

# SCRIPT PROCESSOR REFERENCE MANUAL

OPERATING SYSTEM SOFTWARE  

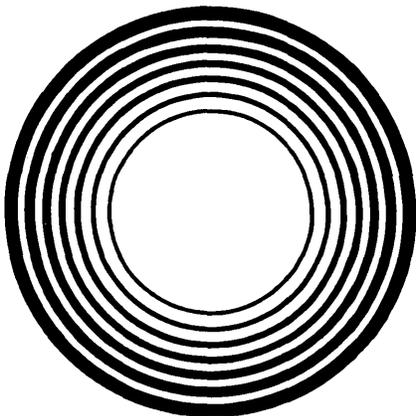
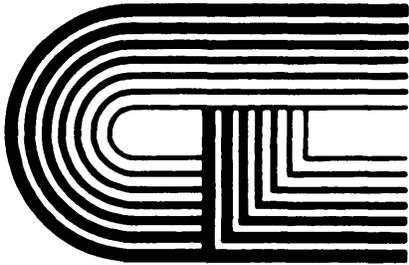
---

MAKES MICROS RUN LIKE MINIS

From  
PHASE ONE SYSTEMS, INC.  

---

OAKLAND, CALIFORNIA



# SCRIPT PROCESSOR REFERENCE MANUAL

Second Edition

Documentation by: C. P. Williams  
Software by: Timothy S. Williams

OPERATING SYSTEM SOFTWARE  

---

MAKES MICROS RUN LIKE MINIS

From  
PHASE ONE SYSTEMS, INC.  

---

OAKLAND, CALIFORNIA

7700 Edgewater Drive, Suite 830  
Oakland, California 94621  
Telephone (415) 562-8085  
TWX 910-366-7139

Second edition, first printing: March, 1980

#### **PROPRIETARY NOTICE**

The software described in this manual is a proprietary product developed by Timothy S. Williams and distributed by Phase One Systems, Inc., Oakland, California. The product is furnished to the user under a license for use on a single computer system and may be copied (with inclusion of the copyright notice) only in accordance with the terms of the license.

Copyright (C) 1980 by Phase One Systems, Inc.

Previous editions copyright 1978, 1979, and 1980 by Phase One Systems, Inc. All rights reserved. Except for use in a review, the reproduction or utilization of this work in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including xerography, photocopying, and recording, and in any information storage and retrieval system is forbidden without the written permission of the publisher.

Z80 is a registered trademark of Zilog, Incorporated.

## **P R E F A C E**

This document is a reference manual for the OASIS SCRIPT word processing language. Prior knowledge in the use of the OASIS EDIT program and the EXEC processor is assumed.

This manual, named SCRIPT, like all OASIS documentation manuals, has the manual name and revision number in the lower, inside corner of each page of the body of the manual. In most chapters of the manual the last primary subject being discussed on a page will be identified in the lower outside corner of the page.

### **Related Documentation**

The following publication provides additional information required in the use of the OASIS SCRIPT language:

**OASIS System Reference Manual**

**OASIS Text Editor Reference Manual**

**OASIS EXEC Language Reference Manual**

## TABLE OF CONTENTS

Section	Page
CHAPTER 1 INTRODUCTION .....	1
1.1 Creating A Script Text File .....	1
1.2 Invoking the Script Processor .....	1
CHAPTER 2 SCRIPT COMMANDS .....	5
2.1 Page Align Command .....	5
2.2 Appendix Command .....	5
2.3 Break Command .....	6
2.4 Case Command .....	7
2.5 Center Command .....	7
2.6 Chapter Command .....	8
2.7 Contents Command .....	8
2.8 File Copy Command .....	8
2.9 Set Character Pitch Command .....	9
2.10 Page Eject Command .....	9
2.11 End of Text Command .....	9
2.12 Line Fill Command .....	10
2.13 Page Footing Command .....	10
2.14 Format Text Command .....	11
2.15 Line Gutter Command .....	11
2.16 Page Heading Command .....	11
2.17 Line Inside Gutter Command .....	12
2.18 Paragraph Indent Command .....	12
2.19 Console Input Command .....	12
2.20 Console Input Variable Command .....	13
2.21 Text Justification Command .....	13
2.22 Left Sided Page Footing Command .....	14
2.23 Line of Characters Command .....	14
2.24 Linking Segmented Files Command .....	14
2.25 Left Margin Command .....	15
2.26 Left Sided Page Heading Command .....	15
2.27 Line Outside Gutter Command .....	15
2.28 Conditional Page Eject Command .....	16
2.29 Paragraph Spacing Command .....	16
2.30 Paragraph Break Command .....	16
2.31 Pause Command .....	17
2.32 Position to Line Number Command .....	17
2.33 Preface Command .....	17
2.34 Read Variable Command .....	18
2.35 Remark Command .....	18
2.36 Right Sided Page Footing Command .....	18
2.37 Right Sided Page Heading Command .....	19
2.38 Right Margin Command .....	19
2.39 Section Command .....	19
2.40 Set Command Character Command .....	21
2.41 Page Number Set Command .....	21
2.42 Set Alternate Space Character Command .....	22
2.43 Set Variable Command .....	22
2.44 Document Size Command .....	22
2.45 Line Skipping Command .....	23
2.46 Line Spacing Command .....	23

## TABLE OF CONTENTS

Section	Page
2.47 Set Special Characters Command .....	24
2.48 Tab Setting Command .....	24
2.49 Document Title Command .....	25
2.50 Type Command .....	25
CHAPTER 3 SPECIAL CHARACTERS .....	27
3.1 Embedded Commands Available in Heading, Footing, and Center .....	27
3.2 Embedded Commands not Available in Heading, Footing, and Center Text ..	27
3.3 Embedded Commands Always Available .....	28
APPENDIX A SCRIPT EXAMPLES .....	31
A.1 Example 1: A Documentation Manual .....	31
A.2 Example 2: Form Letter .....	35
APPENDIX B SCRIPT COMMAND SUMMARY .....	39
APPENDIX C SPECIAL CHARACTER SUMMARY .....	40



## CHAPTER 1

### INTRODUCTION

The OASIS Script processor is a powerful documentation processing system. The processor was not designed for general word processing applications but was designed for documentation and manual generation. (All of the OASIS documentation is generated and maintained via this Script processor.) However the Script processor can be used for many word processing applications.

The general features of the processor include provisions for creating and maintaining: page headings, page footings, line filled and right justified text, title pages, paragraph indentation, variable left and right margins, conditional page ejection, variable line and paragraph spacing, text input from console while processing, access to system date and time, Table of Contents, columnar tabulation, underscored and/or boldface printing, left/right page alignment, interface to external data file, etc.

#### 1.1 Creating A Script Text File

To create text and commands for the Script program to process you must create a file. This file can be created by various OASIS system programs or language processors: BASIC, MACRO, EDIT, TEXTEDIT, etc. Generally the easiest method is by using the EDIT program. The file created must be sequential in format.

The Script file may have any non-reserved file name and type and may reside on any disk drive (or multiple disk drives if the file is segmented).

Script files contain records, each record containing text or a Script command. A Script command record must start with the period character (.) and may only consist of the command word and parameters required by that command (do not mix text with the command record unless specified by a parameter of the command). Script commands are discussed in detail in the following chapter. Script text may not contain any Script commands but may contain embedded command characters. Script command characters are discussed in detail in the chapter "Special Characters".

Script text files may be segmented into several files and linked together for Script processing by using the Script LINK command.

"Boiler-plate" type documents (a document formed by piecing together sections of several documents) may be generated by using the COPY command.

#### 1.2 Invoking the Script Processor

After a Script file is created you must use the CSI SCRIPT command to process the file and generate the output document. The format of the SCRIPT command is:

**SCRIPT <file-name> [<file-type>] [(option ...)]**

Where:

file-name Indicates the file name of the file to be processed.

file-type Indicates the file type. This field is optional and, when not specified, the file type of "SCRIPT" is used. The file type used for the first file is the default file type for all subsequent LINK commands.

## SCRIPT REFERENCE MANUAL

### SCRIPT Options

The options available in the SCRIPT command include the following:

TYPE Indicates that the document generated is to be output to the console output device. This is a default option.

WAIT Indicates that after a full page of text is printed the SCRIPT program is to pause, allowing the operator to insert a new page in the printer, etc. This option has no effect when the option FILE is used.

WORK=<fd> Indicates that the disk to be used for working storage is the disk currently mounted in the drive <fd>. When this option is not specified the system "S" disk will be used.

PRINT[n] Indicates that the document generated is to be output to the primary printer or PRINTERn.

CPI nn Indicates that the printer has been set to nn characters per inch. This option is only necessary for letter quality printers. The default setting for CPI is 10.

REPEAT nn Indicates that the report is to be produced nn times. This option should not be used when an external data file is used in the text output. The default REPEAT count is 1.

FILE Indicates that the document generated is to be output to the file ASSIGNED to channel 2. (See "ASSIGN COMMAND" chapter in the OASIS System Reference Manual.) To insure the proper document size and page numbering you should specify all of the document parameters with the SIZE command.

UPPER Indicates that all alphabetic characters output to the document are to be translated to their upper case equivalents. This option is normally used only when the output device does not have a lower case character set and is not capable of performing the translation itself.

nn [nn] Indicates the from and to range of the printout. The first number indicates the page number of the first page to be printed, the second number indicates the page number of the last page to be printed. When this option is not used the entire report will be printed.

When the Script processor is first invoked by this command the following values are initialized:

Cassmode	Mixed (M)
Fillmode	On
Justify	On
Paraskip	1
Spacing	1
Size	1,80,1,60 (Reset to device parameters if option FILE not specified)
Page	1

## CHAPTER 1: INTRODUCTION

As noted above, when the output from the Script processor is to go to a disk file, the file to be used must be previously assigned (see "ASSIGN COMMAND" in the OASIS System Reference Manual) to channel 2.

If the text to be processed contains one or more READ commands the data file to be used must be previously assigned to channel 3.

The OASIS Script processor is a virtual processing language. This means that the Script processor can process a text file that requires more space than is available in the main memory. The Script processor uses the random access mass storage memory (disk) as an extension of its memory.

The OASIS Script processor is also an optional two pass processor. When two pass processing is specified (see TITLE and CONTENTS commands in following chapter) the Script processor reads and processes the entire text file in the first pass without creating an output document but creating a temporary work file containing the information necessary to create a Table of Contents. During the second pass the text file is reprocessed and the Table of Contents is included, when specified.

When the output is directed to a printer or a disk file (options PRINT or FILE) the Script processor will display on the screen the page number being processed and any LINK commands encountered. This facilitates your monitoring the progress of the text generation.

(This page intentionally left blank)

## CHAPTER 2

### SCRIPT COMMANDS

The Script processor has many commands to assist in the formatting of the output document. The syntax and operation of each of these commands is described in this chapter. All Script commands are lines by themselves and the first character of a command line must be a period. The command words may be in upper or lower case.

Lines of text may not begin with a period character.

#### 2.1 Page Align Command

The ALIGN command allows you to perform condition blank page eject to align a page to a "left" or "right" sided page. A left sided page is an even numbered page; a right sided page is an odd numbered page. The format of the command is:

**.ALIGN <R|L> [<text>]**

Where:

R Indicates that the current page is to be aligned to a right sided page.

L Indicates that the current page is to be aligned to a left sided page.

text Specifies the text to be displayed on the blank page if an extra page has to be ejected to align to the proper side. When <text> is not specified the default message of "(This page intentionally left blank)" is used.

The ALIGN command is generally used before the start of a new chapter or appendix to cause it to be printed on a right (odd numbered) sided page. The OASIS manuals use this command in that way because, when the output is reproduced with double sided copying, the chapters will always start on the right page of a manual which is easier for the user to index to.

For an example of the effect of this command you will have to search through the OASIS manuals looking for the blank pages with the page centered default text. Since this is a conditional eject it is unknown whether a page will be ejected when the text is being edited.

#### 2.2 Appendix Command

The APPENDIX command informs the Script processor that a new appendix is to be started and the title of that appendix. The format of the command is:

**.APPENDIX <level> <title>**

Where:

level Indicates the appendix number. This field is only included in the syntax to be consistent with the SECTION command. The character or characters in this position are ignored by the Script processor (but it must be present). You may use the actual letter code of the appendix or a dummy such as a plus sign (+). It is advised that you use the dummy character. If modifications to your script files are

## SCRIPT REFERENCE MANUAL

made the letter code of the appendix may change, thus causing you to change your APPENDIX commands if letter codes are used in this field. The level field must be preceded and followed by a space.

**title** Indicates the title of the appendix. This text is always folded to upper case by the Script processor before being saved.

When the APPENDIX command is encountered by the Script processor the appendix letter code, title, and beginning page number are saved for inclusion in the Table of Contents. (This occurs during pass zero.)

When the Script output is to a letter quality printer the appendix heading is always printed in boldface.

In addition the following actions are taken:

1. The current appendix letter is incremented by one. If this is the first appendix in the document then the current appendix letter is an "A".
2. The title of the appendix is folded to upper case and the @S variable is assigned the value "APPENDIX " plus the appendix letter code followed by a colon, space, plus the title of the appendix.
3. A page eject is output to the document.
4. The appendix heading is output to the document. The appendix heading consists of two lines, centered on the document page. The first line is the word "APPENDIX" followed by the letter code. The second line is the appendix title.
5. Left and right margins are reset to the .SIZE parameters.
6. The indent value is reset to zero.
7. Fill mode is set on.

### 2.3 Break Command

The BREAK command is a convenience command since its function is always performed whenever the commands: ALIGN, APPENDIX, CENTER, CHAPTER, CONTENTS, EJECT, FILL, FOOT, HEAD, INDENT, LFOOTING, LHEADING, LINE, LMARGIN, NOFILL, PAGE, PREFACE, RFOOTING, RHEADING, RMARGIN, SECTION, and SKIP. The BREAK command causes the text in the current line buffer to be printed unjustified and the new sentence flag is set. The format of the command is:

**.BREAK**

The only known usage of this command would be in the situation where a paragraph break is desired without paragraph spacing being performed.

## 2.4 Case Command

The CASE command informs the Script processor whether to translate characters before output. The format of the command is:

**.CASE U|L|M**

Where:

- U Indicates that characters are to be folded to their upper case equivalents. Embedded case mode command characters will be ignored in this casemode. (See section on Special Characters.)
- L Indicates that characters are to be folded to their lower case equivalents. Embedded command characters may override this casemode. (See section on Special Characters.) The first word of sentences will be capitalized. (The first word of a sentence is the first word of a paragraph or the first word following a space which followed any of these characters: period, question mark, exclamation mark.) This casemode is useful when the editing terminal only has an uppercase character set.
- M Indicates that mixed casemode is to be set. In this casemode characters are not modified by the Script processor. Embedded casemode commands have no effect on the output document (except in the HEADING, FOOTING, and CENTER commands).

When the Script processor is first invoked the casemode is set to mixed (M).

If the global casemode UPPER option was specified in the CSI SCRIPT command the CASE command will have no effect, nor will any embedded casemode commands.

## 2.5 Center Command

The CENTER command specifies that some text is to be centered on the document page. The format of the command is:

**.CENTER <text>**

Where:

- text Indicates the string of characters to be centered and may include leading and/or trailing blanks. This text must be less than 255 characters in length. All of the embedded commands described in the chapter "Special Characters" may be used, except for the embedded line feed.

When the Script processor computes the center of the document page the left and right parameters specified in any LMARGIN and RMARGIN commands are used, not the SIZE command parameters (unless there are no LMARGIN and RMARGIN commands in effect).

The CENTER command causes a paragraph break before and after the centered text. The spacing between paragraphs is determined by the PARASKIP command.

## **SCRIPT REFERENCE MANUAL**

### **2.6 Chapter Command**

The CHAPTER command indicates to the Script processor that a new chapter is to be started. The format of the command is:

**.CHAPTER <level> <title>**

The CHAPTER command operates identical to the APPENDIX command except that chapter levels are numeric, starting with one, and the word "CHAPTER" is used, not "APPENDIX".

### **2.7 Contents Command**

The CONTENTS command specifies that a Table of Contents is to be generated and output at this position. The format of the command is:

**.CONTENTS**

When the CONTENTS command is used you must use the TITLE command also and the TITLE command must be specified prior to the CONTENTS command. (It is the TITLE command that causes the Script processor to make a preliminary first pass of the files, gathering content information.)

The format of the Table of Contents generated by this command is exemplified by the Table of Contents at the beginning of this manual. Lower case Roman numeral page numbers are used (unless the global UPPER option is in effect).

### **2.8 File Copy Command**

The COPY command allows you to copy another Script text file into the text being generated. The format of the command is:

**.COPY <file name> [<file type>]**

When the Script processor encounters a COPY command the file, specified by the file name and type, is searched for on all of the attached disks. When the file is found the scripting process continues with the first record in that file and proceeds until the end of the "copied" file is encountered or an END command is encountered.

Either of these conditions will cause the "copied" file to be closed and the processing continues with the primary file (the text file containing the COPY command that was processed).

A file that is used as a "copied" file may not contain a COPY command.

## 2.9 Set Character Pitch Command

The CPI command allows you to specify the character pitch (number of characters per inch) that the printer is to be set to for subsequent text. This command is only operative when the output device is a letter quality printer specified by the class code of the attachment of the printer (see "ATTACH Command" in the OASIS System Reference Manual). The format of the command is:

**.CPI <value>**

Where:

value Specifies the number of characters per inch that subsequent characters are to be printed at. This value would normally be a 10 (pica) or 12 (elite) but may be any reasonable value.

The value specified is divided into 120 (the number of increments in an inch for letter quality printers). The result will be rounded down to the nearest integer which is used to instruct the printer how much space each character is to take. For example, a CPI setting of 12 will cause the printer to take 10/120 of an inch for each character; a CPI setting of 9 will cause the printer to take 13/120 of an inch for each character. Note that the second example does not come out to exactly 9 characters per inch due to rounding.

## 2.10 Page Eject Command

The EJECT command indicates that a new page is to be started by the Script processor. The format of the command is:

**.EJECT**

The EJECT command does not actually cause a page eject to occur immediately. It does specify that the next text output to the document is to cause a page eject before being output. This means that two EJECT commands in a row will not cause two page ejections. To perform multiple page ejections you will have to have some text between the EJECT commands (blank lines in NOFILL mode, if necessary).

## 2.11 End of Text Command

The END command is a convenience command since its function is to specify that there are no more commands or text records to be processed. This can be accomplished by and end of file without a LINK command. However, it is always a good practice to use the END command because it is a graphic indicator that it was intended to be the end of the text. The format of the command is:

**.END**

**2.12 Line Fill Command**

The FILL command specifies that the lines output to the document are to be filled with words from following lines of text until no more words will fit on the output line. The NOFILL command specifies that the lines output to the document are to have the same words and spacing as the input lines. The format of the commands is:

**.FILL**

**.NOFILL**

The fillmode is set to fill when the Script processor is first invoked. The FILL command does not affect the current status of the justify mode, however, the NOFILL command disables the justify mode (temporarily set to NOJUSTIFY - reenabled by FILL).

The FILL and NOFILL commands do not cause paragraph breaks.

When the fillmode is set to NOFILL any input lines of text are truncated at the current right margin column number. This provides an easy means of printing a line of characters without concern for the current value of the right margin.

**2.13 Page Footing Command**

The FOOTING command informs the Script processor that a page footing is to be established and what the footing is to contain. The format of the command is:

**.FOOTING <text>**

The FOOTING command resets both the LFOOTING text and the RFOOTING text.

The FOOTING command does not cause a page eject but it does change the footing of the current page. All of the allowed embedded commands described in the chapter "Special Characters" may be used in the text. The maximum length of the resulting text is 255.

When the TITLE and CONTENTS commands are used a default FOOTING is set to the following:

**.FOOTING &@,- @P -&**

This causes the current page number to be printed at the bottom of each page, centered on the line, surrounded by hyphens, in boldface (see the footing at bottom of this page).

When the page footing is printed it is always separated from the text by at least one blank line.

To specify that no footing is to be output at the bottom of a page use the FOOTING command with no text following.

### 2.14 Format Text Command

The `FORMAT` command specifies that some text is to be formatted on the document page. The format of the command is:

**`.FORMAT <text>`**

Where:

`text`      Indicate the string of characters to be formatted. Any leading spaces in the text will be ignored. The text must be less than 255 characters in length. All embedded command characters described in the chapter "Special Characters" that apply to `HEADING`, `FOOTING`, and `CENTER` text may be used here, except for the embedded line feed.

A `FORMAT` command causes a paragraph break before and after the command but paragraph spacing is not performed.

### 2.15 Line Gutter Command

The `GUTTER` command instructs the Script processor that a character is to be printed in both gutters of each subsequent line printed. A line gutter is the area to the left or right of a line of print. The format of the command is:

**`.GUTTER [<char>]`**

The `<char>` specified will be printed in both the inside and outside gutter of all subsequent lines until the character is changed to another character or blank. Specifying a blank character turns off the gutter printing.

The `GUTTER` command combines the functions of the `IGUTTER` and `OGUTTER` commands. For an example of the use of this command see the OASIS System Reference Manual. All paragraphs in that manual relating to the multi-user system are marked by vertical bar characters (|) in the gutters.

### 2.16 Page Heading Command

The `HEADING` command informs the Script processor that a page heading exists and specifies what it is. The format of the command is:

**`.HEADING <text>`**

The `HEADING` command resets both the `LHEADING` text and the `RHEADING` text.

All of the allowed embedded commands described in the chapter "Special Characters" may be used in the text. The maximum length of the resulting text must be less than 255 characters. Imbedded tabulation commands (other than the @,) should not be used.

When the `TITLE` and `CONTENTS` commands are used a default `HEADING` is set to the following:

`.HEADING &<title text>@,@,@S@;&`

This specifies that a heading consisting of the title, as specified in the `TITLE` command, is to be printed at the top left hand side of each page; at the top right

## SCRIPT REFERENCE MANUAL

hand side the current chapter or appendix number and title is printed in boldface (see the heading at the top of this page).

When a heading is specified (or the default heading is used) one blank line is used to separate the heading from the text on each page. A HEADING command does not cause any page ejects. The command merely establishes the page heading for subsequent top-of-page printing.

To specify that no heading is to be output at the top of each page use the HEADING command with no text following.

### 2.17 Line Inside Gutter Command

The IGUTTER command operates similar to the GUTTER command discussed previously but only effects the inside gutter of a line. The format of the command is:

**.IGUTTER [<char>]**

The inside gutter of a line is defined as the left area of a line on odd numbered pages and the right area of a line on even numbered pages.

### 2.18 Paragraph Indent Command

The INDENT command specifies the relative indentation of the first word of each paragraph. The format of the command is:

**.INDENT <signed value>**

Indentation is always specified relative to the current LMARGIN value. The Script processor sets the INDENT value to zero (no indenting) when it encounters any of the following commands: APPENDIX, CHAPTER, SECTION, or LMARGIN.

The INDENT value affects only the first word of a paragraph. The first word of a paragraph is the sequence of characters that precede the first occurrence of a space, following a paragraph break.

Specifying a value that would cause printing to the left of the left margin specified in the SIZE command will produce undefined results.

### 2.19 Console Input Command

The INPUT command indicates that a line of text is to be accepted from the console input device. The format of the command is:

**.INPUT [<prompt text>]**

When the Script processor encounters an INPUT command processing of the current text file is suspended. If a prompting message is specified with the INPUT command it is displayed on the console output device. The Script processor then accepts one line of text from the console input device (unless there is data in an EXEC stack - see OASIS EXEC Language Reference Manual). Characters are accepted from the console until a carriage return is typed or 128 characters are entered, whichever occurs first. The text entered is then treated as if it came from the input text file.

After the data is entered processing of the input file continues. It is not

advisable to use the INPUT command when a Table of Contents has been specified as the console input will be requested on each pass. If the text entered in the second pass does not consist of the same number and length of words as entered in the first pass then the Table of Contents may not match the document.

When the input data is coming from an EXEC stack and two pass mode is in effect you must remember that the stack should have information in it for both passes.

## 2.20 Console Input Variable Command

The INPVAR command allows you to specify the value of a text variable at run time. The format of the command is:

**.INPVAR nn [<prompt text>]**

Where:

nn Indicates the variable number to be assigned the value entered from the console. Variables are numbered from 1 through 99.

prompt text Indicates the prompting message to be displayed on the console when the operator is asked to enter the new value.

When the Script processor encounters the INPVAR command processing of the text is suspended; the prompt text, if specified, is displayed on the console; the input prompt character ":" is displayed; the operator is allowed to enter up to 128 characters of text. After the operator has typed the carriage return key or entered the 128th character the text input is assigned to the variable number nn and processing continues. Future references to that variable number in the text will in fact refer to the text that was assigned to the variable.

The text that the operator enters may contain any acceptable embedded commands and/or references to other variables (refer to the chapter on "SPECIAL CHARACTERS").

Refer to the second example in the appendix "SCRIPT EXAMPLES" for an illustration of the usage of this command.

## 2.21 Text Justification Command

The JUSTIFY command indicates that lines of text, after filling, are to be left and right justified by insertion of spaces between words. The NOJUST command indicates that lines of text are not to have spaces added for justification of the line. The format of the commands is:

**.JUSTIFY**

**.NOJUST**

When the JUSTIFY command is specified fillmode is set. In this mode an output line is filled with whole words (the Script processor does not do hyphenation) from the input lines until there is insufficient space for the next word. (This is the function of FILL). If the line does not fill the entire line width the required number of spaces are distributed randomly between words, throughout the line. The preceding paragraph is an example of justified text.

## **2.22 Left Sided Page Footing Command**

The LFOOTING command allows you to specify a page footing for left sided pages. This footing may be different from the right sided page footing (RFOOTING). The format of the command is:

**.LFOOTING [<text>]**

The text of the LFOOTING command, similar to the FOOTING command, may use all of the allowed embedded command described in the chapter "Special Characters". Any underscoring and/or boldfacing specified in the text of this command is separate from the underscoring and/or boldfacing in the body of the page.

The FOOTING command discussed previously is essentially a combination of the LFOOTING command and the RFOOTING command.

To specify that no footing is to be output at the bottom of the left sided pages use the LFOOTING command with no text following.

## **2.23 Line of Characters Command**

The LINE command allows you to generate a full line of one character, from current left margin to current right margin. The format of the command is:

**.LINE [<char>]**

Where:

char      Is the character to be repeated across the entire line. When this character is not specified a line of spaces is generated.

The LINE command is useful for creating heading separators for tables and lists. As an example see the appendixes on "SCRIPT COMMAND SUMMARY" and "SPECIAL CHARACTER SUMMARY" at the back of this manual. The line of equal sign characters at the beginning and end of the two appendixes were generated with the LINE command.

The LINE command provides two advantages: 1) saves space in your text file because only seven characters are required to generate a full line of repetitive characters; 2) the length of the line generated is dynamic: adjusting the current left and right margin expands or contracts the length of the line.

## **2.24 Linking Segmented Files Command**

The LINK command indicates to the Script processor that the input text continues in another file. The format of the command is:

**.LINK <file name> [<file type>]**

When the Script processor encounters a LINK command any remaining lines of text in the current file are ignored. The file, specified by the file name and type is searched for on all of the disks. When the file is found the scripting process continues, using the new file as input. When the file type is omitted the default file type is used. The default file type is the type used by the SCRIPT command for the first file scripted.

**2.25 Left Margin Command**

The LMARGIN command specifies a new left margin to be used for following lines of the document. The format of the command is:

**.LMARGIN <value>**

The value specified may be an absolute column number (unsigned value) or a column relative to the current left margin (signed value). Column numbers are always relative to the document left margin established by the SIZE command.

If a left margin is specified that is to the left of the document left margin undefined results may be produced.

Setting the left margin with this command causes the INDENT value to be reset to zero.

**2.26 Left Sided Page Heading Command**

The LHEADING command allows you to specify a page heading for left sided pages. This heading may be different from the right sided page heading (RHEADING). The format of the command is:

**.LHEADING [<text>]**

The text of the LHEADING command, similar to the HEADING command, may use all of the allowed embedded commands described in the chapter "Special Characters". Any underscoring and/or boldfacing specified in the text of this command is separate from the underscoring and/or boldfacing in the body of the page.

The HEADING command discussed previously is essentially a combination of the LHEADING command and the RHEADING command.

To specify that no heading is to be output at the bottom of the left sided pages use the LHEADING command with no text following.

**2.27 Line Outside Gutter Command**

The OGUTTER command operates similar to the GUTTER command discussed previously but only effects the outside gutter of a line. The format of the command is:

**.OGUTTER [<char>]**

The outside gutter of a line is defined as the right area of a line on odd numbered pages and the left area of a line on even numbered pages.

## **2.28 Conditional Page Eject Command**

The PAGE command specifies to the Script processor that a page eject is to be performed in the document if there is less than a specific number of lines available on the current page. The format of the command is:

**.PAGE <value>**

The value specified indicates the number of lines that must be available on the current page for no page eject to occur. This value does not have any relation to the number of input lines that must fit on the current page.

## **2.29 Paragraph Spacing Command**

The PARASKIP command specifies the number of lines that are to separate each paragraph of the document. The format of the command is:

**.PARASKIP <value>**

When a paragraph break occurs the value specified is multiplied by the current line spacing count (see SPACE command) and the resultant number of blank lines are generated to separate the paragraph in the output document.

When the Script processor is first invoked the value of PARASKIP is one. To indicate that no spacing is to be performed between paragraphs specify a PARASKIP value of 0.

## **2.30 Paragraph Break Command**

A blank or empty input line of text indicates to the Script processor that a paragraph break is to be performed. This is only valid while fillmode is set. (When nofillmode is set the Script processor does not recognize paragraphs.)

When a paragraph break occurs the current paragraph is finished by outputting any words left over from the previous line with no filling or justification performed. (If no words are left over then nothing special is performed at this time.)

The current PARASKIP value is multiplied by the current SPACE value and the resultant number of blank lines are output to the document.

The following commands perform an implied paragraph break:

APPENDIX (after the appendix name is output)  
CENTER (before and after the centered text is output)  
CHAPTER (after the chapter name is output)  
SECTION (before and after printing section heading)

Certain Script commands cause the current paragraph to be terminated without causing a paragraph break or paragraph spacing. Terminating the current paragraph means that any words left over from the last line output are to be output now without any attempt to fill or justify the text. (Refer to the BREAK command.)

### 2.31 Pause Command

The PAUSE command specifies that the Script processor is to wait for a carriage return to be entered on the console input device before processing is continued. The format of the command is:

**.PAUSE [<prompt text>]**

When the Script processor encounters a PAUSE command any prompt message specified is displayed on the console, the bell is sounded, and all processing is suspended until a carriage return is entered on the console. This command would normally be used near the beginning of the document to allow an operator a chance to make final forms alignment changes, etc.

### 2.32 Position to Line Number Command

The POSITION command allows you to specify the line number to advance to on the current page. The format of the command is:

**.POSITION <value>**

Where:

value      Indicates the line number to be positioned to. This value must be within the range of 1 through the page size minus any footing lines (you cannot position past the start of the footing). Values outside of this range will cause a command error.

When the POSITION command is encountered and the value specified is greater than the current line number being printed blank lines will be forced out until the current line number equals the line number specified. If the line number specified is less than or equal to the current line number then the command is ignored.

### 2.33 Preface Command

The PREFACE command specifies the beginning of the document preface text. The format of the command is:

**.PREFACE**

When the PREFACE command is used it must be after a TITLE command and before any CONTENTS or CHAPTER commands. The PREFACE command indicates to the Script processor that all lines of text and commands following, up to the first occurrence of a CONTENTS or CHAPTER command are to be treated as a document preface.

A document preface has a default HEADING of the word "P R E F A C E", centered at the top of the page and a default FOOTING of a centered lower case Roman numeral page number, both in boldface.

## SCRIPT REFERENCE MANUAL

### 2.34 Read Variable Command

The READ command allows you to access an external file, read a record from that file, assigning the fields from the record to internal variables. The format of the command is:

**.READ nn[,nn]...**

Where:

nn Indicates the variable numbers that the fields are to be assigned to. Variables are number from 1 through 99. This command allows you to specify variable number zero (0) but it is used as a place-holder only--ignore field.

Before you can SCRIPT a text file that uses the READ command you must first assign I/O channel 3 to the file to be used (see "ASSIGN COMMAND" in the OASIS System Reference Manual). **The file must be an ASCII sequential format file.**

The field separator in this file is the comma character (,). If a field is to contain the comma character the entire field must be enclosed in quotation marks.

A standard usage for this command is in a text file that generates letters and needs the name and address information from a name and address system.

The READ command(s) should be one of the first commands in a Script file, before any text records.

For an example of this READ command refer to the second example in the appendix on "SCRIPT EXAMPLES" at the back of this manual.

### 2.35 Remark Command

The REMARK command allows you to add remarks to the text file. The format of the command is:

**.REMARK [<text>]**

Remarks might be used to denote places to be updated at a later time or comments explaining what or why certain commands were used, etc.

### 2.36 Right Sided Page Footing Command

The RFOOTING command allows you to specify a page footing for right sided pages. This footing may be different from the left sided page footing (LHEADING). The format of the command is:

**.RFOOTING [<text>]**

The text of the RFOOTING command, similar to the FOOTING command, may use all of the allowed embedded commands described in the chapter "Special Characters". Any underscoring and/or boldfacing specified in the text of this command is separate from the underscoring and/or boldfacing in the body of the page.

The FOOTING command discussed previously is essentially a combination of the LFOOTING command and the RFOOTING command.

To specify that no footing is to be output at the bottom of the right sided pages use the RFOOTING command with no text following.

### 2.37 Right Sided Page Heading Command

The RHEADING command allows you to specify a page footing for right sided pages. This footing may be different from the left sided page footing (LHEADING). The format of the command is:

**.RHEADING [<text>]**

The text of the RHEADING command, similar to the HEADING command, may use all of the allowed embedded commands described in the chapter "Special Characters". Any underscoring and/or boldfacing specified in the text of this command is separate from the underscoring and/or boldfacing in the body of the page.

The HEADING command discussed previously is essentially a combination of the LHEADING command and the RHEADING command.

To specify that no footing is to be output at the bottom of the right sided pages use the RHEADING command with no text following.

### 2.38 Right Margin Command

The RMARGIN command specifies a new right margin to be used for following lines output to the document. The format of the command is:

**.RMARGIN <value>**

The value specified may indicate an absolute column of the document (unsigned value) or a column relative to the current right margin (signed value). Absolute column references are always relative to the document left margin established by the SIZE command. It is advised that only the relative right margins be specified as this allows the format of the document to be changed by only changing the SIZE command.

Specifying a right margin that is to the left of the current left margin will produce undefined results.

### 2.39 Section Command

The SECTION command specifies the beginning of a subsection within a CHAPTER or APPENDIX. The format of the command is:

**.SECTION <level> <title>**

Where:

level Indicates the subsection level being started. A CHAPTER or APPENDIX may have up to four sub-levels of SECTIONS. The level of a SECTION is indicated by the number of periods used in the level field of the SECTION command. This is best illustrated with examples. If the current chapter number is 4 then the following level specifications in SECTION commands will have the indicated effect.

## SCRIPT REFERENCE MANUAL

Level	Effect
===== .+	===== Begin Section 4.1
..+	Begin Section 4.1.1
...+	Begin Section 4.1.2
....+	Begin Section 4.1.3
.....+	Begin Section 4.1.3.1
..+	Begin Section 4.1.4
..+	Begin Section 4.2
..+	Begin Section 4.2.1
5.1.9	Begin Section 4.2.2 (only periods are checked)
+	Begin new Chapter 5 (equivalent to CHAPTER command)

As can be seen the specification of a level resets all lower levels to zero. It is not necessary to use the plus sign, any character or characters other than a period or space would indicate the same. However, it is advised that you not try to specify the actual level of the section because any additions or deletions to the text may change the level number, thus making the maintenance of the text tedious.

**title** Indicates the title of the sub-section. Imbedded casemode commands may be used.

When the **SECTION** command is encountered the section number, title, and page number that the section starts on is saved in a temporary file for inclusion in the Table of Contents. (Occurs on pass one only.)

A **SECTION** command causes an implied **PAGE 4** command to be executed.

When a letter quality printer is used for the Script output section heading are always printed in boldface.

In addition the following events occur:

1. Paragraph break - finish current paragraph, perform any paragraph spacing.
2. Reset left and right margins to document left and right margin. Reset indent value to zero.
3. Output new section level on left side of document, followed by section title.
4. Paragraph break - perform any paragraph spacing.

**2.40 Set Command Character Command**

The SETCOMM command allows you to change the command character from the default period to any character you wish. The format of the command is:

**.SETCOMM [<char>]**

Where:

char Indicates the character that will subsequently be used as the first character of a command record. Specifying no character on this command indicates that the command character is to revert to the default.

The SETCOMM command should only be used when it is necessary for text records to start with the period character. Keep in mind that all commands following the SETCOMM command must start with the character specified, even the SETCOMM command that set the command character back to the default. For example:

```
.LM 10
.SETCOMM :
... The preceding is an elipsis.
... And so is this
:NOFILL
ABCDEFGHIJKLMNPOQRSTUVWXYZ
.|!"#$%&'()*
:SETCOMM
.FILL
```

**2.41 Page Number Set Command**

The SETPAGE command allows you to specify the page number to be used for the current page of output. The format of the command is:

**.SETPAGE nn**

Where:

nn Indicates the value that the page number is to be set to.

This command is very useful for printing updates to existing documentation text or manuals. Since the updated manual may have different page numbers than the existing version it is necessary to set the page number of the pages that will be printed as replacement pages.

When the updated version does not have different page numbers it is generally best to use the from-to option of the SCRIPT command instead of this command.

## SCRIPT REFERENCE MANUAL

### 2.42 Set Alternate Space Character Command

The SETSPACE command allows you to specify that a certain character is to be printed as a space. The format of the command is:

**.SETSPACE [<char>]**

Where:

char Is the character to be translated into a space character. When no character is specified on this command then there will be no alternate space character set.

This command is sometimes needed to print a space that is not to be treated as a word or sentence delimiter. Since the SCRIPT processor always treats the sequence of characters <period><space> as a sentence delimiter and since the SCRIPT processor always separates sentences with two spaces, it is useful to have a character that is not a <space> but is printed as such. For example, to print the name C. P. Williams without a double space after each of the period it is necessary to use the SETSPACE command. Remember to set the space character off when it is no longer needed or else there will be no way to print the character without it being translated to be a space.

### 2.43 Set Variable Command

The SETVAR command allows you to specify the value of a variable to be used later in the text file. The format of the command is:

**.SETVAR nn [<text>]**

Where:

nn Indicates the number of the variable to be set. This number must be between 1 and 99, inclusive.

text Indicates the value that the variable is to be set to. When no text is specified the variable will be set to the null or empty string of characters.

### 2.44 Document Size Command

The SIZE command specifies to the Script processor the document page size. The format of the command is:

**.SIZE [<lm>] [, [<rm>] [, [<top>] [, [<bot>]]]]**

Where:

lm Indicates the left margin column number (absolute).

rm Indicates the right margin column number (absolute).

top Indicates the top line of the document.

bot Indicates the bottom line of the document.

The SIZE command, when used, must be the first line of the text file specified in the SCRIPT command. If the SIZE command is specified in any other position of the input text file an error message is displayed on the console and the SCRIPT program is exited.

If the SIZE command is not specified then the <lm> parameter is set to 1, the <top> parameter is set to 1, the <rm> parameter is set to the linesize of the device specified in the attachment of that device, the <bot> parameter is set to the pagesize of the device specified in the attachment of that device.

Omitting any of the parameters indicates that the default value for that parameter is to be used. When a parameter is omitted but a following parameter is specified the commas must be used to indicate the position of the parameter. For example:

.SIZE 5,30	Default top and bottom
.SIZE ,,3,10	Default left and right
.SIZE 4,,,13	Default right and top

The values set with the SIZE command are those used for the entire document. Any use of the LMARGIN or RMARGIN commands are always relative to the values established by this command.

#### 2.45 Line Skipping Command

The SKIP command specifies to the Script processor that blank lines are to be output to the document. The format of the command is:

**.SKIP <value>**

The value specified is the number of blank lines to be generated as output to the document. This is the only means of outputting blank lines, other than normal line and paragraph spacing, while FILLmode is set. (When NOFILLmode is set input lines are output directly.)

The SKIP command does cause a paragraph break but paragraph spacing is not performed.

If a SKIP command causes a page eject to be output to the document any remaining lines to be skipped are suppressed after the page eject is output.

#### 2.46 Line Spacing Command

The SPACE command specifies to the Script processor the line spacing of the document. The format of the command is:

**.SPACE <value>**

The value indicates the line spacing. 1 is single spacing (no blank lines between printed lines); 2 is double spacing (one blank line between printed lines); etc. When the Script processor is first invoked the line spacing value is one.

Specifying a line spacing value of zero will be translated to a value of one. It is not possible to actually set the line spacing to zero as this would theoretically cause overprinting in the output document.

## SCRIPT REFERENCE MANUAL

### 2.47 Set Special Characters Command

The SPECIAL command allows you to specify most of the characters to be used for the special, embedded commands described in the chapter "Special Commands". The format of the command is:

**.SPECIAL <special characters>**

The <special characters> include the nine special characters that are used in most of the embedded commands. When the SPECIAL command is used all of the special characters must be specified, even if they are not changed.

The special characters that may be changed with this command include:

- @ Script escape character
- ^ Up shift character
- \ Down shift character
- % Tab character
- ~ Absolute tab character
- & Boldface character
- \_ Underscore character
- . Command character
- Alternate space character

The characters to be used for the above functions must be specified in the same sequence as listed.

The SPECIAL command is normally only used to change the absolute tab character (~) and the underscore character (\_) for those systems with console terminals that cannot generate the tilde and/or underscore character.

The SPECIAL command has no effect on the characters output (changing the tab character does not change the output of the percent character) but only on the embedded commands.

For example, after the command `.SPECIAL #^\%*&_.` the Script escape character will be the number symbol (#). To output an up-arrow character you would have to use the sequence: `#^`.

### 2.48 Tab Setting Command

The TABSET command specifies the column numbers to be used for tabulation. The format of the command is:

**.TABSET n1 [n2]...**

Up to sixteen tab positions may be set with the TABSET command. It is not necessary to set all of the positions when the TABSET command is used. If only two tab positions are set then the remaining fourteen positions are cleared.

The column numbers referenced with the TABSET command are relative to the document left margin, and are independent of the currently set LMARGIN value.

When the Script processor is first invoked all tab positions are cleared. For information regarding the use of the tab positions see the CTRL/I and % characters in the Imbedded Commands section.

### 2.49 Document Title Command

The TITLE command specifies the title of the document to the Script processor. The format of the command is:

**.TITLE <text>**

The TITLE command, when used, should be specified before any text is output to the document. The TITLE command actually performs two functions: specifies the title of the document and indicates the beginning of the title page text.

The title of a document should not include any special characters or embedded line feeds. However, it may use embedded casemode commands.

The text of a title page is specified after the TITLE command and before the first occurrence of one of the following commands: PREFACE, CONTENTS, CHAPTER, SECTION, APPENDIX. A title page has no default footing or heading.

### 2.50 Type Command

The TYPE command allows you to display a message on the console and to continue processing, without operator action. The format of the command is:

**.TYPE [<text>]**

Where:

text      Is the message to be displayed on the console.



## CHAPTER 3

### SPECIAL CHARACTERS

The OASIS Script processor uses the commands discussed in the previous chapter to format the output document. Additionally certain characters are used to perform formatting within a line of text. There are three groups of these special characters: embedded commands available only in HEADING, FOOTING, and CENTER text, embedded commands not available in HEADING, FOOTING, and CENTER text, and embedded commands available in all text.

An effort has been made to use characters that do not normally occur in the text of a document. However, if you do have need of using the characters as printing characters then use the Script escape character (@) before each occurrence of a special character. When a Script escape character is encountered by the Script processor it indicates that the character following is to have a different meaning. For example, to cause the output of the Script escape character you must use two escape characters: @@. To output a tilde character you would use an escape followed by the tilde: @~.

If your console terminal is incapable of generating one or more of these special characters refer to the SPECIAL command in the previous chapter.

#### 3.1 Embedded Commands Available in Heading, Footing, and Center

The following character sequences have a reserved meaning only when encountered in the text specified with a HEADING, LHEADING, RHEADING, FOOTING, LFOOTING, RFOOTING, or CENTER commands.

##### Tabulation (@,)

Tabulation in a HEADING or FOOTING has a different meaning than in the body of the document. In a HEADING or FOOTING there are two "tab" positions available. These "tab" positions are actually formatting positions: centered and right justified. The character sequence that indicates tabulation in a HEADING or FOOTING is @, (escape, comma). When the Script processor encounters the first occurrence of the tabulation sequence it indicates that the characters following are to be centered on the line. An example of this is the page number at the bottom of this page. When the Script processor encounters the second occurrence of the tabulation sequence it indicates that the characters following are to be right justified on the line. An example of this is the chapter name at the top right hand portion of the odd numbered pages in this manual.

##### New Lines (@;)

If the HEADING or FOOTING text is to specify a multiple line heading or footing the line feed character or the character sequence @; is used. When the Script processor encounters this character or characters the heading or footing line count is incremented by one and the characters following are left justified on the next line.

#### 3.2 Embedded Commands not Available in Heading, Footing, and Center Text

The following two embedded commands may be used in all text except in the HEADING, LHEADING, RHEADING, FOOTING, LFOOTING, RFOOTING, and CENTER commands.

## **SCRIPT REFERENCE MANUAL**

### **Tab to Tabstop Position (Z)**

To indicate that the next character is to be output at the next position established by the TABSET command the percent character (%) is used. Optionally you may use the control character: CTRL/I; however, this is not advised because the Editor displays the CTRL/I as spaces (except in modify mode) possibly causing confusion. If the output column pointer is past all of the positions established by the TABSET command or if there are no positions established the percent character is translated into a single space character.

### **Tab to Column Number (~)**

To indicate that the next character is to be output at a specific column number the tilde character (~) is used, followed by the number of the column. If the output column pointer is past the column specified then this command is ignored. For example: ~15 will position to column 15 (relative to the left margin of the SIZE command).

### **3.3 Embedded Commands Always Available**

The following characters and character sequences are available in all text, including the text of a HEADING, FOOTING, or CENTER command.

The first four of these embedded command characters relate to the casemode of characters output. These commands are ineffectual if the global UPPER option was specified in the CSI SCRIPT command. Additionally these commands are ineffectual in the body of the document if the casemode is Mixed. Their purpose is to facilitate the casemode specification for console devices that only have upper case character sets. Using these commands on that type of a console graphically displays the casemode of the text.

#### **Capitalize Next Character (^)**

To indicate that the next character is to be capitalized the up-arrow character (^) is used.

#### **Shift to Upper Case (^^)**

To indicate that the characters following are to be folded to upper case a double up-arrow character sequence is used (^^).

#### **Next Character Lower Case (\)**

To indicate that the next character is to be translated to lower case the backslash character (\) is used.

#### **Shift to Lower Case (\\)**

To indicate that the characters following are to be translated to lower case, a double backslash character sequence is used: (\\).

#### **Current Date (@D)**

To include the system date as part of the text the character sequence @D is used. This would be used for including the print date as part of the document. When encountered by the Script processor the current system date is converted to

normalized format (mm/dd/yy) and replaces the special characters in the text. The date always requires eight positions of space.

**Current Page Number (@P)**

To include the current page number as part of the text the character sequence @P is used. When the Script processor encounters this character sequence the current page number replaces these characters. No leading or trailing blanks are included but any leading zeroes of the page number are suppressed.

When a reference is made to the current page number outside of the commands: HEADING, LHEADING, RHEADING, FOOTING, LFOOTING, RFOOTING, the current page number is put into the text but this current page number may not be the actual current page number. This situation will only arise if the line containing the reference to the page number does not fit on the page and a page eject must be performed.

**Current Chapter Name (@S)**

To include the current chapter or appendix name in the text the character sequence @S is used. When the Script processor encounters this character sequence the characters are replaced by the following: the word "CHAPTER " (or "APPENDIX "), the current chapter number (or appendix letter), a colon, the chapter (or appendix) title.

**Current System Time (@T)**

To include the current system time in the text the character sequence @T is used. When the Script processor encounters this sequence the current system time is converted to normalized format and the seconds portion is removed (hh:mm). The resulting five characters replaces the special character sequence.

**Underscore ( \_ )**

To indicate that the characters following are to be underscored the underscore character is used ( \_ ). The first occurrence of the underscore character indicates the beginning of the characters to be underscored and the second occurrence of the underscore character indicates the end of the characters to be underscored. If the output device is not a letter quality printer (capable of physical backspacing) then this command is ignored. Refer to the chapter "ATTACH COMMAND" in the OASIS System Reference Manual for the class codes of known letter quality printers.

**Boldface ( & )**

To indicate that the characters following are to be boldfaced the ampersand character ( & ) is used. The first occurrence of the ampersand character indicates the beginning of the characters to be boldfaced and the second occurrence of the ampersand character indicates the end of the characters to be boldfaced.

This command is ignored if the output device is not a letter quality printer (capable of physical backspacing and software settable pitch control). Refer to the chapter "ATTACH COMMAND" in the OASIS System Reference Manual for the class codes of known letter quality printers.

## **SCRIPT REFERENCE MANUAL**

### **Variables (@n)**

To indicate that the current value of a variable is to be placed in the text the character sequence @n is used where the n is the number of the variable to be used. Variables are numbered from 1 through 99. Variables are defined by one or more of the commands: INPVAR, SETVAR, and READ.

## APPENDIX A

### SCRIPT EXAMPLES

#### A.1 Example 1: A Documentation Manual

The following example Script file is an exact copy of the first two segments used to generate this manual, up through Chapter 1. It illustrates many of the Script commands and, by referencing the first pages of this manual, the output generated by the commands and text.

```
.SIZE 6,88,1,60
.TITLE SCRIPT REFERENCE MANUAL
.SETVAR 97 SCRIPT
.SETVAR 98
.REMARK Variable 97 is the manual name (add space at end when revised)
.REMARK Variable 98 is the manual revision number
.POSITION 16
.CENTER &OASIS&
.SKIP 2
.CENTER &SCRIPT Processor&
.CENTER &Reference Manual&
.POSITION 28
.CENTER &Second Edition&
.COPY AUTHORS.COPY
.EJECT
Second edition, first printing: March, 1980
.COPY PROPRIET.NOTICE
.PREFACE
.SKIP 5
.SPACE 2
This document is a reference manual for the OASIS SCRIPT word
processing language. Prior knowledge in the use
of the OASIS EDIT program and the EXEC processor is assumed.
.SKIP 6
.CENTER &Related Documentation&
.SPACE 1
The following publication provides additional information
required in the use of the OASIS SCRIPT language:

.CENTER &OASIS System Reference Manual&
.CENTER &OASIS Text Editor Reference Manual&
.CENTER &OASIS EXEC Language Reference Manual&
.CONTENTS
.LHEAD &SCRIPT REFERENCE MANUAL&
.RHEAD @, @, &@S&
.COPY INTRO.SCRIPT
.COPY COMMANDS.SCRIPT
.COPY COMMAND2.SCRIPT
.COPY COMMAND3.SCRIPT
.COPY COMMAND4.SCRIPT
.COPY COMMAND5.SCRIPT
.COPY EMBEDDED.SCRIPT
.COPY EXAMPLE.SCRIPT
.COPY COPYEX1.SCRIPT
.COPY EX2.SCRIPT
.COPY SUMMARY.SCRIPT
.COPY COMMENT.FORM
```

## SCRIPT REFERENCE MANUAL

.END

.REMARK File INTRO.SCRIPT

.CHAPTER + INTRODUCTION

.RFOOT &@97@98@,- @P -@,@99&

.LFOOT &@99@,- @P -@,@97@98&

The OASIS Script processor is a powerful documentation processing system. The processor was not designed for general word processing applications but was designed for documentation and manual generation. (All of the OASIS documentation is generated and maintained via this Script processor.) However the Script processor can be used for many word processing applications.

The general features of the processor include provisions for creating and maintaining: page headings, page footings, line filled and right justified text, title pages, paragraph indentation, variable left and right margins, conditional page ejection, variable line and paragraph spacing, text input from console while processing, access to system date and time, Table of Contents, columnar tabulation, underscored and/or boldface printing, left/right page alignment, interface to external data file, etc.

.SECT .+ Creating A Script Text File

To create text and commands for the Script program to process you must create a file. This file can be created by various OASIS system programs or language processors: BASIC, MACRO, EDIT, TEXTEDIT, etc. Generally the easiest method is by using the EDIT program. The file created must be sequential in format.

The Script file may have any non-reserved file name and type and may reside on any disk drive (or multiple disk drives if the file is segmented).

Script files contain records, each record containing text or a Script command. A Script command record must start with the period character (.) and may only consist of the command word and parameters required by that command (do not mix text with the command record unless specified by a parameter of the command). Script commands are discussed in detail in the following chapter. Script text may not contain any Script commands but may contain embedded command characters. Script command characters are discussed in detail in the chapter "Special Characters".

Script text files may be segmented into several files and linked together for Script processing by using the Script LINK command.

"Boiler-plate" type documents (a document formed by piecing together sections of several documents) may be generated by using the COPY command.

.PAGE 8

.SECT .+ Invoking the Script Processor

After a Script file is created you must use the CSI SCRIPT command to process the file and generate the output document. The format of the SCRIPT command is:

.CENTER &SCRIPT <file-name> [<file-type>] [(option ...)]&

Where:

.LM 10

.RM -5

.IND -9

file-name Indicates the file name of the file to be processed.

file-type Indicates the file type. This field is optional and, when not specified, the file type of "SCRIPT" is used.

The file type used for the first file is the default file type for all subsequent LINK commands.

```
.LM 1
.RM +5
.PAGE 6
&SCRIPT Options&
```

The options available in the SCRIPT command include the following:

```
.LM 10
.RM -5
.IND -9
.SETSPACE #
_TYPE Indicates that the document generated is to be output to
the console output device. This is a default option.
```

\_WAIT Indicates that after a full page of text is printed the SCRIPT program is to pause, allowing the operator to insert a new page in the printer, etc. This option has no effect when the option FILE is used.

\_WORK=<fd> Indicates that the disk to be used for working storage is the disk currently mounted in the drive <fd>. When this option is not specified the system "S" disk will be used.

\_PRINT[n] Indicates that the document generated is to be output to the primary printer or PRINTERn.

\_CPI\_#nn Indicates that the printer has been set to nn characters per inch. This option is only necessary for letter quality printers. The default setting for CPI is 10.

\_REPEAT#nn Indicates that the report is to be produced nn times. This option should not be used when an external data file is used in the text output. The default REPEAT count is 1.

\_FILE Indicates that the document generated is to be output to the file ASSIGNED to channel 2. (See "ASSIGN COMMAND" chapter in the \_OASIS System Reference Manual\_.)  
To insure the proper document size and page numbering you should specify all of the document parameters with the SIZE command.

\_UPPER Indicates that all alphabetic characters output to the document are to be translated to their upper case equivalents. This option is normally used only when the output device does not have a lower case character set and is not capable of performing the translation itself.

nn#[nn] Indicates the from and to range of the printout. The first number indicates the page number of the first page to be printed, the second number indicates the page number of the last page to be printed. When this option is not used the entire report will be printed.

```
.SETSPACE
.LM 1
```

## SCRIPT REFERENCE MANUAL

.RM +5  
.PAGE 11

When the Script processor is first invoked by this command the following values are initialized:

.LM 25  
.IND -14  
.RM -5  
.PARA 0  
Cassmode Mixed (M)

Fillmode On

Justify On

Paraskip 1

Spacing 1

Size 1,80,1,60 (Reset to device parameters if option FILE not specified)

Page 1  
.PARA 1  
.LM 1  
.RM +5

As noted above, when the output from the Script processor is to go to a disk file, the file to be used must be previously assigned (see "ASSIGN COMMAND" in the \_OASIS System Reference Manual\_) to channel 2.

If the text to be processed contains one or more READ commands the data file to be used must be previously assigned to channel 3.

The OASIS Script processor is a virtual processing language. This means that the Script processor can process a text file that requires more space than is available in the main memory. The Script processor uses the random access mass storage memory (disk) as an extension of its memory.

The OASIS Script processor is also an optional two pass processor. When two pass processing is specified (see TITLE and CONTENTS commands in following chapter) the Script processor reads and processes the entire text file in the first pass without creating an output document but creating a temporary work file containing the information necessary to create a Table of Contents. During the second pass the text file is reprocessed and the Table of Contents is included, when specified.

When the output is directed to a printer or a disk file (options PRINT or FILE) the Script processor will display on the screen the page number being processed and any LINK commands encountered. This facilitates your monitoring the progress of the text generation.

.END

**A.2 Example 2: Form Letter**

.SIZE 5,72,6,60  
.PAUSE Please align forms so top-of-form is at physical top-of-form.  
.NOFILL  
.LM 44  
Phase One Systems, Inc.  
7700 Edgewater Drive #830  
Oakland, CA 94621

.INPUT Please enter today's date: month name day, year:

.LM 1  
.READ 1,2,3,4  
@1  
@2  
@3

.FILL  
Dear @4:

Phase One Systems appreciates your inquiry regarding the OASIS operating system.

The OASIS operating system is a complete, sophisticated system for the development and operation of integrated application systems on a Z80-based microcomputer system. OASIS has been implemented on several different hardware configurations including Micropolis mini-disk, Shugart single and double density disk, Shugart double sided disk and hard disk systems. Phase One Systems is in the process of converting OASIS for many other hardware configurations.

OASIS includes an integrated file management system to manipulate sequential, direct, and indexed access disk files. Several types of disk drives can be accessed by OASIS including Shugart, PerSci, Micropolis, CalComp, and Pertec (hard disk) drives.

The basic OASIS operating system is a DOS system including utility programs to initialize disks, create, list, erase, copy, and move files, a file editor, and various device drivers.

Available with OASIS is a BASIC interpreter, a macro assembler, a relocatable linkage editor for assembly programs, a text processor, an interactive debugger, and communications programs.

etc. etc. etc.

.LM 44  
.NOFILL  
Sincerely,

**SCRIPT REFERENCE MANUAL**

Howard J. Sidorsky  
President  
.LM 1

Enclosures (3)

cpw:HJS

The following page is a listing of the output generated by the previous example (assuming only one record in the external file) along with the console display:

>ASSIGN 3 NAMES FILE A

>SCRIPT EXAMPLE2 (PRINT

Please align forms so top-of-form is at physical top-of-form.  
Please enter today's date: month-name day, year: March 24, 1980

>

**APPENDIX A: SCRIPT EXAMPLES**

Phase One Systems, Inc.  
7700 Edgewater Drive #830  
Oakland, CA 94621

March 24, 1980

John Q. Public  
1234 SE Main Street  
Oakland, CA 94625

Dear Sir:

Phase One Systems appreciates your inquiry regarding the OASIS operating system.

The OASIS operating system is a complete, sophisticated system for the development and operation of integrated application systems on a Z80-based microcomputer system. OASIS has been implemented on several different hardware configurations including Micropolis mini-disk, Shugart single and double density disk, Shugart double sided disk and hard disk systems. Phase One Systems is in the process of converting OASIS for many other hardware systems.

OASIS includes an integrated file management system to manipulate sequential, direct, and indexed access disk files. Several types of disk drives can be accessed by OASIS including Shugart, PerSci, Micropolis, and Pertec (hard disk) drives.

The basic OASIS operating system is a DOS system including utility programs to initialize disks, create, list, erase, copy, and move files, a file editor, and various device drivers.

Optionally available with OASIS is a BASIC interpreter, a macro assembler, a relocatable linkage editor for assembly programs, a text processor, an interactive debugger, and communications programs.

etc. etc. etc.

Sincerely,

Howard J. Sidorsky  
President

Enclosures (3)

cpw:HJS



## APPENDIX B

### SCRIPT COMMAND SUMMARY

Command	Function
.ALIGN <R L> [<text>]	Perform conditional page alignment.
.APPENDIX <level> <title>	Begin new appendix.
.BREAK	Perform text break.
.CASE U L M	Set casemode.
.CENTER <text>	Center text.
.CHAPTER <level> <title>	Begin new chapter.
.CONTENTS	Insert Table of Contents.
.COPY <file name>	Switch processing to alternate file.
.CPI <value>	Set character per inch value.
.END	Logical end of report.
.EJECT	Begin new page.
.FILL	Set fillmode on.
.FOOTING <text>	Establish page footing.
.FORMAT <text>	Format text.
.GUTTER <char>	Print in line gutters.
.HEADING <text>	Establish page heading.
.IGUTTER <char>	Print in inside line gutter.
.INDENT [+ -]<value>	Set paragraph indent value.
.INPUT [<prompt text>]	Accept text from console.
.INPVAR nn [<prompt text>]	Accept variable from console.
.JUSTIFY	Set justify mode on.
.LFOOTING <text>	Establish left page footing.
.LHEADING <text>	Establish left page heading.
.LINE <char>	Generate full line of one character.
.LINK <file name>	Continue processing in another file.
.LMARGIN [+ -]<value>	Set left margin.
.NOFILL	Set fillmode off.
.NOJUST	Set justify mode off.
.OGUTTER <char>	Print in outside line gutter.
.PAGE <value>	Conditional page eject.
.PARASKIP <value>	Set paragraph spacing value.
.PAUSE [<prompt text>]	Wait for operator.
.POSITION <value>	Position to line number.
.PREFACE	Begin preface pages.
.READ nn[,nn]...	Read variables from external file.
.REMARK [<text>]	
.RFOOTING <text>	Establish right page footing.
.RHEADING <text>	Establish right page heading.
.RMARGIN [+ -]<value>	Set right margin.
.SECTION <level> <title>	Begin sub-section.
.SETCOMM [<char>]	Change/restore command start character.
.SETPAGE nn	Change current page number.
.SETSPACE [<char>]	Change/restore alternate space character.
.SETVAR nn [<text>]	Change value of variable.
.SIZE [<lm>][,<rm>][,<top>][,<bot>]]]	Set document size.
.SKIP <value>	Output blank lines.
.SPACE <value>	Set line spacing value.
.SPECIAL <characters>	Change special character assignments.
.TABSET [n1 [n2]...]	Set tabstop positions.
.TITLE <text>	Set document title.
.TYPE [<text>]	Display message on console.

## APPENDIX C

### SPECIAL CHARACTER SUMMARY

#### Character Function

---

@D	Current system date
@P	Current page number
@S	Current chapter or appendix name
@T	Current system time
@,	Center or right justify (only in HEADING, FOOTING)
@;	New line (only in HEADING, FOOTING)
~	Capitalize next character
^^	Switch to upper case mode
\	Next character lower case
\\	Switch to lower case mode
%	Tab to tabstop position
CTRL/I	Tab to tabstop position
~	Tab to absolute column number
@	Next character not treated as special character
_	Toggle underscore switch
&	Toggle boldface switch
@1	Current value of variable one.
@16	Current value of variable sixteen.
.	
.	
.	
@99	Current value of variable ninety-nine.

---

**Reader's Comments**

Name \_\_\_\_\_ Date \_\_\_/\_\_\_/\_\_\_  
Organization \_\_\_\_\_  
Street \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_ Zip \_\_\_\_\_

Name of manual: \_\_\_\_\_

Did you find errors in this manual? If so, specify with page number.

---

---

---

---

---

---

---

---

Did you find this manual understandable, usable, and well-organized?  
Please make suggestions for improvement.

---

---

---

---

---

---

---

---

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

---

---

---

---

---

---

---

---

Indicate the type of user/reader that you most nearly represent:

- Assembly language programmer
- Higher-level language programmer (BASIC, FORTRAN, etc.)
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Non-programmer interested in computer concepts and capabilities
- Data entry operator

Mail to: OASIS Documentation  
Phase One Systems, Inc.  
7700 Edgewater Drive #830  
Oakland, CA 94621

