

This paper represents the beginning of a 'helpful hints on how to do your job's document. Some of the areas discussed are still incomplete. I will be periodically adding information to the document. If you have any questions or suggestions, please see Tom McGee. Appendix A lists background documents and how to obtain them.

Update_History

Date	Changes
2/8/80	Section 3.0 revised
2/12/80	Appendix A Section 3.3 corrected
2/13/80	Section 1.0 revised
9/11/80	Section 3.5.2.8 revised, Section 3.5.2.8.4 added
12/22/80	Revisions for NDS/VE Phase C

1.0 DUAL STATE NOS/VE DEADSTART

1.0 DUAL_STATE_NOS/VE_DEADSTART

1.1 A170_NOS_DEADSTART

- o The system is configured to run with three FMD units (41, 42 and 43). No 844 drives are needed.
- o Set the D/S panel to deadstart from the primary system disk.
- o Push D/S button
- o Select "D" display
- o Select "H" display
- o Enter CM=10000
- o Enter (CR)
Enter date/time
Wait for deadstart to complete

Note: The deadstart tape which matches this A170 NOS system is found in the tape cabinet back by the curtains, where all the other Integration tapes are kept, labeled DUAL6M.

1.2 CURRENT_DUAL_STATE_CONFIGURATION

- o FMD Unit 43

This unit contains the following:

- A170 NOS (Build 4 level), CTI, CMSE, EI binaries, NOS deadstart files
- Files associated with user number LIBRARY
- Files associated with user number SES
- Files associated with DEV1, REL1, INT1.

- o FMD Unit 41

- A170 NOS (Build 6 level), CTI, MSL, EI binaries, NOS deadstart file

1.0 DUAL STATE-NOS/VE- DEADSTART
1.2 CURRENT DUAL STATE CONFIGURATION

This is a scratch unit

- o FMD Unit 42

This unit contains the following:

- NOS/VE Development Area PL's and Member PL's
- NOS/VE Deadstart Files to be tested (saved in individual user's catalogs)
- Files associated with user number INT2

1.3 DUAL STATE, NOS OPERATION

- 1) The convention used for creating user numbers on NOS/VE is as follows:

- o Your user number will be your initials.
- o Your password will be these 3 letters followed by the letter 'x'.
- o You must see COMSOURCE (R.K. Cooper - x3092) to be assigned a user index

User numbers are created by executing the program "MODVAL" as follows:

- o Type "X.MODVAL".
- o Type "K,m." where m is the MODVAL control point
- o Type "K.C,uuu." where uuu = your user number. Note that the "K." stays on the screen.
- o Type "K.PW=uuuX, FUI=n." where n = your user index
- o Set all other parameters to their maximum values. Do a "+" to see next page - there are 3 pages associated with a user number.
- o Type "K.END." to end creation of that user number. Another user number may now be created.
- o Type "K.END." again to exit MODVAL.

- 2) PF dumping and loading

You may use "SES.DUMPPF" on SN/101 to dump your permanent files to tape, and then load them onto your user number on A170 NOS using "SES.LOADPF". Documentation on how to use these SES procedure and what their parameters are is included in the SES "User's Guide, or they can be obtained by typing:

"SES,HELP.DUMPPF" and "SES,HELP.LOADPF".

1.0 DUAL STATE NOS/VE DEADSTART

1.4 NOS/VE DEADSTART

1.4 NOS/VE_DEADSTART

- o The following file must be available in your catalog on the S2:

TPXXXK contains a NOS/VE deadstart image. This must be a copy of the dual state deadstart images available from the link procedures.

CIMAGE, PPIMAGE, RGIMAGE are "fast" files, which are built from TPXXXK the first time you deadstart NOS/VE. These files are then used on subsequent deadstart attempts. Before a new TPXXXK can be used, these "fast" files must be purged off your user number.

- o Mount the disk labeled "DAHL-Large sector" on 844 unit 0 (other disks will not work). The default disk driver for build M is the NOS/VE driver.

- o Type the following:

X.DIS.
USER,scat,scatx,where scat = INT1 or INT2
CALL, SYEDNVE.
DROP.

- o Bring up dual state:
X.UPMYVE (CAT=mycat, DEV1=scat)
where mycat = user catalog (as before)
scat = system catalog - INT2 or INT1

- o The UPMYVE job will display the following:

REQUEST *K* DISPLAY on the B display

Type K,n. where n is the control point number of the UPMYVE job.

1.5 NOS/VE_TERMINATION

- o Bringing down dual state:

K.*BYEVE.
K.*ENDRUN

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1 ROUTE AN INPUT FILE FROM C170 TO C180

GET,filename.
ROUTE,filename,DC=LP,FC=RH.

The input file which is sent to the 180 must be in 6/12 ASCII (or display code subset). The job file must be a single partition 170 record containing NOS/VE commands. Multi partition input files are not yet supported by NOS/VE so 170 data files used by the program must be obtained through the NOS/VE permanent file GET command. The last command of the job file must be JMXIT in order to cause the job to terminate properly.

1.7.1.1 Route A Print File From C180 to C170

At NOS/VE job termination the job log (dayfile) will be automatically returned to the 170. The job log will be appended to the NOS/VE output file OUTPUT. NOS/VE print files must be written by BAM as 8/8 ASCII RT=W. Print files will be converted from 8/8 ASCII RT=W to Display Code (64 character set - upper case only) when they are sent to the 170. Support for ASCII print files (8/12 ASCII) will be added at a later build. All NOS/VE output files will appear in the 170 output queue (NOS H,D display) with the name IRHFxxx as a banner. In order to route a NOS/VE print file to the 170, the following command must be contained in the 180 job file or be entered from the system console via the K display:

JMROUTE,jobname,filename,PR,REMOTE

jobname - name that the print file will have in the 180 output queue.

filename - name of the local 180 file created by BAM that is to be printed.

PR - specifies that the file is a print file (must always be PR).

REMOTE - name of the 180 family for the print file (must always be REMOTE). The NOS command language must be in HCS mode (HCS command) in order to enter the JMROUTE command.

Example of JMROUTE command:

JMROUTE,LISTING,LINKMAP,PR,REMOTE.

On the C170 side, the printer must be physically and logically on. To logically turn the printer on, under DSD enter:

1.0 DUAL STATE NDS/VE DEADSTART
1.7.1.1 Route A Print File From C180 to C170

DN32.
FORM32, TM.

1.7.1.2 LINK_USER_Command

In order to access C170 permanent files from a C180 job, the NDS/VE job must issue a LINK_USER command. The LINK_USER command specifies the user identification on the 170 under which C170 permanent files will be accessed.

The following must be done to enter the LINK_USER command:

LIU,US=(user,family), PA=password, A=account, PR=project.

The LIU (or LINK_USER) command specifies the NDS user, family and password parameters which are used by IRHF170 to create 170 jobs which access C170 permanent files. All LINK_USER parameters are specified with keywords. The account and project parameters are not required and not utilized.

US (or USER) - This parameter specifies the NDS user number (catalog) and family in which the 170 permanent files reside. 'user' will be the first parameter on the 170 job's USER card and 'family' will be the third parameter. Currently the only family on the S2 Dual State system is called NVE.

PA (or PASSWORD) This parameter specifies the password that is used to login to the user number. 'password' will be the second parameter on the 170 job's USER card.

A or (ACCOUNT) This parameter specifies the account number (charge number) for the 170 job. It will be the first parameter on the 170 job's CHARGE card.

PR or (PROJECT) This parameter specifies the project number for the 170 job. It will be the second parameter on the 170 job's CHARGE card.

Note: When running on the Simulator, the LINK_USER command is not required to use the GET and REPLACE commands.

Example of LINK_USER command:

LIU,US=(FAB,NVE), PA=FABX, A=7136, PR=73E08802.

1.0 DUAL STATE NOS/VE DEADSTART

1.7.1.3 Get A 170 Permanent File From 180

1.7.1.3 Get_A_170_Permanent_File_From_180

The GET command obtains a copy of a permanent file residing on the 170. The 170 permanent file can be either a direct or an indirect access permanent file. The NOS/VE command processor must be in HCS mode (HCS command issued) in order to use the GET command. All parameters on the GET command are positional. Only the 'lfn' parameter is required. A LINK_USER command must be issued (for the 170 family on which the permanent file resides) prior to issuing the GET command. The format of the GET command is:

GET, lfn, pfn, pw, un, fm, cd.

lfn (local file name) - This is the name of the local NOS/VE file to which the 170 permanent file will be transferred.

pfn (permanent file name) - This is the name of the 170 permanent file that is to be accessed. If this parameter is omitted then 'lfn' will be used for the 170 permanent file name.

pw (password) - This is the password that will be used to access the 170 permanent file if a password is required to access the file on the 170.

un (user name) - This is the user name (alternate catalog) on which the 170 permanent file resides.

fm (family) - This is the family on which the 170 permanent file resides. Currently the only 170 family on the S2 Dual State system is NVE.

ca (conversion alternatives) - This parameter specifies the type of conversion that is performed by the IRHF on files transferred from 170 to 180. If this parameter is omitted then a default of B60 will be assumed. Values for this parameter are:

B60: Basic Biary

The full 60 bits of each 170 word are transferred to the lower 60 bits of each 64 bit 180 word. The upper 4 bits of each 64 bit 180 word are set to 0. The file is written to 180 using BAM with Block Type = System (Unblocked) and Record Type = Undefined (RT=U) so no control information is inserted in the file. The 170 logical record structure is dropped (i.e., EDRs are deleted causing the logical records to be packed

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.3 Get A 170 Permanent File From 180

together.

B56: C180 Binary

The lower 56 bits (7 8 bit bytes) of each 170 word are packed into contiguous 8 bit bytes on the 180 (i.e., 7 8 bit bytes from the first 170 word and 1 8 bit byte from the second 170 word go into the first 180 word etc.). The 170 logical record structure (EDRs) are dropped. The way that the 180 file which was transferred from 170 is accessed should correspond to the method used to create it on the 180 originally (assuming that the file originated on the 180).

A6: 6/12 ASCII

A170 6/12 ASCII character files (used by XEDIT and most SES utilities) are converted to 180 8/8 ASCII with Block Type = System (Unblocked) and Record Type = Variable (RT=W). The 170 logical record structure EDRs are dropped.

A8: 8/12 ASCII

170 8/12 ASCII character files are converted to 180 8/8 ASCII with Block Type = System (Unblocked) and Record Type = Variable (RT=W). The 170 logical record structure (EDRs) are dropped.

D64: Display Code 64 Character Set

170 Display Code character files are converted to upper case 180 8/8 ASCII with Block Type = System (Unblocked) and Record Type = Variable (RT=W). The 170 logical record structure EDRs are dropped.

Example of GET command:

```
LIU,US=(FAB,NVE),PA=FABX,A=7136,PR=73E08802.  
GET,TEXT,TEXT612,,,NVE,A6.
```

Note: When the GET command is used on the Simulator, the file specified by the 'pfn' parameter must be a 170 file which is local to the simulator job.

1.7.1.4 REPLACE_A_170_Permanent_File_From_180

The REPLACE command transfers a copy of a 180 local file to a permanent file on the 170. If a permanent file of the same name

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.4 REPLACE A 170 Permanent File From 180

does not exist for the specified user (catalog), a direct access permanent file is created. If a direct access permanent file of the same name already exists in the catalog and the file can be attached with write mode then the existing direct access file is overwritten with the file from the 180. If an indirect access permanent file of the same name already exists in the catalog then the indirect access file is replaced by the file from the 180. An existing indirect access file will not be changed to a direct access file if the user's indirect access file size limit is exceeded. The NOS/VE command processor must be in HCS mode (HCS command issued) in order to use the REPLACE command. All parameters on the REPLACE command are positional. Only the 'lfn' parameter is required. A LINK_USER command must be issued (for the 170 family on which the permanent file resides) prior to issuing the REPLACE command. The format of the REPLACE command is:

REPLACE, lfn, pfn, pw, un, fm, ca.

lfn (local file name) - This is the name of the local NOS/VE file which will be transferred to a permanent file on the 170.

pfn (permanent file name) - This is the name of the 170 permanent file that is to be created or replaced. If this parameter is omitted then 'lfn' will be used for the 170 permanent file name.

pw (password) - This is the password that will be associated with a newly created direct access file or which is used to gain access to an already existing direct or indirect access permanent file.

un (user name) - This is the user name (catalog) on which an existing 170 direct or indirect access file resides. This parameter is illegal if the file does not exist.

fm (family) - This is the family on which the 170 permanent file is to reside. Currently the only 170 family on the S2 Dual State system is NVE.

ca (conversion alternatives) - This parameter specifies the type of conversion that is performed by the IRHF on files transferred from 180 to 170. If this parameter is omitted then a default of B60 will be assumed. Values for this parameter are:

B60: Basic Binary

The lower 60 bits of each 64 bit 180 word are

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.4 REPLACE A 170 Permanent File From 180

transferred to the full 60 bits of each 170 word. The upper 4 bits of each 64 bit 180 word are discarded. The 180 file which is to be transferred should be written by BAM with Block Type = System (Unblocked) and Record Type = Undefined (RT=U). The file is transferred to the 170 as a single logical record (i.e. files with multiple EDRs cannot be created on the 170 from the 180).

B56: C180 Binary

Groups of 7 contiguous 8 bit bytes from the 180 will be transferred to the lower 56 bits of each 170 word (i.e. the first 7 8 bit bytes from the first 180 word go to the lower 56 bits of the first 170 word, the 8th 8 bit byte of the first 180 word and the first 6 8 bit bytes from the second 180 word go to the lower 56 bits of the second 170 word etc.). The way that the 180 file to be transferred is created does not matter because the entire structure of the 180 file is preserved on the 170. The file is transferred to the 170 as a single logical record.

A6: 6/12 ASCII

A 180 8/8 ASCII character file with Block Type = System (Unblocked) and Record Type = Variable (RT=W) is converted to a 170 6/12 ASCII file (used by XEDIT and most SES utilities). The file is transferred to the 170 as a single logical record.

A8: 8/12 ASCII

A 180 8/8 ASCII character file with Block Type = System (Unblocked) and Record Type = Variable (RT=W) is converted to a 170 8/12 ASCII file. The 170 file can be routed directly to the printer with the 170 ROUTE command with the EC=A9 parameter. The file is transferred to the 170 as a single logical record.

D64: Display Code 64 Character Set

A 180 8/8 ASCII character file with Block Type = System (Unblocked) and Record Type = Variable (RT=W) is converted to a 170 Display Code file with lower case characters mapped to upper case. ASCII special characters that do not have a Display Code equivalent are converted to Display Code blanks. The file is transferred to the 170 as a single logical record.

Example of REPLACE command:

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.4 REPLACE A 170 Permanent File From 180

LIU,US=(FAB,NVE),PA=FABX,A=7136,PR=73E08802.
REPLACE,MYFILE,FILEB56,,,NVE,B56.

Note: When the REPLACE command is used on the Simulator, the file specified by the 'pfn' parameter will become a 170 file which is local to the simulator job.

1.7.1.5 Example_Jobs_Representing_Phase_A/B_Library_Creation/Modif

General Notes

- o The HCS and SCL commands have been deleted.
- o Comment capability within command line is not currently available
- o HCS comment line format (starts with single quote) not SCL compatible
- o The B60 conversion parameter on the GET command is utilized to indicate that the A170 file is to be transferred to NOS/VE without conversion
- o The NOS/VE job command stream must use NOS 6/12 ASCII (or display code subset)
- o The B56 conversion parameter on the REPLACE command is utilized to indicate that the 64 bit oriented NOS/VE file is to be transferred to the A170 without truncation of data. 56 C180 data bits are stored in each 60 bit A170 word.
- o The B56 conversion parameter on the GET command is utilized to indicate that an A170 file with 56 C180 data bits per 60 bit A170 word are to be transferred to NOS/VE as a 64 bit oriented NOS/VE file.
- o REPLACE can now be used to replace an existing A170 file
- o GET and REPLACE now allow specification of alternate user names.
- o Parameters to programs must be delimited by quote marks.
- o Files routed from 170 must contain valid 6/12 ASCII data. No previous kludges (% for _) are supported.

1.7.1.5.1 CREATE OBJECT LIBRARY ON NOS/VE AND SAVE IT ON A170 NOS

Notes

- o CLG0170 is A170 permanent file name for file containing object text (in CI data mapping) for modules to be included in the library.
- o CITEXT180 is NOS/VE local file name for file containing object text (in CI data mapping) for modules to be included in the library.
- o IITEXT180 is NOS/VE local file name for file containing object text (in II data mapping) for modules to be included in the library.

1.0 DUAL STATE-NOS/VE DEADSTART
1.7.1.5.1 CREATE OBJECT LIBRARY ON NOS/VE AND SAVE IT ON A170 NOS

- o LIBRARY180 is NOS/VE local file name for the library being created.
- o ILIB170 is A170 permanent file name for file containing the library.

NOS/VE Job Commands

LIU, USER=(jln,NVE),PASSWORD=jlmx,ACCOUNT=notused,PROJECT=notused

```
GET,citext180,clg0170,,,NVE,B50
EXECUTE,, 'citext180,iitext180',,,,CITDII
EXECUTE,,,,,COL
ADD,OBJECT_FILE=iitext180
GENERATE,LIBRARY=library180
QUIT
REPLACE,library180,ilib170,,,NVE,B56
```

1.7.1.5.2 MODIFY A PREVIOUSLY SAVED OBJECT LIBRARY

Notes

- o ILIB170 is A170 permanent file name for file containing the old library
- o LIBRARY180 is NOS/VE local file name for file containing the old library
- o CMOD170 is A170 permanent file name for file containing CI object text for the new module
- o NEWCIMODULE is NOS/VE local file name for file containing CI object text for the new module
- o NEWIIMODULE is NOS/VE local file name for file containing II object text for the new module
- o NEWLIBRARY is NOS/VE local file name for the library being created

NLIB170 is A170 local file name for new library

NOS/VE Job Commands

LIU,USR=(jln,NVE),PASSWORD=jlmx,ACCOUNT=notused,PROJECT=notused

```
GET,library180,ilib170,,,NVE,B56
GET,newcimodule,cmod170,,,NVE,B60
EXECUTE,, 'newcimodule,newiimodule',,,,CITDII
EXECUTE,,,,,COL
ADD,library=library180
REPLACE,OBJECT_FILE=newiimodule
GENERATE,LIBRARY=newlibrary
QUIT
REPLACE,newlibrary,nlib170,,,NVE,B56
QUIT
REPLACE,newlibrary,nlib170,,,NVE,B56
```

1.0 DUAL STATE NDS/VE DEADSTART
1.7.1.5.3 USAGE OF NDS/VE LOADER

1.7.1.5.3 USAGE OF NDS/VE LOADER

LIMITATIONS

- The file(s) specified by the FILE parameter on the EXECUTE command may not be object library files.
- The program option PRESET has no effect since the file system does not yet support presetting.

PROCESS

Create an object text file by compiling a program in real state. Then perform the following steps in virtual state:

- Use the SET_OBJECT_LIST command to specify necessary libraries.
- Acquire any necessary libraries (which are not quoted in text embedded directives) by either:
 - o Creating the library file via the object library generator
 - or
 - o Staging the library file from real state to virtual state using the GET command (with B56 conversion mode specified).
- Stage the object text file from real state to virtual state using the GET command (with B60 conversion mode specified).
- Convert the object text file from the CI data mapping to II data mapping by executing the CITOII utility.
- Load and execute the program via the EXECUTE command.
- Stage the loadmap from virtual state to real state (for printing) by using either:
 - o The REPLACE command (with A6 conversion mode specified) if running on the simulator.
 - or
 - o The JMRUTE command if running on the hardware.

EXAMPLES

The following is an example command sequence for executing a program not requiring any libraries for loading:

Assumptions: all modules to be loaded are contained on the (real state) permanent file 'citxtrs'.

1.0 DUAL STATE-NDS/VE DEADSTART
1.7.1.5.3 USAGE OF NDS/VE LOADER

```
LINK_USER,USER=(jln,NVE),PASSWORD=jlnx,..
ACCOUNT=notused,PROJECT=notused
GET,citxtvs,citxtrs,,,NVE,B60
EXECUTE,, 'citxtvs,iitext',,,,CITDII
EXECUTE,iitext,'program parameters'
JMROUTE,notused,LOADMAP,PR,REMOTE
```

A NDS/VE library containing the CYBIL runtime procedures can be obtained via the following:

```
LINK_USER,USER=(jln,NVE),PASSWORD=jlnx,ACCOUNT=notused,..
PROJECT=notused
GET,CYBILIB,CYBILB,,,DEV1,NVE,B56
```

The following is an example command sequence for executing a program requiring libraries for loading:

Assumptions: the (real state) permanent file 'citxtrs' contains object text generated by the CYBIL CI compiler. The compiler modules reference procedures contained on the user library 'mylib' and the CYBIL run-time library. These libraries have been generated in virtual state and saved in real state.

```
LINK_USER,USER=(jln,NVE),PASSWORD=jlnx,..
ACCOUNT=notused,PROJECT=notused
GET,CYBILIB,CYBILIB,,,DEV1,NVE,B56
GET,mylib,mylib,,,NVE,B56
SET_OBJECT_LIST,ADD=mylib
SET_PROGRAM_OPTIONS,MAP_OPTIONS=(B,E,X,S)
GET,citxtvs,citxtrs,,,NVE,B60
EXECUTE,, 'citxtvs,iitext',,,,CITDII
EXECUTE,iitext,'program parameters'
JMROUTE,notused,LOADMAP,PR,REMOTE
```

1.7.1.5.4 DIFFERENCES BETWEEN PHASE C (BUILD M) AND PHASE A/B (BUI

Notes:

- 1) The 'status' column indicates whether the command/procedure is unchanged from Phase A/B, modified from Phase A/B or not available in Phase A/B (new in Phase C).
- 2) Footnotes are numbered within each section.
- 3) The following summarizes the condition code ranges currently assigned to NVE.

Common Modules	9,000 - 9,999
Common Code Generator	8,000 - 8,999

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4 DIFFERENCES BETWEEN PHASE C (BUILD M) AND PHASE A/B (BUI

Condition Code	Product Identifier	Product Name
1 - 159,999	Reserved	
160,000 - 169,999	AM	Basic Access Methods
170,000 - 179,999	CL	Command Language
180,000 - 189,999	JM	Job Management
190,000 - 199,999	LL	Loader
200,000 - 209,999	MM	Memory Management
210,000 - 219,999	DB	Operating System
220,000 - 229,999	PF	Permanent File Management
230,000 - 239,999	PM	Program Management
240,000 - 249,999	RM	Resource Management
250,000 - 259,999	OF	Operator Facility
260,000 - 269,999	UA	User Administrator
270,000 - 279,999	IC	Interstate Communication
280,000 - 289,999	RH	Remote Host Facility
290,000 - 299,999	OC	Object Code Utilities
300,000 - 309,999	DB	Deadstart/Recovery
310,000 - 319,999	MS	Maintenance Services
320,000 - 329,999	Reserved	
500,000 - 509,999	AA	Advanced Access Method
510,000 - 519,999	AG	ALGOL
520,000 - 529,999	AL	Assembly Language
530,000 - 539,999	AP	APL
540,000 - 549,999	BA	BASIC
550,000 - 559,999	CA	Conversion Aids System
560,000 - 569,999	CB	COBOL
570,000 - 579,999	CY	CYBIL
580,000 - 589,999	FT	FORTRAN
590,000 - 599,999	PA	PASCAL (Wirth)
600,000 - 609,999	PI	PL/1
610,000 - 619,999	SM	Sort Merge
620,000 - 629,999	SY	SYMPL

4. Support of 6-12 ASCII from the console (K display) causes the following changes:

INPUT_	TRANSLATED_IO_	INPUT_	TRANSLATED_IO_
/1	^	/([
/2	"	/)]
/3	#	/+	>
/4	\$	/-	<
/5	(reversed /)	/=	.
/6	;	/*	' (single quote)
/7	?	//	/
/8	{	/,	:
/9	}	/A to /Z	a - z (lower case)
/0	_ (underscore)		

1.0 DUAL STATE NOS/VE DEADSTART

1.7.1.5.4 DIFFERENCES BETWEEN PHASE C (BUILD M) AND PHASE A/B (BUI

(The major incompatibilities with earlier systems are for characters for ; and '. To get a semicolon, type /6, to get a ' (single quote), type /*

example

```
ex utli /* bulk ?/* /6 tstatus
is
ex utli 'bulk' ; tstatus
```

1.7.1.5.4.1 Phase_C_-_Command_Functions

Function_	Status
\$MOD	new
\$CHAR	new
\$CLOCK	new
\$DATE	new
\$FNAME	new
\$INTEGER	new
\$NAME	new
\$ORD	new
\$REAL	new
\$STRING	new
\$STRLEN	new
\$STRREP	new
\$SUBSTR	new
\$TIME	new
\$VAR	new
\$SPECIFIED	new
\$SET_COUNT	new
\$VALUE_COUNT	new
\$RANGE	new
\$DEBUG_MODE_ON	new
\$PARAMETER_LIST	new
\$PARAMETER	new

1.7.1.5.4.2 Phase_C_-_System_Access_Commands

Commands_	Status
Link_User Ph1	unchanged

1.7.1.5.4.3 Phase_C_-_Resource_Management

Command_	Status
REQUEST_TERMINAL	NOS- *1

*1 The REQUEST_TERMINAL command does not properly assign a file

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4.3 Phase C - Resource Management

to a terminal. A RMP\$REQUEST_TERMINAL call issued from the same task that accesses the terminal via BAM must be used to assign a file to a terminal device.

Files assigned to a terminal device can be accessed via the following BAM requests:

- o AMP\$OPEN
- o AMP\$GET_NEXT
- o AMP\$GET_DIRECT
- o AMP\$GET_PARTIAL
- o AMP\$PUT_NEXT
- o AMP\$PUT_DIRECT
- o AMP\$PUT_PARTIAL
- o AMP\$FLUSH
- o AMP\$CLOSE

All terminal attributes can be specified on the RMP\$REQUEST_TERMINAL call but only the following are operational:

- o auto_input
- o format_effectors
- o transparent_mode
- o prout_file
- o prompt_string

1.7.1.5.4.4 Phase C -- File Management

Command_	Status
FILE	new
COPY	new
DUMP_FILE	new
COMPARE	new

1.7.1.5.4.5 Phase C -- Permanent File Management

Command_	Status
HCS GET	unchanged
HCS REPLACE	unchanged
DEFINE	unchanged
ATTACH	unchanged
PURGE	unchanged
CHANGE	unchanged
PERMIT	unchanged
DELETE_PERMIT	unchanged
DEFINE_CATALOG	unchanged
PURGE_CATALOG	unchanged

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4.5 Phase C - Permanent File Management

DELETE_CATALOG_PERMIT new
PERMIT_CATALOG new

1.7.1.5.4.6 Phase C - SCL Statements and Procedures

Command	Status
PROC/PROCEND	new - *5
SET_COMMAND_LIST	new
DISPLAY_COMMAND_LIST	new
REPEAT/UNTIL	new
WHILE/WHILEND	new
DECLARE_VARIABLE	unchanged
REMOVE_VARIABLE	unchanged
BLOCK/BLOCKEND	new
LOOP/LOOPEND	new
FOR/FOREND	new
IF/ELSEIF/ELSE/IFEND	new
CYCLE	new
EXIT	new
INCLUDE	*5
COLLECT_TEXT	*4
DISPLAY_VALUE	unchanged
EXIT_PROC	new
ACCEPT	unchanged
DO	new
HCS	deleted - *1
SCL	deleted - *1
change HCS variable	modified - *2
display HCS variable	modified - *3

*1 The SCL interpreter will now be in control to process commands when a job is initiated. HCS commands will be automatically passed to the HCS command interpreter.

When SCL "comes up" in a job it displays a banner containing the OS name, date and time. The value of these commands are controlled by integration.

*2 The new form of this commands is as follows:

LNS variable name = new_value

*3 The new form of this command is as follows:

LNS variable_name

*4 For some time now the default value for the collect_text command's UNTIL parameter has been () rather than ** as described in the ERS. This was because it was not possible

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4.6 Phase C - SCL Statements and Procedures

to enter from the console (on the S2) a line to be interpreted on the 180 side that began with an asterisk.

Since this problem was alleviated in build L the collect_text command now defaults the UNTIL parameter to **.

- *5 It is now possible to initiate the asynchronous execution of tasks within an SCL PROC or file of INCLUDED commands and have processing of the PROC or file of commands return control before the asynchronous tasks have finished.

1.7.1.5.4.7 Phase C - Object Code Maintenance

Command	Status
CREATE_OBJECT_LIBRARY	unchanged
DISPLAY_LIBRARY	unchanged
SELECT_DISPLAY_LEVEL	unchanged
ADD	unchanged
REPLACE	unchanged
COMBINE	unchanged
CREATE_MODULE	new
BIND_MODULE	new
DELETE	unchanged
CHANGE	unchanged
SATISFY	new
REORDER	unchanged
GENERATE	unchanged
QUIT	unchanged
CI to II Conversion	unchanged
DEFINE_PROGRAM	new

1.7.1.5.4.8 Phase C - User Services

Command	Status
DISPLAY_LOG	new - *1
HCS ZDIS	deleted
HCS FMDAYFIL	deleted

- *1 NOS/VE now creates global logs and allows ASCII logs to be displayed.

The system log can be displayed at the console via the following command:

EXECUTE,,,,,SDZS,TASK_NAME=task1

A job log can be displayed at the console via the following command:

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4.8 Phase C -- User Services

EXECUTE,,,,,ZDIS,TASK_NAME=task2

The following differences exist between the build M DISPLAY_LOG and the description in the 9/26/80 NOS/VE command interface ERS:

- o upon = <local_file_name> instead of output = <file_reference>
- o type = <system!sys!job> instead of type = <system!account!engineering!statistics!job!job_statistic>

1.7.1.5.4.9 Phase C -- File Routing

Command	Status
HCS JMROUTE	*1

*1 JMROUTE of the LOADMAP is no longer required. The LOADMAP will appear on the standard output file by default.

1.7.1.5.4.10 Phase C -- Program Execution

Command	Status
SET_OBJECT_LIST	unchanged
SET_PROGRAM_OPTIONS	*1
DISPLAY_PROGRAM	new
EXECUTE	new - *2
"name call"	new
HCS EXECUTE	*3
Library Loader	unchanged
PUSH_DEBUG_MODE	unchanged
POP_DEBUG_MODE	unchanged
SET_DEBUG_MODE	unchanged
SET_DEBUG_RING	new
TSTATUS	deleted
TMTerm	deleted

*1 All parameters on SET_PROGRAM_OPTIONS are now supported

*2 Programs previously loaded by the NOS/VE loader via the HCS EXECUTE must now utilize the SCL EXECUTE. The parameter passing mechanism is different. The template for a NOS/VE program declaration is the following:

PROGRAM xxx (parameter:pmt\$program_parameters; VAR status: ost\$status)

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4.10 Phase C - Program Execution

where pmt\$program_parameters is defied in deck PMDPARM to be SEQ(*).

In general the parent and child tasks must have some convention regarding the contents of the sequence. If the parent task is SCL (as will be the case when the SCL command EXECUTE is processed) then the contents of the sequence is by convention of type:

clt\$parameter_list_contents

- *3 The DEBUG option on the HCS EXECUTE command is no longer supported. DEBUG mode must be activated via the SCL command SET_DEBUG_MODE.

The NOS/VE loader can no longer be activated by the HCS command EXECUTE. The SCL command EXECUTE must be used to load programs residing on object files (via the NOS/VE loader).

The HCS command EX (also known as EXEC) is no longer supported. HCS "programs" previously loaded into the HCS program library must now be implemented as [XDCL,#GATE] procedures which are built on the library XLJBBB.

Existing HCS programs have been converted and run from the XLJBBB library. The following list shows the NOSVE commands corresponding to the previous HCS commands.

old form	new form
EX RHINPUT,,A	EXECUTE,,,,RHP\$INPUT,TASK_NAME=A
EX RHOUTQ8,,A	EXECUTE,,,,RHP\$OUTPUT,TASK_NAME=B
EX CITOII 'param'	EXECUTE,, 'param' ,,,CITOII
EX COL	EXECUTE,,,,COL
EX OBJLIST 'param'	EXECUTE,, 'param' ,,,OBJLIST
EX IFEXEC,,A	EXECUTE,,,,IFEXEC,TASK_NAME=C

1.7.1.5.4.11 Phase C -- Job Management

Command	Status
SUBMIT	new
DISPLAY_JOB_STATUS	new
DROP_JOB	new
PRINT	new
DROP_FILE	new
DISPLAY_PRINT_STATUS	new
HCS JMEXIT	*1
HCS JMDROP	*2

1.0 DUAL STATE NOS/VE- DEADSTART
1.7.1.5.4.11 Phase C - Job Management

- #1 JMEXIT is no longer required to terminate a batch job. It is required to switch an interactive user back to A170 NOS. (The SCL command LOGOUT can also be used.)
- #2 JMDROP of an executing job will no longer work. The SCL DROP_JOB command must be used.

1.7.1.5.4.12 Phase C -- Command Processing

Procedure_	Status
CLP\$SCAN_PARAM_LIST	unchanged
CLP\$TEST_PARAMETER	unchanged
CLP\$GET_KEYWORD	unchanged
CLP\$GET_SET_COUNT	unchanged
CLP\$GET_VALUE_COUNT	unchanged
CLP\$TEST_RANGE	unchanged
CLP\$GET_VALUE	unchanged
CLP\$DECLARE_VARIABLE	unchanged
CLP\$REMOVE_VARIABLE	unchanged
CLP\$READ_VARIABLE	unchanged
CLP\$WRITE_VARIABLE	unchanged
CLP\$SCAN_COMMAND_FILE	unchanged
CLP\$END_SCAN_COMMAND_FILE	unchanged
CLP\$SCAN_COMMAND_LINE	unchanged
CLP\$PUSH/POP COMMAND LIST	new

1.7.1.5.4.13 Phase C -- Message Generator

Procedure_	Status
DSP\$FORMAT_MESSAGE	unchanged
DSP\$SET_STATUS_ABNORMAL	unchanged
DSP\$APPEND_STATUS_PARAMETER	unchanged

1.7.1.5.4.14 Phase C -- Device Management

Procedure_	Status
RMP\$REQUEST_MASS_STORAGE	new
RMP\$REQUEST_TERMINAL	?

1.7.1.5.4.15 Phase C -- Queued File Management

Procedure_	Status
JMP\$ROUTE	unchanged
JMP\$QSTATUS_TERMINAL_FULL	new
JMP\$QSTATUS_FAMILY_FULL	unchanged

1.0 DUAL STATE-NOS/VE-DEADSTART
1.7.1.5.4.15 Phase C - Queued File Management

JMP\$QSTATUS_FILE_FULL	new
JMP\$QCOUNT_TERMINAL	new
JMP\$QCOUNT_FAMILY	unchanged
JMP\$DROP_QFILE	unchanged
JMP\$DIVERT_TERMINAL	new
JMP\$DIVERT_FAMILY	unchanged
JMP\$DIVERT_QFILE	new
JMP\$ALTER_QFILE	unchanged

1.7.1.5.4.16 Phase C -- Program Execution

<u>Procedure</u>	<u>Status</u>
PMP\$EXIT	new
PMP\$EXECUTE	new
PMP\$TERMINATE	new
PMP\$AWAIT_TASK_TERMINATION	new
PMP\$MODULE_TABLE_ADDRESS	unchanged
PMP\$ENTRY_POINT_TABLE_ADDRESS	unchanged
PMP\$PUSH_JOB_DEBUG_MODE	unchanged
PMP\$POP_JOB_DEBUG_MODE	unchanged
PMP\$SET_JOB_DEBUG_MODE	unchanged
PMP\$JOB_DEBUG_MODE_ON	unchanged
PMP\$PUSH_TASK_DEBUG_MODE	unchanged
PMP\$SET_TASK_DEBUG_MODE	unchanged
PMP\$TASK_DEBUG_MODE_ON	unchanged
PMP\$SET_DEBUG_RING	new
PMP\$DEBUG_RING	new
PMP\$CHANGE_DEBUG_LIBRARY_LIST	new
PMP\$POP_TASK_DEBUG_MODE	unchanged

1.7.1.5.4.17 Phase C -- Program Communication

<u>Procedure</u>	<u>Status</u>
OSP\$AWAIT_ACTIVITY_COMPLETION	new
PMP\$DEFINE_QUEUE	unchanged
PMP\$REMOVE_QUEUE	unchanged
PMP\$CONNECT_QUEUE	unchanged
PMP\$DISCONNECT_QUEUE	unchanged
PMP\$SEND_TO_QUEUE	unchanged
PMP\$RECEIVE_FROM_QUEUE	unchanged
PMP\$STATUS_QUEUE	unchanged
PMP\$STATUS_QUEUES_DEFINED	unchanged
PMP\$GET_QUEUE_LIMITS	unchanged

1.7.1.5.4.18 Phase C -- Condition Processing

<u>Procedure</u>	<u>Status</u>
------------------	---------------

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4.18 Phase C - Condition Processing

PMP\$ESTABLISH_CONDITION_HANDLER	unchanged
PMP\$DISESTABLISH_COND_HANDLER	unchanged
PMP\$CAUSE_CONDITION	unchanged
PMP\$CONTINUE_TO_CAUSE	unchanged
PMP\$TEST_CONDITION_HANDLER	unchanged
PMP\$VALIDATE_PREVIOUS_SAVE_AREA	new
PMP\$SET_EXECUTION_INTERVAL	new
PMP\$ESTABLISH_DEBUG_CFF	new

1.7.1.5.4.19 Phase C -- Program Services

<u>Procedure</u>	<u>Status</u>
PMP\$GET_TIME	unchanged
PMP\$GET_MICROSECOND_CLOCK	unchanged
PMP\$GET_TASK_CP_TIME	unchanged
PMP\$GET_DATE	unchanged
PMP\$GET_USER_IDENTIFICATON	unchanged
PMP\$GET_ACCOUNT_PROJECT	new
PMP\$GET_JOB_NAME	new
PMP\$GET_JOB_ID	new
PMP\$GET_JOB_MODE	new
PMP\$GET_PROGRAM	new
PMP\$GET_TASK_ID	new
PMP\$MANAGE_SENSE_SWITCHES	new
PMP\$GET_DS_VERSION	new
PMP\$GET_PROCESSOR_ATTRIBUTES	unchanged
PMP\$DEFINE_DEBUG_ENTRY	new
PMP\$GET_DEBUG_ENTRY	new
PMP\$MODIFY_DEBUG_ENTRY	new
PMP\$REMOVE_DEBUG_ENTRY	new

1.7.1.5.4.20 Phase C -- Logging

<u>Procedure</u>	<u>Status</u>
PMP\$LOG	*1
PMP\$LOG_ASCII	new
PMP\$DISABLE_ATTACHES_TO_LOG	new
PMP\$ENABLE_ATTACHES_TO_LOG	new
PMP\$GET_LOG_CONTROL	new
PMP\$TERMINATE_LOG	new

*1 logs are now global

1.7.1.5.4.21 Phase C -- File Management

<u>Procedure</u>	<u>Status</u>
Sequential Access	*1

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4.21 Phase C - File Management

Byte_Addressable Access	unchanged
Record Access	*1
Segment Access	?
W_System Specified	unchanged
W_User Specified	new
U_System Specified	unchanged
U_User Specified	new
F_System Specified	unchanged
F_User Specified	new
AMP\$DESCRIBE_NEW_FILE	unchanged
AMP\$FILE	unchanged
AMP\$GET_FILE_ATTRIBUTES	unchanged
AMP\$FETCH	unchanged
AMP\$STORE	unchanged
AMP\$COPY_FILE	new
AMP\$RENAME	new
AMP\$RETURN_LOCAL_FILE	new
AMP\$OPEN	unchanged
AMP\$CLOSE	unchanged
AMP\$FETCH_ACCESS_INFORMATION	unchanged
AMP\$SKIP	*4
AMP\$REWIND	*2
AMP\$WRITE_END_PARTITION	new
AMP\$GET_NEXT	*1, *5
AMP\$GET_DIRECT	unchanged (?)
AMP\$GET_PARTIAL	*1, *6
AMP\$GET_PARTIAL_DIRECT	unchanged (?)
AMP\$PUT_NEXT	*1
AMP\$PUT_DIRECT	unchanged (?)
AMP\$PUT_PARTIAL	*1, *3
AMP\$PUT_PARTIAL_DIRECT	unchanged (?)
AMP\$SEEK_DIRECT	unchanged (?)
AMP\$GET_SEGMENT_POINTER	unchanged (?)
AMP\$SET_SEGMENT_EOI	unchanged (?)
AMP\$SET_SEGMENT_POSITION	unchanged (?)
AMP\$SET_LOCAL_NAME_ABNORMAL	new
AMP\$SET_FILE_INSTANCE_ABNORMAL	new
AMP\$ACCESS_METHOD	new
AMP\$FETCH_FAP_POINTER	new
AMP\$STORE_FAP_POINTER	unchanged (?)

*1 user specified blocking is now supported

*2 AMP\$REWIND
The WAIT parameter on the procedure call is not supported.

*3 AMP\$PUT_PARTIAL
PUT_PARTIAL with the TERM_OPTION = AMPC\$START does not start a new record.

1.0 DUAL STATE NDS/VE DEADSTART
1.7.1.5.4.21 Phase C - File Management

- *4 AMP\$SKIP
If the number of units to skip = 0 and the file is positioned mid-unit the file will remain positioned mid-unit.
- *5 AMP\$GET_NEXT
A GET_NEXT with a working_storage length = 0 will return an undefined file position. A request of zero length is invalid and will result in abnormal termination in later builds.
- *6 AMP\$GET_PARTIAL
A GET_PARTIAL of record type = undefined never returns file position of end-of-record.

1.7.1.5.4.22 Phase C - Permanent File Management

<u>Procedure</u>	<u>Status</u>
PPF\$DEFINE	unchanged - *1
PPF\$ATTACH	unchanged - *1
PPF\$PURGE	unchanged - *1
PPF\$CHANGE	unchanged - *1
PPF\$PERMIT	unchanged - *1
PPF\$DELETE_PERMIT	unchanged - *1
PPF\$DEFINE_CATALOG	unchanged - *1
PPF\$PURGE_CATALOG	unchanged - *1
PPF\$PERMIT_CATALOG	unchanged - *1
PPF\$DELETE_CATALOG_PERMIT	unchanged - *1

- *1 - Build M Permanent File Program Interface Deficiencies
 1. GET,SAVE and REPLACE are not implemented (non-R2 features).
 2. Since no validation facility exist at Build M, there is no official way for master catalogs to be created. The system job does, however, create a master catalog for itself. Therefore the system job and other jobs with the same family name and user name (family1, user1) can perform 180 permanent file operations.
 3. Permanent files on the "180 side" are only permanent until a NDS/VE deadstart.
 4. Usage_selections and share_selections may be specified on the ATTACH request but they will not be honored. Files will never appear busy and hence exclusive access is not possible.
 5. ATTACH returns an existing local file rather than producing an error status.

1.0 DUAL STATE NOS/VE DEADSTART
1.7.1.5.4.22 Phase C - Permanent File Management

6. DEFINE makes an existing local file permanent rather than producing an error status.
7. It is possible to create both a catalog and a file with the same name at a particular level of the catalog tree.
8. The WAIT parameter for ATTACH is always treated as if PFC\$NO_WAIT is specified.
9. Blank names in the GROUP parameter produce an error status.
10. Passwords are not validated for ATTACH, PURGE or CHANGE.
11. Permanent files are never purged because a retention period has expired.

1.7.1.5.4.23 Phase C -- Memory Management

MMP\$ADVISE_IN	unchanged
MMP\$ADVISE_OUT	unchanged
MMP\$ADVISE_OUT_IN	unchanged
MMP\$WRITE_MODIFIED_PAGES	unchanged
MMP\$CREATE_SEGMENT	unchanged
MMP\$DELETE_SEGMENT	unchanged
MMP\$OPEN_SEGMENT	unchanged
MMP\$CLOSE_SEGMENT	unchanged
MMP\$SYSTEM_CREATE_SEGMENT	unchanged
MMP\$SYSTEM_DELETE_SEGMENT	unchanged
MMP\$STORE_SEGMENT_ATTRIBUTES	unchanged
MMP\$SYSTEM_DELETE_ALL	unchanged
MMP\$FETCH_SEGMENT_ATTRIBUTES	unchanged
MMP\$VERIFY_ACCESS	unchanged
MMP\$FREE	unchanged
MMP\$STORE_SEGMENT_ATTRIBUTES	unchanged
MMP\$LOCK_PAGES	*1
MMP\$UNLOCK_PAGES	*1
MMP\$FETCH_PVA_UNWRITTEN_PAGES	*1

*1 stubs have been replaced with code to implement these procedures

1.7.1.5.5 BAM BUILD J DEVIATION FROM ERS

o Data_structures

Partition delimiters are neither generated nor recognized

o Data_procedures

amp\$describe_new_file

1.0 DUAL STATE NDS/VE DEADSTART
1.7.1.5.5 BAM BUILD J DEVIATION FROM ERS

As opposed to the ERS, this request is processed each time a file is opened regardless of whether the file is new or old.

amp\$file

This request can only be issued once per file with predictable results. If a second amp\$file is issued it will overwrite any attributes supplied via amp\$describe_new_file.

amp\$fetch_access_info

Record_length attribute is not returned correctly.

Transfer_count attribute not returned correctly.

Fields error_count, error_status and residual_skip_count are not defined. A default for these attributes will be returned.

amp\$get_segment_pointer

Pointer_kind field of pointer not always returned correctly.

amp\$get_next

Will only transfer the first part of a continuation record created via amp\$put_partial.

amp\$get_partial

If record length is greater than the working storage length, the working storage area will be overwritten to accommodate the record.

Partial_get of F type records is not available.

amp\$rewind

The WAIT parameter on the procedure call is not supported.

FAPs

Cannot pass control to a FAP whose execution bracket is less than 4.

1.8 QSDI_INFORMATION

To create an Express Deadstart Dump (EDD) tape:

1.0 DUAL STATE NOS/VE DEADSTART
1.8 DSDI INFORMATION

- 1) Mount scratch tape (ring in) on a 9-track drive.
- 2) Push D/S button.
- 3) Select U (utilities) display.
- 4) Select E (EDD) display.
- 5) Set channel (S2=13).
- 6) Set ECUU (S2=01uu)

E = equipment

C = 1 for 67X drives
 2 for 66X drives

uu = unit number of the tape drive to be used.
- 7) Answer "non zero inhibits rewind" with a CR.
- 8) Answer "dump number" with a CR.
- 9) Answer "CM/(MB)" with the size of memory you want to dump
(in megabytes) -6 for build M5.
Note: This display may not appear, depending on the
version of EDD being used.
- 10) Answer "dump controlware" with a CR.

To create a listing of the EDD tape:

- 1) REQUEST,DUMP,NT,D=PE,F=S,LB=KU,PO=R,VSN=your choice.
- 2) GET,DSDI/UN=DEV1. (On S/N 101.)

or

GET,DSDI/JN=DEV1. (On S2.)

- 3) Create DSDI directives file:

A DSDI directive file should include the following:

IOUMR.
PROMR.
MEMMR.
PRORF.

W,first_byte_address,last_byte_address,asid. (where the
first_byte_address and last_byte_address are hex byte

1.0 DUAL STATE NOS/VE DEADSTART
1.8 DSDI INFORMATION

addresses and asid is the asid of the segment to be dumped)

4) Execute DSDI:

RFL,60000.
DSDI,M,D,I="input directives file".

5) To run (after the first time):

DSDI,I=n.

(Does not read tape again.)

6) To run interactively:

Same as above, except to do W command must first do:

OUTPUT,LISTFIL.

7) C170 DSDI information can be found in Chapter 10 of the NOS SYSTEM MAINTENANCE Manual.

A170 DSDI info can be found in document ARH3060 -- GID for A170 NOS/S2.

1.9 NAMIAE INFORMATION

Bringing NAM up.

1) At the system console enter:

FNC5,7700.
3.NAM.

2) If IAF is not up at control point 1, enter:

IAF.

3) To send messages to all terminals enter:

3.CFD.MSG,ALL,message.

Bringing NAM down.

1) At the system console, enter:

3.CFD.OI,NE.

2) Turn the teletype by the 2550 off.

1.0 DUAL STATE NOS/VE DEADSTART

1.9 NAMIAF INFORMATION

- 3) To bring IAF down enter:
1.STOP.

1.10 A170_NOS_SHUTDOWN

Before leaving the machine, it is necessary to bring NOS down. If NOS has crashed, a level 3 deadstart must be attempted (see Section 3.1.8) even if the only reason is to bring NOS down. To bring NOS down, do the following:

- 1) Enter:

CHE

The screen will display:

CHECKPOINT SYSTEM.

Enter: carriage return

- 2) Make sure no mass storage device has a checkpoint requested. To do this, enter: E,M. If the display shows there are no "C"s in the status field, then all devices are checkpointed and you may continue.
- 3) Enter:
STEP.
- 4) Push deadstart button.

1.11 INTERIM_MEMORY_LINK_STORAGE_MOVE_CONSIDERATIONS

The following precautions must be taken when running dual state with the Interim Memory Link storage move fix (NOS/VE Build J6 and associated NOS/A170 System).

- 1) Drop IRHF170, PASSON and all permanent file partner jobs before doing *BYEVE (via 2.STOP.)
- 2) Do not rollout IRHF170, PASSON or permanent file partner jobs at any time.
- 3) Before doing a CHECKPOINT SYSTEM, drop IRHF170, PASSON and all permanent file partner jobs (via 2.STOP.)
- 4) If the system crashes a NOS/A170 level 3 deadstart is the preferred action. If for some reason you must do an MCU recovery (REC command) do the following:

1.0 DUAL STATE NOS/VE DEADSTART
1.11 INTERIM MEMORY LINK STORAGE MOVE CONSIDERATIONS

- Clear word 17(8) via:
 - 99.
 - 17,0.
- Enter REC on the MCU console.
- Finish up in A170 only mode (i.e., do not do an UPMYVE), then do a level 3 deadstart. If you bring up NOS/VE again without doing a level 3 deadstart, the results are unpredictable.

1.12 NOS/VE INTERACTIVE FACILITY OPERATION

1.12.1 OPERATOR INITIATION

To bring up the NOS/VE interactive facility do the following:

- 1) Bring up NOS/VE (build M).
- 2) Bring up NOS/A170 networks:
 - FNC5,7700. (may sometimes be skipped)
 - 3.NAM.
- 3) Bring up A170 part of interactive:
 - TAFNVE. (If not yet done)
- 4) When NOS/VE is up, bring up the C180 part of interactive:
 - K.EXECUTE,,,,,IFEXEC,TASK_NAME=IFEXEC.

1.12.2 OPERATOR TERMINATION

To terminate NOS/VE interactive any of the following may be done:

- 3.CFD.DI,AP=TAF. (3 is the NAM control point number)

This is the preferred method. To bring NOS/VE interactive back up, you must first do a 3.CFD.EN,AP=TAF.

- 3.CFD.DI,NE. (2 is the NAM control point number)

This terminates the entire network including IAF,RBF, etc.

1.0 DUAL STATE NOS/VE DEADSTART
1.12.3 OTHER OPERATOR CAPABILITIES

1.12.3 OTHER OPERATOR CAPABILITIES

- To send a "shutdown warning" to all terminals logged on to TAF do:

3.CFD.ID,AP=TAF. (2 is the NAM control point number)
- To send a message to all terminals do:

3.CFD.MSG,ALL,message. (2 is the NAM control point number)
- PASSON has the ability to record various types of diagnostic information. This capability is controlled via the sense switches at the PASSON control point. To turn a sense switch on (off) at control point N do:

N.ONSWX. (N.OFFSWX.)

Where X is the desired sense switch (1 to 6). The PASSON default is all sense switches off. It will take a short period of time before PASSON detects a change in a sense switch and reacts to it. The sense switches currently used by PASSON are:

switch_#	use
1	Network Trace
2	PASSON Logic Trace To Dayfile
3	Memory Link Trace To Dayfile

1.12.4 INTERACTIVE TERMINAL OPERATION

1.12.4.1 Validation To Access NOS/VE

To access NOS/VE via the interactive facility, your user number must be validated to logon to the application named TAF. If you get the message 'ILLEGAL APPLICATION' when trying to logon, you are probably not validated to access TAF. To correct this do the following from the console:

X.MODVAL.

K,N. (N is the MODVAL control point number)
K,U,ccc. (ccc is your user number)

1.0 DUAL STATE NOS/VE DEADSTART
1.12.4.1 Validation To Access NOS/VE

K.AP=ALL.
K.END.
K.END.

If you get the message 'CONNECTION PROHIBITED' it means that only one terminal per user number can be logged on to TAF at one time. This indicates that someone has changed the network configuration - try a different user number until the network configuration is changed.

1.12.4.2 Login_To_NOS/VE

To initially login to NOS/VE via TAF, you must cause the first logon attempt to fail. This can be done by responding to the "FAMILY:" logon prompt with something like: "A,A,A". This must be done because the system will try to connect the terminal to IAF on the first logon attempt no matter what is typed. To access TAF do the following on the second "FAMILY:" prompt:

,user,password,TAF

You can access TAF from IAF by doing "HELLO,TAF" or by answering TAF to the system prompt "APPLICATION:".

1.12.4.3 Terminal Usage

- 1) The slash (/) is a prompt to enter a NOS/VE command. Any normal NOS/VE command can now be entered. The full ASCII character set (lower or upper case and all special characters can be used). Commands do not need to be ended with a period.
- 2) A LOGOUT command will cause the NOS/VE Interactive Job to terminate and it's dayfile (job log) will be returned to the A170 for printing. A new NOS/VE Interactive Job can then be started by responding to the 'APPLICATION:' prompt with TAF.
- 3) Terminal breaks (control-T and control-P) now work. It is possible to terminate a task or to suspend a task and enter a new task to process SCL commands. When a break is entered, the options available to the terminal user are:

R - resume from point of interruption

T - terminate task

S - process SCL commands. To resume execution at the point of interruption use the HCS command TMEXIT

1.0 DUAL STATE NDS/VE DEADSTART

1.12.4.3 Terminal Usage

(temporary).

1.12.4.4 NDS/VE Program Access To The Terminal

- 1) Interactive NDS/VE jobs are able to obtain terminal input through the AMP\$GET_NEXT or AMP\$GET_PARTIAL program interface which can be used by both task services and user ring programs. Interactive programs which use this interface should be able to handle both upper and lower case input in order to make them more convenient to use in both 64 and 96 character set modes.
- 2) Interactive NDS/VE jobs can send output to a terminal through the CLP\$PUT_STND_OUT program interface which can be used by both task services and user ring programs.

.....
2.0 S2 DEVELOPMENT LAB SUPPORT BY INTEGRATION
.....

2.0 S2 DEVELOPMENT LAB SUPPORT BY INTEGRATION

What we have established in the lab so far is the following:

- A 600 tape capacity tape rack for general use (located in near proximity to the 67X tape drives). If your project would like to reserve a section of this tape rack, contact Tim or myself.
- A tape and disk cabinet for storage of system support materials which this project will manage and keep up to date. (If you have been using this cabinet for unauthorized storage - beware. We have the key to the lock!) More will be published about the contents of this cabinet later, and a cabinet index will be posted in the lab to help locate where things are supposed to be placed within the cabinet. This cabinet is currently located against the East wall of the lab, is 6 ft. 8 in. tall, gray in color and with sliding door.
- A two-level documentation rack for system documentation listings. This contains the current build compilation listing interface deck compilation listing (from module named QLND5), listing of the NDS/VE PPU routines, system link map, and various assorted PVE listings. This rack is next to the tape and disk cabinet at this time.
- A desk documentation rack for reference manuals and Tom McGee's collection of "how to" goodies. The objective is to have this reference information at arm's length of the console, but it is currently on top of the two-level unit by the East wall.
- At or near the console is a small notebook containing the NDS System Programmer's Instant, NDS Application Programmer's Instant, and the 180 Instruction codes.

Feel free to examine and use all of the above materials while in the lab. Do not remove or abuse any of these materials. Please notify Tim McGibbon or Mike Carter of any problems or deficiencies of these materials. Leave a note if we are not available.

APPENDIX A NOS/VE BACKGROUND DOCUMENTS

1.0 Hardware Overview

- 1.1 An Introduction to CYBER 180
- 1.2 C180 Instant
- 1.3 Model Independent General Design Specification - ARH1700

2.0 NOS Reference Manuals

- 2.1 XEDIT V3.0 - 60455730
- 2.2 IAF V1.0 User's Guide - 60455260
- 2.3 NOS Reference Manual - Vol 1, 60435400 - Vol 2, 60445300
- 2.4 NOS Instant
- 2.5 NOS Operators Guide - 60435600
- 2.6 NOS Diagnostic Handbook
- 2.7 NOS A170 ERS
- 2.8 NOS A170 GID - ARH3060

3.0 NOS/VE Reference Documents

- 3.1 Program Interface ERS - obtained from Karen Rubey
- 3.2 Command Interface ERS - obtained from Karen Rubey
- 3.3 NOS/VE User's Guide - obtained by the following:
ATTACH,HUG/UN=DAH
SES.FORMAT I=HUG,L=LIST,LOCAL,TEXTFORM
SES.PRINT LIST
- 3.4 NOS/VE Procedures and Conventions - obtained by
SES,MAD.LISTPC
- 3.5 JFS Deadstart/180 Operating Procedures
- 3.6 Listing of all NOS/VE Modules - obtained by
SES,DEV1.LISTNVE. See Integration Procedures Notebook
for details.

3.7 Listing of all NOS/VE type declarations - obtained by doing, SES,DEV1.LISTNVE M=QLNOS.

3.8 NOS/VE Code Transmittal/PL Maintenance Procedures

See Integration Procedures Notebook.

3.9 NOS/VE Internal Interface Maintenance Procedures

Memo available from S.C. Wood.

3.10 Integration Procedures Notebook

Obtained by:

Acquire,IPNDOC2/UN=MDC. SES.PRINT,IPNDOC2.

4.0 Tools Reference Documents

4.1 PASCAL-X Interactive Debugger - ARH3142

4.2 SES User's Guide - ARH1833

4.3 PASCAL-X Specification - ARH2298

4.4 C180 Assembler ERS - ARH1693

4.5 Simulator ERS - ARH1729

4.6 VEGEN ERS - ARH2591

4.7 VELINK ERS - ARH2816

4.8 Simulated I/O ERS - ARH3125

4.9 Object Code Utilities ERS - ARH2922

4.10 CYBIL Implementation Dependent Handbook - ARH3078

5.0 Dual State Cookbook

To acquire additional copies of this document enter:

ACQUIRE DOCUMEN/UN=SKT512
SES.FORMAT I=DOCUMEN TXTFORM.

Table of Contents

1.0 DUAL STATE NOS/VE DEADSTART	1-1
1.1 A170 NOS DEADSTART	1-1
1.2 CURRENT DUAL STATE CONFIGURATION	1-1
1.3 DUAL STATE, NOS OPERATION	1-2
1.4 NOS/VE DEADSTART	1-3
1.5 NOS/VE TERMINATION	1-3
1.6 BRING UP C170 REMOTE HOST	1-4
1.7 BRING UP C180 REMOTE HOST	1-4
1.7.1 ROUTE AN INPUT FILE FROM C170 TO C180	1-4
1.7.1.1 Route A Print File From C180 to C170	1-5
1.7.1.2 LINK_USER Command	1-6
1.7.1.3 Get A 170 Permanent File From 180	1-7
1.7.1.4 REPLACE A 170 Permanent File From 180	1-8
1.7.1.5 Example Jobs Representing Phase A/B Library Creation/Modif	1-11
1.7.1.5.1 CREATE OBJECT LIBRARY ON NOS/VE AND SAVE IT ON A170 NOS	1-11
1.7.1.5.2 MODIFY A PREVIOUSLY SAVED OBJECT LIBRARY	1-12
1.7.1.5.3 USAGE OF NOS/VE LOADER	1-13
1.7.1.5.4 DIFFERENCES BETWEEN PHASE C (BUILD M) AND PHASE A/B (BUI	1-14
1.7.1.5.5 BAM BUILD J DEVIATION FROM ERS	1-27
1.8 DSDI INFORMATION	1-28
1.9 NAMIAF INFORMATION	1-30
1.10 A170 NOS SHUTDOWN	1-31
1.11 INTERIM MEMORY LINK STORAGE MOVE CONSIDERATIONS	1-31
1.12 NOS/VE INTERACTIVE FACILITY OPERATION	1-32
1.12.1 OPERATOR INITIATION	1-32
1.12.2 OPERATOR TERMINATION	1-32
1.12.3 OTHER OPERATOR CAPABILITIES	1-33
1.12.4 INTERACTIVE TERMINAL OPERATION	1-33
1.12.4.1 Validation To Access NOS/VE	1-33
1.12.4.2 Login To NOS/VE	1-34
1.12.4.3 Terminal Usage	1-34
1.12.4.4 NOS/VE Program Access To The Terminal	1-35
 2.0 S2 DEVELOPMENT LAB SUPPORT BY INTEGRATION	 2-1
 APPENDIX A NOS/VE BACKGROUND DOCUMENTS	 A1