July 3, 2012

MEMORANDUM

To: SCRIPT Users

Subject: UOW SCRIPT - VERSION(3.2) 78JAN13

From: Mrs. Sandi Ward
Department of Computing Services
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1

Disclaimer:

Although this programme has been tested by its developer, no warranty, expressed or implied, is made by the developer or the University of Waterloo, as to the accuracy and functioning of the programme and related programme material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the developer or the University of Waterloo, in connection therewith.

Distribution of the current version of SCRIPT will be made to those users who submit a 1200 foot or larger reel of tape and a check or purchase order for twenty-five dollars ($25 in Canadian funds) to cover costs of distribution. The distribution tape is the result of doing IEHMOVEs from 2314 disk to tape. The tape will be 9-track and 800 or 1600 BPI depending upon request. It is labelled 'SCRIPT' and contains the following datasets.

DSNAME	SEQ	CONTENTS
SCRIPT.TEXT	1	Object decks SPACE=(3200,(80,,3)) RECFM=FB,LRECL=80,BLKSIZE=3200
SCRIPT.FMEMO	2	Documentation (Recfm=FB, Unsequenced for CMS) SPACE=(3360,(500,,12)) RECFM=FB,LRECL=80,BLKSIZE=3360
SCRIPT.MEMO	3	Documentation (Recfm=VB, Unsequenced for OS) SPACE=(3360,(220,,12)) RECFM=VB,LRECL=84,BLKSIZE=3360
SCRIPT.SOURCE	4	Source SPACE=(3360,(200,,3)) RECFM=FB,LRECL=80,BLKSIZE=3360
SCRIPT.MACLIB	5	Macros and Copy Code SPACE=(3360,(800,,20)) RECFM=FB,LRECL=80,BLKSIZE=3360
SCRIPT.LOAD	6	Load module SPACE=(7294,(40,,3)) RECFM=U,LRECL=7294,BLKSIZE=7294
SCRIPT.SYSHYPH	7	Hyphenation Exception Dictionary SPACE=(3360,(20)) RECFM=VB,LRECL=80,BLKSIZE=3360
SCRIPT.TSO	8	Alternate TSO Interface material SPACE=(3360,(120,,3)) RECFM=FB,LRECL=80,BLKSIZE=3360

Following is suggested JCL to assemble and linkedit one new member '??' into SCRIPT for use under OS. Note that source members CMSCRIPT and TSOCRIPT are not assembled this way.

```
//??  JOB  '############,T=1,P=200'
//*  SCRASLK - TO ASSEMBLE AND LINK A SCRIPT MODULE
//STEP1  EXEC  PGM=ASMGASM,REGION=150K,
//             PARM='TERM'
//SYSLIB   DD  DSN=SCRIPT.MACLIB,DISP=SHR
//         DD  DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD  DSN=&&SYSUT1,DISP=(NEW,DELETE,DELETE),
//         SPACE=(CYL,(6,1)),UNIT=2314
//SYSUT2   DD  DSN=&&SYSUT2,DISP=(NEW,DELETE,DELETE),
//         SPACE=(CYL,(8,2)),UNIT=(2314,SEP=SYSUT1)
//SYSUT3   DD  DSN=&&SYSUT3,DISP=(NEW,DELETE,DELETE),
//         SPACE=(CYL,(8,2)),UNIT=(2314,SEP=(SYSUT1,SYSUT2))
//SYSPRINT DD SYSOUT=A
//SYSTERM  DD SYSOUT=A
//SYSLIN   DD DSN=SCRIPT.TEXT(??),DISP=OLD
//SYSIN    DD DSN=SCRIPT.SOURCE(??),DISP=SHR
//LKED    EXEC  PGM=LINKEDIT,REGION=180K,
//              PARM='LIST,XREF,NCAL,SIZE=(180K,50K)'
//SYSPRINT  DD  SYSOUT=A
//SYSUT1    DD  DSN=&&SYSUT,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1,1)),UNIT=2314
//MODLIB    DD  DSN=SCRIPT.TEXT,DISP=SHR
//SYSLMOD   DD  DSN=SCRIPT.LOAD,DISP=OLD
//SYSLIN    DD  *
         INCLUDE   MODLIB(SCRIPTW)
         INCLUDE   MODLIB(SCRIPTCW)
         INCLUDE   MODLIB(SCRIPTFO)
         INCLUDE   MODLIB(SCRIPTHY)
         INCLUDE   MODLIB(SCRIPTMC)
         INCLUDE   MODLIB(SCRIPTOU)
         INCLUDE   MODLIB(SCRIPTPM)
         INCLUDE   MODLIB(SCRIPTRD)
         INCLUDE   MODLIB(SCRIPTSY)
         INCLUDE   MODLIB(SYSINT)
         ENTRY     SCRIPTW
         NAME      SCRIPT(R)
/*
```

The inclusion of SCRIPTHY, the hyphenation routine, may be deleted to save some 12K in the size of the module. Harmless Unresolved External References will result and the Hyphenation (.HY) facilities may not be used.

Following is a suggested procedure for running SCRIPT in OS batch.

```
//SCRIPT   PROC PROG=SCRIPT,SCRIPT=,SIZE=128K
//SCRIPT   EXEC PGM=&PROG,PARM='&SCRIPT',REGION=&SIZE
//*
//*      S C R I P T
//*
//STEPLIB   DD  DSN=SCRIPT.LOAD,DISP=SHR
//SYSHYPH   DD  DSN=SCRIPT.SYSHYPH,DISP=SHR
//SYSPRINT  DD  SYSOUT=A
//SYSTERM   DD  SYSOUT=A
```

Following is a suggested procedure for formatting a SCRIPT manual in OS batch.

```
//SCRIPTMN PROC PROG=SCRIPT,MEM=SCRIPT,SCRIPT=,SIZE=160K
//         EXEC PGM=&PROG,PARM='&SCRIPT',
//             REGION=&SIZE
//*
//*      TO FORMAT AN OFFLINE SCRIPT MANUAL
//*
//STEPLIB   DD  DSN=SCRIPT.LOAD,DISP=SHR
//SYSHYPH   DD  DSN=SCRIPT.SYSHYPH,DISP=SHR
//SYSPRINT  DD  SYSOUT=A
//SYSTERM   DD  SYSOUT=A
//SYSLIB    DD  DSN=SCRIPT.MEMO,DISP=SHR
//SYSIN     DD  DSN=SCRIPT.MEMO(&MEM),DISP=SHR
```

Then to format a SCRIPT Reference Guide, run the following job:

```
//SCRIPTMN  JOB  '#,T=2,P=200'
//     EXEC  SCRIPTMN
```

Note that using the default PASS=1 option causes the table of contents to print at the end of the manual. To make the table of contents print at the start, after the header page, add PASS=2 to the parm field.

Member SYSPUB within SCRIPT.MEMO contains a sample set of SCRIPT high level Macros for formatting a thesis or other research report in chapters and sections with an automatic Table of Contents and Figure, Footnote and Table numbering. To format documentation for SYSPUB that uses SYSPUB with automatic hyphenation:

```
//SYSPUBMN  JOB  '#,T=1,P=50'
//     EXEC  SCRIPTMN,MEM=SYSPUBMN,SIZE=250K
```

To format an Introduction to SCRIPT that uses SYSPUB with automatic hyphenation:

```
//SCRINTRO  JOB  '#,T=1,P=50'
//     EXEC  SCRIPTMN,MEM=SCRINTRO,SIZE=250K
```

To format a SCRIPT Techniques Guide that uses SYSPUB with automatic hyphenation:

```
//SCRIPTNI  JOB  '#,T=1,P=50'
//     EXEC  SCRIPTMN,MEM=SCRIPTNI,SIZE=250K
```

To format a sample SCRIPT Reference Card:

```
//SCRIPTCD  JOB  '#,T=1,P=20'
//     EXEC  SCRIPTMN,MEM=SCRIPTCD
```

The following discourse concerns generating and running SCRIPT under CMS.

Since the distribution tape is in unloaded IEHMOVE format the data must be restored using the IEHMOVE facilities of the TAPEPDS and TAPEMAC commands, except SCRIPT.SYSHYPH which is a sequential file.

Source member CMSCRIPT is the SCRIPT interface routine. Assembling CMSCRIPT requires SCRIPT.MACLIB and CMSLIB. The CMSCRIPT TEXT deck produced includes an ordered table of V-cons which forces the CMS LOADER to automatically include all of the required SCRIPT TEXT decks. To generate the SCRIPT MODULE which includes SCRIPT's interface routine perform the following:

```
LOAD CMSCRIPT (CLEAR
GENMOD SCRIPT (FROM CMSCRIPT
```

To assemble the source decks of SCRIPT requires SCRIPT.MACLIB and the OS macros from the CMS equivalent of SYS1.MACLIB. The SCRIPT source decks that you may wish to assemble all have filenames of pattern 'SCRIPT*' and 'SYSINT'.

Under CMS there is support to type a file "SCRIPT MEMO Y" in response to a filename containing a '?'. The following is a suggestion for this file:

```
FORMAT:   SCRipt fname ( option1 ... optionN )
                 where fname has filetype of SCRIPT
                 and fname '?' for HELP.

OPTIONS:  (Capital letters indicate minimum length,
           default comes first)

TErminal (or ONline  NOFIle) - output goes to terminal
MEmo     (or ONline  FIle  ) - output goes to 'fname MEMO'
DIsk     (or OFfline FIle  ) - output goes to 'fname LISTING'
PRinter  (or OFfline NOFIle) - output goes to printer
NOPRint                      - no output produced

NOADjust/ADjust  - thirty columns from left margin
NOCEnter/CEnter  - same as ADJUST
NOCOnt/COntinue  - continue in spite of errors
NODEBUG/DEBUG    - for program debugging features
FOrmat/NOFOrmat  - to disable text formatting
LOCal/GLObal     - to define positional call arguments
NOMArk/MArk      - to indicate start of text lines
NONUmber/NUmber  - to show filename and record number
PROFile/NOPROFile - to imbed "PROFILE SCRIPT"
NOQUiet/QUiet    - to suppress ONLINE version message
NOSCREen/SCREen  - for ONLINE output to a CRT
SIX/EIGHT        - initial six or eight lines per 11" page
NOSTAT/STAT      - statistics on error file
NOSTop/STop      - wait at bottom of each page
NOTRans/TRans    - uppercase conversion of output
ONEPass/TWOPass  - number of passes to produce output
UPper/NOUPper    - reference names to uppercase
WAit/NOWAit      - prompt for first ONLINE page insertion

ADjust=<0|nn>     - adjust output from left margin
BMargin=<6|nn>    - default BOTTOM MARGIN
CENTER=<0|nn>     - same as ADJUST
COntinue=<0|n>    - continue for number of errors
DArk=<1|n>        - DARK Offline Output
FFCHannel=<1|n>   - printer channel skip to align page
FFTOp=<3|n>       - line number of page with channel skip
FMargin=<1|nn>    - default FOOTING MARGIN
FNSize=<200|nn>   - max outstanding footnote lines
HMargin=<1|nn>    - default HEADING MARGIN
HSFSover=<9|nn>   - Top Title/Bottom Title overlap
LINenum/LEGalnum=<0|nn> - print page line number in column nn
LLength=<60|nn>   - default LINE LENGTH
NUmber=<0|nn>     - same as NUMBER in column nn
PAge=<1|mm<:nn>>  - output from page mm to page nn
PASses=<1|nn>     - number of passes to produce output
PLength=<66|nn>   - default PAGE LENGTH
RMSize=<200|nn>   - max records per remote
SEQCol=<240|nn>   - sequence column on variable files
SRLen=<150|nn>    - max length of character value
TABLeft=<0|1>     - left tab adjustment
TMargin=<6|nn>    - default TOP MARGIN

&string value     - set reference value externally
```

The following concerns an experimental program called NSCATS that can convert most of an ATS Archive Document to SCRIPT Input format (RECFM=V, unsequenced). The source is in SCRIPT.SOURCE(NSCATS) and should be compiled with PL1/F. The load module is in SCRIPT.LOAD(NSCATS). To use it requires the following JCL:

```
//NSCATS  JOB  '#,T=1,P=20'
//       EXEC  PGM=NSCATS,REGION=100K
//STEPLIB   DD  DSN=SCRIPT.LOAD,DISP=SHR
//SYSPRINT  DD  SYSOUT=A
//SYSUT1    DD  DSN=ATS.ARCHIVE,UNIT=2400,VOL=SER=??????,
//              DISP=OLD,LABEL=(2,BLP),
//              DCB=(RECFM=F,LRECL=3100,BLKSIZE=3100)
//SYSUT2    DD  DSN=SCRIPT.INPUT,UNIT=2314,VOL=SER=??????,
//              DISP=(NEW,KEEP),SPACE=(TRK,(10,10)),
//              DCB=(RECFM=VB,LRECL=244,BLKSIZE=3520)
```

This version of SCRIPT should run successfully under TSO. A sample TSO interface routine is provided in source member TSOCRIPT. This routine is not completely up to date -- see comments at start of code. A second TSO interface routine which is larger and more up to date is found in file 'SCRIPT.TSO'. See member 'NOTES' within this file for more information. To make SCRIPT compatible with the TSO Editor, sequencing in first eight columns of variable length files is supported via the SEQCOLUMN= parameter.

This current version of SCRIPT was assembled with Assembler (G) V2L7a. Though not every routine was tested, it is expected that Assembler (H) or the IBM VS/VM System Assembler, Assember (XF) may also be used. It should be noted that any possible future versions may again be resequenced and possibly restructured.

This tape is being distributed with no warranty, expressed or implied. The Department of Computing Services is interested in receiving bug reports and ideas for extensions. Moreover, we ask that any extensions made to this package be reported and if possible, that the source changes be made available.