i> is omitted, 1 is assumed. The six possible title lines are the same for top titles and bottom titles. Bottom titles are numbered from bottom to top; top titles are numbered from top to bottom. Therefore, top title 1 sets the same storage buffer as bottom title 6. |
| <i>part1</i>  | Is the portion of the title to be left justified.                                                                                                                                                                                                                                                                                                                 |
| <i>part2</i>  | Is the portion of the title to be centered between the left and right margins.                                                                                                                                                                                                                                                                                    |
| <i>part3</i>  | Is the portion of the title to be right justified.                                                                                                                                                                                                                                                                                                                |
| <i>/</i>      | Is any delimiter character that does not appear in <i>part1</i> , <i>part2</i> , or <i>part3</i> .                                                                                                                                                                                                                                                                |

**Initial Setting:** Top All 1 ///Page &

### Restrictions

1. Every occurrence of the page number symbol in *part1*, *part2*, and *part3* is replaced with the current page number on each page where a title appears, unless .PN OFF or .PN OFFNO is in effect. The character designated as the page number symbol can be changed with the .PS [Page Number Symbol] control word or the .DC [Define Character] control word.
2. Symbol substitution and character translation set up by the .TR [Translate Character] control word is done on *part1*, *part2*, and *part3* when the .RT control word is processed, not on every page.
3. The three parts of the title are used to form the actual title that is to be saved, and no part can be more than 120 characters in length. This title can be printed at the top or bottom of each subsequent output page, if space has been allocated for it by using the .HS or .FS control words.

4. The specific location of the top titles on the page is controlled by the .TM [Top Margin] and .HM [Heading Margin] control words; the number of top titles to be used on each page is controlled by the .HS [Heading Space] control word.
5. The specific location of the bottom titles on the page is controlled by the .BM [Bottom Margin] and .FM [Footing Margin] control words; the number of bottom titles to be used on each page is controlled by the .FS [Footing Space] control word.
6. Any title can be changed by including another .RT control word later in the file.
7. The default top title, printed on each page of output after page one, is  
.PAGE &  
which is right-justified at the top of the page. This title can be suppressed with the .PN OFF control word.
8. This control word takes effect on the page after it is encountered.
9. The length of the title is that of the line length as it is set by the .LL control word.
10. The parameters can be specified in any order, and if contradictory options are specified, only the latest one is used. The first character that is not recognized as an option is taken as a delimiter.

`.rt t 'heading' 'page &'`

The heading and the current page number are printed at the top of all subsequent pages, unless the heading space is set to zero.

---

## .SL [Set Line Space]

### Purpose

This control word defines the vertical distance from the baseline of the current line to the baseline of the following line.

### Format

►► .SL—*vsiz*e—————►◄

### Parameters

*vsiz*e     Is the vertical size of all of the following formatted output lines until another .SL control word redefines the vertical size.

**Initial Setting:** One logical device print line.

**Default:** One print line.

### Restrictions

1. The vertical size of formatted output lines is set by the .SL control word to the nearest approximation of the requested size that is possible on the current logical device.
2. The .SL value is used for formatted lines and for requests in lines for the following control words:

- .CC [Conditional Column Begin]
- .CP [Conditional Page Eject]
- .SK [Skip]
- .SP [Space]

In all other control words that have a vertical dimension expressed in lines, such as .PL [Page Length] and .TM [Top Margin], the size of the request is based on the size of a print line on the current logical device. For example, if the logical device prints 8 line per inch, the control .TM 4 would set the top margin to 4 lines, or one-half inch, regardless of the .SL control value.

3. Additional space is placed above each line, and is discarded if the line falls at the top of a column.
4. The A parameter of the .SP [Space] and .SK [Skip] control words does not affect spacing set with the .SL control word.



---

## Appendix B. Related Publications and Products

---

### Related Products

The following products are related to DCF. For information about ordering any of these products, contact your local IBM Branch Office.

- **MARKUP:** A PC-based editor that helps you create GML documents. Refer to *MARKUP User's Guide and Tutorial*, which you can order with the MARKUP product, 6476161.
- **Publishing Systems BookMaster:** A host-based application that runs under control of DCF and is designed for high-volume in-house publishing applications. This application contains a superset of the GML Starter Set.
- **Publishing Systems BookManager BUILD and BookManager READ:** Host-based IBM licensed programs that let you create and use online books and documents at your terminal in a VM/CMS system.
- **OfficeVision:** OfficeVision provides an integrated electronic office that delivers a broad range of office functions for business communications with the help of an integrated set of services.
- **Standard Generalized Mark-up Language (SGML) TextWrite OS/2 Edition:** TextWrite is a software program that writers can use to create and modify SGML-compliant documents.
- **TextTagger:** TextTagger is a software program that analyzes electronic documents and inserts tagging to comply with the Department of Defense's Computer-Aided Acquisition and Logistics Support (CALS) initiative.
- **SGML Translator DCF Edition:** A program that parses, validates, and translates SGML documents so they can be formatted and printed by DCF.
- **Print Services Facility (PSF):** A licensed program that combines print data with resources to manage and control data transmitted to IBM page printers.

---

### Optional Features

For information about ordering these optional features, contact your local IBM Branch Office:

- **SCRIPT Mathematical Formula Formatter (SMFF):** An optional feature of DCF that can be ordered separately. SMFF makes it possible to construct and display mathematical and scientific formulas on page printers.
- **Office Document Feature (ODF):** An optional feature of DCF that can be ordered separately. ODF allows documents that have been created with an office system to be printed on DCF-supported printers.
- **Document Composition Facility/Double Byte (DCF/DB):** An optional feature of DCF that can be ordered separately. DCF/DB allows for the inclusion and printing of double-byte characters in four supported languages with DCF.
- **DCF Viewer Windows Kit:** An optional feature of DCF that can be ordered separately. The DCF Viewer Windows Kit enables you to browse and print DCF documents and resources on your workstation.

## Publication Library Guide for the Document Composition Facility

The following table lists the Document Composition Facility publications by number as they relate to user tasks. “DCF Publications” on page 482 lists the titles and the order numbers that correspond to the numbers listed in the table.

| Number                                                           | User Tasks                                           | Typical Audience                                         | Brief Description                                                                                                                             |
|------------------------------------------------------------------|------------------------------------------------------|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| (1) (2) (3) (20)                                                 | Planning and introducing DCF/DLF                     | Users, system planners                                   | Provide a general overview of text processing, library facility, and available books.                                                         |
| (3) (4) (5) (12) (16) (20)                                       | Formatting documents (using the GML starter set)     | Novice to experienced end users                          | Provide an introduction to the Generalized Markup Language (GML) starter set and describes the GML starter set tags and SCRIPT/VS messages.   |
| (6)                                                              | Creating bar codes with DCF/GML                      | Experienced end users                                    | Provides information about using GML to create bar codes.                                                                                     |
| (9) (10) (11) (12) (17) (18) (19) (21)                           | Formatting documents (using SCRIPT/VS control words) | Knowledgeable to experienced end users                   | Describe the function and use of all SCRIPT/VS control words, macros, diagnostic aids, and the formatting features and messages.              |
| (14) (15)                                                        | Converting RFTDCA for SCRIPT/VS formatting           | Novice to experienced RFTDCA users                       | Describe the function and use of the optional Office Document Feature, including diagnostic aids and messages.                                |
| (4) (5) (7) (9) (10) (11) (19)                                   | Modifying GML starter set <sup>13</sup>              | Document administrator and text programmer <sup>14</sup> | Contain material about GML starter set tags, SCRIPT/VS control words, and GML starter set modifications.                                      |
| (4) (5) (7) (8) (9) (10) (11) (16) (19)                          | Creating GML application processing functions        | Document administrator and text programmer <sup>14</sup> | Provide information about designing your own GML and about GML concepts, GML starter set tags, SCRIPT/VS control words, and usage guidelines. |
| (10) (11) (12) (13) (19) (24) (25) (26) (27) (28) (29) (31) (32) | Installing, modifying, and maintaining DCF           | System programmer                                        | Give information on error isolation, program tailoring, and use of SCRIPT/VS.                                                                 |
| (22)                                                             | Creating mathematical formulas with SMFF             | Experienced end users                                    | Describes the function and use of the SCRIPT Mathematical Formula Formatter (SMFF), including .EQ control word and messages.                  |

<sup>13</sup> Central Programming Service support and maintenance are provided **only** on the unmodified GML applications shipped with DCF. If you modify any of these GML applications shipped with DCF, it is recommended that you also maintain an **unmodified** copy for diagnostic purposes.

<sup>14</sup> The document administrator is responsible for defining markup conventions and procedures for an organization. The text programmer implements application processing functions (APFs) that provide the processing specified by the document administrator.

| Number | User Tasks                                                          | Typical Audience                | Brief Description                                                                             |
|--------|---------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------------|
| (23)   | Creating memos, transparencies, and schedules with GML applications | Novice to experienced end users | Describes the use of the memo, transparencies, and schedule applications, including messages. |
| (30)   | Creating DCF documents with SBCS and DBCS                           | Novice to experienced users     | Describes the use of SBCS and DBCS with DCF.                                                  |

## DCF Publications

### Number Titles and Order Numbers

|      |                                                                                                                                            |
|------|--------------------------------------------------------------------------------------------------------------------------------------------|
| (1)  | <i>About DCF</i> , G520-6362.                                                                                                              |
| (2)  | <i>Document Composition Facility and Document Library Facility General Information</i> , GH20-9158.                                        |
| (3)  | <i>Document Composition Facility: Introduction to Generalized Markup Language</i> , G544-3192.                                             |
| (4)  | <i>Document Composition Facility: Generalized Markup Language Starter Set User's Guide</i> , SH20-9186.                                    |
| (5)  | <i>Document Composition Facility: Generalized Markup Language Starter Set Reference</i> , SH20-9187.                                       |
| (6)  | <i>Document Composition Facility: Bar Code User's Guide</i> , S544-3115.                                                                   |
| (7)  | <i>Document Composition Facility: Generalized Markup Language Starter Set Implementation Guide</i> , SH35-0050.                            |
| (9)  | <i>Document Composition Facility: SCRIPT/VS User's Guide</i> , S544-3191.                                                                  |
| (10) | <i>Document Composition Facility: SCRIPT/VS Text Programmer's Guide</i> , SH35-0069.                                                       |
| (11) | <i>Document Composition Facility: SCRIPT/VS Language Reference</i> , SH35-0070.                                                            |
| (12) | <i>Document Composition Facility SCRIPT/VS Messages</i> , SH35-0048.                                                                       |
| (13) | <i>Document Composition Facility: Diagnosis Guide and Reference</i> , LH40-0209.                                                           |
| (14) | <i>Document Composition Facility: Office Document Feature User's Guide</i> , G544-3129.                                                    |
| (15) | <i>Document Composition Facility: Office Document Feature Reference</i> , S544-3130.                                                       |
| (16) | <i>Using the Document Composition Facility</i> , SR21-0515 (Training Course 32291).                                                        |
| (17) | <i>Using DCF with the 4250 Printer</i> , SR20-8486 (Training Course 32908).                                                                |
| (18) | <i>Using DCF with Page Printers</i> , SR21-1211 (Training Course 32243).                                                                   |
| (19) | <i>Document Composition Facility—Release 3 (SCRIPT/VS) for Document Administrators and Text Programmers</i> , SR20-7525 (Training Course). |
| (20) | <i>Document Composition Facility (SCRIPT/VS) Student Text</i> , SC20-1894 (Training Course).                                               |
| (21) | <i>Document Composition Facility: TSO Enhancements Update Guide</i> , G544-3345.                                                           |
| (22) | <i>Document Composition Facility: SCRIPT Mathematical Formula Formatter User's Guide</i> , S544-3306.                                      |
| (23) | <i>Document Composition Facility: Generalized Markup Language (GML) Applications User's Guide</i> , G544-3305.                             |
| (24) | <i>Document Composition Facility: MVS Program Directory</i> , G544-3669.                                                                   |
| (25) | <i>Program Directory for use with DCF and SMFF for VM</i> , G544-3670.                                                                     |
| (26) | <i>Document Composition Facility: VSE Program Directory</i> , G544-3671.                                                                   |
| (27) | <i>Document Composition Facility: ODF Program Directory for MVS</i> , G544-3687.                                                           |
| (28) | <i>Document Composition Facility: ODF Program Directory for VM</i> , G544-3686.                                                            |
| (29) | <i>Document Composition Facility: SMFF Program Directory for MVS</i> , G544-3685.                                                          |
| (30) | <i>Document Composition Facility: Double Byte User's Guide</i> , S544-3795                                                                 |
| (31) | <i>Document Composition Facility: Double Byte Program Directory for MVS</i>                                                                |
| (32) | <i>Document Composition Facility: Double Byte Program Directory for VM</i>                                                                 |

The following are also available:

- *Document Composition Facility: GML Starter Set Quick Reference*, SX26-3719.
- *Document Composition Facility: SCRIPT/VS Text Programmer's Quick Reference*, SX26-3723.
- *Document Composition Facility Post-Processor Examples*, S544-3484.
- *Document Composition Facility*, SH35-0086 (binder).

**Note:** The majority of DCF publications are available as machine-readable BookManager built BOOKS and as source files on the *Printing and Publishing Collection* CD-ROM, SK2T-2921.

## Related Publications

You should use the following publications to evaluate the use of DCF in different operating environments:

- *IBM Virtual Machine/System Product: Introduction*, GC19-6200.

This publication contains an introduction to CMS (Conversational Monitor System), which is one of the interactive systems that SCRIPT/VS operates with.

- *IBM Virtual Machine/System Product: Terminal User's Guide*, GC19-6206.

This publication describes the various terminal types supported by VM/SP for those who plan to use VM/SP in their operations.

- *OS/390 TSO/E User's Guide*, SC28-1968.

This publication gives detailed user information about OS/390 TSO (Time Sharing Option), which is one of the interactive systems that SCRIPT/VS operates with. It describes the TSO EDIT facility and related facilities for text entry, text editing, and data set management.

- *A Guide to IBM's Advanced Function Printing*, G544-3095.

This publication describes the use of a licensed program (PSF, DCF, GML, OGL, GDDM, and PMF) and the use of a subset of a licensed program in conjunction with the IBM Advanced Function Printing (AFP) printers, including the IBM 3800 Printing Subsystem Models 3 and 6 and the IBM 3820 Page Printer.

- *Advanced Function Printing Software: General Information*, G544-3415.

This publication defines Advanced Function Printing (AFP), describes the features and functions of the AFP licensed programs, and shows how the programs work together. It is intended for the people in your organization who plan for, install, use, and maintain IBM's AFP software products. It also contains a list of the AFP publications.

- *TSO Extensions Version 2 User's Guide*, SC28-1880

This publication gives detailed information on general TSO/E functions using data sets and running programs in TSO/E. It includes information on how to run foreground commands from a background job.

If you install DLF in the MVS environment, you need a copy of *OS/VS2 Access Method Services*, GC26-3841.

If you install DLF in the VSE environment, you need a copy of *VSE/VSAM Access Method Services: User's Guide and Reference*, SC24-5144.

## Related Publications

---

# Glossary

---

## Source Identifiers

This publication includes terms and definitions from:

- The *American National Dictionary for Information Processing Systems*, copyright 1982 by the Computer and Business Equipment Manufacturers Association (CBEMA). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *ISO Vocabulary—Information Processing* and the *ISO Vocabulary—Office Machines*, developed by the International Organization for Standardization, Technical Committee 97, Subcommittee 1. Definitions of published segments of the vocabularies are identified by the symbol (I) after the definition; definitions from draft international standards, draft proposals, and working papers in development by the ISO/TC97/SC1 vocabulary subcommittee are identified by the symbol (T) after the definition, indicating final agreement has not yet been reached among participating members.

---

## References

The following cross-references are used in this glossary:

### Deprecated term for

Indicates that the term should not be used (because it is obsolete, misleading, ambiguous, or jargonistic) and refers to the preferred term. For a deprecated term, the commentary contains only this reference; the deprecated term is not defined.

### Synonymous with

Appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning. The commentaries of the less desirable or less specific terms refer back to the preferred term with the *Synonym for* reference words.

### Synonym for

Appears in the commentary of a less desirable or less specific term and identifies the preferred term that has the same meaning.

### Contrast with

Refers to a term that has an opposite or substantively different meaning.

**See** Refers to a multiple-word term in which this term appears.

### See also

Refers to related terms that have similar (but not synonymous) meanings.

## A

**Adobe Document Structuring Conventions.** A standard subset of conventions that allow PostScript page descriptions to be accepted as input by many programs and resource managers.

**Adobe Font Metrics (AFM) files.** Files containing PostScript font information that SCRIPT/VS uses to format documents for PostScript devices.

**advanced function printing (AFP).** The ability of licensed programs to use the all-points-addressable concept to print text and images on a printer.

**alignment.** The horizontal placement of text in a column or cell.

**all-points addressability.** The capability to address, reference, and position text, overlays, and images at any defined point on the printable area of the paper. See *page device* and contrast with *line device*.

**all-points addressable (APA).** In computer printing, pertaining to the ability to address and print or not print each picture element (pel) on a page.

**alphameric.** Synonym for alphanumeric.

**alphanumeric.** Pertaining to a character set that contains letters, digits, and usually other characters, such as punctuation marks. (A) Synonymous with alphameric.

**alphanumeric character set.** A character set that contains both letters and digits and may contain control characters and special characters. (T) Synonymous with alphameric.

**alphanumeric character subset.** A character subset that contains both letters and digits and may contain control characters, special characters, and the space character. (I) (A) Synonymous with alphameric.

**alphanumeric string.** A sequence of characters consisting of the letters a through z and the numerals 0 through 9.

**ampersand.** The & character.

When an ampersand begins a character string, SCRIPT/VS assumes the character string is a symbol name. If the symbol name is defined, SCRIPT/VS replaces the symbol with its value (unless symbol substitution is off).

**APF.** Application processing function.

**application processing function (APF).** In GML processing, the processing that is performed when a document element or attribute is recognized. In SCRIPT/VS, an APF is implemented as a sequence of control words, possibly intermixed with text and symbols, in one of three forms: macro definition, value of a symbol, or imbedded file.

**application program interface (API).** The formally-defined programming language interface which is between an IBM system control program or a licensed program and the user of the program.

**ascender.** (1) In a font, the distance from the baseline to the top of the character. See *maximum ascender*. (2) The part of a lowercase letter that rises above the body of the letter. Letters with ascenders are b, d, f, h, k, l, and t.

**ASCII.** American National Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8-bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A) Contrast with *EBCDIC*.

**Note:** IBM has defined an extension to ASCII code (characters 128–255) that uses all 8 bits. DCF uses this 8-bit extension.

**ATMS-II.** Advanced Text Management System.

**ATMS-III.** Advanced Text Management System.

**attribute.** A characteristic of a document (or document element) other than its type or content. For example, the security level of a document or the depth of a figure.

**attribute label.** In GML markup, a name of an attribute that is entered in the source document when specifying the attribute's value.

## B

**back matter.** In a book, those sections such as glossary or an index that are placed after the main chapters or sections.

**balancing.** In multicolumn formatting, the process of making column depths on a page approximately equal

by redistributing the text in the columns. See also *vertical justification*.

**bar code.** A code representing characters by sets of parallel bars of varying thickness and separation that are read optically by transverse scanning.

**baseline.** In a font, the imaginary line on which the bottom of each successive character is aligned.

**basic document element.** In a general document, one of a group of elements that occurs frequently; for example: note, paragraph, and definition list.

**batch environment.** An environment in which noninteractive programs are executed.

**binding edge.** The edge of a page to be bound, stapled, or drilled. Defined with the BIND option of the SCRIPT command.

**body.** Of a printed page, the portion between the top and bottom margins that contains the text.

**boldface.** A heavy-faced type. Also, printing in this type.

**bottom margin.** On a page, the space between the body or the running footing, if any, and the bottom edge of the page.

**%%BoundingBox: comment.** An Adobe Document Structuring Conventions comment that contains integers describing the coordinates of the lower-left and upper-right corners of the bounding box in the default user coordinate system. The %%BoundingBox: comment coordinates are used by SCRIPT/VS to place the PostScript image in the space reserved within a DCF document.

**bounding box coordinates.** The coordinates of the lower-left and upper-right corners of an imaginary box surrounding an image in the default user coordinate system. The bounding box coordinates are used by SCRIPT/VS to place a PostScript image in the space reserved within a DCF document.

**break.** An interruption in the formatting of input lines so that the next input line is printed on a new output line.

**bug.** An error in a program or in a document markup.



## C

**call.** To transfer control to a procedure, program, routine, or subroutine.

**camera-ready copy.** Copy which is ready for photographic typesetting.

**caps.** Capital letters. See also *initial caps*.

**caption.** Text accompanying and describing an illustration.

**case-sensitive.** The relevance of a group of letters is uppercase or lowercase. ABC is different from Abc, which is different from ABc.

**CDPF.** Composed Document Printing Facility.

**cell.** A single unit within a table, in which text or other expressions may appear. A cell is always rectangular and is usually bounded by horizontal and vertical rules.

**centimeter (cm).** A measurement equal to one hundredth of a meter; 0.39 inch.

**character.** A member of a set of elements that is used for the representation, organization, or control of data. Characters may be letters, digits, punctuation marks, or other symbols.

**character arrangement table.** An array of data that translates input data into printable characters and identifies associated character sets and graphic character modification modules.

**character printer.** (1) The identification of characters by automatic means. (l) (A) (2) The identification of geographic, phonic, or other characters by various means, including magnetic, optical, or mechanical means. (A)

**character set.** A finite set of different characters that is complete for a given purpose. For example, in printing, the characters that constitute a font.

**character space.** The horizontal space of a character. This size depends on the character font and the device on which the character is printed.

**character spacing.** The space between characters in a word.

**cicero.** In the Didot point system, a unit of 0.1776 inch (4.512 millimeters) used in measuring typographical material.

**CMS.** Conversational monitor system—an interactive processor that operates within VM/370.

| **CMYK.** The color values cyan, magenta, yellow, and black used with the .CR [Color] control word.

**code page.** A font library member that gives the associated code points and character identifiers.

**code point.** A 1-byte code representing one of 256 potential characters.

**coded font.** (1) The combination of a code page and a font library. (2) A font library member that is fully described in terms of typeface, point-size, weight, width, and attribute.

**color separation.** The process of making separate masters of a document for color printing.

**column.** A vertical arrangement of characters or other expressions.

**column balancing.** The process of redistributing lines of text among a set of columns so that the amount of text in each column is as equal as possible.

**column width.** The width of each text column on a page. Specified with the .CW [Column Width] control word. (In multicolumn formatting, all columns on a page usually have the same width.)

**command.** A request from a terminal or a request specified in a batch processing job for the performance of an operation or the execution of a particular program. For example, a request given at a terminal for SCRIPT/VS to format a document or for an editor to edit a line of text.

**comment.** A control word line that is ignored by SCRIPT/VS. Such lines begin with either .\* or .cm.

**composed text.** Text that has been formatted and that contains control information to direct the presentation of the text on page printers.

**composite.** The act or result of formatting a document.

**composite rotation.** The total amount of rotation done by the printer to place text in the correct orientation on the page.

**composition.** The act or result of formatting a document.

**compositor.** A person or program that composes text.

**concatenation.** An operation that forms an output line containing as many words as the column width allows, by placing the first words from an input line after the last words from the preceding input line. When words from an input line would reach beyond the right margin

and hyphenation cannot be performed, they are placed at the beginning of the next output line, and so on.

**control word.** An instruction within a document that identifies its parts or indicates to SCRIPT/VS how to format the document. See also *macro*.

**control word line.** An input line that contains at least one control word.

**control word statement.** A control word and its parameters.

**Conversational Monitor System (CMS).** A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities, and operates only under control of the VM/370 VM control program.

**copy group.** A portion of a form definition that defines a set of modifications that can be used when printing a page.

**current left margin.** The left limit of a column that is in effect for formatting. Each column's left margin is specified with the .CD [Column Definition] control word. However, the current left margin (that is, the left boundary for an output line) might vary to the right of the column's left margin when indentation is changed with the .IN [Indent], .UN [Undent], .IL [Indent Line], and .OF [Offset] control words.

**current line.** The line in a source document at which a computer program (such as an editor or a formatter) is positioned for processing.

## D

**debug.** (1) To detect, diagnose, and eliminate errors in computer programs and SCRIPT/VS documents. (T)  
(2) Synonymous with checkout, troubleshoot.

**default value.** A value assumed by a computer program when a control word, command, or control statement with no parameters is processed. In GML processing, the value assumed for an attribute when none has been specified.

**deferred control words.** SCRIPT/VS control words that are processed after the text has been placed on the page.

**descender.** (1) In a font, the distance from the baseline to the bottom of the character. See *maximum descender*. (2) The part of a letter that falls below the body of the letter. Letters with descenders are g, j, p, q, y, and Q.

**destination.** (1) Any point or location, such as a node, station, or particular terminal, to which information is to

be sent. (2) An external logical unit LU or application program to which messages or other data are directed.

**dictionary.** A collection of *word stems* that is used with the spelling verification and automatic hyphenation functions.

**Didot point system.** A standard printer's measurement system on which type sizes are based. A Didot point is 0.0148 inch (0.376 millimeter). There are 12 Didot points to a cicero. See also *cicero* and *point*.

**document.** A publication or other written material. See also *output document* and *source document*.

**document administrator.** A person who is responsible for defining markup conventions and procedures for an organization.

**document conversion processor.** A computer program that processes a machine-readable document that includes formatting controls written in one formatter language, to produce a machine-readable document that includes formatting controls appropriate for another formatter language.

**document library.** A set of VSAM data sets, accessible in a batch environment, that contain documents and related files.

**dot leader.** A set of periods that fills in the space between two pieces of split text such as a chapter title and its page number in a table of contents.

**download.** To transfer data from a processing unit to an attached device such as a microcomputer for processing. Contrast with *upload*.

**duplex.** A mode of copying or printing on both sides of a sheet.

## E

**EBCDIC.** Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

**edit.** To add, change, delete, or rearrange data and to perform operations such as code conversion and zero suppression.

**editor.** (1) See *linkage editor*. (2) See *editor program*.

**editor program.** (1) A computer program designed to perform such functions as rearrangement, modification, and deletion of data in accordance with prescribed rules. (2) Contrast with *linkage editor*.

**eject.** In formatting, a skip to the next column or page.

**element.** Any part of a document: a single character, a word, or a sentence. Also refers to any part of a document you can identify with a GML tag (tagged element), such as a paragraph, figure, or heading.

**em.** A unit of measure equal to the width or the height of the character “m” in a particular font.

**en.** A unit of measure usually equal to one-half the width of an em. For many typefaces, the average width of lower case characters tends to be equal to the width of an en.

**enabled.** Used in reference to a tag, it means that the tag is mapped to its appropriate APF.

**Encapsulated PostScript.** Any file containing the PostScript Page Description Language that conforms to Adobe 2.0 Document Structuring Conventions and follows conventions defined by Adobe Systems to allow the file to be included by other applications. Encapsulated PostScript can be included in a DCF document by means of the .PO [PostScript] control word.

**epifile.** The portion of a profile (after a .EF control word) that is processed *after* the main document has been processed.

**escapement.** (1) Movement of one character space between the paper carrier and typing or printing position, parallel with the typing or printing line. (2) The unit of vertical or horizontal movement that is built into a device. For the 4250, that value is 1/600th of an inch; for the 3800 Model 3 and the 3820 Page Printer, that value is 1/240th of an inch; and for PostScript devices, that value is 1/72000th of an inch.

**exposure.** The amount of risk associated with a schedule item or items.

**extended symbol processing.** The processing of a symbol whose value causes the rest of the line to be stacked and later processed as a new input line.

## F

**factor.** A dimensionless scalar value used to form a product with another value. Factors can also be expressed as percentages.

**FBA.** Fixed-block-architecture.

**figure space.** (1) A unit of measure equal to the width of the “en” space in a particular font. (2) In the Document Composition Facility, the width of the figure (0).

**fill character.** A character used to occupy a space; for example, blanks used to fill up the space left by tabbing.

**float.** (1) A keep (group of input lines kept together) whose location in the source file can vary from its location in the printed document. (2) Of a keep, to be formatted in a location different from its location in the source file.

**flush.** Having no indentation.

**fold.** (1) To translate the lowercase characters of a character string into uppercase. (2) To place that portion of a line that does not fit within a column on the next output line.

**folio.** Page number.

**font.** A font library member that contains characters that must be used in conjunction with a code page font library member.

**font object.** A member of a font library. In CMS, a file whose filetype matches the name of the font library. In MVS, a member of a partitioned data set (PDS).

**font set.** The set of fonts to be used in formatting a source document.

**footer.** Text that appears at the bottom of every page of a document, for example, a page number.

**footing.** Words located at the bottom of the text area. See also *running footing*.

**footnote.** A note of reference, explanation, or comment, placed below the text of a column or page, but within the body of the page above the running footing.

**foreground.** The environment in which interactive programs are executed. Interactive processors reside in the foreground.

**format.** (1) The shape, size, and general makeup of a printed document. (2) To prepare a document for printing in a specified format.

**formatter.** (1) A computer program that prepares a source document for printing. (2) That part of SCRIPT/VS that formats input lines for a particular type of logical device.

**formatting mode.** In document formatting, the state in which input lines are concatenated and the resulting output lines are justified.

**FORMDEF.** Synonym for form definition.

**form definition (FORMDEF).** A resource object that defines the characteristics of the form which include: overlays to be used, text suppression, position of page data on the form, and number and modifications of a page. Synonymous with FORMDEF.

**front matter.** In a book, those sections (such as preface, abstract, table of contents, list of illustrations) that are placed before the main chapters or sections.

## G

**general document.** A type of document whose description can apply to a variety of documents, from memoranda to technical manuals. It can be used as a catch-all category for documents that do not conform to any other type description.

**Generalized Markup Language (GML).** A language that can be used to identify the parts of a source document without respect to a particular processing system.

**GML delimiter.** A special character that denotes the start of GML markup. In the starter set, it is initially a colon (:).

**GML end tag delimiter.** A special character that denotes the end of GML markup. In the starter set, it is initially a period (.).

**GML interpretation.** Recognizing the start or end of an element (or an attribute label), associating it with an APF, and executing the APF. In SCRIPT/VS, interpretation is performed jointly by SCRIPT/VS and by APFs.

**graphic character modification (GCM) module.** Modules that contain scan patterns of IBM-supplied character sets, user-defined character sets, or both without respect to particular processing.

**graphical data display manager (GDDM).** An IBM licensed program that creates page segments.

**gutter.** In multicolumn formatting, the space between columns.

## H

**hanging indentation.** The indentation of all lines of a block of text following the first line, which is not indented the same number of spaces. Specified with the .OF [Offset] or .UN [Undent] control word.

**heading.** Words located at the beginning of a chapter or section or at the top of a page, above the first line of text on the page. See also *head-level* and *running heading*.

**heading segment.** An element that begins with a heading, followed by basic document elements and lower-level heading segments.

**head-level.** The typeface and character size associated with the words standing at the beginning of a chapter or chapter topic.

**hexadecimal.** Pertaining to a numbering system based on 16, using the sixteen digits 0, 1, . . . 9, A, B, C, D, E, and F. For example, hexadecimal 1B equals decimal 27. See also *EBCDIC*.

**highlighting.** Emphasis associated with a document element. In formatting, highlighting is usually expressed by changing a font, overstriking, underscoring, or capitalizing the highlighted element.

**horizontal justification.** The redistribution of horizontal white space at the end of a line of text to the spaces between the words and letters of the line in order to exactly fill the width of the column with the text.

## I

**IEBIMAGE.** A utility program that creates and maintains various 3800 Printing Subsystem Model 1 modules (for example, character arrangement table and graphic character modification (modules) and stores them in SYS1.IMAGELIB.

**image.** (1) A likeness or imitation of an object, such as a picture or logo. (2) A PostScript file that can contain any combination of images and text.

**impact printer.** A printer, in which printing is the result of mechanical impacts. (I) (A)

**implied paragraph structure.** An element that begins with an implied paragraph; that is, one for which you do not specifically enter a paragraph tag. The existence of the paragraph is understood from the existence of the implied paragraph structure, for example, as with notes, figure captions, and lists.

**indent.** To set typographical material to the right of the left margin, while still retaining the original (fixed) margin settings.

**indentation.** The action of indenting. The condition of being indented. The blank space produced by indenting. Specified with the .IN [Indent], .IR [Indent Right], .UN [Undent], .OF [Offset], and .IL [Indent Line] control words. See also *hanging indentation*.

**initial value.** A value assumed by SCRIPT/VS for a formatting function until the value is explicitly changed with a control word. The *initial value* is assumed even before the control word is encountered, whereas the

*default* value is assumed when the control word is issued without parameters. See also *default value*.

**initialize.** (1) In programming languages, to give a value to a data object at the beginning of its lifetime. (2) To set counters, switches, addresses, or contents of storage to zero or other starting values at the beginning of, or at prescribed points in, the operation of a computer routine.

**inline space.** The amount of horizontal white space that usually occurs between words in a line.

**input device.** Synonym for input unit.

**input line.** A line, as entered into a source file, to be processed by a formatter.

**input unit.** A device in a data processing system by means of which data can be entered into the system. (I) (A) Synonymous with input device.

**interactive.** Pertaining to an application in which entries call forth a response from a system or program, as in an inquiry system. An interactive system might also be conversational, implying a continuous dialog between user and system. Interactive systems are usually communicated with via terminals, and they respond to commands. See also *foreground*.

**interactive environment.** The environment in which an interactive processor operates.

**Interactive System Productivity Facility (ISPF).** A dialog manager for interactive applications that provides control and services to allow processing of the dialogs in different host environments.

**intercharacter space.** Extra horizontal white space inserted *between* characters of a word. This space is in addition to the space included as part of the characters by the designer of the font.

**interword space.** See *word space*.

**ISPF.** Interactive System Productivity Facility.

**italic.** A type style with characters that slant to the right.

## J

**job control language (JCL).** A control language used to identify a job or describe its requirements to the operating system.

**job control statement (JCS).** A statement in a job that is used in identifying the job or describing its requirements to the operating system.

**justification.** See *horizontal justification* and *vertical justification*.

**justify.** To control the printing positions of characters on a page so that the left-hand and right-hand margins of the printing are regular. (I) (A) See *left-justify* and *right-justify*.

## K

**kanji.** The nonphonetic Japanese writing system. In a font representing kanji characters, each character is represented by a double-byte code. Contrast with *katakana*.

**katakana.** A character set consisting of symbols used in one of the two common Japanese phonetic alphabets. Each character is represented by one byte. Contrast with *kanji*.

**keep.** In a source document, a collection of lines of text to be printed in the same column. When the vertical space remaining in the current column is insufficient for the block of text, the text is printed in the next column. In the case of single-column format, the next column is on the next page.

## L

**layout.** The arrangement of matter to be printed. See also *format*.

**leader.** (1) Dots or hyphens (as in a table of contents) used to lead the eye horizontally. (2) The divider between text and footnotes on a page, usually a short line of dashes.

**left-hand page.** The page on the left when a book is opened; usually even-numbered.

**left-justify.** To control the printing positions of characters on a page so that the left-hand margin of the printing is regular. (I) (A)

**legend.** An explanatory list of the symbols, lines, and other components of a schedule.

**ligature.** A single character (piece of type or font raster) that represents two or more input characters: *ff* and *ffi* are examples of characters that may be represented by (printed as) a ligature.

**line device.** Any of a class of printers that accept one line of text from the host system at a time. SCRIPT/VS supports such line devices as the 1403 Printer and 3800 Model 1.

**line printer.** (1) A device that prints a line of characters as a unit. (l) (A) (2) Contrast with character printer, page printer.

**line space.** The vertical distance between the baseline of the current line and the baseline of the preceding line.

**line spacing.** See *line space*.

**logical output device.** The combination of a physical output device and such logical variables as page size and number of lines per vertical inch (for line devices). A specification of 1403W6 is an example of a logical output device.

**logical page.** Synonym for page.

**lowercase.** Pertaining to small letters as distinguished from capitals, for example: a, b, g rather than A, B, G.

## M

**machine-readable.** Pertaining to data a machine can acquire or interpret (read) from a storage device, or a data medium, or other source.

**maclib.** See *macro library*

**MACLIB library.** A library that contains macros, copy files, or source program statements for use under CMS.

**macro.** See *macro instruction*

**macro instruction.** (1) An instruction that when executed causes the execution of a predefined sequence of instructions in the same source language. In SCRIPT/VS, a macro is a sequence of one or more control words, symbols, and input lines. A macro's definition can be recursive. (2) Synonymous with macrostatement.

**macro library.** A library of macro definitions used during macro expansion. The form the library takes will vary by environment, being a MACLIB in CMS, a PDS in TSO, and so on.

**macrostatement.** Synonym for macro instruction.

**macro substitution.** During formatting, the substitution of control words, symbols, and text for a macro.

**map.** To associate a tag with an APF, using the .AA [Associate APF] control word.

**margin.** (1) The space above, below, and on either side of the body of a page. (2) The left or right limit of a column.

**mark up.** (1) To determine the markup for a document. (2) To insert markup into a source document.

**markup.** Information added to a document that enables a person or system to process it. Markup information can describe the document's characteristics, or it can specify the actual processing to be performed. In SCRIPT/VS, markup consists of GML tags, attribute labels and values, and control words.

**markup-content separator.** A delimiter used in GML markup that indicates the end of the markup and the beginning of the text. The default markup content separator for GML is a period (.).

**maximum ascender.** The maximum height from the baseline of sequential characters to the top mark of the tallest character in a font character set.

**maximum descender.** The maximum depth from the baseline of sequential characters to the bottom mark of any character in the font character set.

**MCS.** Markup/content separator.

**meter (m).** 1.0936 yards; 3.2808 feet; 39.3696 inches.

**millimeter (mm).** One thousandth of a meter; 0.04 inch.

## N

**National Language Support.** Translation requirements affecting parts of devices and licensed programs; for example, rules for translation of message text, and for conversion of symbols such as the US dollar sign to the UK pound sign.

**nonimpact printer.** A printer, such as the 3800 Printing Subsystem, in which printing is not the result of mechanical impacts, but is instead produced by another process such as laser beam, ink jet, or electroerosion. The 3800 Printing Subsystem, for example, uses a laser based technology, and the 4250 Printer uses an electroerosion process.

## O

**object.** A sequential collection of control records that represents documents, pages, fonts, and so on.

**offset.** (1) To indent all lines of a block of text, except the first line. (2) The indentation of all lines of a block of text following the first line.

**option.** Information entered with a SCRIPT command to control the execution of SCRIPT/VS.

**orientation.** In the 3800 Printing Subsystem Models 3 and 8, the number of degrees an object is rotated relative to a reference; for example, the orientation of printing on a page, relative to the page coordinates. See also *text orientation*.

**output device.** Synonym for output unit.

**output document.** A machine-readable collection of lines of text or images that have been formatted, or otherwise processed, by a document processor. The output document can be printed or it can be filed for future processing.

**output line.** A line of text produced by a formatter.

**output unit.** A device in a data processing system by which data can be received from the system. (I) (A) Synonymous with output device.

**overlay.** A collection of predefined data such as lines, shading, text, boxes, or logos that can be merged with variable data on a page while printing.

## P

**page.** A collection of data that can be printed on a physical sheet of paper. Synonymous with logical page.

**PAGEDEF.** Page definition.

**page definition (PAGEDEF).** (1) A resource, specified in the print data set JCL, that defines the rules for transforming the input to pages and text controls. (2) In the 3800 Print Management Facility, a member of a partitioned data set that contains the formatting instructions for a print data set, although it can be used for any compatible print data set.

**page device.** A device that prints a formatted page that has graphics and text merged.

**page printer.** (1) Any of a class of devices that accept composed pages, constructed of composed text and images, and prints one page as a unit. SCRIPT/VS supports such page printers as the 4250 Printer, the 3800 Model 3, and the 3820 Page Printer. (2) Contrast with *character printer*, *line printer*.

**page segment.** (1) An object that can contain text and images and be included on any addressable point on a page or electronic overlay. It assumes the environment of the object it is included in. (2) A library member that contains the definition of the page segment.

**paginate.** To number pages.

**paragraph unit.** An element that has the same structure as a paragraph. In a general document, the

paragraph units are: paragraph, note, and paragraph continuation.

**parameter.** (1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) The syntax of some SCRIPT/VS control words includes parameters, which establish the properties of a formatting function or a printed page. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted.

**part.** In a general document, a part is a zero-level heading segment. See also *heading segment*.

**patch PSC element.** A PSC element that is used temporarily to modify the normal output.

**pel.** (1) An element of a raster pattern about which a toned area on a photoconductor can appear. See also *raster pattern*. (2) The unit of horizontal measurement for the 3800 Printing Subsystem and the 4250 Printer. On the 3800 Printing Subsystem Model 1, one pel equals approximately 1/180th inch. On the 3800 Model 3 and the 3820 Page Printer, one pel equals approximately 1/240th inch. On the 4250 Printer, one pel equals approximately 1/600th inch.

**pel density.** The number of picture elements per inch of linear measurement.

**physical output device.** A physical device, that stores, prints, or displays data, such as a terminal, a disk file, a line printer, or a nonimpact printer. The 1403 is an example of a physical output device.

**pica.** A unit of about 1/6 inch used in measuring typographical material. It is similar to a cicero in the Didot point system.

**pitch.** On a typewriter, the distance between corresponding points of two equal characters that are typed immediately adjacent to one another.

**point.** (1) A unit of about 1/72 of an inch used in measuring typographical material. There are 12 points to the pica. (2) In the Didot point system, a unit of 0.0147 inch. There are 12 Didot points to the cicero.

**PostScript devices.** Any of a class of devices that accept the PostScript page description language, such as the IBM 4019 LaserPrinter. When used with DCF, PostScript devices must be configured to accept 8-bit ASCII.

**PostScript language.** A programming language designed to convey a description of virtually any desired page to a printer. It possesses a wide range of graphic operators that may be combined in any manner.

**PostScript image file.** Any file containing encapsulated PostScript that is imbedded in a DCF document by means of the .PO [PostScript] control word. PostScript image files can include any combination of images or text.

**profile.** (1) In SCRIPT/VS processing, a file that is imbedded before the primary file is processed. It can be used to control the formatting of a class of source documents. When processing GML markup, the profile usually contains the mapping from GML to APFs and the symbol settings that define the formatting style. (2) In the DLF library, a collection of information that identifies a batch SCRIPT/VS user (user profile) or a document processor (attribute profile) or that defines certain library parameters (system profile).

**proportional spacing.** The spacing of characters in a printed line so that each character is allotted a space proportional to the character's width.

## R

**ragged right.** Pertaining to text that is not right-justified. See also *left-justify*.

**ragged left.** Pertaining to text that is not left-justified. See also *right-justify*.

**reference element.** In a general document, an element whose content is a reference to another element that is generated by an APF. There are five: figure reference, footnote reference, heading reference, index entry reference, and list item reference.

**required blank.** A character that prints as a blank, but does not act as a word separator.

**required space.** In word processing, a space or blank that must not be removed when adjusting a line or paragraph of text. A character that prints as a blank, but does not act as a word separator. Synonym for required blank.

**residual text.** The line of text following the markup/content separator of a GML tag.

**RFTDCA.** Revisable-Form-Text Document Content Architecture specifies how IBM office systems interchange documents that are in revisable form. This architecture defines the structure of the data streams that represent revisable-form-text documents within the office system or network.

| **RGB.** The color values red, green, and blue used with  
| the .CR [Color] control word.

**right-hand page.** The page on the right when a book is opened; usually odd-numbered.

**right-justify.** To control the positions of characters on a page so that the right-hand margin of the printing is regular. (I) (A)

**row.** A horizontal arrangement of characters or other expressions on a printed page.

**rule.** (1) A straight horizontal or vertical line used, for example, to separate or border the parts of a figure or box. (2) A solid or patterned line of any weight, extending horizontally across the row or vertically down the column.

**running footing.** A footing that is repeated above the bottom margin area on consecutive pages or on consecutive odd-numbered or even-numbered pages in the text area of the page. Synonymous with footer.

**running heading.** A heading that is repeated below the top margin area on consecutive pages or on consecutive odd-numbered or even-numbered pages in the text area of the page.

## S

**SCRIPT/VS.** The formatter component of the Document Composition Facility. SCRIPT/VS provides capabilities for text formatting and document management, macro processing and symbol substitution, and GML tag recognition and processing.

**section.** Each part of the output page when an output page has two or more single-column parts with the same or different column-widths, or a single-column part and a multicolumn part, or two or more different multicolumn parts.

**segment.** An object containing composed text and images, prepared before formatting and included in a document when it is printed.

**separation masters.** The process of making separate masters of a document for the purpose of special printing (such as multi-part forms or multiple-color printing).

**set.** Used in reference to a symbol; it implies the .SE [Set Symbol] control word.

**set size.** The set size of a given typeface determines the number of characters that will fit in a line of a given width when it is printed or set.

**shading.** Highlighting an area on the page by varying graded density.

**showpage.** A PostScript command that transmits the current page to the current output device, causing any marks on the page to actually appear.



**slip end date.** The end date of a schedule or project that has moved or *slipped* to a later time than originally planned.

**slip start date.** The start date of a schedule or project that has moved or *slipped* to a later time than originally planned.

**small caps.** Capital letters in the same style as the normal capital letters in a font, but approximately the size of the lowercase letters.

**source document.** A machine-readable collection of lines of text or images that is used for input to a computer program.

**space.** A blank area separating words or lines.

**space unit.** A unit of measure of horizontal or vertical space. In GML markup, the em is used when a measure that is relative to the current font size is required. When an absolute measure is required, as in specifying the depth of a figure, recommended space units are inches (nnI), millimeters (nnW), picas/points (nnPnn), or Ciceros/Didot points (nnCnn), where nn is the number of units. See also *em*, *pica*, *point*, *Cicero*, and *Didot point system*.

**starter set.** An example of GML support that is provided with the Document Composition Facility. It consists of a document-type description for general documents, a profile, and a library of APFs.

**string.** A sequence of elements of the same nature, such as characters considered as a whole.

**structure.** A characteristic of a document (or element) that expresses the type and relationship of the elements of the content. See also *content* and *element*.

**structured field.** A self-identifying string of bytes, analogous to a logical record. A structured field consists of an introducer, which identifies and characterizes the structured field, and data or parameters.

**symbol.** A name in a source document that can be replaced with something else. In SCRIPT/VS, a symbol is replaced with a character string. SCRIPT/VS can interpret the character string as a number, a character string, a control word, or another symbol.

**symbol substitution.** During formatting, the replacement of a symbol with a character string that SCRIPT/VS can interpret as a value (numeric, character string, or control word) or as another symbol.

**SYSVAR.** An option of the SCRIPT command that permits the user to specify values for symbols. In the

starter set, SYSVAR symbol values determine whether certain processing variations will occur, such as heading numbering, duplex formatting, and two-column printing.

## T

**tab.** (1) (noun) A preset point in the typing line of a typewriter-like terminal. A preset point in an output line. (2) (noun) A tab character, X'05'. (3) (verb) To advance to a tab for printing or typing.

**table.** (1) An array of data each item of which can be unambiguously identified by means of one or more arguments. (I) (A) (2) An arrangement of cells in rows and columns.

**tag.** In GML markup, a name for a type of document or document element that is entered in the source document to identify it. For example, *.p.* is the tag used to identify a paragraph.

**terminal.** A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

**text item.** Explicitly marked elements that occur within text, such as within a paragraph unit. In a general document, for example, quotations and phrases are text items.

**text orientation.** A description of the appearance of text as a combination of print direction and character rotation.

**text programmer.** One who implements APFs that provide the processing specified by the document administrator. In SCRIPT/VS, this involves writing SCRIPT/VS macros and organizing macro libraries and profile files so that the appropriate composition will be done for each tag.

**text line.** A line that contains only text.

**text variable.** A symbol whose final value is to be treated only as text.

**time sharing option (TSO).** An option on the operating system; for System/370, the option provides interactive time sharing from remote terminals.

**token.** A string of characters that is treated as a single entity. In SCRIPT/VS, a parameter passed to a macro in one of the local variables &\*1, ... &\*n.

**top margin.** On a page, the space between the body or running heading and the top edge of the page.

**transparency.** A master or copy on material that transmits light without diffusion.

**translation table.** A table that provides replacement characters for characters that cannot be printed or that substitutes characters from a different font. For example, the 256-byte portion of the character arrangement table that translates the user's data code for a character recognizable by the 3800 Printing Subsystem Model 1.

**TRC.** Table reference character (TRC) in printer SYSOUT data sets, a second control byte, following the carriage control byte, which indicates which font the record is to be printed in. The presence of TRCs is indicated by the JCL parameter DCB=OPTCD=J.

**TSO.** Time-sharing option.

**typeface.** All type of a single style. There might be several fonts (different sizes) with the same typeface or style.

**typeface family.** A collection of fonts of a common typeface that vary in size and style.

**typeset.** (1) To arrange the type on a page for printing. (2) Pertaining to material that has been set in type.

**type posture.** A typeface style variation indicating whether a typeface is upright (as in Roman) or slanted to the right (as in italic or cursive).

**type size.** A measurement in pitch or points of the height and width of a graphic character in a font. For example, the vertical height (point size) of a given typeface, such as 10 point.

**type style.** The form of characters within a set of the same font, for example: elite, or pica. (T) Attributes such as posture, weight, and width may vary in a type style.

**type weight.** (1) The degree of boldness of a typeface series, caused by different thicknesses of the strokes that form a graphic character. (2) One of the many attributes of a font, others, for example: being size and typeface.

**type width.** The horizontal size (set size) of a given typeface. The width may be given in units of measurement, such as set 9 point, or it may be descriptive: ultra condensed, condensed, expanded, and so on.

## U

**underscore.** (1) A line printed under one or more characters. (2) To place a line under one or more characters; to underline.

**unformatted mode.** (1) In document formatting, the state in which each input line is processed and printed without formatting. Other SCRIPT/VS control words remain in effect and are recognized. (2) In document printing, using the UNFORMAT option, the state in which each input line (control words as well as text) is printed as it exists in the input, in the order in which it is processed. No formatting is done.

**unique identifier (ID).** In a general document, an attribute whose value serves as a name that can be used to refer to the element. See also *reference element*.

**unit space.** The minimum amount of additional spacing acceptable for purposes of horizontal justification, as specified by the font designer.

**upload.** To transfer data from a device such as a microcomputer to a processing unit. Contrast with *download*.

**uppercase.** Pertaining to capital letters as distinguished from small letters, for example: A, B, G rather than a, b, g.

## V

**variable.** A quantity that can assume any of a given set of values.

**variable text.** For the .VT [Variable Text] control word, text to be inserted in a formatted document by a postprocessor.

**vertical justification.** Redistribution of the extra vertical white space at the end of a column between lines of text, so as to make the columns appear to be the same length.

## W

**widow.** A line or word by itself ending a paragraph.

**word space.** The white space placed between words in a line. This is sometimes referred to as an interword space.

**word spacing.** The space between words in a line. See also *word space*.

**writable character generation module (WCGM).** In the 3800 Printing Subsystem, a 64-position portion of character generation storage that holds the scan elements of a single character set. There are two WCGMs in the basic 3800, and optional added storage provides two more.

## **X**

**XPO.** See *exposure*



---

# Index

## Special Characters

- ... [Set Label] control word
  - specifying with the control word 60
- .AA [Associate APF] control word
  - specifying with the control word 62
- .AM [Adjust Margin] control word
  - specifying with the control word 64
- .AN [And] control word
  - specifying with the control word 66
- .AP [Append] control word
  - effect of UNFORMAT option on 48
  - specifying with the control word 68
- .AR [Area] control word
  - specifying with the control word 70
- .BC [Balance Columns] control word
  - specifying with the control word 74
- .BF [Begin Font] control word
  - specifying with the control word 75
- .BL [Blank] control word
  - specifying with the control word 78
- .BM [Bottom Margin] control word
  - specifying with the control word 79
- .BR [Break] control word
  - specifying with the control word 80
- .BX [Box] control word
  - specifying with the control word 81
- .CB [Column Begin] control word
  - specifying with the control word 91
- .CC [Conditional Column Begin] control word
  - specifying with the control word 92
- .CD [Column Definition] control word
  - specifying with the control word 93
- .CE [Center] control word
  - specifying with the control word 95
- .CG [Copy Group] control word
  - specifying with the control word 97
- .CL [Column Line Length] control word
  - specifying with the control word 99
- .CM [Comment] control word
  - specifying with the control word 100
- .CO [Concatenate Mode] control word
  - specifying with the control word 462
- .CP [Conditional Page Eject] control word
  - specifying with the control word 101
- .CR [Color] control word
  - specifying with the control word 102
- .CS [Conditional Section] control word
  - effect of UNFORMAT option on 47
  - specifying with the control word 105
- .CT [Continued Text] control word
  - specifying with the control word 107
- .CW [Control Word Separator] control word
  - specifying with the control word 463
- .DA [Define Area] control word
  - specifying with the control word 108
- .DC [Define Character] control word
  - specifying with the control word 112
- .DF [Define Font] control word
  - specifying with the control word 130
- .DH [Define Head Level] control word
  - specifying with the control word 136
- .DI [Delay Imbed] control word
  - specifying with the control word 466
- .DL [Dictionary List] control word
  - specifying with the control word 140
- .DM [Define Macro] control word
  - defining a macro with 145
  - specifying with the control word 142
  - using inline form of 144
  - using subscripted form of 144
- .DR [Define Rule] control word
  - specifying with the control word 151
- .DS [Double Space Mode] control word
  - specifying with the control word 154
- .DU [Dictionary Update] control word
  - ADD parameter of 155
  - DEL parameter of 155
  - specifying with the control word 155
- .DV [Define Variables] control word
  - specifying with the control word 157
- .EC [Execute Control] control word
  - specifying with the control word 161
- .EF [End of File] control word
  - providing epifile facility with 41
  - specifying with the control word 163
- .EL [Else] control word
  - specifying with the control word 164
  - using for alternative processing 164
  - using in macros 164
  - using with .IF 164
- .EM [Execute Macro] control word
  - specifying with the control word 166
- .EQ [Equation] control word
  - specifying with the control word 169
- .ES [Extra Space] control word
  - specifying with the control word 167
- .EZ [EasySCRIPT] control word
  - specifying with the control word 468
- .FL [Float] control word
  - specifying with the control word 170
- .FM [Footing Margin] control word
  - specifying with the control word 470

|                                                   |          |  |
|---------------------------------------------------|----------|--|
| .FN [Footnote] control word                       |          |  |
| effect of UNFORMAT option on                      | 47       |  |
| specifying with the control word                  | 172      |  |
| .FO [Format Mode] control word                    |          |  |
| specifying with the control word                  | 174      |  |
| .FS [Footing Space] control word                  |          |  |
| specifying with the control word                  | 471      |  |
| .FV [Format Vertically] control word              |          |  |
| specifying with the control word                  | 177      |  |
| .GO [Goto] control word                           |          |  |
| effect of UNFORMAT option on                      | 47       |  |
| specifying with the control word                  | 179      |  |
| using to branch to files or macros                | 179      |  |
| .GR [Group] control word                          |          |  |
| specifying with the control word                  | 181      |  |
| .GS [GML Services] control word                   |          |  |
| specifying with the control word                  | 183      |  |
| .H0 [Heading Level 0] control word                |          |  |
| specifying with the control word                  | 190      |  |
| .HM [Heading Margin] control word                 |          |  |
| specifying with the control word                  | 472      |  |
| .HR [Horizontal Rule] control word                |          |  |
| specifying with the control word                  | 192      |  |
| .HS [Heading Space] control word                  |          |  |
| specifying with the control word                  | 473      |  |
| .HW [Hyphenate Word] control word                 |          |  |
| specifying with the control word                  | 195      |  |
| .HW [Hyphenate] control word                      |          |  |
| specifying with the control word                  | 196      |  |
| .IC [Intercharacter Spacing] control word         |          |  |
| specifying with the control word                  | 201      |  |
| .IE [Index Entry] control word                    |          |  |
| specifying with the control word                  | 202      |  |
| .IF [If] control word                             |          |  |
| specifying with the control word                  | 203      |  |
| using for conditional processing                  | 203      |  |
| using SYSOUT comparand with                       | 203      |  |
| using SYSPAGE comparand with                      | 203      |  |
| .IF [If] control word.                            |          |  |
| using with substitution off                       | 204      |  |
| .IL [Indent Line] control word                    |          |  |
| specifying with the control word                  | 206      |  |
| .IM [Imbed] control word                          |          |  |
| effect of UNFORMAT option on                      | 48       |  |
| specifying with the control word                  | 208      |  |
| .IN [Indent] control word                         |          |  |
| specifying with the control word                  | 210      |  |
| .IR [Indent Right] control word                   |          |  |
| specifying with the control word                  | 217      |  |
| .IS [Inline Space. control word                   |          |  |
| specifying with the control word                  | 219      |  |
| .IT [Input Trace] control word                    |          |  |
| specifying with the control word                  | 221      |  |
| .IX [Index] control word                          |          |  |
| specifying with the control word                  | 224      |  |
| .JU [Justify Mode] control word                   |          |  |
| specifying with the control word                  | 226      |  |
| .KP [Keep] control word                           |          |  |
| specifying with the control word                  | 227      |  |
| .LB [Leading Blank] control word                  |          |  |
| specifying with the control word                  | 230      |  |
| .LI [Literal] control word                        |          |  |
| specifying with the control word                  | 231      |  |
| .LL [Line Length] control word                    |          |  |
| specifying with the control word                  | 233      |  |
| used in setting value of .CL [Column Line Length] |          |  |
| control word                                      | 233      |  |
| used with running headings and footings, and      |          |  |
| footnotes                                         | 233      |  |
| .LO [Lead-Out] control word                       |          |  |
| specifying with the control word                  | 234      |  |
| .LS [Line Spacing] control word                   |          |  |
| specifying with the control word                  | 235      |  |
| .LT [Leading Tab] control word                    |          |  |
| specifying with the control word                  | 242      |  |
| .LY [Library] control word                        |          |  |
| MAC parameter of                                  | 243      |  |
| specifying with the control word                  | 243      |  |
| SYM parameter of                                  | 243      |  |
| .MC [Multicolumn Mode] control word               |          |  |
| specifying with the control word                  | 244      |  |
| .ME [Macro Exit] control word                     |          |  |
| specifying with the control word                  | 245      |  |
| .MG [Message] control word                        |          |  |
| specifying with the control word                  | 246      |  |
| .MS [Macro Substitution] control word             |          |  |
| specifying with the control word                  | 248      |  |
| .NF [No Formatting] control word                  |          |  |
| specifying with the control word                  | 249      |  |
| .NL [Null Line] control word                      |          |  |
| specifying with the control word                  | 252      |  |
| .OC [Output Comment] control word                 |          |  |
| specifying with the control word                  | 258      |  |
| .OF [Offset] control word                         |          |  |
| specifying with the control word                  | 260      |  |
| .OI [Overlay Include] control word                |          |  |
| specifying with the .OI [Overlay Include] control |          |  |
| word                                              | 262      |  |
| specifying with the control word                  | 262      |  |
| .OR [Or] control word                             |          |  |
| specifying with the control word                  | 274      |  |
| using to check multiple conditions                | 274      |  |
| using with .IF                                    | 274      |  |
| .PA [Page Eject] control word                     |          |  |
| EVEN parameter of                                 | 276      |  |
| NOSTART parameter of                              | 276      |  |
| ODD parameter of                                  | 276      |  |
| specifying with the control word                  | 276, 278 |  |
| using to reset page numbers                       | 276      |  |
| using to start a new page                         | 276      |  |

|                                                  |     |  |
|--------------------------------------------------|-----|--|
| .PI [Put Index] control word                     |     |  |
| specifying with the control word                 | 279 |  |
| .PL [Page Length] control word                   |     |  |
| specifying with the control word                 | 282 |  |
| .PM [Page Margins] control word                  |     |  |
| specifying with                                  | 284 |  |
| .PN [Page Numbering Mode] control word           |     |  |
| specifying with the control word                 | 286 |  |
| .PO [PostScript] control word                    |     |  |
| specifying with the control word                 | 289 |  |
| .PP [Paragraph Start] control word               |     |  |
| specifying with the control word                 | 474 |  |
| .PS [Page Number Symbol] control word            |     |  |
| specifying with the control word                 | 475 |  |
| .PT [Put Table of Contents] control word         |     |  |
| specifying with the control word                 | 294 |  |
| .PW [Page Width] control word                    |     |  |
| specifying with the control word                 | 296 |  |
| .PX [Prefix] control word                        |     |  |
| specifying with the control word                 | 297 |  |
| .QQ [Quick Quit] control word                    |     |  |
| specifying with the control word                 | 299 |  |
| .QU [Quit] control word                          |     |  |
| specifying with the control word                 | 300 |  |
| .RC [Revision Code] control word                 |     |  |
| specifying with the control word                 | 301 |  |
| .RD [Read Terminal] control word                 |     |  |
| specifying with the control word                 | 308 |  |
| .RE [Restore Environment] control word           |     |  |
| specifying with the control word                 | 309 |  |
| .RF [Running Footing] control word               |     |  |
| effect of UNFORMAT option on                     | 47  |  |
| specifying with the control word                 | 310 |  |
| .RH [Running Heading] control word               |     |  |
| effect of UNFORMAT option on                     | 47  |  |
| specifying with the control word                 | 314 |  |
| .RI [Right Adjust] control word                  |     |  |
| specifying with the control word                 | 318 |  |
| .RN [Reference Numbers] control word             |     |  |
| specifying with the control word                 | 320 |  |
| .RT [Running Title] control word                 |     |  |
| specifying with .RT [Running Title] control word | 476 |  |
| .RV [Read Variable] control word                 |     |  |
| setting symbols with                             | 321 |  |
| .SA [Save Environment] control word              |     |  |
| specifying with the control word                 | 323 |  |
| .SB [Shift Baseline] control word                |     |  |
| specifying with the control word                 | 325 |  |
| .SC [Single Column Mode] control word            |     |  |
| specifying with the control word                 | 327 |  |
| .SD [Shading Definition] control word            |     |  |
| specifying with the control word                 | 328 |  |
| .SE [Set Symbol] control word                    |     |  |
| specifying with the control word                 | 337 |  |
| SUBSTR                                           | 337 |  |
| .SI [Segment Include] control word               |     |  |
| specifying with the control word                 | 343 |  |
| .SK [Skip] control word                          |     |  |
| specifying with the control word                 | 351 |  |
| .SL [Set Line Space] control word                |     |  |
| specifying with the control word                 | 478 |  |
| .SM [Separation Master] control word             |     |  |
| specifying with the control word                 | 353 |  |
| .SO [STAIRS/VS Output] control word              |     |  |
| specifying with the control word                 | 354 |  |
| .SP [Space] control word                         |     |  |
| specifying with the control word                 | 356 |  |
| .SS [Single Space Mode] control word             |     |  |
| specifying with the control word                 | 358 |  |
| .SU [Substitute Symbol] control word             |     |  |
| specifying with the control word                 | 359 |  |
| .SV [Spelling Verification] control word         |     |  |
| specifying with the control word                 | 361 |  |
| .SX [Split Text] control word                    |     |  |
| specifying with the control word                 | 363 |  |
| .TA [Table] control word                         |     |  |
| specifying with the control word                 | 365 |  |
| .TB [Tab Setting] control word                   |     |  |
| specifying with the control word                 | 374 |  |
| .TC [Table of Contents] control word             |     |  |
| specifying with the control word                 |     |  |
| .TD [Table Definition] control word              |     |  |
| specifying with the control word                 | 379 |  |
| .TE [Terminal Input] control word                |     |  |
| effect of TWOPASS option on                      | 47  |  |
| specifying with the control word                 | 393 |  |
| .TH [Then] control word                          |     |  |
| specifying with the control word                 | 395 |  |
| .TI [Translate Input] control word               |     |  |
| specifying with the control word                 | 397 |  |
| .TM [Top Margin] control word                    |     |  |
| specifying with the control word                 | 399 |  |
| .TP [Tab Position] control word                  |     |  |
| specifying with the control word                 | 400 |  |
| .TR [Translate Character] control word           |     |  |
| specifying with the control word                 | 404 |  |
| .TS [Translate String] control word              |     |  |
| specifying with the control word                 | 406 |  |
| .TU [Translate Uppercase] control word           |     |  |
| specifying with the control word                 | 408 |  |
| .TY [Type on Terminal] control word              |     |  |
| specifying with the control word                 | 411 |  |
| .UC [Underscore and Capitalize] control word     |     |  |
| specifying with the control word                 | 412 |  |
| .UD [Underscore Definition] control word         |     |  |
| specifying with the control word                 | 414 |  |
| .UN [Undent] control word                        |     |  |
| specifying with the control word                 | 416 |  |
| .UP [Uppercase] control word                     |     |  |
| specifying with the control word                 | 418 |  |

- .US [Underscore] control word
  - specifying with the control word 420
- .UW [Unverified Word] control word
  - specifying with the control word 422
- .VR [Vertical Rule] control word
  - specifying with the control word 423
- .VT [Variable Text] control word
  - specifying with the control word 426
- .WF [Write to File] control word
  - ERASE parameter of 428
  - IMBED parameter of 428
  - specifying with the control word 428
  - TAG parameter of 428
- .WS [Word Space] control word
  - specifying with the control word 431
- .WZ [Widow Zone] control word
  - specifying with the control word 434
- '
  - See control word modifier
- \* 100
  - (semicolon)
    - See control word separator
- @user-option 48
  - description 48
- &\$PASS symbol 35
- &\$PSNO symbol 35
- &SD' 347
- &SW' 347
- &SYSPLS the character string to use as the separator
  - for a 281
- &SYSPRS 281
- &SYSSEE 279
- &SYSSEEALSO 279
- +
  - See continuation character

## A

- addenda dictionaries
  - adding words to 155
  - building 155
  - deleting words from 155
  - specifying 140
  - used for spelling verification 140
  - using for hyphenation 140
  - using for spelling verification 155
- adjusting left margin 64
- algorithmic hyphenation routine
  - not using 196
  - specifying 196
- alphabetic numbers
  - using for page numbers 286
- APFs
  - for GML tags 62
- appending files
  - naming the files to be appended 68

- appending files (*continued*)
  - symbols set when files are appended 68
  - using the .AP [Append] control word 68
- application processing functions
  - See APFs
- Arabic numbers
  - using for page numbers 286
- areas
  - defining with the .DA [Define Area] control word 108
  - ending 70
  - starting 70
- ATMS-III
  - environment restrictions 12
  - file naming conventions 9
  - input file characteristics 10
- attributes
  - of a symbol's value 450

## B

- balancing columns 74
- baseline
  - shifting of 325
- BIND option
  - description of 24
- binding
  - specifying 24
- blank lines
  - processing of 78
- blanks
  - processing input lines that begin with 230
  - required 118
- BookManager BUILD and READ 479
- BookMaster 479
- bottom margins
  - specifying 79
- box character sets
  - specifying 82
    - 32A 82
    - 32T 82
    - 38C 82
    - APL 82
    - GPC 82
    - TNC 82
    - TRM 82
    - VAN 82
- boxes
  - centering text within 84
  - drawing with rules 87
  - ending 81
  - nested 85
  - specifying 81
- breaks
  - causing 80
  - control words that cause breaks 441



## C

- capitalization
  - of text 48, 412, 418
  - using the .UC [Underscore and Capitalize] control word 412
  - using the .UP [Uppercase] control word 418
- carriage control characters
  - placed in output data stream 258
- cells, table
  - arranging 381
  - defaults 383
  - defining with .TD [Table Definition] control word 379
  - rotating 382
  - rotation 379
  - specifying minimum depth 382
  - specifying widths 381
  - starting with .TA [Table] control word 365
- centering text 95
- change bars
  - placement of 301
  - specifying 301
- changes
  - color xiii
  - defining objects xiii
  - grouping xiii
  - navigating xiii
  - overlay rotation xiii
- CHARS option
  - description of 24
- CMS
  - environment restrictions 11
  - file naming conventions 8
  - input file characteristics 9
- color
  - defining 102
- color defining
  - using the .CR [Color]. control word 102
- color definition
  - .BX control word 81
  - .DA control word 109
  - .TD control word 382
- column balancing
  - using the .BC [Balance Columns]. control word 74
- column line length
  - setting with the .CL [Column Line Length] control word 99
  - specifying with the .LL [Line Length] control word 233
- columns
  - balancing 74
  - conditional 92
  - defining with the .CD [Column Definition] control word 93
  - multicolumn 244
  - saving current definition of 327
- columns (*continued*)
  - single 327
  - specifying 327
  - temporary 327
  - starting 91
- comments
  - adding to a SCRIPT/VS file 100
  - placed in output data stream 258
  - using “.” to enter 100
- concatenation
  - canceling 174
  - restoring 174
- conditional processing
  - effect of TWOPASS option on 47
  - using the .AN control word 66
  - using the .EL [Else] control word 164
  - using the .IF [If] control word 203
  - using the .OR [Or] control word 274
  - using the .TH [Then] control word 395
  - with .GO as subject of IF statement 179
- conditional sections
  - defining with the .CS [Conditional Section] control word 105
- continuation
  - of text lines 107
- continuation character
  - defining 116
- CONTINUE option
  - description of 25
  - effect on TWOPASS option 47
- control word modifier 54
- control word separator
  - changing 54
  - effect of 53
  - system symbol for 54
  - using 53
- control words
  - associating with shading definition 262
  - deferred 279
  - notational conventions 55
  - reading syntax diagrams 56
  - separator 53
  - syntax of 53
  - take effect on next page 441
  - that cause breaks 441
  - that end a keep, float, footnote, named area, table 442
  - that ensure page is started 442
  - Type 1 55
- control words, unsupported
  - unsupported control words 461
- conventions, notational 55
- copy group
  - use with PSF 97
- courses
  - self-study 482

courses (*continued*)

training 482

CTF option

description of 25

## D

DCF publications 482

library guide 480

DCF Viewer Windows Kit 479

DCF/DB 479

DDUT option 18

decimal numbers

using for page numbers 286

defaults for logical output devices 23

deferred control words

.PI [Put Index] 279

.PT [Put Table of Contents] 294

.SE [Set Symbol] when used with & 337

defining columns 102

defining macros 142

defining symbols 337

defining text variables 157

name restrictions 157

defining variables 157

name restrictions 157

DEST option

description of 25

DEVICE option

description of 26

effect on TERM option 46

using to specify STAIRS/VS 46

diagnostic aids

tracing 221

displaying output at a terminal 46

DLF batch environment

using SCRIPT/VS in 11

Document Composition Facility/Double Byte  
(DCF/DB) 479

Document Library Facility

environment restrictions 12

double spacing 154

DSMUTTOC file

table of contents entries saved in 376

DSMUTWTF file 428

## E

end of file

simulating 163

ending processing

with the NOCONT option 37

environment restrictions

ATMS-III 12

CMS 11

Document Library Facility 12

environment restrictions (*continued*)

TSO 11

VSE 12

epifile 163

equation

.EQ control word 169

error messages

printing 36

error processing

defaults for 18

extra spaces

formatting of 167

specifying the width of 167

## F

features, optional

DCF Viewer Windows Kit 479

Document Composition Facility/Double Byte  
(DCF/DB) 479

Office Document Feature (ODF) 479

SCRIPT Mathematical Formula Formatter  
(SMFF) 479

file names

in ATMS-III 9

in CMS 8

in TSO 12, 8

FILE option

description of 30

files

characteristics of 9, 10

in ATMS-III 10

in CMS 9

in TSO 10

files, primary input

naming conventions 8

fill characters

between split text 363

between tab positions 400

floats

description of 170

formatting of 170

placement of 170

font library

specifying 33

with FONTLIB option 33

FONTLIB option

description of 33

specifying 33, 34

in ATMS-III 34

in CMS 33

in MVS 34

in TSO 33

in VSE 34

using 33

- fonts
  - defining with the .DF [Define Font] control word 130
  - provided with SCRIPT/VS 455
  - restoring 278
  - specifying attributes 132
  - specifying CODEPAGE 133
  - specifying COLOR 133
  - specifying of 130
  - specifying OLDATTR 132
  - specifying OLDWEIGHT 132
  - specifying OLDWIDTH 132
  - specifying point size 131
  - specifying rotation 132
  - specifying TYPE 131
  - specifying typeface 131
  - specifying weight 132
  - specifying width 132
  - specifying with CHARS option 24
  - starting 75
  - supplied with the 3800 Printing Subsystem 456
- footer, table
  - starting with .TA [Table] control word 365
- footers, table 368
- footings
  - See running footings
- footnotes
  - providing a leader for 172
  - rules for entering 172
  - specifying 172
- forcing a new page 101, 276
- formatting
  - beginning of 174
  - ending of 174
- formatting environment
  - named 309
  - naming 309, 323
  - restoring 309
  - saving 323
    - by name 323
- formatting mode 174
  - vertical 177
- formatting passes
  - indicated with the &\$PASS symbol 35, 47
  - indicated with the &\$PSNO symbol 35
  - specifying with the FPASSES option 35
  - specifying with the TWOPASS option 47
- FPASSES option
  - and cross-references 378
  - description of 35
  - with table of contents 378
- fractional numbers
  - using for page numbers 286
- fractional pagination 286
- full-stop characters 114

## G

- Generalized Markup Language
  - See GML tags
- GML tags
  - associating with APFs 62
- groups
  - for navigation with AFP Workbench 181
  - specifying the beginning of 181
  - specifying the end of 181

## H

- hanging indention
  - using offset mode 260
- head levels
  - characteristics of 136
  - defining with the .DH [Define Head Level] control word 136
  - formatting of 190
  - redefining with the .DH [Define Head Level] control word 136
  - spacing for 136
- header, table
  - starting with .TA [Table] control word 365
- headers, table 368
- headings
  - See head levels
  - See running headings
- help online
  - SCRIPT command help 7
  - CMS environment 7
  - TSO environment 7
- horizontal rules
  - for tables 380
  - specifying 192
- horizontal space
  - inserting between words 219
- hyphenation
  - changing the frequency 196
  - controlling 196
  - of single words 195
- hyphenation language 140

## I

- imbedding files
  - naming the files to be imbedded 208
  - using the .IM [Imbed] control word 208
- including files
  - of PostScript images 289
- indenting
  - output lines 206, 210, 217
- indention
  - from the right margin 217
  - hanging 260
  - of a single line or paragraph 206

- indentation (*continued*)
  - permanent 210
  - temporary 210
  - using offset mode 260
- index
  - creating with the .IX [Index] control word 224
  - sort sequence language 140
  - starting on new page 224
  - starting on same page 224
- index entries
  - creating 279
  - generating page numbers for 279
  - including multi-levels of 279
  - single 202
  - sort sequence of 279
  - specifying page numbers for 279
  - specifying text of 279
- INDEX option
  - description of 35
- index sort sequence 225, 280
- index sort sequence language 140
  - specifying 140
- index terms
  - building a sort key 280
  - cross referencing 279
  - multiple references to 279
  - sort sequence 225
  - sort sequence language 225
  - sort sequence of 279
  - specifying 279
- indexes
  - See also* index entries
  - See also* index terms
  - automatically generating 224
  - building 279
  - creating 224
  - creating entries for 279
  - enabled with the INDEX option 279
  - including cross references 279
  - including multi-level entries 279
  - including page references 279
  - specifying headings for 202
  - starting on new page 224
  - starting on same page 224
- inline segments 343
- inline space
  - inserting of 219
  - used to perform tabs 219
- input file 7
  - processing using SCRIPT/VS 7
- input files
  - characteristics of 9, 10
    - in a CMS environment 9
    - in a TSO environment 10
    - in an ATMS-III environment 10
  - formatting of 7

- input files (*continued*)
  - naming conventions 8, 9, 12
    - ATMS-III 9
    - CMS 8
    - TSO 8, 12
- input line, maximum characters 55
- input lines
  - beginning with a blank 230
  - beginning with a tab 242
  - concatenation of 174
  - prefixing of 297
  - processed as text 231
  - tracing of 221
- input processing
  - tracing of 221
- input trace
  - output lines generated by 221
- input translation 397
- intercharacter space
  - inserting of 201
  - used to emphasize text 201

## J

- justification
  - horizontal 196
    - with the .HW [Hyphenate] range control word 196
  - vertical 177, 234
    - using the .LO [Lead-Out] control word 234
- justifying output lines 226

## K

- keeping blocks of text together 227
- keeps
  - delayed 227
  - floating 227
  - formatting environment of 227
  - inline 227
  - regular 227
  - specifying 227

## L

- labels
  - setting 60
  - use of with the .GO [Goto] control word 60
- language
  - for hyphenation 140
  - for index sort sequence 140
  - for spelling verification 140
- leading blanks
  - processing of 230
- left margin
  - adjusting 64

- LIB option
  - description of 35
- library guide
  - for DCF publications 480
- line lengths
  - setting with the .CL [Column Line Length] control word 99
- line printers 22
- line spacing
  - specifying 235
- logical output devices
  - defaults for 23
  - formatting considerations for 19
  - table of 28

## M

- macro definitions 142
- macro libraries
  - specifying with LIB option 35
- macro names 54
- macros
  - defining symbols within 142
  - defining with the .DM [Define Macro] control word 142
  - disabling processing of 248
  - enabling processing of 248
  - ending the processing of 245
  - executing when macro substitution is off 166
  - invoked like control words 54
  - length of names 54
  - naming conventions 142
  - processing 142
  - processing with the .MS [Macro Substitution] control word 248
  - rules for symbol substitution within 142
  - using with substitution off 144
- margins
  - bottom 79
    - specifying with the .BM [Bottom Margin] control word 79
  - changing 64, 284
  - effect of BIND option on 284
  - specifying 79
  - specifying for even-numbered pages 284
  - specifying for odd-numbered pages 284
  - specifying with the .PM [Page Margins] control word 284
  - top 399
    - specifying with the .TM [Top Margin] control word 399
- marking updated material 301
- MARKUP 479
- markup content separator
  - defining 114

- maximum characters in input line 55
- MCS
  - See markup content separator
- measurement units
  - See spunit
- MESSAGE option
  - description of 36
- multicolumn format
  - defining 93
  - resuming processing of 244
  - starting 91
- multiple formatting passes 35

## N

- named areas
  - definition of 108
  - ending 70
  - rotation of 109
  - specifying depth of 109
  - specifying font for 109
  - specifying width of 109
  - starting 70
  - types of 108
- named fonts
  - describing 75
- new page, specifying 101, 276
- NOCONT option
  - description of 37
- NODDUT option 18
  - description of 37
- NOPROF option
  - description of 37
- NOSEGLIB option
  - description of 37
- NOSPIE option
  - description of 38
- notational conventions 55
- NOWAIT option
  - description of 38
- null lines
  - processing of 252
- NUMBER option
  - description of 38
  - effect on UNFORMAT option 48
- numbering pages 286

## O

- obsolete control words
  - .BT control word 461
  - .CO control word 461
  - .CW control word 461
  - .DI control word 461
  - .EB control word 461
  - .EP control word 461

obsolete control words (*continued*)

.ET control word 461  
.EZ control word 461  
.FI control word 461  
.FM control word 461  
.FS control word 461  
.FT control word 461  
.HE control word 461  
.HM control word 461  
.HN control word 461  
.HS control word 461  
.JU control word 461  
.NB control word 461  
.NC control word 461  
.NJ control word 461  
.OB control word 461  
.OP control word 461  
.OT control word 461  
.PP control word 461  
.PS control word 461  
.RT control word 461  
.SL control word 461  
.SY control word 461  
.TB control word 461  
.TT control word 461

ODF 479

Office Document Feature (ODF) 479

OfficeVision 479

offsetting text 260

online help

message help 7

CMS environment 7

TSO environment 7

SCRIPT command help 7

CMS environment 7

TSO environment 7

optional features

DCF Viewer Windows Kit 479

Document Composition Facility/Double Byte  
(DCF/DB) 479

Office Document Feature (ODF) 479

SCRIPT Mathematical Formula Formatter  
(SMFF) 479

OPTIONS option

description of 38

options, SCRIPT command

abbreviations 20

output

displaying at a terminal 46

obtaining printed 40

specifying destination of 20

output devices

logical 19

formatting considerations for 19

physical 19

formatting considerations for 19

output file

writing to 428

output files

printing part of 39

output lines

centering 95

double-spacing of 154

formatting of 174, 249

using the .FO [Format Mode] control word 174

using the .NF [No Formatting] control word 249

indenting 206, 210, 217

justification of 174

reference numbers for 320

single spacing of 358

output lines, generated by tracing 221

output lines, justification 226

output translation 404

overdraw options 175

overlays, including 262

## P

page composition

multicolumn format 244

single-column format 327

page devices 22

4250 Page Printer, 3800 Printing Subsystem3. and

3820 Page Printer 22

SCRIPT/VS support for 22

page eject mode 276

page ejects, conditional 101

page layout

effect of SCRIPT/VS control words 438

page length

specifying with the .PL [Page Length] control  
word 282

page margins

specifying with the .PM [Page Margins] control  
word 284

page numbers

alphabetic 286

Arabic 286

decimal 286

fractional 286

including prefixes for 286

Roman 286

setting current 286

specifying with the .PN [Page Numbering Mode]  
control word 286

PAGE option

description of 39

page overlays

.OI [Overlay Include] control word 262

page width

specifying with the .PW [Page Width] control  
word 296

- pagination
  - forcing an even-numbered page 276
  - forcing an odd-numbered page 276
  - fractional 286
- parts of the page 438
- patterns
  - SCREEN 329
  - STANDARD 329
- performance considerations
  - for label processing 60
  - in iterative processing 179
  - in spelling verification 361
- physical output devices
  - formatting considerations for 19, 22
- postprocessor
  - using with variable text 426
- PostScript 289
- primary input file
  - naming conventions 8
- PRINT option
  - description of 40
- Print Services Facility (PSF) 479
- printer classes
  - line printers 22
  - page devices 22
- printer output, obtaining 40
- printers
  - line 22
  - page devices 22
- printing
  - table of contents
    - on new page 376
    - on same page 376
- printing index
  - starting on new page 224
  - starting on same page 224
- printing part of an output document 39
- processing
  - ending immediately 299
  - ending with the NOCONT option 37
  - immediate end of 299
- products, related
  - See related products
- PROFILE option
  - description of 40
- PSF 479
- PSOUT option
  - description of 41
- publication, related (see related publications)
- publications
  - for DCF 482

## Q

- QUIET option
  - description of 42

## R

- reading syntax diagrams 56
- reference numbers
  - for output lines 320
- related products
  - BookManager BUILD and READ 479
  - BookMaster 479
  - MARKUP 479
  - OfficeVision 479
  - Print Services Facility (PSF) 479
  - SGML TextWrite 479
  - SGML Translator DCF edition 479
  - TextTagger ProcessMaster edition 479
  - TextTagger Workstation edition 479
- related publications 479
  - descriptions 483
- required blanks 118
- revision codes
  - for single lines 306
  - nested 305
  - placement of 301
  - specifying 301
  - with tables 368
- right margin
  - aligning text with 318
- Roman numerals
  - using for page numbers 286
- ROTATE parameter
  - .OI control word 263
  - angle 263
- rotation of named areas 109
- rows, table
  - defining with .TD [Table Definition] control word 379
  - starting with .TA [Table] control word 365
- rows, tables
  - defaults 383
- rules
  - defining with the .DR [Define Rule] control word 151
  - horizontal 192, 380
    - specifying with the .HR [Horizontal Rule] control word. 192
    - specifying with the .TD [Table Definition] control word 380
  - specifying with the .TD [Table Definition] control word 380
  - vertical 380, 423
    - specifying with the .TD [Table Definition] control word 380
    - specifying with the .VR [Vertical Rule] control word 423
- running footings
  - defining for even-numbered pages 310
  - defining for odd-numbered pages 310
  - definition of 310

- running footings (*continued*)
  - placement on a page 310
  - processing of 310
  - redefining 310
  - specifying 310
  - suppressing 310
  - where to define 310

- running headings
  - defining for even-numbered pages 314
  - defining for odd-numbered pages 314
  - definition of 314
  - placement on a page 314
  - processing of 314
  - redefining 314
  - specifying 314
  - suppressing 314
  - where to define 314

## S

### SCRIPT command

- descriptions 7—48
- issuing as an
  - ATMS-III command 8
  - CMS command 8
  - TSO command 8
- options 17, 18, 20, 439
  - abbreviations 20
  - defaults 17
  - mutually exclusive 18
  - summary of 439
  - syntax 17
- printing lowercase letters as uppercase 48
- specifying as destination of output 20
- syntax of
  - user defined options 48
- using 7—48

### SCRIPT command help

- CMS environment 7
- TSO environment 7

### SCRIPT command options

- abbreviating 20

### SCRIPT Mathematical Formula Formatter (SMFF)

- program 479

### SCRIPT/VS

- control words
  - deferred 279, 294, 337
  - guidelines for entering in an input file 53
  - marking up a document with 53
  - notational conventions 55
  - See individual control words
  - separator 53
  - syntax of 53
  - that cause a break 441
  - that end a keep, float, footnote, named area, table 442
  - that ensure page is started 442

### SCRIPT/VS (*continued*)

- control words (*continued*)
  - that take effect on next page 441

- Type 1 55

- in a DLF batch environment

- in a TSO batch environment

- in interactive environment

- logical devices 28

- macros 54

- obsolete control words 461

- .BT control word 461

- .CO control word 461

- .CW control word 461

- .DI control word 461

- .EB control word 461

- .EP control word 461

- .ET control word 461

- .EZ control word 461

- .FI control word 461

- .FM control word 461

- .FS control word 461

- .FT control word 461

- .HE control word 461

- .HM control word 461

- .HN control word 461

- .HS control word 461

- .JU control word 461

- .NB control word 461

- .NC control word 461

- .NJ control word 461

- .OB control word 461

- .OP control word 461

- .OT control word 461

- .PP control word 461

- .PS control word 461

- .RT control word 461

- .SL control word 461

- .SY control word 461

- .TB control word 461

- .TT control word 461

- terms for parts of page 438

- utility files 438

- with DLF

- with TSO

### SCRIPT/VS logical devices

- logical line devices 27

- logical page devices 28

- logical PostScript devices 28

### SEARCH option

- description of 42

### SEGLIB option

- description of 43

### segment library

- bypassing with NOSEGLIB option 37

- search options for 18

- specifying with SEGLIB option 43



- segments
  - including in a document 343
  - placing inline 343
  - specifying depth of 347
    - using the &SD' symbol attribute 347
  - specifying width of 347
    - using the &SW' symbol attribute 347
  - specifying with the .SI [Segment Include] control word 343
- self-study courses 482
- separation masters
  - defining with the .SM [Separation Master] control word 353
- SEPMASSTR
  - defining 353
- SEPMASSTR option
  - description of 44
- SEPMASSTR parameter
  - .DF control word 133
  - .DR control word 151
  - .OI control word 263
  - .PO control word 289
  - .SD control word 328
  - .SI control word 343
  - token 263
- setting labels 60
- setting tabs 400
- SGML TextWrite 479
- SGML Translator DCF edition 479
- shading
  - patterns 329
  - variations 329
- shading definition
  - .BX control word 81
  - .DA control word 108
  - .SD control word 328
  - .TD control word 382
  - shading 382
- simple indentation 210
- single index entries 202
- single spacing 358
- SMFF 479
- sort sequence
  - index 225
- sort sequence language
  - specifying for index 140
- sort sequence, index 280
- space units 58
  - See also* spunit
  - rounding of 58
  - types of 58
- spacing
  - using the .BL [Blank] control word 78
  - using the .LO [Lead-Out] control word 234
  - using the .LS [Line Spacing] control word 235
  - using the .SB [Shift Baseline] control word 325
- spacing (*continued*)
  - using the .SK [Skip] control word 351
  - using the .SP [Space] control word 356
  - with the .DS [Double Space Mode] control word 154
  - with the .SS [Single Space Mode] control word 358
- special characters
  - defining with the .DC [Define Character] control word 112
- specifying formatting passes
  - FPASSES option 35
  - TWOPASS 47
- SPELLCHK option
  - description of 45
- spelling verification
  - starting 361
  - stopping 361
- Spelling verification language 140
- splitting text 363
- STAIRS/VS
  - using the CTF option for 25
- starting a new page 101, 276
- starting an even-numbered page 276
- starting an odd-numbered page 276
- stop characters, full 114
- STOP option
  - description of 45
- subscripts
  - creating 325
- Summary of Amendments xiii
- superscripts
  - creating 325
- symbol libraries
  - specifying with LIB option 35
- symbols
  - assigning values to 337
  - beginning with an \* 338
  - containing special characters 338
  - defining 337
  - effect of TWOPASS option on 47
  - effects of substitution 340
  - for current page number 337
  - in arithmetic expressions 340
  - name restrictions 337
  - substitution values for 337
  - using to set current page number 340
- syntax diagrams, reading 56
- SYOFF option 18
  - description of 45
- SYON option 18
  - description of 45
- SYSVAR option
  - description of 45

## T

### table

- defaults 383

### table of contents

- adding entries to 294
- automatically generating 376
- printing of 376
- processing of 376
- producing 47
- resetting the page numbers 376
- specifying with the .TC [Table of Contents] control word 376
- starting on new page 376
- starting on same page 376
- using the DSMUTOC file 376

### tables

- breaking a table 365

#### cells

- defining with .TD [Table Definition] control word 379
- rotating 379
- starting with .TA [Table] control word 365
- defining with .TD [Table Definition] control word 379
- formatting rotated cells 367
- formatting text inside 366
- rotating 379

#### rows

- defining with .TD [Table Definition] control word 379
- starting with .TA [Table] control word 365
- specifying horizontal rules 380
- specifying vertical rules 380
- splitting a table 365, 367
- starting with .TA [Table] control word 365
- table footer
  - starting with .TA [Table] control word 365
- table footers 368
- table header
  - starting with .TA [Table] control word 365
- table headers 368

### tabs

- aligning 400
- center aligning 400
- fill characters for 400
- leading 242
- processing of 242
- left aligning 400
- right aligning 400
- setting 374, 400
- specifying with the .TB [Tab Setting] control word 374
- specifying with the .TP [Tab Position] control word 400

### TERM option

- description of 46

- terminals, displaying output at 46

- terms for parts of the page 438

### text

- entering at the terminal 393

### text libraries

- specifying with TLIB option 46

### text lines

- continuation of 107
- printing on current page 97, 226

- text string translation 406

### text variables

- defining 157
- name restrictions 157

- TextTagger ProcessMaster edition 479

- TextTagger Workstation edition 479

### TLIB option

- description of 46

- tracing of input processing 221

- training courses 482

### translation

- of input 397
- of output 404
- of text strings 406

### TSO

- environment restrictions 11
- file naming conventions 12—13
- input file characteristics 10
- input file naming conventions 8, 12

### TSO batch environment

- using SCRIPT/VS in 11

### two formatting passes

- See TWOPASS option

### TWOPASS option

- description of 47

- Type 1 control words 55

## U

### underscoring

- definition using the .UD [Underscore Definition] control word 414

- using the .US [Underscore] control word 420

### underscoring and capitalizing

- using the .UC [Underscore and Capitalize] control word 412

### UNFORMAT option

- description of 47

### units of measure

- See *also* spunit
- space units 58

### UPCASE option

- description of 48

### Uppercase string translation

## V

- variable text
  - inserting into source document 426
  - using postprocessors 426
  - with page printers 426
  - with the .VT [Variable Text] control word 426
- variables
  - defining 157
  - name restrictions 157
- variations
  - pattern 329
  - shading 329
- vertical distribution
  - with the .BC [Balance Columns] control word 74
- vertical formatting
  - with the .FV [Format Vertically] control word 177
- vertical justification
  - as affected by the .LO [Lead-Out] control word 234
  - as affected by the .LS [Line Spacing] increment control word 236
  - as affected by the .LS [Line Spacing] skip control word 236
  - as affected by the .LS [Line Spacing] space control word 236
  - as affected by the .LS [Line Spacing] text control word 236
- vertical rules
  - for tables 380
  - specifying 423
- vertical space
  - at the bottom of a page 79
- vertical spacing
  - using the .BL [Blank] control word 78
  - using the .LO [Lead-Out] control word 234
  - using the .LS [Line Spacing] control word 235
  - using the .SB [Shift Baseline] control word 325
  - using the .SK [Skip] control word 351
  - using the .SP [Space] control word 356
- VSE
  - environment restrictions 12

## W

- what's new xiii
- widow control 434
- widow zones
  - processing of 434
- word spaces
  - controlling the width of 431
- word spacing 431



---

# Readers' Comments — We'd Like to Hear from You

**Document Composition Facility**  
**SCRIPT/VS Language Reference**  
**Release 4.0**

**Publication No. SH35-0070-07**

Use this form to provide comments about this publication, its organization, or subject matter. Understand that IBM may use the information any way it believes appropriate, without incurring any obligation to you. Your comments will be sent to the author's department for the appropriate action. Comments may be written in your language.

**Note:** IBM publications are not stocked at the location to which this form is addressed. Direct requests for publications or for assistance in using your IBM system, to your IBM representative or local IBM branch office.

|                                         | Yes   | No    |
|-----------------------------------------|-------|-------|
| • Does the publication meet your needs? | _____ | _____ |
| • Did you find the information:         |       |       |
| Accurate?                               | _____ | _____ |
| Easy to read and understand?            | _____ | _____ |
| Easy to retrieve?                       | _____ | _____ |
| Organized for convenient use?           | _____ | _____ |
| Legible?                                | _____ | _____ |
| Complete?                               | _____ | _____ |
| Well illustrated?                       | _____ | _____ |
| Written for your technical level?       | _____ | _____ |
| • Do you use this publication:          |       |       |
| As an introduction to the subject?      | _____ | _____ |
| As a reference manual?                  | _____ | _____ |
| As an instructor in class?              | _____ | _____ |
| As a student in class?                  | _____ | _____ |
| • What is your occupation?              |       |       |

\_\_\_\_\_  
Thank you for your input and cooperation.

**Note:** You may either send your comments by fax to 1-800-524-1519, or mail your comments. If mailed in the U.S.A., no postage stamp is necessary. For residents outside the U.S.A., your local IBM office or representative will forward your comments.

**Comments:**

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

\_\_\_\_\_  
Phone No.



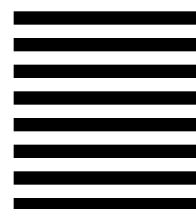
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



## BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development  
The IBM Printing Systems Company  
Department H7FE Building 003G  
P O Box 1900  
BOULDER CO 80301-9817



Fold and Tape

Please do not staple

Fold and Tape





Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SH35-0070-07

