SHARE PROGRAM LIBRARY AGENCY

PROGRAM NUMBER

035008

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257

SHARE PROGRAM LIBRARY SUBMITTAL FORM

SHARE PROGRAM LIBRARY AGENCY
Triangle Universities Computation Center
Post Office Box 12076
Research Triangle Park, North Carolina
27709 USA
Attention: Mr. Joe Ragland

SPLA CONTROL NUMBER:

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the "SHARE Program Library Standards Manual".

(1) Program Number (to be filled in by SPLA) 360D-03.5.008

(2) System Type (machine) 370

(3) Search Key ..... NSCRIPT- PRODUCES TEXT DATASETS in MANUSCRIPT FORM

(4) Programming Language ..... ASSEMBLER

(5) Author's Name and Address ..... Dr. William Dwyer
Math Dept. Yale University
New Haven, Ct.

(6) Direct Inquiries to Name and Address (if different than Author) ..... Mr. Robert C. Daley
Director of PDO
39-451, MIT
77 Mass. Ave. Cambridge, Ma. 02139

(7) Title of Program ..... NSCRIPT - produces text datasets in manuscript form

(8) Submitter's Installation Membership Code..... MI

(9) Submitter's Own Program Identification and Suffix(Optional)...

(10) Primary Subject Code..... 03 05

(11) Operating or Monitor System Required OS/360

(12) New or Revision Code (if revision, show prior Program Number in Item 1)..

(13) Year Completed..... 1973

(14) Date of Submittal..... DECEMBER 1973

(15) Documentation (number of original pages submitted)..... 111

(16) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

SHARE PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide:

a. Purpose
b. Programming Language used
c. Version and modification level or release number
d. Field of application
e. Type of routine (main program, subroutine, etc.)
f. Specific description of machine requirements

ABSTRACT

The quality and accuracy of a document depend greatly on the ease with which revisions can be made to the document. This statement is particularly true of technical documentation (into which class the present manual falls), which should always accurately reflect the status of the things they describe.

It is natural, particularly at MIT, that computer software solutions to the problems of document production should be devised. CTSS's "runoff", MULTICS's "runoff", CMS's "SCRIPT" represent similar such solutions.

"NSCRIPT" is an outgrowth of SCRIPT intended for use under CMS on a System/360 model 67 running under CP/67. It's set of command words encompasses most of those belonging to SCRIPT, MULTICS's "runoff", and TSO's FORMAT. In most cases they perform identical functions and have the same symbolic notation.

NSCRIPT running under 360/OS/TSO, which was developed by the MIT Programming Development Office, has the same outward appearance as it did when running under CMS. Except for the fact that OS I/O conventions make it somewhat more

(Please attach additional pages if necessary).....Total pages attached 1

Permission to Publish

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program."

(17) Signature of Submitter and Date Robert C Daley 11/27/73

(18) Signature of Installation Addressee Joseph R. Steinberg 1/7/74

difficult to use, it still has all the capabilities that it had with CMS.

Several powerful features are available with NSCRIPT that are not available with TSO's FORMAT:

1. The ability to enter footnotes at convenient places in the input. Footnotes are saved up and printed at the bottoms of output pages.

2. The ability to use symbolic "reference names" to simplify numbering and cross-referencing.

3. The ability to define both heading and footing lines for both even- and odd-numbered pages.

4. The ability to specify format control information or text dynamically (during printout).

5. The ability to use Roman Numerals (instead of Arabic) in page numbers and, in conjunction with the heading and footing controls, to place page numbers in a variety of places on the output page.

6. The ability to specify translation table pairs.

7. The ability to control the output conditionally.

8. The ability to define "remote sequences", which are invoked at specified places in the output.

NSCRIPT consists of 2 modules, a command processor (prompter) for use with TSO, and a program for processing NSCRIPT files which can be invoked either by the TSO prompter of by a batch job.

Since NSCRIPT can treat TAB characters internally, several (optional) modification to the TSO EDITOR for TAB processing are included. The modifications include the addition of a SCRIPT file type which is similar to a text file type except for line length and TAB processing.

Massachusetts Institute of Technology
Programming Development Office
Cambridge, Massachusetts 02139

To: Recipients of the NSCRIPT program

Date: December 13, 1973

Subject: NSCRIPT installation tape

The NSCRIPT distributation tape contains 4 physical files on a standard labeled tape (VOL=SER=SCRIPT). The first file is a sequential file and contains sample JCL for installing the NSCRIPT program and for making optional modifications to TSO for a SCRIPT dataset type and for special tab processing for SCRIPT dataset types. The first file contains approximately 130 card images and has a block size of 1600.

The remaining three files are partitioned datasets unloaded from 2314's using IEHMOVE.

The files and their contents are listed below.

File 1 JCL DSNAME=SCRIPT.JCL

File 2 Load library DSNAME=SCRIPT.LOAD

NSCRIPT	NSCRIPT command processor
SCRIPT	NSCRIPT program

File 3 Source library DSNAME=SCRIPT.SRC

NSCRIPT	NSCRIPT command processor
NSCRIPTA	
SCRIPT	NSCRIPT program
SCRIPTFO	
SCRIPTLK	
SCRIPTFD	
SGIKJOEB	IEBUPDTE deck for R21.7 MODGEN for TAB modification
MITEBELE	TAB modification R21.7
IKJEBELE	Zap for TAB modification R21.7

File 4 NSCRIPT document DSNAME=MEMO.SCRIPT

The library from which the NSCRIPT memo can be produced in batch or from TSO.

The TSO command to produce it is as follows:

```
nscript memo(nscript) lib offline twopass
```

This will produce a dataset MEMO.LIST which should be printed with a TN train.

To use the SCRIPT file type, it must be defined at SYSGEN with the following attributes:

```
DSTYPE    SCRIPT
FIXED     0-252
BLOCK     1680
FORMAT    VAR
VAR       252-255
CONVERT   ASIS
PRMPTR    NSCRIPT
USERSRC   DATASET
```

The TAB modification is composed of two parts. First, the TSO editor is modified (module IKJEBELE) to process tab characters differently if the dataset type is SCRIPT and TAB OFF has been specified. Under these conditions, the EDITOR will not replace tab characters with a number of spaces, but will leave the tab character intact in the line. The tab character prints as a colon when listed; however, the NSCRIPT program will recognize the tab characters and process them according to the '.tb' control word. The second part of the TAB modification is in the SYSGEN specification of the SCRIPT dataset type. There is no SYSGEN option to specify TAB OFF as a default TAB specification nor is there a means to specify the tab locations for user defined dataset types. A modification the the MODGEN macro SGIKJ0EB tests for a dataset type of SCRIPT and forces a default tab specification of TAB OFF and locations 10,20,30,40,50,60, and 70. This modification allows the SCRIPT dataset type to automatically retain tab characters in the input lines unless the user specifies TAB ON during his edit session. By specifying TAB ON, he reverts to translating tab characters to spaces assuming tab locations of 10,20,30,40,50,60,and 70.

The SCRIPT dataset type and the TAB modification are optional and in no way affect the performance of the NSCRIPT program or command processor.

Attached is a listing of the card images on the first file of the tape.

SCRIPT.JCL

```
//*                                                                     00000010
//*       INSTALL NSCRIPT  PROGRAM                                      00000020
//*                                                                     00000030
//*       PRE-ALLOCATE THE DATASETS                                     00000040
//*                                                                     00000050
//   EXEC  PGM=IEFBR14                                                  00000060
//A  DD  DSN=SCRIPT.LOAD,UNIT=2314,SPACE=(TRK,(20,20,2)),               00000070
//       DISP=(NEW,CATLG),                                             00000080
//       VOL=SER=234106                                                00000090
//B  DD  DSN=SCRIPT.SRC,UNIT=2314,SPACE=(TRK,(100,20,2)),               00000100
//       DISP=(NEW,CATLG),                                             00000110
//       VOL=SER=234106                                                00000120
//C  DD  DSN=MEMO.SCRIPT,UNIT=2314,SPACE=(TRK,(40,20,5)),               00000130
//       DISP=(NEW,CATLG),                                             00000140
//       VOL=SER=234106                                                00000150
//*                                                                     00000160
//*       RESTORE FROM TAPE                                             00000170
//*                                                                     00000180
// EXEC PGM=IEHMOVE                                                    00000190
//SYSPRINT DD SYSOUT=A                                                 00000200
//SYSUT1  DD  UNIT=SYSDA,VOL=SER=234105,DISP=OLD                       00000210
//TP DD UNIT=TAPE9,LABEL=(,SL),DCB=DEN=2,DISP=(OLD,PASS),              00000220
//   VOL=SER=SCRIPT                                                    00000230
//DISK001  DD  UNIT=2314,VOL=SER=234106,DISP=OLD                       00000240
 COPY DSNAME=SCRIPT.SRC,TO=2314=234106,FROM=TAPE9=(SCRIPT,2),FROMDD=TP 00000250
 COPY DSNAME=SCRIPT.LOAD,TO=2314=234106,FROM=TAPE9=(SCRIPT,3),FROMDD=TP 00000260
 COPY DSNAME=MEMO.SCRIPT,TO=2314=234106,FROM=TAPE9=(SCRIPT,4),FROMDD=TP 00000270
//*                                                                     00000280
//*    ASSEMBLE AND LKED THE SOURCE                                     00000290
//*                                                                     00000300
//   EXEC ASMFC                                                        00000310
//ASM.SYSIN  DD  DSN=SCRIPT.SRC(NSCRIPT),DISP=SHR                      00000320
//   EXEC  ASMFCL                                                      00000330
//ASM.SYSIN  DD  DSN=SCRIPT.SRC(NSCRIPTA),DISP=SHR                     00000340
//LKED.SYSLMOD  DD  DSN=&&GOSET(NSCRIPT)                               00000350
//LKED.SYSIN  DD  *                                                   00000360
 ENTRY NSCRIPT                                                       00000370
//*                                                                     00000380
//*                                                                     00000390
//   EXEC  ASMFC                                                       00000400
//ASM.SYSIN  DD  DSN=SCRIPT.SRC(SCRIPT),DISP=SHR                       00000410
//   EXEC  ASMFC                                                       00000420
//ASM.SYSIN  DD  DSN=SCRIPT.SRC(SCRIPTFC),DISP=SHR                     00000430
//   EXEC  ASMFC                                                       00000440
//ASM.SYSIN  DD  DSN=SCRIPT.SRC(SCRIPTLK),DISP=SHR                     00000450
//   EXEC  ASMFCL                                                      00000460
//ASM.SYSIN  DD  DSN=SCRIPT.SRC(SCRIPTRD),DISP=SHR                     00000470
//LKED.SYSLMOD  DD  DSN=&&GOSET(SCRIPT)                                00000480
 ENTRY SCRIPT                                                        00000490
//*                                                                     00000500
//*       ASSEMBLE MIT TAB PROCESSING MODIFICATION                      00000510
//*                                                                     00000520
//*                                                                     00000530
//MIT  EXEC  ASMFCL,PARM.LKED='LIST,LET,XREF,RENT,REUS,REFR'           00000540
//ASM.SYSIN  DD  DSN=SCRIPT.SRC(MITERELE),DISP=SHR                     00000550
//LKED.SYSLMOD  DD  DSN=&&GOSET                                        00000560
//LKED.PASELIB  DD  DSN=SYS1.CMDLIB,DISP=SHR                           00000570
```

SCRIPT.JCL

```
//LKED.SYSIN  DD  *                                                     00000580
 INCLUDE BASELIB(IKJEBELE)                                              00000590
 NAME IKJEBELE(R)                                                       00000600
 REPLACE MITEBELE                                                       00000610
 INCLUDE SYSLMOD(IKJEBELE)                                              00000620
 INCLUDE SYSLMOD(IKJEBELE)                                              00000630
 ENTRY IKJEBELE                                                         00000640
 IDENTIFY MITEBELE('TAB MODIFICATION')                                  00000650
 NAME IKJEBELE(R)                                                       00000660
//*                                                                     00000670
//*    APPLY ZAP TO ACTIVATE TAB MODIFICATION                           00000680
//*                                                                     00000690
//  EXEC  PGM=IMASPZAP                                                  00000700
//SYSPRINT  DD  SYSOUT=A                                                00000710
//SYSLIB    DD  DSN=&&GOSET,UNIT=2314,                                  00000720
//              DISP=(OLD,PASS)                                         00000730
//SYSIN     DD  DSN=SCRIPT.SRC(IKJEBELE),DISP=SHR                       00000740
//*                                                                     00000750
//*                                                                     00000760
//*                                                                     00000770
//*   LIST THE UPDTS DECK FOR SGIKJEB                                   00000780
//*                                                                     00000790
//*        THIS UPDATE TO MODGEN SHOULD BE APPLIED PRIOR TO A SYSGEN -  00000800
//*             THE SYSGEN SHOULD INCLUDE A SPECIFICATION FOR A SCRIPT  00000810
//*             DATASET TYPE (SEE COVER LETTER OR BELOW FOR ATTRIBUTES) 00000820
//*                                                                     00000830
//*        IF IT IS NOT POSSIBLE TO INSTALL THE SCRIPT DATASET          00000840
//*        TYPE AT A SYSGEN, THERE ARE 2 POSSIBILITIES.                 00000850
//*                                                                     00000860
//*        1.   EXTRACT THE EDIT MACRO OUTPUT FOR IKJEBEPD FROM STAGE 2;00000870
//*             RECODE IT TO INCLUDE THE SCRIPT DATASET TYPE            00000880
//*             WITH THE FOLLOWING ATTRIBUTES:                          00000890
//*                                                                     00000900
//*                  DSTYPE - SCRIPT                                    00000910
//*                  FIXED  - 0-252                                     00000920
//*                  BLOCK  - 1630                                      00000930
//*                  FORMAT - VAR                                       00000940
//*                  VAR    - 252-255                                   00000950
//*                  CONVERT- ASIS                                      00000960
//*                  PROMPTF - NSCRIPT                                  00000970
//*                  USERSRC- DATASET                                   00000980
//*                                                                     00000990
//*                  THEN UPDATE MODGEN WITH THE UPDTS LISTED HERE,     00001000
//*                  AND REASSEMBLE IKJEBEPD USING THE UPDATED MODGEN.  00001010
//*                                                                     00001020
//*        2.   IF THERE IS ROOM IN THE IKJEBEPD CSECT OF IKJEBEPS,     00001030
//*             SUPERZAP THE CORRECT CODES INTO THE TABLE.              00001040
//*                                                                     00001050
//  EXEC  PGM=IEBPTPCH                                                  00001060
//SYSPRINT  DD  SYSOUT=A                                                00001070
//SYSUT1    DD  DSN=SCRIPT.SRC,DISP=SHR                                 00001080
//SYSUT2    DD  SYSOUT=A                                                00001090
//SYSIN  DD  *                                                          00001100
 PUNCH TYPORG=PO,MAXNAME=4                                              00001110
 MEMBER NAME=SGIKJOEB                                                   00001120
//*                                                                     00001130
//*       PRODUCE THE NSCRIPT MANUAL IN BATCH                           00001140
```

SCRIPT.JCL

```
//*                                                                    0000115C
//  EXEC  PGM=SCRIPT,PARM='CENTERED2PASS'                              0000116C
//STEPLIB  DD  DSN=SCRIPT.LOAD,DISP=SHR                                0000117C
//SYSPRINT DD  SYSOUT=A,DCB=BLKSIZE=133                                0000118C
//CONTROL  DD  DSN=MEMC.SCRIPT(CONTROL),DISP=SHR                       0000119C
//SYSLIB   DD  DSN=MEMC.SCRIPT,DISP=SHR                                0000120C
//SCRIPTIN DD  DSN=MEMC.SCRIPT(NSCRIPT),DISP=SHR                       0000121C
//*                                                                    0000122C
//*          THE SCRIPT FILE SHOULD BE PRINTED ON A TN TRAIN           0000123C
//*                                                                    0000124C
//SCRIPT   DD  SYSOUT=A,DCB=BLKSIZE=2000                               0000125C
```

On November 8, 1978 I received and installed program number 360D-03.5.008 which is the text editor package called NSCRIPT. When testing began I kept getting program checks whenever the 'SR' control work was encountered.

All vitrual storage (VS) systems use the high order byte of register 15 as a flag byte; in the initialization section of NSCRIPT it loads 'R15' into 'BASE' as a base register. In the 'SR' control word source code there are 'LNR' and 'LPR' operations on the base register, thus the module loses its addressability. The fix for this is to change the 'LR' (load register) to a 'LA' (load address) operation in the initialization SECTION OF NSCRIPT.

IE. source statement number 123

```
LSR BASE,R15      set base register
Change To:
LA BASE,0(0,15) set base register
```

Kent White
Technical Services
ACT Computer Services Ltd.
16900-107 Avenue
P.O. Box 2405
Edmonton, Alberta
Canada T5J 2S1

Applications Program Series
AP-55 revision 2

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
INFORMATION PROCESSING CENTER

NOVEMBER 21, 1973

NSCRIPT REFERENCE GUIDE

TABLE OF CONTENTS

Line Format Control Words:

Margin and Line Spacing Control Words:

Page Control and Numbering Control Words:

Special Text Handling Control Words:

Symbolic Reference Control Words:

Appendices:

PREFACE

The quality and accuracy of a document depend greatly on the ease with which revisions can be made to the document. This statement is particularly true of technical documentation (into which class the present manual falls), which should always accurately reflect the status of the things they describe.

It is natural, particularly at MIT, that computer software solutions to the problems of document production should be devised. CTSS's "runoff", MULTIC's "runoff", CMS's "SCRIPT" represent similar such solutions.

This manual describes "NSCRIPT", an outgrowth of SCRIPT intended for use under CMS on a System/360 model 67 running under CP/67. Its set of command words encompasses most of those belonging to SCRIPT, MULTIC's "runoff",and TSO's FORMAT. In most cases they perform identical functions and have the same symbolic notation.

NSCRIPT running under 360/OS/TSO, which was developed by the MIT Programming Development Office, has the same outward appearance as it did when running under CMS. Except for the fact that OS I/O conventions make it somewhat more difficult to use, it still has all the capabilities that it had with CMS.

Several powerful features are available with NSCRIPT that are not available with TSO's FORMAT:

(1) The ability to enter footnotes at convenient places in the input. Footnotes are saved up and printed at the bottoms of output pages.

(2) The ability to use symbolic "reference names" to simplify numbering and cross-referencing.

(3) The ability to define both heading and footing lines for both even- and odd-numbered pages.

(4) The ability to specify format control information or text dynamically (during printout).

(5) The ability to use Roman Numerals (instead of Arabic) in page numbers and, in conjuction with the heading and footing controls, to place page numbers in a variety of places on the output page.

(6) The ability to specify translation table pairs.

(7) The ability to control the output conditionally.

(8) The ability to define "remote sequences", which are invoked at specified places in the output.

Other slight modifications have been made and are fully described in the body of this manual.

NOTES ON THIS DOCUMENT

Footnote Numbering:

In order to allow the greatest flexibility in the placement and numbering of footnotes and in order to take advantage of the superscript characters available on an IBM 1403 printer with a TN train mounted, several things were done:

(1) A footnote counter was kept as two reference names, n0 and n1, which represented the units and tens position of a decimal number, respectively. At the beginning of output, n0 and n1 were both given the value 0. Just prior to each footnote reference, the control line ".im add" was used to cause the following script file to be "executed":

```
.ur .sr n0 &n0 + 1
.ur .ur .sr n1 &n1 + 0&&&n0.x
.ur .ur .sr n0 &n0 - &&&n0.x.0
.ur .sr nfn &n1.&n0.
.ur .ur .sr snfn '&&&n0..'
.ur .ur .&&&n1.y.sr snfn '&&&n1..&snfn.'
```

(2) A file named "SUPER SCRIPT" was created which contained the following:

```
.sr / '&(.&snfn'
.sr // '.&).'
.sr ( '('
.sr ) ')'
.sr 0 '0'
   .
   .
   .
.sr 9 '9'
```

The reference names "/", "//", "10x", and "0y" were given the initial values "(&nfn.)", "", "1", and "'cm'", respectively.

At the beginning of output the user was asked (via .PR) to enter (via .RC) the line ".im super" if output was to the off-line printer or "" (nothing) otherwise.

(3) Each footnote reference of the form "<word><ref>" was entered as follows:

```
.ur .ur .ur <word>&/.&//.
```

CENTER
CE

Causes output to be centered horizontally on the page. This option is recommended for output printed on the off-line printer but if it is used to center on-line output it will increase the time required to type the output by at least 50 percent.

CONTROL(* or control-dataset-name)

Control-dataset-name defines the dataset read at the request of .RC control words. If the control dataset name is unquoted and has only a single index level, it is suffixed with ".TEXT" by the system (i.e., NAME becomes NAME.TEXT). If the CONTROL option is not specified, the default is that input will be accepted from the terminal.

LIBRARY(dataset-name-list)

This option specifies that all of the datasets in the dataset-name-list are to be concatenated into a single SYSLIB for use by NSCRIPT. The SYSLIB dataset is searched for members specified in .AP (apend) or .IM (imbed) control words. The datasets must be partitioned. If any dataset name is unquoted and contains only a single index level, it is suffixed with ".TEXT". If the LIBRARY option is entered without a dataset name list, the primary input dataset (which in this case must be partitioned) is passed to NSCRIPT as a single SYSLIB. (Note that the input dataset name must point to a specific member of this partitioned dataset.)

MESSAGE(* or message-dataset-name)

Message-dataset-name defines the dataset in which NSCRIPT will place error messages. If this operand is not specified, MESSAGE(*) is the default, and messages will be printed on the terminal. If the message dataset name is unquoted and has only a single index level, it is suffixed with ".MESSAGE".

which produces a reference of the form "<word>(1)" for on-line output or <word>[1] for off-line output.

(4) Each footnote began:

.ur (&nfn.) ...

Table of Contents:

(1) Each line in the table of contents was a ".UR" containing an instance of a mnemonic reference name, which was set to the page number of the start of the section via ".sr ... %".

(2) The TWOPASS option was used to produce the copy you are reading.

Manual Dating:

The date which appears on the cover sheet was entered via .RC (READ CONTROL line) to insure accuracy.

File Organization:

The following members of a TEXT dataset were used during production:

NSCRIPT	Contained general description, options, usage, and control word summary.
NSCRTC	Contained cover sheet, table of contents, preface, and this section. Imbedded by NSCRIPT.
AP, etc.	Contained detailed descriptions of all control words. Imbedded by NSCRFC1.
NSCRFC1	Contained necessary ".SR", ".PA", and ".IM" control words to cause insertion of the individual control word descriptor files and to generate values for reference names used in the table of contents. Imbedded by NSCRIPT.
SUPER	As described above.

CENTER
CE

Causes output to be centered horizontally on the page. This option is recommended for output printed on the off-line printer but if it is used to center on-line output it will increase the time required to type the output by at least 50 percent.

CONTROL(* or control-dataset-name)

Control-dataset-name defines the dataset read at the request of .RC control words. If the control dataset name is unquoted and has only a single index level, it is suffixed with ".TEXT" by the system (i.e., NAME becomes NAME.TEXT). If the CONTROL option is not specified, the default is that input will be accepted from the terminal.

LIBRARY(dataset-name-list)

This option specifies that all of the datasets in the dataset-name-list are to be concatenated into a single SYSLIB for use by NSCRIPT. The SYSLIB dataset is searched for members specified in .AP (apend) or .IM (imbed) control words. The datasets must be partitioned. If any dataset name is unquoted and contains only a single index level, it is suffixed with ".TEXT". If the LIBRARY option is entered without a dataset name list, the primary input dataset (which in this case must be partitioned) is passed to NSCRIPT as a single SYSLIB. (Note that the input dataset name must point to a specific member of this partitioned dataset.)

MESSAGE(* or message-dataset-name)

Message-dataset-name defines the dataset in which NSCRIPT will place error messages. If this operand is not specified, MESSAGE(*) is the default, and messages will be printed on the terminal. If the message dataset name is unquoted and has only a single index level, it is suffixed with ".MESSAGE".

NOWAIT
NO

Causes output to begin immediately without waiting for the first page to be adjusted(3).

NUMBER
NU

Causes the name of the current SCRIPT file (that given by the "input-dataset-name" argument or one now being used as a result of .im or .ap control lines) together with the current line number within that file to be placed in the left margin of the output(4). If the input dataset is line numbered, the line number is also printed in the left margin.

OFFLINE
OF

Causes the output to be edited and formatted for the off-line printer. The output is directed to an on-line dataset whose name is constructed from that of the primary input dataset by replacing the last index level with ".LIST". It is recommended that the CENTER option also be used when producing off-line output, as the result is easier to use in copying machines. It is also recommended that OFFLINE output be used when proof-reading the file, as on-line output is very slow.

---

(3) The NOWAIT option need not be used when producing off-line output, as NSCRIPT realizes that no adjustment of the first page is required.
(4) NUMBER also implies CENTER or ADJUST. Otherwise, enough room would not be available to hold the file name and item number.

PAGE(page-number)

Causes printout to begin with the page number specified. The entire file is scanned and formatted until the correct page is reached, but output does not begin until it is reached. The page number must be three digits or less. The PAGE option may be useful for restart purposes or for checking single pages in the file(5).

PRINT(* or output-dataset-name)

Output-dataset-name defines the dataset into which the formatted output produced by NSCRIPT is placed. If the output dataset is not quoted and contains only a single index level, it is suffixed with ".LIST".

Note that if PRINT(*) is specified, the TERMINAL option is assumed. If the OFFLINE option is specified, output from NSCRIPT is by default directed to a disk dataset whose name is constructed from that of the primary input dataset by replacing the last index level with ".LIST", and PRINT(output-dataset-name) need only be specified if a different dataset name is desired. If PRINT(output-dataset-name) is specified, OFFLINE is selected by default. If neither the TERMINAL nor OFFLINE options is specified, PRINT(*) is assumed.

SINGLE
SI

Only a single page of output is produced. This page will be the first page of the file, or if PAGE has been specified, that page.

STOP
ST

Causes a pause at the bottom of each page when producing on-line output. This pause allows the user time to insert the next page. The pause is terminated by him with a carriage return.

---

(5) See the description of the SINGLE option.

PAGE(page-number)

Causes printout to begin with the page number specified. The entire file is scanned and formatted until the correct page is reached, but output does not begin until it is reached. The page number must be three digits or less. The PAGE option may be useful for restart purposes or for checking single pages in the file(s).

PRINT(* or output-dataset-name)

Output-dataset-name defines the dataset into which the formatted output produced by NSCRIPT is placed. If the output dataset is not quoted and contains only a single index level, it is suffixed with ".LIST".

Note that if PRINT(*) is specified, the TERMINAL option is assumed. If the OFFLINE option is specified, output from NSCRIPT is by default directed to a disk dataset whose name is constructed from that of the primary input dataset by replacing the last index level with ".LIST", and PRINT(output-dataset-name) need only be specified if a different dataset name is desired. If PRINT(output-dataset-name) is specified, OFFLINE is selected by default. If neither the TERMINAL nor OFFLINE options is specified, PRINT(*) is assumed.

SINGLE
SI

Only a single page of output is produced. This page will be the first page of the file, or if PAGE has been specified, that page.

STOP
ST

Causes a pause at the bottom of each page when producing on-line output. This pause allows the user time to insert the next page. The pause is terminated by him with a carriage return.

---

(5) See the description of the SINGLE option.

Usage:

Appendix A contains an unformatted listing of a sample input file.

The NSCRIPT command causes the specified file to be edited and formatted under control of control lines contained within the file. Output can be directed to the user's terminal or to a permanent dataset from which it can be printed on the terminal or printed on the off-line printer by either submitting a background job or using the TSO OFFLINE command.(7) Certain differences in the processing occur depending on where the output is directed and are described below under individual headings.

Each line read from the disk file is inspected for a first character equal to ".", which identifies a control line. Control lines are not output, but are interpreted to specify the format of the output. Control lines must be entered as single lines in the file. The control word may be specified in either upper or lower case (or a mixture of the two) and must be separated from its arguments (if any) by one or more spaces.

Control lines may appear anywhere in the file, and have effect on all output produced after their appearance. Input data should not be placed in a control line, as it may be erroneously interpreted as an operand and it will certainly be lost.

The TRANSLATE option is needed if output is to be directed to an off-line printer which is not equipped with upper and lower case letters(8). In conjuction with the UNFORMATTED option, TRANSLATE provides a means of printing the original file on a printer which does not have the TN-chain mounted.

---

(7) The TSO OFFLINE command, which should not be confused with the OFFLINE option of the NSCRIPT command, is described in OS-25.

(8) The TN-chain has upper and lower case letters and 58 special characters.

TERMINAL
TERM

Causes output lines to be formatted for an on-line terminal. This option is mutually exclusive of the OFFLINE option. TERMINAL is selected by default if PRINT(*) is specified or assumed.

TRANSLATE
TR

Causes character translation using a translate table that contains as default lower to upper case mapping. Other types of translation may be specified with the .TR control word described later in the manual.

TWOPASS

The TWOPASS option causes NSCRIPT to make one complete pass through the input file(6), performing all indicated formatting but not producing any output. A second pass is then made, during which output is produced. This option is primarily useful for handling a file which uses reference names defined later in the input.

UNFORMATTED
UN

The specified file is printed along with all control lines. No formatting or editing takes place. UNFORMATTED output is similar to the output one would obtain by using CMS PRINTF or TSO LIST.

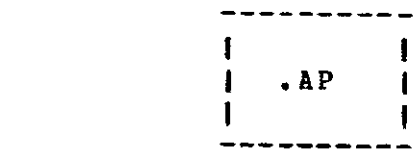
---

(6) A complete pass is made regardless of whether or not the single option is specified.

on the off-line printer. Canonicalization was performed automatically by the CMS Editor if a filetype of SCRIPT was used, but unfortunately, the TSO Editor does not have this capability. If a word or character is to be underscored using the TSO Editor the following technique can be used to establish the proper sequence and consequently will conform to the canon.

INPUT
xUxnxdxexrxsxcxoxrxe
change /x/_/ all
Underscore

Note: Just after typing the underscore a backspace was typed which caused the third slash to overprint (provided a backspace is not the character kill character).

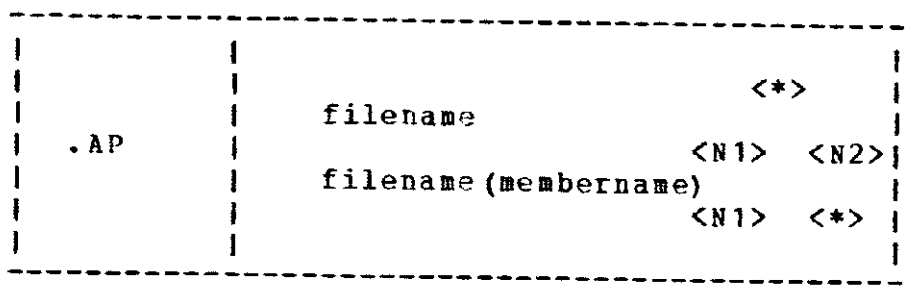


APPEND control

Purpose:

The APPEND control word allows an additional TEXT or SCRIPT file to be appended to the file just printed.

Format:



Usage:

When the .AP control word is encountered, the current TEXT or SCRIPT file will be closed. Items N1 through N2 of the "filename" will be read and printed as a continuation of output already produced.

Each filename that has not been used in a previous IMBED or APPEND Control word must have an associated "DD Statement" or TSO FILE that references a dataset containing variable length records. If the Filename is not ALLOCATEd(21) or the "DD Statement" is missing NSCRIPT will write an error message before closing all opened files.

Defaults:

The entire file is read in starting at the first line and no break is created.

---

(21) This is a TSO Command. Refer to the IBM TSO Command Language Reference manual, GC28-6732.

Notes:

(1) "*" may be coded in place of "N1 N2" to indicate that the entire file is to be read. If no arguments appear after the filename, "*" is assumed.

(2) If N1 is coded, "*" may be coded in place of N2 to indicate that the entire file starting with item N1 is to be read. If the third argument is omitted, "*" is assumed.

(3) The .AP control word only allows files to be appended to the end of the current file. If it is desired to insert file contents at some point besides the end, the .IM control word should be used.

(4) If .AP FILENAME is used to supply the file name, a four step procedure is followed in searching for the referenced file. In order, a search is conducted for

-1- A DD statement (or TSO file) "FILENAME" which points to a sequential dataset(22).

-2- A DD statement (or TSO file) "FILENAME" which points to a partitioned dataset, which itself contains a member "FILENAME".

-3- A member "FILENAME" in an active partitioned dataset. (The active partitioned datasets are searched for member "FILENAME" in reverse order of opening.)

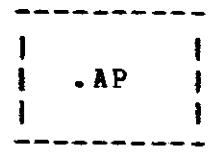
-4- A member "FILENAME" in the optional "SYSLIB" partitioned dataset.

If nothing is found a message is printed out and an empty file is APpended.

If .AP FILENAME(MEMBERNAME) is used to supply the file name, step -3- of the search is skipped, and in steps -2- and -4-, the partitioned datasets in question are searched for a member named "MEMBERNAME" not "FILENAME".

---

(22) A DD statement (or TSO file) which refers to a specific member of a partitioned dataset is considered to point to a sequential dataset.



Examples:

(a) .AP CHA4CONT
The file named CHA4CONT will be read and formatted for output as a continuation of the current TEXT file. If the output is being created in batch mode a "DD Statement" similar to the statement below would have to be included in the job setup for input of this file.

```
//CHA4CONT DD DSNAME=U.M1234.5678.CHA4CONT.TEXT,
//            DISP=SHR
```

If CHA4CONT was a member of a library, it could be referenced by the LIBRARY option. In batch, a SYSLIB DD statement must be included.

```
//SYSLIB   DD  DSN=U.M1234.5678.library-name.TEXT,
//         DISP=SHR
```

(b) .AP FIG2 12 *
The file named FIG2 TEXT will be read starting with item 12 and output as a continuation of the current TEXT file.

ARABIC control

Purpose:

The ARABIC control word causes page numbers to use arabic numerals.

Format:

Usage:

The .AR control word causes all page numbers produced hereafter in headings or footings to be printed in arabic numerals.

Defaults:

Page numbering in arabic is in effect until an ".ro" is encountered. A break is not created by this command.

Notes:

(1) Arabic numerals are the normal mode.

The PAGE(page-number) option may be used in conjuction with the SINGLE option to selectively format and print one portion of a manuscript. This option may also be used when restart is required of the output(9).

On-line Procedures:

Output can be produced on the user's terminal in two ways:

(1) By invoking the NSCRIPT command with the desired options, or

(2) For an input dataset whose last level qualifier is SCRIPT(10), by using the RUN subcommand of the TSO editor. No options can be specified.

Except for references to the use of options, the following describes the steps that are used to produce output on the user's terminal for either method.

(1) Unless the NOWAIT option was specified, the immediate response to issuing the NSCRIPT command is the following line(11):

LOAD PAPER; HIT RETURN

The user should remove the ordinary terminal paper and insert the paper on which he desires his output to appear. The paper should be positioned in such a way that the top edge of the paper is just below the typeball runner. Output will begin as soon as he types a carriage return(12).

---

(9) As might occur when system fails during on-line output of a very long file or when the user's own errors cause output to terminate.

(10) If the RUN subcommand is issued while editing a TEXT dataset, the message

TEXT DATASET CANNOT BE RUN

will be printed, and the user is returned to EDIT mode.

(11) If the TWOPASS option is supplied, this message is not produced until the beginning of the second (output) pass.

(12) Note that the paper will be ready for typing on the top line following the carriage return.

(2) If the STOP option was supplied, output will cease at the end of each page. The user should insert the next piece of paper, aligning it as described in (1), and type a carriage return[13].

(3) At the conclusion of the last page of output, the carriage will be spaced several lines into the following page before the READY message is typed[14]. The user may re-insert the terminal paper at this time.

Appendix B contains sample on-line output from the input file in Appendix A.

Off-line Procedures:

NSCRIPT output can be printed on the off-line printer in one of three ways:

(1) By using the TSO SUBMIT command to execute a background job.
(2) By submitting an over-the-counter batch job.
(3) Buy using the OFFLINE (or PRINT) option of NSCRIPT and the TSO OFFLINE command.

Appendix C contains the Job Control Language statements necessary to use methods (1) and (2).

---

(13) This procedure differs from the previous version of SCRIPT, which required the user to align to the line on which the heading was to be printed.
(14) If STOP is specified, the carriage will pause at the end of the last page.

To use the TSO OFFLINE command, NSCRIPT must be invoked with the OFFLINE option[15]. If the OFFLINE option is used, the dataset created will be named by replacing the last index level of the primary input dataset with ".LIST". (If the PRINT(output-dataset-name) option is also specified, output-dataset-name will have ".LIST" as its last level qualifier.) The dataset created can be specified in the TSO OFFLINE command, along with any desired operands, to produce output on the off-line printer. (Refer to OS-25 for a description of the TSO OFFLINE command.) N.B. Unless the NSCRIPT TRANSLATE option was specified with the default translation to upper case, the TRAIN(TN) option of the OFFLINE command must be used to obtain output printed with both upper and lower case letters. Failure to specify TRAIN(TN) or the TRANSLATE option of NSCRIPT will result in unreadable output.

When output is to be formatted for the off-line printer:

(1) The first page of output is positioned at a page crease (perforation) and then normal first page processing is initiated. If a heading is to be printed on the first page, ".he"[16] must be the first line in the primary input dataset.

(2) The second and succeeding pages of an OFFLINE file may be printed starting on the first line, which should generally be just below the perforation.

Error Messages:

All messages whether error or informative are written to the DDNAME SYSPRINT.

For error messages produced by incorrect control lines, the following message is printed:

ERROR OCCURRED AFTER READING xxxx LINES OF FILE xxxxxxxx.
Along with the line causing problems.

---

(15) On-line output directed to a file will not print properly on the off-line printer. Conversely, off-line file output will not print properly on the on-line terminal.

(16) See the description of the heading control words later in the manual.

INCORRECT PARAMETER LIST
An invalid parameter has been specified for a control word, or a required parameter is missing.

OPEN FOR DDNAME xxxxxxxx WAS UNSUCESSFUL.
Cannot find a reference to the file in question.

DISK ERROR WHILE READING
A permanent I/O error has been encountered while reading the current file.

ILLEGAL CONTROL CARD ENCOUNTERED
An unrecognizable control line has been encountered in the input file(17).

TOO MANY FOOTNOTES OR REMOTE LINES NOW QUEUED.
More than 100 footnote or remote lines are now queued. Probable cause of this diagnostic is a failure to close a footnote with a ".fn end" or a remote with a ".rm".

NO MORE CORE AVAILABLE - REDUCE IMBED NESTING
This message is given when .im (imbed) or .rc (read control) nesting (recursion) is too deep or too many imbeded files are active at once. Reduce the nesting of .im's and retry the command.

TOO MANY REFERENCE NAMES.
This message indicates that more than 200 reference names have been requested.

INVALID CONTROL WORD PARAMETER FIELD
The argument portion of a control line is invalid. This could possibly be caused by a character being used where only a numeric argument is acceptable.

OUTPUT LINE TOO LONG OR PRINTER ERROR
Caused by sending a line longer that 133 characters to the printer or because DCB parameters were incorrectly specified.

---

(17) This error is commonly caused by presenting an input line, not intended to be a control line, which begins with ".".

**** A TERMINAL ERROR HAS OCCURRED ...
This message indicates an error in the NSCRIPT command module. The appearance of this message should be promptly reported to System Maintenance personnel and a copy of the SCRIPT file should be saved for their use in diagnosing the difficulty. Usually, the user can bypass the problem by changing his SCRIPT file.

Creation of Input for NSCRIPT:

All input must be in the form of variable length records which makes their creation through a typerwriter type of terminal almost mandatory. Input lines can have a maximum length of 255 characters(18). NSCRIPT will format these input lines into lines of length specified by ".11" or 60 characters as default(19).

The TSO editor, which must be used to create the input dataset, permits two file types: TEXT and SCRIPT. The characteristics of the TEXT dataset type are described in the EDIT section of the TSO Command Language Reference (GC28-6732). SCRIPT is similar to TEXT, except that its line length is 240 (not 255) and the tab default is TAB OFF(20). The TAB OFF default in the SCRIPT file type permits tabs to remain in the text and be processed by NSCRIPT under control of the .TB control word.

Formatting of a line with characters from a following line can be prevented by creating what is called a "break". A break is caused explicitly by use of leading blanks (one is sufficient), a tab, or the break command word (.br). Implicit breaks are caused by any of the commands that cause normal line spacing to be interrupted. Breaking capabilities for each command word can be found in its complete description in the remainder of this manual.

Lines containing backspaces must be canonicalized so that the correct number of character position will be typed per line and underscores will over-print properly

---

(18) It is possible to input more than 130 characters per line if backspaces and underscores are used.
(19) If the sequence underscore, backspace, character is found, the underscore and backspace will not be counted.
(20) If TAB ON is specified using the TAB subcommand of the editor for a SCRIPT file type, the resulting dataset functions the same as a TEXT dataset.

BOTTOM MARGIN control

Purpose:

The BOTTOM MARGIN control word specifies the number of lines which are to appear between the bottom of the output page and the last line of ordinary or footnote text.

Format:

.BM	n

Usage:

At the bottom of all subsequent output pages (including the current page), n lines will appear between the bottom of printed text(23) and the physical bottom of the page.

Any footing line will appear within these n lines.

Defaults:

This command does create a break. Unless otherwise specified n = 3 will be in effect. When this command word is encountered and the operand is omitted n = 1 will be assumed.

Notes:

(1) See also the description of the .FM (Footing Margin) control word.

(2) At no time may the value set in .BM be smaller or equal to the value set in .FM.

---

(23) Footnote text included.

BREAK control

Purpose:

BREAK causes the immediately previous line to be typed without filling in with words from the next line.

Format:

Usage:

BREAK is used to prevent concatenation of lines such as paragraph headings or the last line of a paragraph. It causes the preceding line to be typed as a short line if it is shorter than the current line length.

Defaults:

This command is in effect and cannot be "turned off" except if it is the line immediately following a ".if" that is false. Then it is skipped.

Notes:

(1) Many of the other control words act as a BREAK. No BREAK is necessary when one of these is present. In the description of each control word it states, in the section titled Defaults, whether that control word acts as a break or not.

(2) A leading blank or tab on a line has the effect of a BREAK immediately before the line.

(3) If NO CONCATENATE is in effect, all lines appear to be followed by a BREAK.

Example:

(1) Heading:
.br
First line of paragraph ...

This part of the file will be printed as:

.BR

Heading:
First line of paragraph ...

If the BREAK control word were not included, it would be typed:

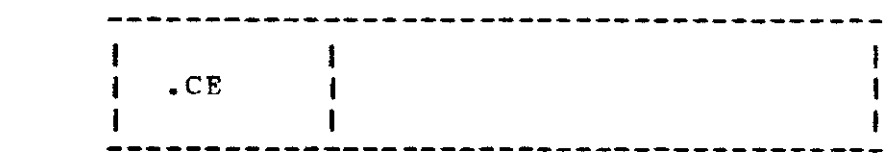
Heading: First line of paragraph ...

CENTER control

Purpose:

The line following the CENTER control word will be centered between the margins.

Format:



Usage:

The next line in the input file, including its leading or trailing blanks, will be centered between the left and right margins.

Defaults:

This command does cause a break. Any input lines or parts of lines, that have not been typed will be typed without filling in words from any following lines.

Notes:

(1) If the line to be centered is longer than the current line length, it will be truncated and not centered.

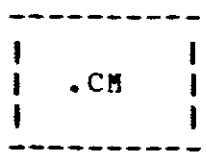
(2) The left and right margins are the value of any indent value (.IN) and the current line length, respectively.

Example:

(1) .CE
Other Methods

When this line is typed, the characters "Other Methods" will be centered:

Other Methods



COMMENT control

Purpose:

The COMMENT control word is ignored and may be used to enter comments into a script file.

Format:

.CM	<anything>

Usage:

The .CM control word allows comments to be stored in the SCRIPT file. These comments may be seen whenever the file is printed (using LIST, OFFLINE, PRINT or the UNFORMATTED option).

Comment lines may be used to store unique identifications for use during editing of the file.

Defaults:

This command does not cause a break.

Notes:

(1) Since only the first two characters of a control line (exclusive of the ".") are examined to determine what control word is present, and since undefined reference names have null-string ("") values, a reference name can be given the value 'CM' and used to control conditional recognition of other control words. (See Example 2.)

Examples:

(1) .CM .NF used below.

The comment line will be seen when examining the unformatted printout of the script file and will serve to explain what is going on to the reader.

(2) .sr stop0 'CM'
.ur .ur .&&stop&n..im setup

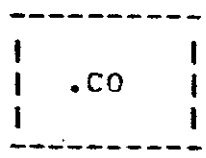
If the reference name "n" has any value other than 0, the line will be:

.im setup

If the reference name "n" has the value 0, the line will be:

.cmim setup

which is, of course, interpreted as a comment and ignored.



CONCATENATE control

Purpose:

CONCATENATE cancels a previous NO CONCATENATE control word and causes output lines to be formed by concatenating input lines and truncating at the nearest word boundary to fit within the specified line length.

Format:

Usage:

The CONCATENATE control word specifies that output lines are to be formed by shifting words to or from the next input line. The resulting line will be as close to the line length as is possible without exceeding it or splitting a word.

Defaults:

This command creates a break and is in effect until an ".nc" is encountered.

Notes:

(1) CONCATENATE is the normal mode.

(2) Output produced with CONCATENATE in effect is similar to what a normal typist produces.

CONDITIONAL PAGE control

Purpose:

The CONDITIONAL PAGE control word causes a page eject to occur if less than a specified number of lines remain on the page.

Format:

.CP	n

Usage:

The .CP control word will cause a page eject to occur if "n" lines do not remain on the current page.

This request is especially useful:

(1) Before a .SP control word which is used to leave room for a drawn figure. Use of .CP will guarantee that all of the spaces are contiguous(24).

(2) Preceding a section heading to insure that the heading will not be left alone at the bottom of the page.

Defaults:

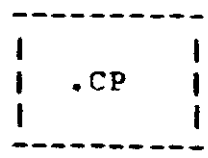
This command will not cause a break. If n is omitted the command will be ignored.

Notes:

(1) If no operand is specified, the .CP control word is ignored.

---

(24) Note that .LS YES must be specified if the spaces entered in this way are to be left on the next page.



(2) By "remainder of current page" it is meant "the area between the last typed line and the beginning of saved footnote text (if any) or the current bottom margin (if no footnotes are saved)."

DON'T COUNT control

Purpose:

The DON'T COUNT control word causes all output produced directly by the immediately following text line to not be counted for purposes of page length control.

Format:

Usage:

If output is off-line, all output produced by the next text line will be overprinted on the last printed line.

If output is on-line, output produced by the next text line will be typed normally, but will not be counted towards the page length.

The .DC control word is primarily useful to allow the user to store equations and such in his SCRIPT file when two or more typeballs are necessary during output.

Defaults:

This command will cause a break before processing the immediately following text line and it will be in effect only for that line. Subsequent lines will be processed normally.

Notes:

(1) Only text lines which immediately follow a .DC are not counted.

DOUBLE SPACE control

Purpose:

The DOUBLE SPACE control word causes a line to be skipped between each line of typed output.

Format:

Usage:

Subsequent output lines will be double spaced.

Defaults:

This command is not in effect unless encountered. It does cause a break to occur when specified.

Notes:

(1) Single spacing is the normal mode.

(2) .SP control words encountered while .DS is in effect will produce twice the normal number of blank lines.

(3) Footnote lines are never double spaced.

(4) The operand to the .CP control word is not doubled when .DS is in effect.

EMPTY PAGE control

Purpose:

The EMPTY PAGE control word is used to control suppression of empty (except for headings and footings) pages.

Format:

.EM	YES
	NO

Usage:

Empty pages(25) can be generated in a number of ways and are not normally printed by NSCRIPT.

.EM YES specifies that empty pages are to be printed. .EM NO specifies that empty pages are not to be printed.

Defaults:

A break will not be created and empty pages will not be printed unless ".em yes" or ".em" is encountered.

Notes:

(1) If the operand is omitted, "YES" is assumed, since the user presumably intends to accomplish something with the control word.
(2) .EM NO is the normal mode.
(3) The first page of an on-line file is considered empty, whereas the first page of an OFFLINE file is not.
(4) If pages are suppressed as a result of .EM NO, page numbers will still increment unless .PN OFFNO has been specified.

---

(25) Pages which would, if printed, contain only headings, footings, and possibly footnotes.

.FD

ODD FOOTING control

Purpose:

The ODD FOOTING control word specifies three items of title information to be printed at the bottom of odd-numbered pages.

Format:

	S1
.FD	
	'S1'S2'S3'

Where S1, S2 and S3 are character strings not containing quotation marks. Any of the fields may be omitted, but the quotation marks must be included to indicate missing fields, e.g., 'S1''S3'.

Usage:

The .FD control word is used in a way similar to the .HE control word. The title items defined with .FD will be printed in a footing line near the bottom of odd-numbered pages.

The footing line is generated by:

(1) Substituting the current page number for each appearance of the escape character "%".
(2) Left-adjusting S1.
(3) Centering S2.
(4) Right-adjusting S3.

Defaults:

A break will not be created when this command is encountered. Unless otherwise specified, ".fd ''''" will be in effect.

Notes:

(1) No individual item in the footing data may be longer than 60 characters.
(2) S3 may overlay S2 if necessary, and S2 may overlay S1 if necessary.

(3) The default value of the footing line is "'''''".
(4) The quotation marks are used to delimit the character strings, not to surround them.

.FE

FOOTING control

Purpose:

The FOOTING control word specifies three items of title information to be printed at the bottom of even- and odd-numbered pages.

Format:

.FE	S1	
	'S1'S2'S3'	

Where S1, S2 and S3 are character strings not containing quotation marks. Any of the fields may be omitted, but the quotation marks must be included to indicate missing fields, e.g., 'S1''S3'.

Usage:

The .FE control word is used in a way similar to the .HE control word. The title items defined with .FE will be printed in a footing line near the bottom of even- and odd-numbered pages.

The footing line is generated by:

(1) Substituting the current page number for each appearance of the escape character "%".
(2) Left-adjusting S1.
(3) Centering S2.
(4) Right-adjusting S3.

Defaults:

A break will not be created when this command word is encountered. Unless otherwise specified ".fe ''''" will be in effect.

Notes:

(1) No individual item in the footing data may be longer than 60 characters.
(2) S3 may overlay S2 if necessary, and S2 may overlay S1 if necessary.

(3) The default value of the footing line is "'''''".
(4) The quotation marks are used to delimit the character strings, NOT to surround them.

FOOTING MARGIN control

Purpose:

The FOOTING MARGIN control word specifies the number of blank lines which are to be left between the bottom of formatted text and any footing line.

Format:

.FM	n

Usage:

The .FM control word defines the footing margin, which is the number of blank lines which will be left between the bottom of formatted text and any footing line on all pages.

If the footing margin is n and the bottom margin is m, the bottom of each page will appear as follows:

(1) n blank lines,
(2) The footing line (if any),
(3) m - n - 1 blank lines.

Defaults:

This command word creates a break when encountered and until then n = 1 will be in effect. If the operand is omitted n = 1 will be assumed.

FOOTNOTE control

Purpose:

The FOOTNOTE control word allows the user to enter a footnote which will be printed at the end of the current page.

Format:

.FN	BEGIN
	END

Usage:

When the .fn begin control word is encountered, the current values of all relevent control variables are saved and NSCRIPT prepares to accept footnote text.

When the .fn end control word is encountered, the values saved by the .fn begin are restored and formatting of output continues.

NSCRIPT saves enough room at the bottom of the page to print the footnotes and three separator lines. If enough room does not exist, footnotes will be continued at the bottom of the next page. Any control words (except .fn) may appear within the footnote. All page eject controls are ignored, space (.sp) requests in footnotes are in forced single space mode and all other requests apply only to the footnote(26).

Defaults:

This command does not create a break.

---

(26) Format controls in effect when the .fn begin is encountered remain in effect during formatting of the footnote unless explicitly overriden. (E.g., if indentation of the text has been requested with an .IN control word, the footnote will also be indented.)

.FN

Notes:

(1) Footnotes are always single spaced on output.
(2) No more than 100 formatted footnote lines may be waiting for output at any time.
(3) The .fn begin control does not act as a break. The next regular input line(27) will be concatenated with the previous line.
(4) No footnotes will appear on a page if at least 4 lines are not available at the end of the page. If footnotes are generated within 5 lines of the bottom, they will be saved for the next page.
(5) The line which precedes a footnote definition will always be printed on the current page. It can never be displaced to the next page by accumulation of footnotes.
(6) Footnotes appear before the bottom margin set in the .bm control.
(7) Footnote numbering, as used in this manual, is described in the section "NOTES ON THIS DOCUMENT", page vii.

Example:

(a) As Anderson -1-
.fn begin
.in 5
.un 5
-1- Anderson, D.A.,
Grundlagen der Madchenjadgt, MIT Press, August 1932.
.fn end
has noted, this phenomenon is indigeneous to ...

Can be used to obtain output similar to that shown here:

(a) As Anderson -1- has noted, this phenomenon is indigineous to ...

---

(27) That is, the first line after the .fn end.
-1- Anderson, D.A., Grundlagen der Madchenjadgt, MIT Press, August 1932.

FORMAT control

Purpose:

The FORMAT control word combines the effect of CONCATENATE and JUSTIFY.

Format

.FO .FI	

Usage:

The FORMAT control word is a short-hand way to specify CONCATENATE and JUSTIFY. This control word specifies that output lines are to be formed by shifting words to or from the next line (concatenate) and padded with extra blanks to produce an even right margin (justify).

Defaults:

This command does create a break. It is in effect unless a ".nf" is encountered.

Notes:

(1) Since FORMAT is the normal mode, the control word is used only to cancel a previous NO FORMAT control word.

(2) The .FI control word is provided for compatability with FILL in TSO's FORMAT and MULTIC's "runoff". FILL performs and functions the same as .FO.

EVEN FOOTING control

Purpose:

The EVEN FOOTING control word specifies three items of title information to be printed at the bottom of even-numbered pages.

Format:

	S1
.FV	
	'S1'S2'S3'

Where S1, S2 and S3 are character strings not containing quotation marks. Any of the fields may be omitted, but the quotation marks must be included to indicate missing fields, e.g., 'S1''S3'.

Usage:

The .FV control word is used in a way similar to the .HE control word. The title items defined with .FV will be printed in a footing line near the bottom of even-numbered pages.

The footing line is generated by:

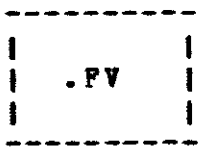
(1) Substituting the current page number for each appearance of the escape character "%".
(2) Left-adjusting S1.
(3) Centering S2.
(4) Right-adjusting S3.

Defaults:

A break will not be created when this command is encountered and until then ".fv '''" will be in effect.

Notes:

(1) No individual item in the footing data may be longer than 60 characters.
(2) S3 may overlay S2 if necessary, and S2 may overlay



S1 if necessary.

(3) The default value of the footer line is "'''''".

(4) The quotation marks are used to delimit the character strings, NOT to surround them.

```
----------
|        |
|  .HD   |
|        |
----------
```

ODD HEADING control

Purpose:

The ODD HEADING control word is used to define three headings to be printed at the top of odd-numbered pages.

Format:

```
-----------------------------------------
|         |    S1                       |
|  .HD    |                             |
|         |    'S1'S2'S3'               |
-----------------------------------------
```

Where S1, S2 and S3 are character strings not containing quotation marks. Any of the fields may be omitted, but the quotation marks must be included to indicate missing fields, e.g., 'S1''S3'.

Usage:

The .HD control word is used in the same way as the .HE control word. The headings defined with .HD will appear only on odd-numbered(28) pages, however.

The heading line is generated by:

(1) Substituting the current page number for each appearance of the escape character "%".
(2) Left-adjusting S1.
(3) Centering S2.
(4) Right-adjusting S3.

Defaults:

This command does not create a break. Unless otherwise specified ".hd '''PAGE %'" will be in effect.

---

(28) Odd-numbered pages are those which have the binding to the left of the left-hand margin in a book.

Notes:

(1) Appearance of .HD does not affect the headings defined for even-numbered pages by .HE or .HV.
(2) .HD may be used in conjuction with .HV (even heading) to produce output suitable for double-side printing.
(3) The quotation marks are used to delimit the character strings, NOT to surround them.
(4) No individual item in the heading data may be longer than 60 characters.
(5) S3 may overlay S2 if necessary, and S2 may overlay S1 if necessary.

Example:

```
.hd 'Department of Alchemy''ATN-05-3-70-%'
.hv 'ATN-05-3-70-%''Department of Alchemy'
```

Could be used to obtain output in which the number of the technical paper is always in the upper outside corner and the department name is always in the upper inside corner.

.HE

HEADING control

Purpose:

The HEADING control word is used to define three headings to be printed at the top of both even- and odd-numbered pages.

Format:

.HE	S1
	'S1'S2'S3'

Where S1, S2 and S3 are character strings not containing quotation marks. Any of the fields may be omitted, but the quotation marks must be included to indicate missing fields, e.g., 'S1''S3'.

Usage:

The .HE control word specifies 3 items of character data to be printed near the top of all subsequent output pages(29).

The heading line is generated by:

(1) Substituting the current page number for each appearance of the escape character "%".
(2) Left-adjusting S1.
(3) Centering S2.
(4) Right-adjusting S3.

If the top margin is m and the heading margin is n, the top of the output page will consist of:

(1) m - n - 1 blank lines,
(2) the heading line, and

---

(29) Including the current page if nothing has yet been written, except the first page of OFFLINE output if the top and heading margins are such that the heading is required to appear within the first 3 lines.

(3) n blank lines.

Defaults:

A break is not created by this command. Unless otherwise specified ".he '''PAGE %'" will be in effect.

Notes:

(1) No individual item in the heading data may be longer than 60 characters.
(2) S3 may overlay S2 if necessary, and S2 may overlay S1 if necessary.
(3) The default value of the header line is "'''PAGE %'".
(4) The quotation marks are used to delimit the character strings, NOT to surround them.
(5) Appearance of the .HE control word overrides any previous values of the heading items. If no items are specified, no heading line at all will be generated(30). Furthermore, items which are omitted become null and do not retain their previous values. Thus, to obtain the same effect that one had with SCRIPT by specifying ".he xxx", it is necessary to specify ".he 'xxx''PAGE %'".

Example:

```
.he 'Department of Alchemy''ATN-05-3-70-%'
```

---

(30) Setting a null header is the only way to achieve the effect one would have obtained with ".pn off" under the previous version of SCRIPT.

.HM

HEADING MARGIN control

Purpose:

The HEADING MARGIN control word specifies the number of blank lines which are to be left between the heading line and the first line of text.

Format:

.HM	n

Usage:

The heading line (if any) produced on subsequent output pages will be separated from the first line of text by n blank lines.

If the current top margin is m and the heading margin is n, the top of each page will appear:

m - n - 1 blank lines
The heading line
n blank lines

Defaults:

This command creates a break and until it is encountered n = 1 will be in effect. If the operand is omitted n = 1 will be assumed.

Notes:

(1) The heading margin must always be strictly less than the top margin.

EVEN HEADING control

Purpose:

The EVEN HEADING control word is used to define three headings to be printed at the top of even-numbered pages.

Format:

.HV	S1	
	'S1'S2'S3'	

Where S1, S2 and S3 are character strings not containing quotation marks. Any of the fields may be omitted, but the quotation marks must be included to indicate missing fields, e.g., 'S1''S3'.

Usage:

The .HV control word is used in a way similar to .HE. The headings defined by .HV are, however, printed only on even-numbered pages(31).

The heading line is generated by:

(1) Substituting the current page number for each appearance of the escape character "%".
(2) Left-adjusting S1.
(3) Centering S2.
(4) Right-adjusting S3.

Defaults:

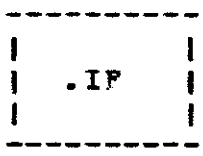
This command does not create a break when encountered. Unless otherwise specified ".hv '''PAGE %'" is will be in effect.

---

(31) Even-numbered pages are bound on the right margin.

Notes:

(1) Appearance of .HV does not affect headings defined for odd-numbered pages via .HE or .HD.
(2) See the description of .HE and .HD for further notes and examples.
(3) The quotation marks are used to delimit the character strings, NOT to surround them.



CONDITIONAL control

Purpose:

The IF control word causes the next input line to be processed or not depending on the result of a comparison.

Format:

.IF	'S1' <op> 'S2'	
	n1 <op> n2	

Where:

S1 and S2 are character strings not containing blanks,
n1 and n2 are signed decimal integers, and
<op> is one of the following:

=	>
¬=	<=
<	>=

Usage:

If the comparison operation is true, the immediately next input line is processed normally. Otherwise, the next input line is skipped (ignored).

If the first operand begins with the character "'", the comparison will be between two characters strings using the standard EBCDIC collating sequence.

If the first operand does not begin with "'", the comparison will be between two signed decimal integers.

Defaults:

This command does not create a break when encountered.

Notes:

(1) If the two character strings to be compared have unequal lengths, the comparison will result in the shorter string being considered less than the longer.

Example:

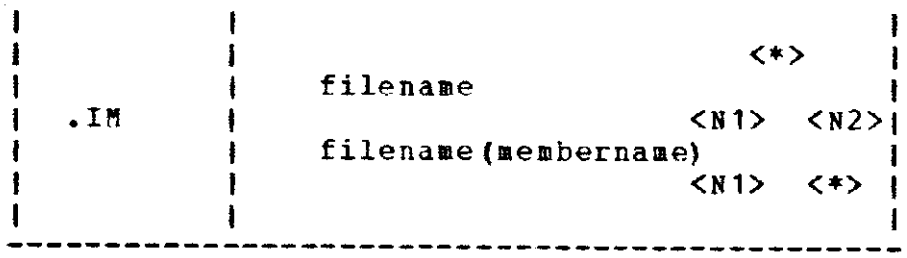
(1)
```
.sr a % / 2 * 2 - %
.ur .if &a = 0
```
Will cause the next input line to be processed if the current page number is even.

IMBED control

Purpose:

The IMBED control word is used to insert the contents of a specified file into the printout of another file.

Format:



Usage:

The first argument names a text file whose contents are to be included in the input. Each Filename or DDNAME that has not been used in a previous IMBED or APPEND Control word must have an associated "DD Statement" or TSO FILE that references a dataset containing variable length records. If the Filename is not ALLOCATEd(32) or the "DD Statement" is missing NSCRIPT will write an error message befor closing all opened files.

N1 gives the item number of the first line of the imbedded file to be read, N2 gives the item number of the last line to be read. Instead of "N1 N2", the single character "*" may be used to indicate that the entire file is to be included. The "*" may also be used instead of N2 to indicate that the rest of the file (that is, items N1 through the end) are to be included.

The .IM and .AP control words perform similar functions, but .IM allows the contents of a second file to be inserted into the printout of an existing file rather than appended to it. Imbedding may be used to insert standard sets of control words at desired spots, or to control formatting of a long document out of

---

(32) This is a TSO Command. Refer to the IBM TSO Command Language Reference manual, GC28-6732.

```
----------
|        |
|  .IM   |
|        |
----------
```

If .IM FILENAME(MEMBERNAME) is used to supply the file name, step -3- of the search is skipped, and in steps -2- and -4-, the partitioned datasets in question are searched for a member named "MEMBERNAME" not "FILENAME".

Examples:

(a) .IM CHAPTER4
The contents of the file named CHAPTER4 will be included in the input. When the end of CHAPTER4 is reached, printout of the current file will continue.

(b) .IM CONTROL 2 *
All but the first line of CONTROL will be imbedded. Note that CONTROL is the Filename used by ".rc" and ".rd". This can be used to give added flexibility to the NSCRIPT file. For example, suppose there are several ".rc's" within the file and it is executed both on-line and off-line. Because ".rc" always starts reading at the first line the following technique can be used.

```
.sr test 1
.pr If output is off-line, enter ".sr test 0",
.pr otherwise nothing.
.rc
.pr Enter Date of this manual.
.ur .if &test = 0
.im control 2 2
.ur .if &test = 1
.rc
```

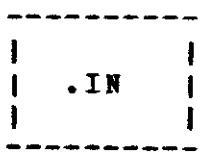
CONTROL would reference a dataset if the output was directed off-line, and the first line of that dataset would be ".sr test 0". If the output is directed to the offline printer a "DD Statement" similar to the one below would have to be included in the job setup for input of this file.


```
//CONTROL   DD   DSNAME=U.M1234.5678.CONTROL.TEXT,
//                DISP=SHR
```

If CONTROL was a member of a library it could be referenced by the LIBRARY option. In batch a SYSLIB DD statement must be included.

```
//SYSLIB DD DSN=U.M1234.5678.library-name.TEXT,
//             DISP=SHR
```

(c) .IM FOOTNOTE 12 15
Lines 12, 13, 14, and 15 of FOOTNOTE will be imbedded.



INDENT control

Purpose:

The INDENT control word causes the left margin of the printout to be indented a specified number of spaces.

Format:

.IN	n

Usage:

The .IN control word causes printout to be indented "n" spaces from the left margin. This indentation remains in effect for all subsequent lines(34) until another .IN is encountered.

".IN 0" or ".IN" will cancel the indentation and cause printout to continue at the original left margin.

Defaults:

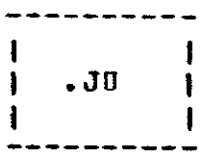
When this command word is encountered it creates a break and until then n = 0 is in effect. When the operand is omitted n = 0 is assumed.

Notes:

(1) the .IN request causes any offset (.OF) setting or undent (.UN) setting to be cleared.

---

(34) Including new paragraphs, footnotes, and new pages. It remains in effect even if .NF is specified.



JUSTIFY control

Purpose:

The JUSTIFY control word causes output lines to be padded with extra blanks so that the right margin is justified.

Format:

Usage:

The .JU control word specifies that all subsequent output lines are to be formed by padding with extra blanks to cause the right margin to be justified.

Defaults:

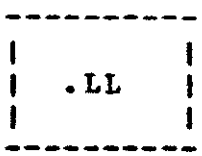
This command creates a break and is in effect until an ".nj" is encountered.

Notes:

(1) Since JUSTIFY is the normal mode, this control word is used to cancel a previous NO JUSTIFY control word or the NO JUSTIFY part of a NO FORMAT control word.

(2) If a line exceeds the current line length, and NO CONCATENATE is in effect, the line is printed as is.

(3) This control word is seldom used without CONCATENATE, which is the same as specifying FORMAT which combines the two functions.



LINE LENGTH control

Purpose:

The LINE LENGTH control word specifies the number of horizontal character positions which are to be printed in subsequent output lines.

Format:

.LL	n

Usage:

The .LL control sets the length of subsequent output lines to "n" characters.

Defaults:

When this command word is encountered it creates a break. Unless otherwise specified n = 60 characters per line (including blanks) will be in effect. If no argument is supplied then n = 1 will be assumed.

Notes:

(1) Practically all terminals and printers print 10 characters per horizontal inch. The default value of 60 is sufficient to give 1.5" left and 1" right margins on 8.5" paper.

.LS

LEADING SPACES control

Purpose:

The LEADING SPACES control word is used to control suppression of blank lines at the beginning of a page.

Format:

.LS		YES
		NO

Usage:

Blank lines may occassionally be the first lines of formatted output to be printed on a page. For example, a ".sp 10" may have appeared 5 lines from the bottom of the previous page. NSCRIPT does not normally print such blank lines(35).

Since blank lines may sometimes be desirable at the beginning of a page(36) the .LS control is provided.

.LS YES allows blank lines to begin a page. .LS NO prevents blank lines from beginning a page.

Defaults:

This command does not create a break and ".LS NO" is in effect until ".LS YES" or ".LS" is specified. (If the operand is omitted, "YES" is assumed.)

Notes:

(1) If the operand is omitted, "YES" is assumed.
(2) .LS NO is the normal mode.
(3) Leading spaces are allowed on the first page of printed output.
(4) This discussion applies only to formatted text, not to headings or top margins.

---

(35) Except at the top of the first page of output.
(36) For example, when centering a figure on a page.

NO CONCATENTATE control

Purpose:

The NO CONCATENATE control word stops words from shifting to or from the next line.

Format:

Usage:

The NO CONCATENATE control word stops words from shifting to and from the next line. There will be a one-to-one correspondence between the words in input and output lines.

Defaults:

This command word does create a break and is not in effect until encountered.

Notes:

(1) Concatenation will be completed for the lines which precede the .NC control word, since it also acts as a BREAK.

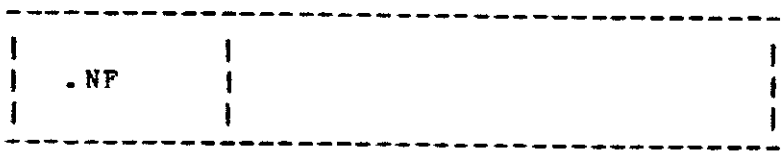
(2) If JUSTIFY is in effect, the right margin will still be adjusted.

NO FORMAT control

Purpose:

The NO FORMAT control word causes lines to be typed as they appear in the input by negating the effect of JUSTIFY and CONCATENATE.

Format:



Usage:

The .NF control word is a short-hand way to specify .NJ and .NC. Subsequent output lines will by typed exactly as they appear in the input until a .FO, .JU, or .CO control word in encountered.

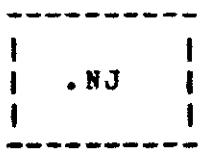
Tables and equations are more easily entered if .NF is used to suppress formatting just before them.

Defaults:

This command does create a break and is not in effect until encountered.

Notes:

(1) The diagnostic "LINE TOO LONG OR PRINTER ERROR" is commonly caused by failing to specify .FO after a section produced under .NF.



NO JUSTIFY control

Purpose:

The NO JUSTIFY control word stops justification of the right margin.

Format:

Usage:

The .NJ control word causes NSCRIPT to cease inserting extra spaces into lines in order to justify the right margin. If CONCATENATE is in effect, the output lines will be as long as possible without exceeding the current line length and without breaking words in the middle. The resulting output will be similar to what an ordinary typist would produce.

Defaults:

This command does create a break and is not in effect until encountered.

.OF

OFFSET control

Purpose:

The OFFSET control word causes all but the first line of a section to be indented.

Format:

.OF	n

Usage:

The .OF control word is used to indent the left side of the printout "n" spaces after printing the next output line. The indentation remains in effect until a BREAK or until another .OF control word is encountered.

The .OF control word may be used within a section which is also indented with the .IN control. Any subsequent .IN control word causes the indentation set by .OF to be cleared.

If it is desired to start a new section with the same offset as the previous section, it is necessary to repeat the .OF request.

Defaults:

This command does create a break when encountered. If no argument is supplied then n = 0 will be assumed.

Notes:

(1) If the following sequence is input:

```
.of m
<text>
.of n
<text>
```

The second offset will be added to the first if m and n are different. If m and n are the same, this addition will not occur. Phrased differently: if m ≠ n, all but the first line of the first section

together with the first line of the second section will be indented m spaces and all but the first line of the second section will be indented m + n spaces. If m = n, both sections will be indented m spaces.

Examples:

(1) Many of the sections of this manual(37) are produced by a sequence like this:

```
<section header>
.in 5
.of 4
(1) <text>
.of 4
(2) <text>
   .
   .
   .
.in
```

---

(37) Including the example text.

.OF

.PA

PAGE EJECT control

Purpose:

The PAGE control word causes an output page eject.

Format:

.PA	n

Usage:

When the .PA control word is encountered, the rest of the current page is skipped, any saved footnote lines are printed, the footer line is printed, and a new page is begun whose number is "n".

Defaults:

This command word does create a break when encountered. If the operand is omitted then the current page number plus one (% + 1) is assumed.

Notes:

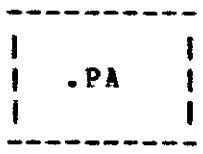
(1) If the STOP option was specified in the control line which invoked NSCRIPT, output will cease at the end of the current page to allow the typist to insert the next page of paper.

(2) If the current page is empty and .EM NO is in effect, either by default or through explicit statement, the .PA control has no effect other than causing the page number to be incremented or reset. Consequently, heading lines are not printed until there is a text line about to be printed which allows dynamic changes to page headings and possibly remote lines imbedded in the text of that page.

(3) Heading lines, margins, etc., must be specified before the first line of text which will appear on

the new page.

(4) If no argument is supplied, the page number will be incremented by 1.



the new page.

(4) If no argument is supplied, the page number will be incremented by 1.

ODD PAGE FORCE control

Purpose:

The ODD PAGE control word causes output to continue on an odd-numbered page.

Format:

Usage:

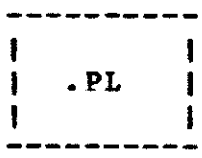
The .PD control word causes one or two page ejects so that output continues on an odd-numbered page.

Defaults:

This command will create a break when encountered.

Notes:

(1) If empty-page suppression is on (.EM NO), the page number will be incremented to the next odd page and output will continue on the next physical page.



PAGE LENGTH control

Purpose:

The PAGE LENGTH control word specifies the physical size of the output page in units of typewriter lines.

Format:

.PL	n

Usage:

The .PL control word allows the use of various paper sizes for output.

Normal 8.5" x 11" paper is 66 lines long on a IBM 1403 printer, a IBM 2741 type of terminal, or any other device which types 6 lines per vertical inch.

Defaults:

This command word will create a break and unless otherwise specified n = 66 will be in effect. If the command is encountered and the operand is omitted n = 1 will be assumed.

Notes:

(1) Use of the .PL control word for any other purpose than specifying the actual physical size of the output page is discouraged. The TOP MARGIN and BOTTOM MARGIN control words should be used to control the dimensions of printed text.

.PN

PAGE NUMBERING control

Purpose:

The PAGE NUMBER control allows the user to control the incrementing of page numbers.

Format:

.PN	OFFNO
	ON

OFFNO — Suppresses incrementing of page numbers.

ON — Causes incrementing of page numbers to resume.

Usage:

The .PN control word is used to control the automatic incrementing of page numbers. If OFFNO is specified, page numbers will not increase as subsequent pages are output. If ON is specified, incrementing will resume.

Defaults:

This command word will not create a break. Unless otherwise specified "ON" will be in effect.

Notes:

(1) If page incrementing is suppressed, the even and odd page force control words (.PV and .PD) function exactly like .PA.

(2) The only way to eliminate page numbers from appearing on the output is to remove the "%" escapes from heading and footing lines.

PRINT control

Purpose:

The PRINT control word causes one line of information to be typed on the user's terminal.

Format:

Usage:

If output is to the user's terminal, the .PR control is ignored(38). Otherwise, the entire operand field is printed on the terminal.

The .PR control may be of use, for example, immediately preceding a .RC (Read Control) control word as a reminder of what to do.

Defaults:

This command word will not create a break when encountered.

Example:

```
.pr Type Name and Address (3 lines)
.rc
.br
.rc
.br
.rc
.br
```

---

(38) Except during the first pass of a TWOPASS run, when .PR causes output normally.

EVEN PAGE FORCE control

Purpose:

The EVEN PAGE control word causes output to continue on an even-numbered page.

Format:

Usage:

The .PV control word causes one or two page ejects so that output continues on an even-numbered page.

Defaults:

This command word will create a break when encountered.

Notes:

(1) If empty-page suppression is on (.EM NO), the page number will be incremented to the next even page and output will continue on the next physical page.

RIGHT ADJUST control

Purpose:

The RIGHT ADJUST control word causes the next input line to be right adjusted in the output line.

Format:

Usage:

The next input text line, including any leading or trailing blanks, will be right adjusted in the output line.

Defaults:

This command word will create a break when encountered.

Notes:

(1) If the next line is longer than the current line length, it will be truncated.

Example:

(1) .ra
(1.5)
Produces:

(1.5)

READ CONTROL control

Purpose:

The READ CONTROL control word allows the user to enter control or input lines(39) during processing of the input file.

Format:

.RC	1	
	n	

Usage:

When the .RC control word is encountered, the typeball is jiggled, the user's terminal keyboard is unlocked, and n lines are accepted and processed as if they had been in the input file. The lines thus read may be prose or control information.

Defaults:

The ".rc" control word does not act as a break in itself. However, control words read under its control may act as breaks. If the operand is omitted then n = 1 is assumed.

Notes:

(1) If output is being placed on the online terminal, the typing element will space but not print(40). The user should manually space the carriage back one line to leave it properly aligned after the

---

(39) Note that the .RD control word merely unlocks the keyboard to allow the user to type. It does not look at what the user types.

(40) The typing element will print during the first pass of a TWOPASS run even if output is online.

carriage return which ends the line he types.
(2) A completely null line is taken to mean that the user is done typing input lines. The .RC is terminated regardless of whether n lines have been read. A null line signals and End Of File (EOF).
(3) The user might wish to use .RC in order to dynamically specify control words or prose.
(4) The DDNAME CONTROL is used for input for the ".rc" control word.

Example:

```
.cm user will enter name and address here
.rc 4
```

READ control

Purpose:

The READ control word allows the user to type a line on the output page during NSCRIPT output.

Format:

.RD	n

Usage:

When the .RD control word is encountered during output to the user's terminal, NSCRIPT will spin the user's typeball and unlock his keyboard until "n" carriage returns have been typed. The line typed is ignored completely except for purposes of counting lines on the current page.

When the .RD control word is encountered during output to the offline printer, NSCRIPT will insert n blank lines in the output.

This control word may be useful to allow addresses to be inserted in form letters or to allow the user to change typeballs.

Defaults:

This command word does create a break when encountered. If the operand is omitted n = 1 is assumed.

Notes:

(1) If output is offline and the .RD is received within n lines of the bottom margin and ".LS NO" is in effect, spaces will not appear at the top of the next page. If however, output is online and the other two conditions are met, spaces will appear at the top of the next page.

REMOTE control

Purpose:

The REMOTE control word allows the user to save one or more input lines, which will be automatically imbedded at a specific place on the current page, the next page, or subsequent pages.

Format:

.RM	n	SAVE NOSAVE DELETE

Where:

n is a decimal integer in the range TOP MARGIN < n < BOTTOM MARGIN

Usage:

The input lines between the first .RM and the next .RM are saved. When the next line n is to be printed, the saved lines are automatically interpreted.

If NOSAVE is specified or assumed by default, the saved lines are erased after the first use.

If SAVE is specified, the remote sequence is saved and invoked on line n of every page until a .RM n DELETE is received specifying the same line number.

The .RM control word is useful for defining multi-line headers or for causing automatic insertion of figures.

Following the imbedding of a remote sequence, line formatting options are restored to their values prior to the insertion of the remote.

.RM

Defaults:

This command word will not create a break when encountered. If both operands are omitted the first operand will be set to the TOP MARGIN plus one, i.e. (n = TM + 1), and the second operand will become "NOSAVE". If only the first operand is specified, the second will become "NOSAVE".

Notes:

(1) Any lines whatever may appear within the .RM (except another .RM) and are interpreted when the remote sequence is inserted.

(2) The combined total number of footnote and remote lines may not exceed 100.

(3) Remote sequences are "triggered" when line n of the output page is about to be printed. While a remote sequence is being used for input, no other remote may be triggered. If two or more remotes specify the same line number, only the oldest will be triggered.

(4) A .RM DELETE deletes the oldest sequence for the specified line number.

(5) If no arguments are presented, a line number one bigger than the current top margin is assumed.

Example:

(1) Assume a TOP MARGIN of 6. The following sequence places a figure at the top of the next page:

```
.rm 7
.in
   (Figure)
.ce
FIGURE ONE: Water Concentration in Lake Erie as
.ce
a Function of GNP.
.rm
```

ROMAN control

Purpose:

The ROMAN control word causes page numbers to use lower-case roman numerals.

Format:

Usage:

The .RO control word causes all page numbers produced hereafter in headings or footings to be printed in lower-case roman numerals.

Defaults:

This command word will not create a break and unless encountered will not be in effect.

SPACE control

Purpose:

The SPACE control word generates a specified number of blank lines before the next typed line.

Format:

.SP	n

Usage:

The SPACE control word causes n blank lines to be typed. If the end of the page is reached before satisfying the request, the remaining spaces may or may not be typed on the next page depending on whether .LS YES has been specified. If DOUBLE SPACE is in effect, twice the specified number of spaces are typed.

Defaults:

This command does create a break when encountered and if the operand is omitted n = 1 will be assumed.

Notes:

(1) .SP control words within footnotes cause blank lines to be generated within the printout of the footnote at the bottom of the page.

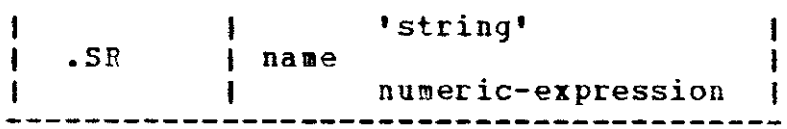
(2) If the operand is omitted, 1 is assumed.

SET REFERENCE control

Purpose:

The SET REFERENCE control word allows the user to assign a character or numeric value to a symbolic reference name.

Format:



Usage:

The reference name named by the first operand is given the value of the second operand. If the reference name does not exist, it will be created.

The .SR control word may be used for a variety of purposes. For example:

(1) Symbols may be defined as having as their value the number of the page on which they appear and can be used to construct a table of contents.
(2) Symbols can be assigned to count the number of equations being used and to number the equations as they appear on the output.

The reference name may be any character string of length 7 or fewer which does not contain a blank or period. The name is converted to upper case prior to use.

If a character string is the assigned value, it must consist of 8 or fewer characters and must not include blanks. Quotes within the string need not be doubled.

The form of the numeric-expression which may be used as the assigned value is as follows:

```
<expression> =:: <term> | <expression> <op> <term>
<term> =:: <escape> | <decimal-integer>
<op> =:: + | - | * | /
```

.SR

At least one blank must separate all items in the expression. That is, "5 + % - 7" is valid, while "5+%-7" is not. One unary "+" or "-" is allowed to immediately(41) precede the number. Operators are performed left to right.

The following <escape>'s are recognized:

(1) "%" Current page number.
(2) "<*" Current line number of file.
(3) "<l" Current line number of page.

Defaults:

This command word will not create a break when encountered. If one or both operands are omitted it will be treated as an error.

Notes:

(1) No more than 200 reference names may be defined during one run of NSCRIPT.

(2) See the description of the .UR control word and the TWOPASS option for further hints on the use of .SR.

Example:

(1) The sequence:

```
.sr eqnno 0
   .
   .
   .
.ur .sr eqnno &eqnno + 1
.ur .sr xeqn &eqnno
.ur x = erfc(a - bc)                              (&eqnno.)
.ur .sr eqnno &eqnno + 1
.ur y = erfc(a + bc)                              (&eqnno.)
   .
   .
   .
.ur Using the previous result for x (Equation &xeqn.)...
```

---

(41) With no intervening blanks.

```
----------
|        |
|  .SR   |
|        |
----------
```

Produces:

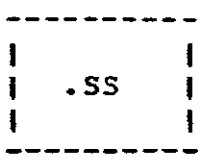
x = erfc(a - bc) (1)
y = erfc(a + bc) (2)
Using the previous result for x (Equation 1)...

(2) The sequence:

```
.sr a 0
.sr b -5
.sr c 'a'
.sr d 'b'
.ur .ur .sr &c &&&d + 1
.ur a = &a..
.ur b = &b..
```

Produces:

```
a = -4.
b = -5.
```



SINGLE SPACE control

Purpose:

The SINGLE SPACE control word causes output to be single spaced.

Format:

Usage:

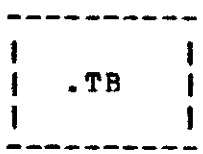
Subsequent output lines will be single spaced.

Defaults:

When this command word is encountered it will create a break. Unless otherwise specified ".ss" will be in effect.

Notes:

(1) Since SINGLE SPACE is the normal mode, this control word is used to undo the effect of a previous .DS.



TABULATION SET control

Purpose:

The TAB SETTING control word specifies the tabs stops to be assumed for the following lines when converting the TAB character (X'05') generated by the typewriter TAB key into the appropriate number of spaces.

Format:

.TB	T1 T2 ... Tn

Usage:

Tabulation characters present in the SCRIPT file are expanded by NSCRIPT into one or more blanks to simulate the effect of several logical tab positions. The .TB control word specifies the locations of these logical tab stops.

A .TB control word with no operands causes reversion to the default tab settings.

Defaults:

This control word will create a break. Until ".tb" is encountered with operands the tab columns are 5, 10, 15, ..., 80. They will be in effect from the beginning or when ".tb" is encountered without operands.

Notes:

(1) The tab stops must be monotonically increasing.

(2) The SCRIPT dataset type must be used with .TB. If a TEXT dataset type is used for the input dataset, tabs must be specified with the TAB subcommand of the TSO editor, not the NSCRIPT control word .TB.

(3) To prevent inconsistencies when entering data containing tabs, the physical tab settings on the terminal should be the same as those specified in the .TB control word.

TOP MARGIN control

Purpose:

The TOP MARGIN control word specifies the number of lines which are to be placed between the physical top of the output page and the first line of text.

Format:

Usage:

Subsequent output pages will begin with m lines (which may include a heading line) before the first line of text.

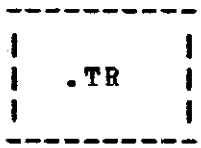
Defaults:

This command word will create a break and until encountered m = 5 will be in effect. When encountered without an operand m = 1 will be assumed. This may result in an error.

Notes:

(1) The TOP MARGIN must always be strictly greater than the HEADING MARGIN.

(2) The margin specified by .TM will apply to the current page if no output has yet been produced on it.



TRANSLATE control

Purpose:

The TRANSLATE control word allows the user to specify the contents of the translate table used to implement the TRANSLATE option.

Format:

.TR	S1 T1 <S2 T2 ... >

Where S1, T1, etc., are single characters or two-digit hexadecimal numbers using upper or lower case letters.

Usage:

If the TRANSLATE option was specified in the command line which invoked NSCRIPT, all subsequent output lines will be printed with all occurrences of "S1" replaced by "T1", etc.

The .TR control word is primarily of use when output must use a different character set than was used to create the SCRIPT or TEXT files. For example, the user may print on line a file which uses special characters not available on the terminal(42) or to use a "correspondence coded" secretary's typewriter typeball with a different type style to be used in place of the normal "terminal coded" typeball by using a translate table such as the table in Appendix D.

Defaults:

This command will not create a break when encountered. Lower case to upper case translation will be in effect until ".tr" is encountered with operands.

---

(42) For example, the TN-train's superscript characters are not available on the 2741 terminal.

.TR

Notes:

(1) Heading, footing, and footnote lines are translated under control of the translate table current when the line is output.

(2) NSCRIPT control lines are never translated.

(3) Translate pairs remain active until explicitly re-specified.

(4) The default translate table takes lower case letters into upper case letters and leaves all other characters the same.

(5) Hexadecimal numbers are recognized by the presence of 2 characters (instead of one) and may use upper or lower case letters A-F.

(6) The last pair in a .TR line may consist of only one argument, which indicates that the corresponding character is to be translated into itself (left unchanged).

Examples:

(1) .TR 8D (9D) B0 0 ... B9 9
Causes the TN-train's superscript parentheses and numbers to print as ordinary parentheses and numbers.

(2) .TR % 7B
Causes occurrences of the character "%" to be replaced by the character X'7B' (the "#" character, which is not easy to enter into a file because it can be the TSO character erase character).

(3) .TR 40 ?
Causes all blanks in the file to be typed as "?" on output.

(4) .TR 05 40
This is probably an attempt to remove all tabulation characters from the file, but it has no effect since all tab characters are removed prior to printing the output.

(5) .TR 40 ? $
Causes all blanks in the file to be typed as "?" on output, and the "$" to appear as "$" (i.e., to be translated into itself.)

UNDENT control

Purpose:

The UNDENT control word forces the next output line (only) to start a specified number of columns to the left of the current indent.

Format:

.UN	n

Usage:

The .UN control word causes only the next output line to begin n spaces further left than the current indentation.

The .UN control word serves a similar purpose as .OF, but in a different way. The choice between the two is largely a matter of personal preference.

Defaults:

This command will create a break when encountered. If the operand is omitted then n = 0 will be assumed.

Notes:

(1) The undentation may not exceed the current indentation.

Examples:

(1) The following two sequences are equivalent:

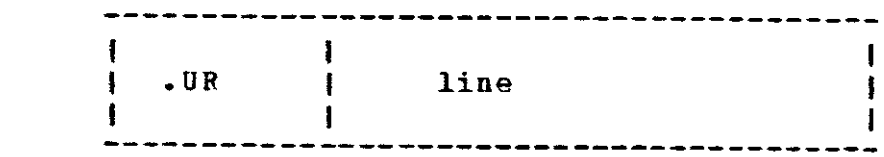
```
.in 5
.of 4
<text>

.in 9
.un 4
<text>
```

USE REFERENCE control

Purpose:

The USE REFERENCE control word causes an input line to be reformatted by substituting the current values of specified reference names in the line and to be processed as if it had been in the input.

Format:



Usage:

The line which begins one blank after the .UR control word is reformatted by replacing occurrences of strings of the form "&name" with the current value of the reference name called "name".

A reference name instance is denoted by the appearance of a "&" followed by the name of the desired reference name followed by either a blank or a period. The current value of the reference name is converted (if necessary) to a character string and substituted for the string "&name." (if a period is used for delimiting) or the string "&name" (if a blank is used).

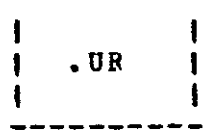
The line thus constructed is processed as if it had been in the original input stream.

Defaults:

The ".ur" control word does not act as a break itself. However, control words within the line may create a break when the line is reinterpreted after substituting symbolic references with their values. If line is omitted a blank line is assumed.

Notes:

(1) See the description of the .SR control word for additional information.



(2) If reference names appearing in the line have not yet been assigned a value via .SR, a null character string value will be assumed. Note, however, the effect of the "TWOPASS" option previously described. Note also that such reference names do not count toward the total of 200.

(3) Text &'s in a .UR line are indicated by "&&". Only one "&" appears in the reformatted line.

(4) Negative values are allowed for numeric reference names and result in a signed integer suitable for use with the .SR control word.

(5) The TAB character is <u>not</u> a blank.

SAMPLE INPUT TO NSCRIPT

```
.he 'AP-55-2'APPENDIX B'Page %'
.ce
SAMPLE OUTPUT FROM NSCRIPT
.rm 45
.ra
EXAMPLE OF REMOTE CONTROL
.rm
.sp 2
"Oh nuts!  I have to type this whole page over".  That
complaint is probably heard many times in the course of
preparing manuscripts--reports, letters, minutes,and so on.
Then there is the problem of making duplicate original
copies--of a typed
manuscript.  When we use an ordinary typewriter,
text-processing, as this is called, can be a time-consuming
and harrassing job.  The NSCRIPT facility, which operates
under TSO(*),
.fn begin
(*) Time Sharing Option for OS/360
.fn end
was written to handle these procedures for you;
you merely type in the first version and make
any necessary corrections.  By
using the "commands" that you type in with your lines of
text, the computer prints out as many "first" copies as you
wish, with margins, spacing, indentation, etc., performed
in accordance with your commands.
.sp 1
One of the most useful features of NSCRIPT is right-margin
justification, as in book and newspaper printing.  This
means that your text is spaced as evenly as possible
between the margins of your printed page, filling in with
blanks where  necessary.  The printed line is
under control of a "line-length" command; any
short line that you type in will "grab" words
from a longer line above or below it, and fill out the
line with blanks where necessary to avoid splitting words
at the margins.  This is the ordinary (or default) mode
under NSCRIPT.  You can, however, have lines printed out
exactly as you type them in, with no justification
performed by NSCRIPT, by including a command that instructs
the computer to turn off the format mode.  This is useful
for typing figures and charts.  Line justification can
later be respecified if desired.
.cm Last line of Input Text.
```

SAMPLE OUTPUT FROM NSCRIPT

"Oh nuts! I have to type this whole page over". That complaint is probably heard many times in the course of preparing manuscripts--reports, letters, minutes,and so on. Then there is the problem of making duplicate original copies--of a typed manuscript. When we use an ordinary typewriter, text-processing, as this is called, can be a time-consuming and harrassing job. The NSCRIPT facility, which operates under TSO(*), was written to handle these procedures for you; you merely type in the first version and make any necessary corrections. By using the "commands" that you type in with your lines of text, the computer prints out as many "first" copies as you wish, with margins, spacing, indentation, etc., performed in accordance with your commands.

One of the most useful features of NSCRIPT is right-margin justification, as in book and newspaper printing. This means that your text is spaced as evenly as possible between the margins of your printed page, filling in with blanks where necessary. The printed line is under control of a "line-length" command; any short line that you type in will "grab" words from a longer line above or below it, and fill out the line with blanks where necessary to avoid splitting words at the margins. This is the ordinary (or default) mode under NSCRIPT. You can, however, have lines printed out exactly as you type them in, with no justification performed by NSCRIPT, by including a command that instructs the computer to turn off the format mode. This is useful for typing figures and charts. Line justification can later be respecified if desired.

EXAMPLE OF REMOTE CONTROL

(*) Time Sharing Option for OS/360

CONTROL STATEMENTS FOR SUBMITTED OR BATCH JOBS

```
/*FORMAT   PR,DDNAME=SCRIPT,TRAIN=TN
/*FORMAT   PR,DDNAME=SYSPRINT,TRAIN=TN
//stepnam1 EXEC PGM=SCRIPT,PARM=(options)
//SYSPRINT DD  SYSOUT=A,DCB=BLKSIZE=138
//SCRIPT   DD  SYSOUT=A,DCB=BLKSIZE=2000
//SCRIPTIN DD  DSNAME=U.projno.progno.name1.TEXT,DISP=SHR
//CONTROL  DD  DSNAME=U.projno.progno.name2.TEXT,DISP=SHR
//SYSLIB DD DSNAME=U.projno.progno.name3.TEXT,DISP=SHR
/*
```

Where:

options — The PARM parameter on the EXEC statement contains the "options" which are as specified and used as described on page 1, with the following exceptions. The CONTROL, LIBRARY, MESSAGE, and PRINT options cannot be used. The TWOPASS option must be specified as 2PASS. The PAGE(page-number) option must be specified as PAGExxx, where xxx ia a three digit number which must include leading zeros (e.g., PAGE003).

projno — corresponds to the user's project or problem number.

progno — corresponds to the user's programmer number.

name1, name2 — are the names the user gave to the datasets. The last level qualifier may be either TEXT or SCRIPT.

DDNAMES — required by NSCRIPT and their functions are listed below:

SCRIPTIN — is for the primary input TEXT or SCRIPT dataset. The dataset name referenced should be equivalent to the Online Command Format's dataset name parameter being the dataset qualification level preceeding TEXT.

SYSPRINT — is for output of messages such as those from ".pr" and those initiated by errors. It is the equivalent of the MESSAGE dataset.

CONTROL is used for input and responses to messages printed in SYSPRINT. The NSCRIPT control words ".rc", ".rd" use this DDNAME to recieve input lines. If ".rc", ".rd" are not used this DD Statement can be omitted.

SYSLIB is searched for members specified in .AP or .IM control words. This DD statement is optional.

SCRIPT is for the formatted and edited output from NSCRIPT.

Notes:

(1) Datasets used by NSCRIPT must have variable length records. Any BLKSIZE that does not span tracks can be used for the input datasets. Output datasets should have a minimum blocksize of 138 and can have a maximum equal to that allowed by ASP.

(2) If the SYSLIB option is not used, each IMBED or APPEND filename must have an additional DD Statement included in the job setup with the filename used as the DDNAME.

(3) If TRANSLATE is specified then the TN train is not required(provided you use the default translate table) and the "/* FORMAT" ASP statements should not be included in the job setup.

"CORRESPONDENCE CODED"
SECRETARY'S TYPERWRITER TYPEBALL
TRANSLATE TABLE

```
.tr 1 y 2 2 3 3 4 8 5 4 6 6 7 5 8 7 9 # 0 9 - . & %
.tr q f w $ e u r n t @ y i u t i o o q p d @ <
.tr a p s r d v f c g a h z j & k w l y $ * # ;
.tr z 0 x / c x v l b , n s m j , g . k / h
.tr = b ; e : E ' m * > ( " ) ( _ ¬ + B
.tr Q F W ! E U R N T ¢ Y I U T I O O Q P D ¢ '
.tr A P S R D V F C G A H Z J + K W L Y ! - " M
.tr Z ) X ? C X V L B | N S M J ? H
.tr < 40 > 40 | 40 ¬ 40
.tr 8d 40 9d 40 b0 40 b1 40 b2 40 b3 40 b4 40 b5 40
.tr b6 40 b7 40 b8 40 b9 40 00 40 ¢ 40 02 40
```