FULL SCREEN EDITOR

UNDER NOS 2.1

83/06/28

SOFTWARE RELEASE BULLETIN

SMD130521

1.0 Full Screen Editor

This version of the Full Screen Editor (FSE) provides NOS V2.1 users with most of the editing capabilities to be released in NOS V2.2. These capabilities include:

- o Direct keyboard manipulation of text for screen terminals.
- o Editing directives for hard-copy and screen terminals.
- o Redefinable function keys.
- o Editor procedure libraries, including iterative operations and automatic configuration on startup.
- o On-line HELP.
- o UNDO capability for reversing changes to the text.
- o MARK capability for delimiting text to be moved, copied, etc.
- o Menu displays for global locate/replace.
- o Split screen displays for multiple file editing.
- o Support for 6-bit, 6/12 bit, and 8/12 bit character sets.
- o Word processing functions such as centering lines, splitting lines, joining lines, and paragraph justification.
- o Full screen support for CDC 721, CDC 722, and ANSI X3.64 terminals such as the DEC VT100*, Zenith Z19**, and Heathkit H19***.

* DEC is a registered trademark of the Digital Equipment Corporation.

** Zenith Z19 is a product of the Zenith Corporation.

*** Heathkit H19 is a product of the Heath Corporation.

The NOS FSE User's Guide in its final printed form will be available from Literature Distribution Services (LDS) with the release of NOS V2.2 (publication number 60460420). Included with the NOS V2.1 release is a copy of the preliminary draft of the NOS FSE User's Guide for NOS V2.2. The NOS V2.1 version of FSE differs from the NOS V2.2 documentation as follows:

o Selecting screen mode

The SCREEN command does not exist in NOS V2.1 IAF. Thus, FSE has no way to determine the terminal model at the start of each edit session, and will initialize as a line editor by default. The user can then switch to screen mode with one of the following four directives:

1. SET SCREEN 721
2. SET SCREEN 722
3. SET SCREEN VT100
4. SET SCREEN Z19 (also to be used for Heatkit H19 terminals)

The user can use other techniques to indicate terminal type. The FSE command (i.e., control statement) can accept editing directives in the comment field, as in:

```
FSE,MYFILE.SET SCREEN 721
```

The other alternative, which should be used only by the user expecting to always use FSE from the same terminal model, is to include a SET SCREEN directive in the STARTUP procedure of the FSEPROC file.

- FSE command in non-batch IAF subsystems

  The NOS V2.1 IAF has no command table entry for FSE, thus the FSE command will be recognized only in the batch subsystem. Sites can easily develop IAF modifications to add FSE to the command table for the other subsystems.

- No-auto-drop file status

  The NOS V2.1 version of FSE will not establish the no-auto-drop attribute for the ZZZWORK file. This means that you cannot resume a previous edit session if the ZZZWORK file was released from your job by commands like NEW, OLD, CLEAR, RETURN(*), and UNLOAD(*).

- No multi-user implementation

  The NOS V2.2 release of FSE will include an optional multi-user subsystem version of FSE. This subsystem is not provided in the NOS V2.1 version. There is no user incompatibility due to this restriction, but the performance of the NOS V2.1 single-user version will be less than that of the NOS V2.2 multi-user version.

The Program Library for the NOS V2.1 FSE is identical to that for NOS V2.2; we will not resequence the PL for NOS V2.2. PSRs will be accepted against the NOS V2.1 FSE and corrective code will be available in a future NOS V2 release and through SOLVER.

2.0 Release Materials

NOS V2.1 FSE resides on the tape known as REL1K. REL1K contains all the binary, program library, procedure, and data files necessary to load NOS FSE V2.1. It has the following characteristics: 7 track (800 bits per inch (bpi)) or 9 track (1600 characters per inch (cpi)), binary recording mode, FSE1P0*NOS580577 as file identifier in HDR1 label, and ten files:

File 1 - Installation procedure (CCL) verification job

File 2 - Absolute binary of FSEEX Executive

File 3 - FSEPL program library in MODIFY format

File 4 - FSE build procedure (CCL)

File 5 - Edit help file (6/12 ASCII format)

File 6 - Default edit procedure file

File 7 - Edit teach file (6/12 ASCII format)

File 8 - Sample FSE edit file

File 9 - Sample FSE directive file

File 10 - Verification output for sample problem in files 8 and 9

3.0 Installation Procedure

Listings of the installation procedure, and the FSE build procedure may be obtained from REL1K by executing the following job:

```
    Job Control Card.
    USER,username,password,familyname.
    CHARGE,*.
                            HY
    LABEL,TAPE,VSN=REL1K,D= PE ,R,L=FSE1P0*NOS580577.
    COPYSBF,TAPE.
    SKIPF,TAPE,2.
    COPYSBF,TAPE.
    7/8/9
    6/7/8/9
```

NOS V2.1 FSE is installed by executing a job (described below) which invokes the CCL procedure contained in the first record of file 1 on the release tape REL1K. This job will copy the NOS V2.1 FSE product to permanent files, then execute the FSE sample problem to verify the installation. The permanent files associated with each file on the release tape are:

	Name	Type
File 1	Not retained on a permanent file	
File 2	FSELGO	D/A
File 3	FSEPL	D/A
File 4	BLDFSE	I/A
File 5	FSEHELP	D/A
File 6	FSEPROC	I/A
File 7	FSTEACH	I/A
File 8	Not retained on a permanent file	
File 9	Not retained on a permanent file	
File 10	Not retained on a permanent file	

The NOS V2.1 FSE binary and other needed permanent files are installed by executing the following job at the system console:

```
X.DIS.
SUI,userindex.
                              HY
LABEL,TAPE,VSN=REL1K,D= PE ,R,L=FSE1PQ*NOS580577.
COPYBF,TAPE,INSTALL.
BEGIN,,INSTALL.
```

The dayfile will show a verify good when the sample output (LISTOUT) is compared to the sample test output (LIST).

Once verification is complete, the binary for FSEEX (FSELGO) can be SYSEDITed onto the running system by doing the following:

```
ATTACH,FSELGO.
SYSEDIT,B=FSELGO,I=0,C.
```

If a SYSEDIT is inappropriate, then adding the FSELGO binary to the deadstart tape through LIBEDIT can be accomplished by doing the following:

```
COMMON,SYSTEM.
LIBEDIT,P=SYSTEM,B=FSELGO,Z.+*I,ABS/EDIT,FSEEX
```

This will add the absolute binary FSEEX after the existing EDIT binary and write a new system file to file NEW. File NEW can then be copied to a deadstart tape.

DROP.

3.1 Build Procedure

The NOS V2.1 FSE build procedure (BLDFSE) is provided so that sites may build their own binaries of FSEEX. The previous installation procedure (INSTALL) must still be run since it generates permanent files needed by FSE (i.e., FSEPL, FSEHELP, FSEPROC, and FSTEACH). FSE can be built by executing the following job at the system console:

```
X.DIS.
SUI,userindex.
ATTACH,OPL/UN=LIBRARY.   (composite NOS PL, or
                          wherever it is maintained)
BEGIN,,BLDFSE.
```

This procedure will produce the following four files:

1. LGO = Binary of FSEEX.

2. FSEOUT = Listing of FSEEX.

3. COMPOUT = Listing of FSELIB.

4. LOADOUT = Load map of FSEEX.

Again this binary can either be SYSEDITed or LIBEDITed onto the running system. Refer to the steps on page 6.

```
DROP.
```

3.2 File Placement

Three files should be stored under user name LIBRARY - the FSEHELP file, the FSTEACH file, and the default FSEPROC procedure file. Use the MOVEPF utility called by the following formats of the DSD entries. You must run MOVEPF at the system console.

```
X.MOVEPF(UI=userindex,DI=377776,F1=FSEHELP)
X.MOVEPF(UI=userindex,DI=377776,F1=FSEPROC)
X.MOVEPF(UI=userindex,DI=377776,F1=FSTEACH)
```

FSEHELP is direct access and FSEPROC and FSTEACH are indirect access. After moving these files to user name LIBRARY, make all of them public with read permission.

3.3 Notes and Cautions

The installation procedure will PURGE permanent files which match the names of FSELGO, FSEPL, BLDFSE, FSEHELP, FSEPROC, and FSTEACH.