THE SOFTWARE DISPATCH REVIEW

XVM/DOS V1A

MARCH 1976

DEC-XV-XSPSA-A-D

digital

THE SOFTWARE DISPATCH REVIEW

XVM/DOS V1A

MARCH 1976

DEC-XV-XSPSA-A-D

The material in this document is for information purposes only and is subject to change without notice. Digital Equipment Corporation assumes no responsibility for any errors which may appear in this document. Software binaries and sources are provided only under licenses. The standard terms and conditions, OEM Agreement, and/or quantity discount agreement contain the licenses for all binaries other than for the DECsystem-10.

digital

SOFTWARE SERVICES

Copyright 1976 DIGITAL EQUIPMENT CORPORATION

Betty Steinfeld
Editor

Sharon Lewis
Publications Coordinator

TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS

COMPUTER LABS
DDT
DEC
DECUS
DIGITAL
EDGRIN
FLIP CHIP
FOCAL
GLC-8
KA10
LAB-K
PDP
SABR
TYPESET-8
UNIBUS
DECTAPE
IDACS
INDAC-8
QUICKPOINT
DIBOL
EDUSYSTEM
OMNIBUS
OS/8
COMTEX-11
CDP
RAD-8
RSX
RSTS
DECCOMM
PHA
LAB-8
DECSET
DECSYSTEM-10
MASSBUS
TYPESET-11
CLASSIC
DECAL
XVM

XVM/DOS

1.0 INTRODUCTION

2.0 FILING
2.1 System Components
2.2 Version Numbers and Support Categories
2.3 System Documentation
2.3.1 General Systems Manuals
2.3.2 Language Manuals
2.3.3 Utility Programs

3.0 SOFTWARE PERFORMANCE REPORTS
3.1 Software Performance Report Guidelines

4.0 XVM/DOS SYSTEM NOTES

1.0 INTRODUCTION

The Software Dispatch Review is a compendium of information which provides a customer with a maintenance notebook on current software documentation and the status of known software problems. The notebook is supplemented with articles in the monthly Software Dispatch which should be filed in the appropriate sections of the Software Dispatch Review.

It is recommended that users make all the published patches. Users with source files are asked to make the indicated source level changes and suitably update their software system.

2.0 FILING

This introductory material should be filed at the beginning of the notebook.

A system has been devised to help you file each article in its proper place. The key is the following figure:

<table>
<tr><td colspan="2">SOFTWARE PRODUCT
(1)</td><td colspan="2">VERSION
(1A)</td></tr>
<tr><td colspan="2">COMPONENT
(2)</td><td>VERSION
(2A)</td><td>EDIT
(2B)</td></tr>
<tr><td colspan="2">SUBPROGRAM OR ADDITIONAL INFORMATION
(2C)</td><td>SEQUENCE
(3)</td><td>PAGE
OF
(3A)</td></tr>
<tr><td>NEW
(4)</td><td>REPLACEMENT ARTICLE
(5)</td><td colspan="2">ORIGINAL DATE
(5A)</td></tr>
</table>

Coding Block.

Each month the Software Dispatch should be taken apart and inserted into the notebook.

First, the articles are classified by software product (1). All articles should be filed under the appropriate major heading.

Secondly, the software product is broken down by its components (2). See section 2.1 for the list of software components.

Articles pertain to the program with the latest version (2A) or edit (2B), whichever is applicable. Source level changes will always affect the edit.

All programs which print out their version number do so initially as V1A000. A new update release would cause the letter to be changed to "B". This is the major release level of the monitor. Whether a change to a program is given as a source change, the three digit code in the version number is incremented by one; e.g., V1A001.

Finally, the article is referenced by sequence number (3). As an article is added to each component, it is assigned the next higher sequence number.

Additional information in the coding block is presented to further clarify the article and is not specifically for filing:

(1A) Version of the software product.

(2A) Version number of the component.

(2B) Edit number.

(2C) Other information helpful to the user.

(3A) Page number and pages in the article.

(4) An "X" in this block indicates a new article.

(5) A number in this block indicates an article republished for revision or correction and specifies the number of the revision. For example, the second revision of an article which originally appeared in June 1974 is shown in the following figure.

(5A) Original date of a revised article.

NEW	REPLACEMENT ARTICLE	ORIGINAL DATE
	2	June 1974

Coding block showing the second revision.

2.1 System Components

Articles concerning changes to manuals will be coded and filed along with changes to programs. For example, an article containing a change to the CHAIN program could be followed by an article amending the CHAIN and EXECUTE manual. Changes to the monitor manuals and other general system documents will appear under the code Monitor.

XVM/DOS	Additional Information	Comment
ABSL11		PDP-11 Absolute Loader
BOOTSTRAP	RFBOOT	RF Disk Bootstrap
	RKBOOT	RK Disk Bootstrap
	RPBOOT	RP Disk Bootstrap
BUILD	BUILD	
	CODE	XVM Installation Aid
CARD READER HANDLERS	CD.DOS	Card Reader Handler
CHAIN	CHAIN	
	EXECUTE	
DDT		
DECTAPE HANDLERS	DTA.	
	DTC.	
	DTD.	
	DTE.	
	DTF.	
DISK HANDLERS	DKA.	RF Disk Handlers
	DKB.	
	DKC.	
	DKL.	
	DPA.	RP Disk Handlers
	DPB.	
	DPC.	
	DPL.	
	RKA.	RK Disk Handlers
	RKB.	
	RKC.	
	RKL.	
DOSSAV		System Save/Restore Program
DUMP		
EDITOR		
EDITVP		
EDITVT	FOCAL	
FOCAL	FNEW	
FORTRAN		General FORTRAN Issues

FORTRAN COMPILER		Compiler
FORTRAN OTS		Object time system
		Utility routines
LINKING LOADER	.LOAD	Relocatable file loader
LINE PRINTER HANDLERS	LPA.15	XVM Line Printer Handler
	LPU.	PDP-11 Line Printer Handler
LK35 KEYBOARD HANDLER	LKA	
MACRO XVM	MACRO	XVM Assembler
	CREF	Cross reference program
MCLOAD	MACIMG	
MAC11	MACINT	
	M11.8	
	M11.12	
MAGTAPE	MTA.	Magtape handler
	MTC.	Magtape handler
	MTDUMP	
	MTF.	Magtape handler
MONITOR	DOSNRM	Nonresident MONITOR
	RESMON	Resident Monitor
PAPERTAPE PUNCH HANDLERS	PPA	
	PPB	
	PPC	
PAPERTAPE READER HANDLERS	PRA.	
	PRB.	
PATCH		
PIP		
PIREX		PDP-11 Peripheral Processor Executive
PLOTTER HANDLER	XYA.	Plotter Handler for XY11/XY311
QDMP XVM		XVM Core Dumper
QFILE		
SPLGEN		
SPLOAD	SPLIMG	
SPOOL	SPOL15	
	SPOL11	
SRCCOM		
SYSTEM		All system information that does not fall under any other component will appear here.
SGEN		
SYSTEM LOADER	.SYSLD	System Program Loader
UDMP11		PDP-11 Core Dumper
UPDATE		
VP15 GRAPHICS	VPA.	
	VPA.S	
	FORT	
	NUVAL	
	VECTOR	

VT15 GRAPHICS	VTA.	
	VTPRIM	
	LTORPB	
	DYLDR	
	TRACK	
	ROTATE	
	CIRCLE	
WRITING TABLET HANDLER	VWA.	
8TRAN		
BOSS	B.PRE	Preprocessor
	NRBOSS	Nonresident monitor
BOSS PROCEDURE FILE		The additional information may contain any of the procedure files listed in the BOSS XVM Users Manual.

2.2 Version Numbers and Support Categories

Language & Utilities Category 1		Device Handlers Category 1		Monitors Category 1	
Program	Version	Program	Edit Number	Program	Version
ABSL11	(EDIT) 103	BOOTSTRAPS		XVM/DOS	V1A000
BOSS	V1A000	RFBOOT	017		
CHAIN	V1A000	RKBOOT	020		
DDT	V1A000	RPBOOT	009		
DOSSAV	V1A000	CARD READER HANDLERS			
DUMP	V1A000	CD.DOS	134		
EDITOR	V1A000	DECTAPE HANDLERS			
EDITVP	V1A000	DTA.	129		
EDITVT	V1A000	DTC.	103		
EXECUTE	V1A000	DTD.	108		
FOCAL	V1A000	DTE.	108		
FORTRAN	V1A000	DTF.	108		
LOADER	V1A000	DISK HANDLERS			
MACRO XVM	V1A000	DOSRFA	159		
MAC11	V1A000	LINE PRINTER HANDLERS			
MTDUMP	V1A000	LPA.15	150		
PATCH	V1A000	LPU.	124		
PIP	V1A000	LK35 KEYBOARD HANDLER			
PIREX	154	LKA.	002		
SPLGEN	V1A000	MAGTAPE HANDLERS			
SPLOAD	V1A000	MTA.	108		
SPOOL	V1A000	MTC.	101		
SGEN	V1A000	MTF.	114		
SRCCOM	V1A000	PAPERTAPE PUNCH HANDLERS			
UPDATE	V1A000	PPA.	102		
8TRAN	V1A000	PPB.	101		
		PPC.	101		
		PAPERTAPE READER HANDLERS			
		PRA.	100		
		PRB.	100		
		PLOTTER HANDLER			
		XYU.	036		
		VP15 GRAPHICS HANDLER			
		VPA.	113		
		VT15 GRAPHICS HANDLER			
		VTA.	006		
		WRITING TABLET HANDLER			
		VWA.	005		

2.3 System Documentation

2.3.1 General System Manuals

	Order Code
BOSS XVM USERS MANUAL	DEC-XV-OBUAA-A-D
XVM/DOS USERS MANUAL	DEC-XV-ODMAA-A-D
XVM/DOS SYSTEM MANUAL	DEC-XV-ODSAA-A-D
XVM/DOS KEYBOARD COMMAND GUIDE	DEC-XM-ODKBA-A-D
XVM UNICHANNEL SOFTWARE MANUAL	DEC-XV-XUXMA-A-D
SGEN XVM UTILITY MANUAL	DEC-XV-USUTA-A-D
XVM/DOS READ ME FIRST USERS GUIDE AND MASTER INDEX	DEC-XV-ODGIA-A-D
XVM/DOS V1A SYSTEM INSTALLATION GUIDE	DEC-XV-ODSIA-A-D

2.3.2 Language Manuals

FORTRAN IV XVM LANGUAGE MANUAL	DEC-XV-LF4MA-A-D
FORTRAN IV XVM OPERATING ENVIRONMENT MANUAL	DEC-XV-LF4EA-A-D
FOCAL XVM LANGUAGE MANUAL	DEC-XV-LFLGA-A-D
MACRO XVM ASSEMBLER LANGUAGE MANUAL	DEC-XV-LMACA-A-D
MAC11 XVM ASSEMBLER LANGUAGE MANUAL	DEC-XV-LMLAA-A-D

2.3.3 Utility Programs

CHAIN XVM/EXECUTE XVM UTILITY MANUAL	DEC-XV-UCHNA-A-D
DDT XVM UTILITY MANUAL	DEC-XV-UDDTA-A-D
EDIT/EDITVP/EDITVT XVM UTILITY MANUAL	DEC-XV-UETUA-A-D
LINKING LOADER XVM UTILITY MANUAL	DEC-XV-ULLUA-A-D
MTDUMP XVM UTILITY MANUAL	DEC-XV-UMTUA-A-D
PATCH XVM UTILITY MANUAL	DEC-XV-UPUMA-A-D
PIP XVM UTILITY MANUAL	DEC-XV-UPPUA-A-D
SRCCOM XVM UTILITY MANUAL	DEC-XV-USRCA-A-D
UPDATE XVM UTILITY MANUAL	DEC-XV-UUPDA-A-D
VP15A XVM GRAPHICS SOFTWARE MANUAL	DEC-XV-GVPAA-A-D
VT15 XVM GRAPHICS SOFTWARE MANUAL	DEC-XV-GVTAA-A-D
8TRAN XVM UTILITY MANUAL	DEC-XV-UTRNA-A-D

3.0 SOFTWARE PERFORMANCE REPORTS

Each new installation is provided with Software Performance Report (SPR) forms. The SPR form enables users to suggest enhancements to or report problems with Digital Equipment Corporation software or documentation. When a problem is encountered, an SPR should be completed and mailed to the local SPR Center. (See inside back cover.)

Responses will be sent to the name and address appearing on the form. Additional SPR forms may be obtained by writing to the local SPR Center.

3.1 Software Performance Report Guidelines

The following is a guideline for completing Software Performance Reports (SPRs) so that adequate information is included.

For all XVM/DOS systems, please completely fill out the SPR form. Subscription and warranty customers must enter their system serial number on the SPR form. It is important that we know the machine configuration--including the system disk type, the amount of core in use and the peripherals on the machine. The name and version number of the system program in use, if any, is absolutely essential. An adequate and clear description of the problem is very important and will certainly speed processing of the SPR. Two of the best ways of supplementing the description is to include the terminal printout that shows the problem as well as an actual copy of the user program that caused the problem if one is involved.

Further, UNICHANNEL-15 should include this additional information, if possible:

1. The UNICHANNEL-15 configuration, including the core size of the PDP-11 and PDP-11 peripherals.

2. The assembly parameters used to create the PIREX and SPOOLER in use.

3. A listing of PIREX or SPOOLER if any user modifications have been made.

4. A dump of both PDP-11 core (use UDMP11) and XVM core (use QDMP XVM). UDMP11 requires an LP11/LS11/LV11 line printer.

4.0 XVM/DOS V1A SYSTEM NOTES

	Sequence
BOOTSTRAP	
Programming Note Regarding .GET & PUT MACROs	1†
BOSS	
Restrictions in $CRT and $ADD Usage	1†
CARD READER HANDLER	
Error Messages for PDP-15 NONUNICHANNEL CARD READERS	1†
CHAIN	
Notes on the Operation of CHAIN	1†
Multiple Entry Points Handled Incorrectly in Overlays	2
DECTAPE HANDLERS	
Reading Beyond End-of-File	1†
Support of .TRAN Function on PDP-8/10/11 Tapes	2†
DISK HANDLERS	
Renaming Files	1
FORTRAN	
Usage Restriction and Precaution	1†
Use of ADSS vs XVM/DOS Magtape Handler MTF	2†
A Change in the Handling of Carriage Control Characters in Nonprinter Devices by the FORTRAN Object Time Routine BCDIO	3
Programming Note to Eliminate Carriage Return	4†
Using System Buffers for FORTRAN OTS Double Buffering	5
FORTRAN COMPILER	
Modes of Statement Function Variables	1†
Programming to Avoid Incorrect Mode Typing of Functions Declared EXTERNAL	2†
Variables Declared INTEGER are Typed as LOGICAL	3†
DOUBLE INTEGER Literals Incorrectly Defined	4†
Transferring Control to a FORMAT Statement	5†
Restriction of Variables in Named COMMON	6†
Displaced Error Reports	7†
End of Compilation Message	8

†Replacement Article.

	Sequence
FORTRAN OTS	
Programming Note Regarding Argument Address GET Routine, .DA	1†
Numerical Restriction in AMOD and DMOD	2†
IOPS31 Errors When Referencing the Last Element of an Array	3
First Character of Each Record Sometimes Truncated On Mass Storage Files	4
LINKING LOADER	
Programming Note	1
Incomplete Loading of Subroutines with Multiple Entry Points	2
BATCH .DAT Slots	3
BATCH I/O Device Use	4
System Build Leaves Erroneous .DAT Assignment	5
MACRO XVM	
Programming Note on Use of T Switch	1†
MACRO E Switch Does Not Give Correct Lines in Console Device Under BOSS	2
Edits to Bring MACRO to XVM/RSX Release Level	3
MONITOR	
Teletype Handler .INIT Function Limitations	1†
.DAT -12 Set Up	2
PATCH	
MICLOG Before Patching on Disk	1
FORTRAN Programs Cannot be Made System Programs	2
PIP	
IOPS0 after (H) Copy to System Disk	1
SYSTEM	
Conversion from ADSS-15: Programs May Not Fit	1
Using API Level 4	2
System Manual Not Complete	3

†Replacement Article.

Programming Note Regarding .GET & .PUT MACROs

The unit number field in an object code product before XVM/DOS V1A000 with .GET and .PUT monitor calls is ignored in the XVM/DOS system. ↑Q AREAS will be utilized if present only on the current system device unit Ø. XVM/DOS .GET/.PUT MACROs do not allow unit number fields.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
BOOTSTRAP	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	1	1 1
NEW ☐ REPLACEMENT ARTICLE 2	ORIGINAL DATE	
	Pre-1973	

Restrictions in $CRT and $ADD Usage

PROBLEM:

The following sequence of cards causes BOSS XVM to crash with an IOPS11 error.

```
$ADD XYZ
$ADD YYY

   or

$CRT XYZ
$CRT YYY
```

RESTRICTION:

Two $ADD cards or $CRT cards cannot be adjacent. The system will not handle this properly.

A user can get around this by inserting any legal BOSS command card(s) other than $CRT, $ADD, $JOB and $END between the two cards.

Error Messages for PDP-15/XVM NON-UNICHANNEL CARD READERS

When a hopper-empty, stacker-full or reader-not-ready condition arises, the following message is printed:

IOPS4 CD NOT READY

To recover, clear the condition, ready the reader, and type CTRL R.

If a card with an illegal punch is encountered the following message is printed:

IOPS4 CD-ILLEGAL PUNCH

To recover, punch a correct card, discard the incorrect card, place the new card in the input hopper, and type CTRL R to continue.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
CARD READER HANDLER	N/A	134
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	1	1 OF 1
NEW ☐ REPLACEMENT ARTICLE 1	ORIGINAL DATE	
	August 1974	

Notes on the Operation of CHAIN

1. Routines or subroutines declared as part of the resident code by using the library indicator (#) in the command string are entered into a dummy global symbol table. Hence, it is necessary for a ".GLOBL name" identical to the name accompanying the "#" to be present within the routine to be called. An identical file name is not sufficient; in fact, file names are ignored when searching for library indicator (#) routines. In the absence of this declaration, the error message "UNRESOLVED GLOBAL" will result.

2. CHAIN scans the user library (.LIBR5) before scanning the system library (.LIBR) to load library routines and satisfy unresolved globals.

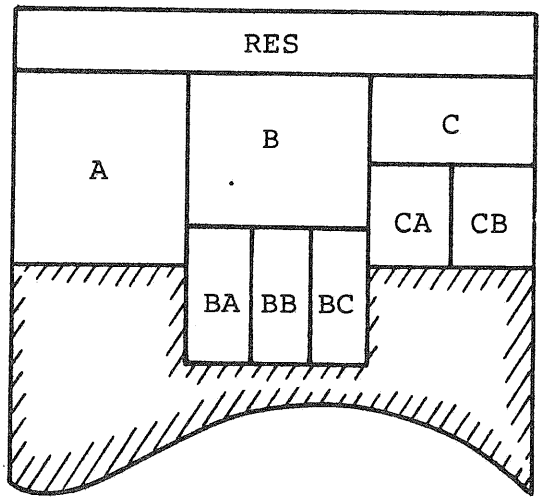
3. CHAIN resolves globals in a manner similar to the LINKING LOADER.

Restrictions in building an overlay structure:

Pages 7-5 through 7-8 of the Chain/Execute Manual define a set of rules that govern the building of an overlay structure through the definition of links and structures.

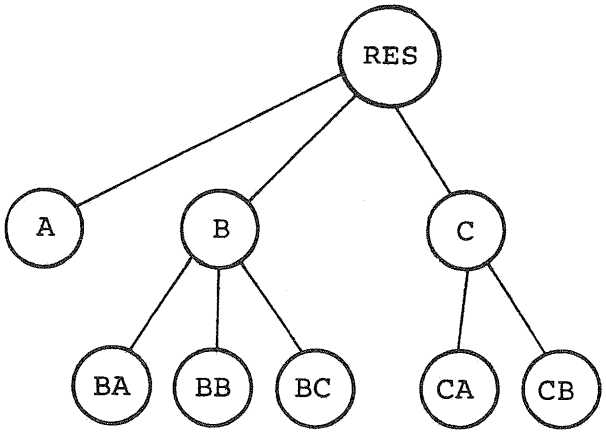
There are a few overlay structures that cannot be defined within the framework of these rules. Any attempt to define one of these overlay structures will result in the printout of any one of the appropriate error messages.

An example of one such overlay structure that cannot be built follows:



high core

low core

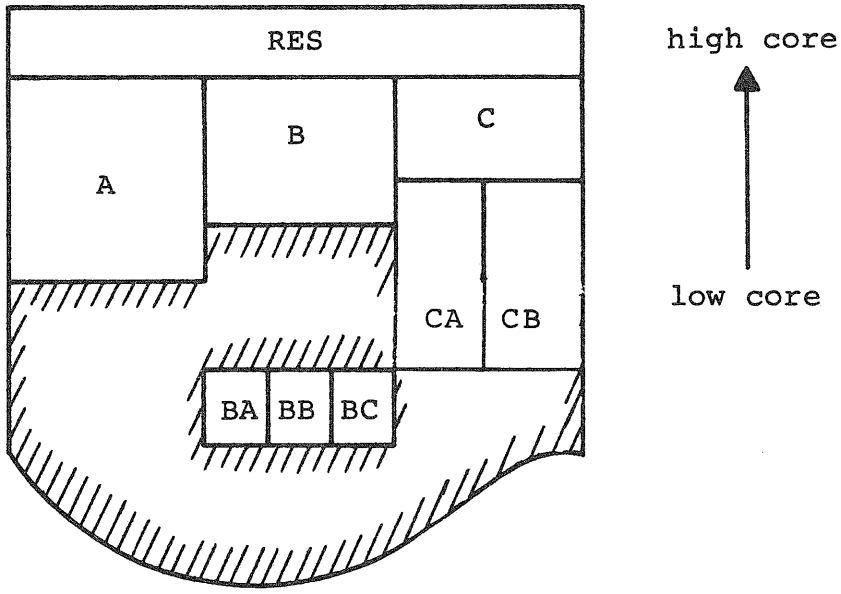


equivalent tree structure

The following attempt to define this overlay structure will cause the error message indicated to be printed out.

```
>LA=A
>LB=B,BA
>LC=C,CA
>BA:BB:BC
↑COMPONENT NAME USED AS LINK NAME ---BA
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
CHAIN	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION CHAIN XVM/EXECUTE XVM UTILITY MANUAL DEC-XV-UCHNA-A-D	SEQUENCE 1	PAGE OF 2 3
NEW ☐ REPLACEMENT ARTICLE 3	ORIGINAL DATE May 1973	



It is possible to define the above overlay structure by the definitions:

```
>A:B,BA
>B:C,CA
>CA:CB
>BA:BB:BC
```

In general there can be only one link in an overlay structure whose link components, if any, are allocated contiguously in core (as in link "C, CA"). Hence, there will be holes in core if an overlay structure like the one above is built.

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
CHAIN		N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION CHAIN XVM/EXECUTE XVM UTILITY MANUAL DEC-XV-UCHNA-A-D		SEQUENCE 1	PAGE OF 3 3
NEW	REPLACEMENT ARTICLE	ORIGINAL DATE	
	3	May 1973	

Multiple Entry Points Handled Incorrectly in Overlays

PROBLEM:

When used in overlays, program units with multiple entry points cause DUPLICATE GLOBAL DEF or UNRESOLVED GLOBAL errors to be output by CHAIN.

RESTRICTION:

Program units with multiple entry points can not be placed in overlays but must be placed in the resident code.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A000	
COMPONENT	VERSION	EDIT
CHAIN	V1A000	179
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	2	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

Reading Beyond End-of-File

The last block of a file on DECtape has a forward data link of 777777. If the user tries to read past this block, DTA will return the end-of-file sequence (ØØ1ØØ5,776773) in the user buffer. Subsequent .READs will continue to pass back the same two words. This corrects a problem that occurred in PIP when reading in dump mode and when the ØØ1ØØ5,776773 sequence was part of the data being transferred, with still more data following it.

NOTE: The above change has not been made to the DTC., DTD., DTE., or DTF. DECtape Handlers.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
DECTAPE HANDLERS	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
DTA.	1	1 1
NEW ☐ REPLACEMENT ARTICLE 2	ORIGINAL DATE Pre-1973	

Support of .TRAN Function on PDP-8/1Ø/11 Tapes

The DECtape handlers in the system (except for DTC. and DTF.) can be utilized to transfer data from PDP-8/1Ø/11 tapes using the .TRAN function.

In order to realize the above, a minor source modification is needed. The instruction,

```
AND (7777 /clear possible erroneous data
```

must be inserted after the location DTSRCK+2, i.e., after the instruction,

```
LAC  DTBCA
```

in DTA., DTD. & DTE.

This is necessary because these tapes have 12-bit block numbers with extraneous data in the most significant bits.

Note that since the file structure on these tapes is different from the XVM/DOS file structure, only the .TRAN function can be used.

This information is provided for the user's convenience only and is not a feature supported by DEC.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
DECTAPE HANDLERS	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
DTA., DTD., DTE.	2	1 1
NEW ☐ REPLACEMENT ARTICLE 2	ORIGINAL DATE	
	November 1972	

Renaming Files

PROBLEM:

All Disk Handlers allow a file to be renamed to the same name as an existing file.

DISPOSITION:

Should this happen, use PIP to rename the files, noting that PIP will always access the first file it finds in a directory with the specified name.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
DISK HANDLERS	N/A	150
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
DOSRFA	1	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

Usage Restriction and Precaution

PROBLEM:

The FORTRAN IV Language Manual DEC-XV-LF4MA-A-D describes the usage of object time FORMAT specifications. It is not allowed, however, to use the name of an array that appears in a SUBROUTINE statement parameter list as an array name that is referenced by an I/O statement; that is, in the following program, the construction is correct.

```
      DIMENSION IBUF(1Ø),FORM(1Ø)
      DATA FORM(1) /5H(4I1Ø/
      DATA FORM(2) /5H)    /
      DO 1Ø I = 1,1Ø
1Ø    IBUF(I) = I
      NSZ=4
      WRITE (6,FORM)  (IBUF(I),I=1,NSZ)
      CALL PRINT (IBUF,NSZ,FORM)
      PAUSE
      END
```

The first four elements of IBUF will be printed according to the format specified in the array FORM. If this is attempted in the subroutine PRINT, shown below, an OTS 12 will occur.

```
      SUBROUTINE PRINT (IBUF,NSZ,FORM)
      DIMENSION IBUF(1), FORM(1Ø)
      WRITE (6,3) (FORM(I),I=1,1Ø)
3     FORMAT(1X,1ØA5)
      WRITE (6,FORM) (IBUF(I),I=1,NSZ)
      RETURN
      END
```

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
FORTRAN		N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION		SEQUENCE	PAGE OF
		1	1 2
NEW	REPLACEMENT ARTICLE	ORIGINAL DATE	
	2	March 1973	

Usage Restriction and Precaution

To avoid this problem, it would be necessary to create some array in PRINT and to copy FORM into it. The former array could then be specified in the WRITE statement.

As a further precaution, always enclose your FORMAT specification in parentheses when using this technique. Note that this was done in the DATA statement of the main program on the previous page.

Use of ADSS vs XVM/DOS Magtape Handler MTF

PROBLEM:

1. The MTF. handler released with ADSS-15 V5A does not calculate checksums for IOPS binary records. MTF. released with XVM/DOS does calculate checksums. Users attempting to read old data tapes under XVM/DOS FORTRAN may run into problems.

2. Under XVM/DOS MTF. has a default buffer size of 255_{10}. The ADSS-15 version has a fixed buffer size of 56_{10}.

3. The buffer size in XVM/DOS MTF. may be changed under program control by referencing the global "MTBSIZ." (See XVM/DOS Users Manual.)

4. Under XVM/DOS, MTF. checks for record-length errors prior to calculating checksums; and, if the record length is less than 255 (i.e., 56), MTF. returns to the user without calculating a checksum.

5. MTA and MTF generate EOF in the same mode as the current data transfer mode. This may cause some problems in applications written prior to DOS-V3B.

The above inconsistencies create the following possibilities when reading old data tapes under XVM/DOS FORTRAN and MTF.

	MTBSIZ	OTS Error	Data Read In
One physical record per logical record	255_{10}	None	Good
	56_{10}	11	--
More than one physical Record per logical record	255_{10}	None	Bad
	56_{10}	11	--

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
MTF.	2	1 2
NEW ☐	REPLACEMENT ARTICLE 2	ORIGINAL DATE
		Pre-1973

Use of ADSS-15 vs XVM/DOS Magtape Handler MTF

SOLUTION:

The user may copy an old data tape using the following program.

```
10    CALL SET56
      READ (1, END = 11) LIST
      CALL SET255
      WRITE (2) LIST
      GO TO 10

11    CALL SET255
      WRITE (2) LIST
      CALL CLOSE (1)
      CALL CLOSE (2)
      STOP
      END
```

SET56 and SET255 are two MACRO XVM subroutines:

```
                .GLOBL SET56,MTBSIZ,.FM

SET56           0
                LAC  (70)
                DAC* MTBSIZ
                DAC* .FM
                JMP* SET56
                .END

                .GLOBL SET255,MTBSIZ,.FM

SET255          0
                LAC  (377)
                DAC* MTBSIZ
                DAC* .FM
                JMP* SET255
                .END
```

NOTE: Do not modify these programs. Insure that tapes are at load point before starting.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
MTF.	2	2 2
NEW ☐ REPLACEMENT ARTICLE ☐ 2	ORIGINAL DATE	
	Pre-1973	

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

HISTORY: Operation on DOS V1A, DOS V2A, and RSX PLUS III

The PDP-15 FORTRAN I/O system has always had a somewhat inconvenient behavior concerning the first character (the carriage control character) of formatted ASCII records. The BCDIO program, which actually builds or decodes the output or input lines under control of the specified FORMAT statement, always interpreted the first character of each output record as a carriage control character. Upon output, BCDIO examined the first character placed in the buffer and always performed the following translation:

Character Found	Translated to	Meaning to LP Handler
'1'	FF, 14_8	Skip to top of form
'+'	DLE, $2Ø_8$	Overprint
'Ø'	DC1, 21_8	Double space
' ' (space)	LF, 12_8	Single space
anything else	LF, 12_8	Single space

The implications of this method are obvious: Output to the line printer and console terminal worked as desired; output to the papertape punch was good, since a more or less standard format for ASCII paper tape is LF-data-CR. However, disk, DECtape, and magtape files also had a carriage control character preceding each record; or, if the unwary user did not

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 1 16
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

provide a space to be used for carriage control, the first character of his data was lost.

This action is documented in the footnote on page 6-5 of the FORTRAN Language Manual, DEC-XV-LF4MA-A-D.

On input, when BCDIO was decoding an input line, it first dropped line feeds, form feeds, etc. (actually it dropped any character less than 40_8). No action was taken when there was no leading line feed character.

The conclusion to all of this is that:

1. On output, the first character of each line was translated by BCDIO:

First Character	Translated to	Meaning
'1'	FF, 14_8	Skip to top of form
'0'	DC1, 21_8	Double space
'+'	DLE, 20_8	Overprint
' ' (space)	LF, 12_8	Single space
any other	LF, 12_8	Single space

before the line was written to any device.

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 2 16
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

2. An extra character position must have been provided in FORMAT statements even when the intended destination device was a mass storage unit. For example:

```
    WRITE (3,1ØØ) A,B,C
1ØØ FORMAT (1X,3F6.2)
```

   This was to prevent BCDIO from destroying the first data digit.

3. Since carriage control character translations always occurred, a program which produced a report intended for the line printer or console terminal which used forms control characters '1', '+' or 'Ø' in addition to ' ' produced a PIPable file if the .DAT slot normally used for the printer was assigned to a disk. In other words, if a disk was assigned rather than a line printer, a file was created by the OTS which contained the translated listing if the file were PIPed to the printer.

4. A different FORMAT statement was needed to read a record than was used to write the record. The input FORMAT needed to describe a record which was one character is shorter (specifically, missing the first character) than the record described by the output FORMAT. Using the example from above:

```
    READ (3,1Ø1) A,B,C
1Ø1 FORMAT(3F6.2)
```

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 3 16
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

This was because the leading line feed produced by BCDIO in the output case was stripped on input by BCDIO rather than retranslated to a ' ' (space) character.

5. The same FORMAT statement may be used to read data from any device, whether the device has leading line feeds (mass storage, paper tape, etc.) or not (terminal, card reader, etc.).

The problem with this method, of course, was the need to use a different FORMAT statement for writing and reading the same record. While this restriction may be annoying, it did not appear to be a violation of the USA FORTRAN standard, X3.9 1966:

7.1.3.2 READ and WRITE Statements

(page 16, col. 1, line 3)

Records may be formatted or unformatted. . . . When an unformatted or formatted READ statement is executed, the required records on the identified unit must be, respectively, unformatted or formatted records.

This states that once a record has been written with a FORMAT statement, it must be read with a FORMAT statement, but the FORMAT statements need not be the same.

In the January 27, 1975 revision of the ANSI FORTRAN standard (X3J3/61 FORTREV 75-01-27), there is an interesting statement concerning the use of column 1 for carriage control:

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 4 16
NEW [x] / REPLACEMENT ARTICLE [ ]	ORIGINAL DATE Janaary 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

12.9.5.2.3. Printing of Formatted Records

The transfer of information in a record to certain devices determined by the processor is called "printing." If a formatted record is printed, the first character of the record is not printed. The remaining characters of the record, if any, are printed in one line beginning at the left margin.

12.9.5.2.3d

A PRINT statement does not imply that printing will occur and a write statement does not imply that printing will not occur.

OPERATION ON DOS-15 V3AØØØ, DOS-15 V3BØØØ, RSX PLUS III V1A AND RSX PLUS III V1B

Beginning with the version 44 FORTRAN system (May 1973) a subtle, but far-reaching change was made to BCDIO. The purpose for the change was to make it possible to read and write the same record on mass storage devices using the same FORMAT statement. While this was a noble enough goal, the modifications were not properly researched or documented.

The change implemented was to simply convert a line feed character to a space character on input, before beginning to process the record. This produces an action complementary to that taken on output and does in fact achieve the immediate goal.

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 5 16
NEW [x] / REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

For example:

```
        WRITE(3,1ØØ)A,B,C
    1ØØ FORMAT(1X,3F6.2)
        .
        .
        .
        READ(3,1ØØ)A,B,C
```

The above example will actually write then read a file using the same format.

PROBLEM:

Obviously, making a change such as this one without providing adequate user documentation and conversion aids causes a user compatibility and conversion issue. Aside from this problem, there is another concerning device independence.

Clearly, if BCDIO is going to convert a leading line feed to a space on input conversions, this extra space must be accounted for in the associated FORMAT statement. For purposes of discussion, assume this extra space is handled by a "1X" conversion. Now, an attempt is made to read a record in the following format:

```
XXX.XXXXX.XXXXX.XX
```

The FORMAT statement might be:

```
FORMAT(3F6.2)
```

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 6 16
NEW [x] / REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

If the record is on a mass storage device and the operating system is DOS-15 V3AØØØ, the FORMAT must be:

```
FORMAT(1X,3F6.2)
```

However, to run this program with the card reader as the input device rather than disk, the program may not run as the input device anticipated. Assuming the three values to be converted by the "F" conversion began in column 1 of the input cards, the program will get the wrong input values because there is no leading line feed for BCDIO to map to a space to satisfy the "1X" conversion.

In this manner, the changed BCDIO destroys some previously available measure of device independence at the FORTRAN level. Additionally, since BCDIO was changed to recognize only a line feed on input and convert it to a space, if a record were written to a mass storage device headed by any of the other translated carriage control characters (i.e., FF, DC1, or DLE), these characters are stripped on input, rather than retranslated to a space or some other character. This created a somewhat inconsistent treatment of the situation.

In summary, the conversion created the following problems:

1. Users of earlier systems are now faced with the prospect of converting programs and/or data files.

2. Source program mobility among PDP-15 FORTRAN users is lost, or at least hampered.

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE: 3 / PAGE OF: 7 16
NEW: x / REPLACEMENT ARTICLE:	ORIGINAL DATE: January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

3. Input device independence to a FORTRAN program is sacrificed to a large extent.

4. Using the first character of mass storage device records for a carriage control character is questionable at best and certainly of limited utility.

5. The treatment of the first character of mass storage device records on input is inconsistent and a potential source of confusion.

6. Use of the first character of all records for carriage control seems to increase the effort required when converting a program from another machine to run on a PDP-15.

7. Essentially, all customers who received their PDP-15 within the last two years have written their programs to run with the changed BCDIO.

8. Due to point 7, any further discussion of the particular way BCDIO was changed is now largely academic since an approximately equal portion of the active customer base is affected by changes to either the earlier or later versions of BCDIO.

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 8 16
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

ALTERNATIVES:

Given that the problem deserves some action to completely solve the "same FORMAT" problem and to clear up as many of the attendant problems as possible, the following is a list of alternative solutions.

1. Do nothing. This choice is ruled out by the given assumption that the situation will be improved.

2. Go back to the old way. This option puts all burden of conversion on UNICHANNEL customers. It also required the conversion by UNICHANNEL customers with no incentive; that is, nothing is gained for the effort in conversion.

3. Stay with the new way. Again, this places all the burden on only one group, the older customers. This approach also carries with it the disadvantages outlined in section III.

4. Change BCDIO to convert all translated carriage control characters (20, 21, 12, and 14 OCTAL) back to their respective characters (+, Ø, space, and 1) on input. If one of the above characters is not the first character in the input buffer, then BCDIO will supply a space character at the front of the buffer. This approach will correctly handle the same FORMAT problem initially addressed. At the same time, the device independence issue is resolved.

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 9 16
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

The problems with this are that again, only the older customers are forced to change. Additionally, this would require a "1X" at the start of each FORMAT, which seems a bit contrived.

5. Change BCDIO so it does not perform carriage control translation on output to nonprinter devices, leaving the conversion as is to printer-type devices, and so that on input all carriage control codes (12, 14, 20, 21, etc.) are dropped. This solves the different FORMAT problem and restores device independence. This solution involves conversion also, but it requires everyone to do some amount of work to come up to the new standard.

THE PROPOSED SOLUTION:

It is our opinion that the most desirable solution is the last alternative. It is the most consistent with the spirit of the ANSI standard with other large scale computer systems and with many other DIGITAL systems.

Under the new system, when using a nonprinter device, the same FORMAT statement could be used for both input and output. The first character of the record will not be modified by the object time system. Thus, either of the following FORMAT statements would be acceptable for output:

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 10 16
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

```
100  FORMAT(1X,F6.2)  or  100  FORMAT(F6.2)
```

The only restriction would be that the FORMAT statement used for input must specify the same number of characters as the output statement did (in other words, they must be the same).

In addition, device independence will be maintained, as a leading line feed (produced by input from an old-style disk file, for example) will be simply dropped, making the record appear the same as one from the card reader.

The most important considerations are those concerning the conversion efforts involved. This change can be discussed relative to converting the following classes of items:

1. DOS V1A,V2A programs that both write and read files.
2. DOS V1A,V2A programs that only write files.
3. DOS V1A,V2A programs that only read files.
4. DOS V1A,V2A programs that do no file I/O.
5. DOS V1A,V2A data files.
6. DOS V3 programs that both write and read files.
7. DOS V3 programs that only write files.
8. DOS V3 programs that only read files.
9. DOS V3 programs that do no file I/O.
10. DOS V3 data files.

The following sections discuss the conversion procedure for each of the above cases.

SOFTWARE PRODUCT	VERSION
DOS/XVM	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 11 16
NEW [x] / REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

1. DOS V1A,V2A programs that both read and write a file.

   For these programs, file I/O FORMAT statements are of the following type:

```
     WRITE(1,1ØØ)A                    READ(1,1Ø1)A
1ØØ  FORMAT(1X,F6.2)            1Ø1   FORMAT(F6.2)
```

   The recommended conversion is to eliminate the first character conversion in the FORMAT statement.

```
     WRITE(1,1ØØ)A                    READ(1,1Ø1)
1ØØ  FORMAT(F6.2)               101   FORMAT(F6.2)
```

   Or simply use the READ FORMAT statement for output also.

2. DOS V1A,V2A programs that only write files.

   The some conversion as used in 1 above: Delete the 1X or its equivalent in output FORMAT statements.

3. DOS V1A,V2A programs that only read files.

   No conversion is required.

4. DOS V1A,V2A programs that do not file I/O.

   No conversion is required.

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 12 16
NEW [x] / REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

5. DOS V1A,V2A data files.

   No conversion is required; however, files generated under V2A will have a leading line feed character on each record. While these do no harm being read under either the old or new system, they could be deleted for aesthetic reasons.

6. DOS V3 programs that both write and read a file.

   No conversion is required for these programs, but it is suggested for reasons of device independence and aesthetics. File I/O statement for V3 programs would have the form:

```
     WRITE(1,1ØØ)A                    READ(1,1ØØ)A
1ØØFORMAT(1X,F6.2)
```

   If the 1X (or equivalent) were dropped from all file I/O FORMAT statements, device independence with nonmass storage devices would be restored, as well as saving space on the mass storage media:

```
     WRITE(1,1ØØ)A                    READ(1,1ØØ)A
1ØØ FORMAT(F6.2)
```

7. DOS V3 programs that only write data.

   No conversion is required, but the same considerations apply as in 6 above. Note that a user should generally change all programs if he changes any. Again, this conversion is highly desirable.

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 13 16
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

8. DOS V3 programs that only read files.

   Case 1: Reading old files

   Since a leading line feed is no longer converted to a space on input, these programs must be modified to reflect this change. The recommended conversion will be to delete the 1X or its equivalent from all file input FORMAT statements.

   Case 2: Reading new files

   Since a blank is no longer changed to a line feed on output but is left a blank, there are no changes required. However, it is strongly recommended that all mass storage files no longer allocate the first character for carriage control, so if the programs which wrote the files have been changed as recommended, then the programs which read the files must be changed also.

9. DOS V3 programs which do not file I/O. No conversion is required.

10. DOS V3 data files:

    If the programs are converted as recommended, no change to the old data files is necessary. However, if the carriage control space is kept in all file FORMAT statements, then

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 14 16
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

all old data files will have to have leading line feeds converted to spaces.

The careful reader will have noticed that in all previous systems, it was possible to assign a disk to a FORTRAN logical unit which may have usually been assigned to a line printer. The resulting file could then be PIPed directly to the printer. Under the new system, this would produce a single spaced listing with the ASCII carriage control character being printed in column 1. This is because the ASCII code to internal carriage control code translation is no longer being performed by BCDIO to a mass storage device.

To retain this capability, a new switch is being added to PIP which can be used on a TRANSFER command to perform the standard FORTRAN carriage control character translation.

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 / PAGE OF 15 16
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

A Change in the Handling of Carriage Control Characters on Nonprinter Devices by the FORTRAN Object Time Routine BCDIO

Note that paper tape is being considered as a nonprinter device and, as such, will not have carriage control translation performed by BCDIO. However, some users rely on the off-line listing of paper tapes on an ASR teletype. This requires that each data record is bounded by a Line Feed and Carriage Return. Fortunately, this capability is retained, since the punch handler preceded each record with a Line Feed if it does not already have one. Although this scheme will prevent the correct off-line listing of overprinted, double-spaced, and top-of-form records, these features never worked in the past either, since the special internal carriage control codes were not recognized by the ASR terminal.

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE 3 | PAGE 16 OF 16
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE January 1976

Programming Note to Eliminate Carriage Return

There is a FORTRAN programming technique to effectively eliminate the carriage return following a WRITE to the teletype. Follow the query line with an integer (using A1 format) which is initialized with an altmode (in 7 bit ASCII). If a READ is desired at the end of the line, it must be through a .DAT slot differing from that on which the WRITE was issued (which avoids a reINIT by FIOPS). Also, the READ .DAT slot must have been previously INITed (done, for example, by a REWIND).

```
      DATA IALT/#764ØØØ/
      REWIND 3
      WRITE (4,4ØØ) IALT
4ØØ   FORMAT (1X, 'NUMBER PLEASE:', A1)
      READ (3,) N
      .
      .
      .
      END
```

This results in a FOCAL type read, viz,

NUMBER PLEASE:

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	4	1 1
NEW ☐ REPLACEMENT ARTICLE ☐ 2	ORIGINAL DATE	
	July 1972	

Using System Buffers for FORTRAN OTS Double Buffering

The object of double buffering FORTRAN I/O is increased throughput by increased I/O computation overlap. The price paid for this performance increase is less free memory for user programs.

The memory space required for double buffering is requested from the pool of system buffers via .GTBUF macros. This means that users will generally require more buffers to be allocated than were necessary in older systems.

Different devices are buffered in different ways. First, disks are not double buffered at all, owing to their speed.

Second, one system buffer, usually 500_8 words, is partitioned into as many individual line buffers as possible for all other devices except DECtape and Magnetic tape. These line buffers are partitioned as each device is used, and the length of the line buffer allocated is determined by the maximum buffer size returned by each handler in the .INIT macro.

Due to the complexity of recovering and reallocating space in this system buffer, the device line buffer remains allocated even after it is closed via ENDFILE or CALL CLOSE. When all space in this buffer is allocated, any further devices which are opened are simply not double buffered. No indication of this status is available to the FORTRAN programmer.

Finally, in the case of DECtape and Magnetic tapes, one entire system buffer is allocated via .GTBUF each time a new tape unit is opened. This is of course required due to the long record lengths possible with these two devices. Again, for reasons of simplicity and efficiency these double buffers are not returned when the devices are closed.

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE: 5 / PAGE OF: 1 2
NEW: X / REPLACEMENT ARTICLE:	ORIGINAL DATE: March 1976

Using System Buffers for FORTRAN OTS Double Buffering

As an example of the number of buffers used for a given program, consider a hypothetical program which uses two disk files, one DECtape file, the line printer, and card reader. This program will require a minimum of five buffers if all devices are to be open simultaneously. Recall that one buffer is used by the disk and DECtape handlers for each active file.

There will be an additional buffer for the DECtape double buffers, plus one buffer for the line printer and card reader double buffer for a total of 5 buffers.

Note that if users have written special handlers to be used with FORTRAN I/O statements, the handlers must return a valid maximum buffer size with the .INIT macro. Also, if a user determines the increased overhead in FIOPS and system buffers not to warrant the increased performance, double buffering may be assembled out of the OTS modules by following instructions in the Assembly Parameters section of the XVM/DOS Installation Guide.

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE: 5 / PAGE OF: 2 OF 2
NEW: X / REPLACEMENT ARTICLE:	ORIGINAL DATE: March 1976

Modes of Statement Function Variables

PROBLEM:

IMPLICIT mode declarations of variables appearing in a Statement Function definition fail, i.e., these variables will assume default mode.

SOLUTION:

To circumvent this compiler error, use an explicit mode declaration for such variables.

SOFTWARE PRODUCT		VERSION
XVM/DOS		V1A
COMPONENT	VERSION	EDIT
FORTRAN COMPILER	V1A000	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
F4M, FPF4M	1	1 1
NEW / REPLACEMENT ARTICLE		ORIGINAL DATE
[ ] / [1]		March 1973

Programming to Avoid Incorrect Mode Typing of Functions Declared EXTERNAL

PROBLEM:

The compiler is presently typing as INTEGER any function named in an EXTERNAL statement. This has bad effects only where a call to such a function is generated in the same program in which it is declared EXTERNAL and where the function should not generate an INTEGER result.

RESTRICTION:

Restrict your use of the EXTERNAL statement to naming functions that are used only as subroutine or other function parameters; despite the fact that they are thought to be INTEGER by the main program, the correct address of the function will be passed. However, if a function is involved in some program, do not declare it an EXTERNAL statement in this same program. The function will then retain its EXTERNAL characteristic (a fact that is determined by the compiler by the context in which it is used), will not lose its mode, and can be passed as a subroutine parameter.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN COMPILER	V1A000	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
F4M, FPF4M	2	1 1
NEW / REPLACEMENT ARTICLE	ORIGINAL DATE	
[ ] / [2]	April 1973	

Variables Declared INTEGER are Typed as LOGICAL

PROBLEM:

Variables explicitly declared INTEGER will actually be typed as LOGICAL. This causes problems only in data directed output of such variables.

SOLUTION:

To avoid the problem do not use the explicit INTEGER declaration for variables that will be output with data directed WRITE statements.

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
FORTRAN COMPILER		V1A000	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION		SEQUENCE	PAGE OF
F4M, FPF4M		3	1 1
NEW	REPLACEMENT ARTICLE	ORIGINAL DATE	
	2	December 1973	

SOFTWARE DISPATCH

XVM/DOS

DOUBLE INTEGER Literals Incorrectly Defined

PROBLEM:

Large magnitude negative DOUBLE INTEGER literals are being incorrectly defined. This problem will exist, for example, in a DATA statement initialization of a DOUBLE INTEGER variable.

SOLUTION:

To avoid the problem, define the literal as a positive value and negate in an assignment statement, as follows:

```
DOUBLE INTEGER  DI
DATA   DI/1234567/
   .
   .
   .
DI = -DI
   .
   .
   .
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN COMPILER	V1A000	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
F4M,FPF4M	4	1 1
NEW ☐ REPLACEMENT ARTICLE 2	ORIGINAL DATE	
	December 1973	

Transferring Control to a FORMAT Statement

PROBLEM:

The Fortran compiler fails to flag as an error GOTOs to FORMAT statements.

NOTE:

When a FORMAT statement is compiled, the binary output begins with a JMP to the location immediately following the code produced from the FORMAT statement. Control may be passed to a FORMAT statement in a variety of ways. Consider the following two examples:

```
      J=Z                          J=Z    .
      GOTO 1ØØ                     I=I+1
       .                     1ØØ FORMAT (....)
       .                         J=5
       .
1ØØ FORMAT(....)
      J=5
```

In either of these cases, the JMP instruction beginning the expansion of the FORMAT statement transfers control to the expansion of the line J=5. Hence, whenever control is transferred to a FORMAT statement, the FORMAT statement behaves similarly to a numbered CONTINUE statement. This problem is not something which should be of major concern to users of FORTRAN; however, it is something of which they should be aware.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN COMPILER	V1A000	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
F4M, FPF4M	5	1 1
NEW / REPLACEMENT ARTICLE	ORIGINAL DATE	
[ ] / 1	April 1974	

SOFTWARE DISPATCH

XVM/DOS

Restriction of Variable Names in Named COMMON

PROBLEM:

The compiler is unable to differentiate a symbol table entry for a named COMMON block from a variable of the same name in the same COMMON block.

Thus,

```
COMMON  BLKS/BLKS,.../
```

will generate >Ø4C< errors.

RESTRICTION:

Since the name of a named COMMON block is not actually used in a FORTRAN program, simply change the COMMON block name to a unique symbol. This same change must also be made to every subprogram which references the named COMMON block.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN COMPILER	V1A000	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
F4M, FPF4M	6	1 1
NEW ☐ REPLACEMENT ARTICLE 1	ORIGINAL DATE	
	July 1974	

Displaced Error Reports

PROBLEM:

Error messages may be printed several lines below the line in which the compiler detected the error.

The problem results from the fact that comment lines are not detected by the main logic of the compiler. Thus, error reports will be displaced by the number of comment lines following the line in error.

RESTRICTION:

Any errors detected by the compiler for a given line will be printed after any comment lines which may follow the erroneous line.

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
FORTRAN COMPILER		V1A000	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION		SEQUENCE	PAGE OF
F4M, FPF4M		7	1 OF 1
NEW	REPLACEMENT ARTICLE	ORIGINAL DATE	
	1	July 1974	

End of Compilation Message

TEMPORARY RESTRICTION:

The FORTRAN compiler outputs the End of Compilation message twice if .DAT -12 is assigned to CM or TT.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN COMPILER	V1A000	069
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
F4M, FPF4M	8	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE March 1976	

Programming Note Regarding Argument Address GET Routine, .DA

Because of the multiple entry feature in FORTRAN XVM, the argument address fetch subroutine .DA does "double indirection" when both fetching and storing addresses passed from a main program to a subroutine. Users should be aware that if bit zero in the storage address cell of the subroutine is set, another level of indirection is performed. Thus, MACRO subroutines using the argument address cells as scratch cells may cease to function under FORTRAN XVM.

Example:

```
        .TITLE  MACRO SUBROUTINE CALLED FROM FORTRAN
        .GLOBAL SUBR,.DA
SUBR    XX              /ENTRY POINT
        JMS*    .DA     /CALL ARG. GET ROUTINE
        JMP     .+4     /EXPECTING 3 ARGUMENTS
A       .DSA            /ADDR. OF FIRST
B       .DSA            /ADDR. OF SECOND
C       .DSA            /ADDR. OF THIRD
```

After the return from .DA, locations A, B, and C contain the addresses of the arguments to be passed. It is a common practice to then execute a statement of the following form:

```
        LAC*    A       /GET THE ARGUMENT ITSELF
        DAC     A       /AND PUT IT IN POINTER WORD.
```

If the value of the first argument happened to be negative, then bit Ø would be a 1. The next time SUBR is called, .DA would interpret the fact that bit Ø of the pointer word A is a 1 to mean that the address of the argument should not be put in the pointer word itself, but that the address is to be put in the word whose address is in the pointer word. As a result, this programming practice will cause unpredictable results.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN OTS	N/A	11
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
.DA	1	1 1
NEW ☐ REPLACEMENT ARTICLE 1	ORIGINAL DATE	
	July 1972	

SOFTWARE DISPATCH

XVM/DOS

Numerical Restrictions in AMOD and DMOD

PROBLEM:

In the existing FORTRAN documentation there is stated a restriction on the range of values that AMOD and DMOD can accommodate. In either of these, for a call of the form AMOD (ARG1,ARG2), one of the internal operations is computing the result of ARG1/ARG2, and then converting this result to a single precision integer. This latter operation limits the useful range of ARG1/ARG2 to less than 2^{17}, i.e., less than 131Ø72. When this condition is not met, an OTS 11 error occurs, the program continues, and the results of AMOD or DMOD are not generally predictable.

SOLUTION:

This restriction can be relieved somewhat by the following example. (This is not to be construed to be a supported software feature of the FORTRAN Object time system). Considering the case of DMOD, code and compile:

```
DOUBLE PRECISION FUNCTION DMOD (ARG1,ARG2)
DOUBLE PRECISION ARG1,ARG2, D
DOUBLE INTEGER J
J = ARG1/ARG2
D = J
DMOD = ARG1 - D*ARG2
RETURN
END
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN OTS	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
AMOD, DMOD	2	1 2
NEW ☐	REPLACEMENT ARTICLE 1	ORIGINAL DATE December 1972

Numerical Restrictions in AMOD and DMOD

Explicitly state this program file name in the loader command string, and it will be loaded instead of the FORTRAN Library routine DMOD. It extends the largest useful value of ARG1/ARG2 to be less than 34,359,738,368.

An equivalent routine may be written to replace AMOD by replacing "DOUBLE PRECISION" with "REAL" and "DMOD" with "AMOD" at all locations in which each appears.

IOPS31 Errors When Referencing the Last Element of an Array

PROBLEM:

Under certain circumstances an IOPS31 error can occur when referencing the last element of an array which is in a common block located in extended memory, if that element is the last word in memory.

DISPOSITION:

A temporary solution is to either lower the logical memory size (MEMSIZ command) or add a dummy variable at the end of the common block in question.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
FORTRAN OTS	N/A	052
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
BCDIO	3	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

First Character of Each Record Sometimes Truncated On Mass Storage Files

PROBLEM:

If a file is created on a mass storage file, then eventually reopend for output (to be rewritten) in the same core load, the first character of each record may be truncated as it is written into the new file. This problem pertains only to output files; a file may be reopened for input with no difficulty.

SOLUTION:

Due to a coding error in FIOPS, not all of the device information is retained in some internal tables following a CLOSE. These source changes to FIOPS, making the edit number Ø43, correct this problem.

```
/EDIT #042  24-NOV-75
/       ADD A NEW ASSEMBLY PARAMETER, '&NODBL' TO SUPPRESS
/       DOUBLE BUFFERING, AND MAKE DOUBLE BUFFER THE DEFAULT.
/
/EDIT #043  21-JAN-76
/       FIX BUG IN .DSK ROUTINE WHICH CAUSED THE 'NON-PRINTER
/       DEVICE' BIT TO GET LOST IF A FILE WAS RE-OPENED FOR
/       OUTPUT.
/
/
/
```

(I)

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN OTS	N/S
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE / PAGE OF
FIOPS	4 / 1 OF 3
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE
	March 1976

First Character of Each Record Sometimes Truncated On Mass Storage Files

```
          SKP
FCT1      0
         CAL     0          /.INIT IOPS ROUTINE.
         .DSA    1
         .DSA    FCFRR
.FC6          .DSA    0
I        LAC      SLOT     /(RKB-043) RE-BUILD UTBITS
         JMS      .DSK     /(RKB-043) TO REFLECT 'NON-PRINTER' DEVICE
         .IFDEF %DBL
/*****************************************************************************
                STL
                LAC .FC6   /TEST FOR TTY

                SAD (42   /LINE BUFFER SIZE OF TTY
                LRS 1   /IF YES, SET BIT 1
                LRS 1   /SET BIT 0 TO INDICATE 1ST TIME
                AND (700000
                DAC RW   /OR IN LATER
        /
                LAC SLOT
                TAD (.RTABL-1
                DAC T.1
        FC..5   LAC* T.1
                SAD (777777)
                JMP FC..4
                SZA
                JMP FC..2
                .ENDC
I       /(RKB-043)
        /(RKB-043) THE FOLLOWING FOUR LINES WERE DELETED:
        /       .IFDEF %DOS15
C       /       LAC SLOT
        /       JMS .DSK          /IF DEVICE IS A DISK, AC = -1 ON RETURN
        /       .ENDC
I               .IFDEF %DBL
                LAC      .DSKF    /(RKB-043) GET INICATOR FROM .DSK
                SZA!CLC
                JMP FC..2         /IT'S  A DISK; INSTAL A -1
                LAW -375   /TEST FOR MS
                TAD .FC6
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	
FORTRAN OTS	N/A	
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
FIOPS	4	2 3
NEW: X / REPLACEMENT ARTICLE:	ORIGINAL DATE	
	March 1976	

First Character of Each Record Somtimes Truncated On Mass Storage Files

```
        SMA     /MS?
        JMP FC..6       /YES; TRY FOR A FULL BUFFER
        LAC BUFEND
        SZA     /IS THERE A NON-MS BUFFER YET?
        JMP FC..3       /YES; TRY TO PARTITION IT  (FURTHER)

FC11    SAD (777777)    /CLEAR AC IF NOT A DSK
        CLA
        DAC     .DSKF
        LAC     FC9     /(RKB-043) GET THE DAT SLOT
        TAD     (STAB-1) /(RKB-043) FIND THE STAB ENTRY
        DAC     FC4     /(RKB-043) USE THIS WORD TEMORARILY
        LAC*    FC4     /(RKB-043) IS THIS SLOT BEING RE-OPENED?
        SZA             /(RKB-043) SKIP IF YES
        JMP     FC11.2  /(RKB-043) NO, ITS ALREADY OPEN
        LAW     -375    /(RKB-043) NEED TO RE-DETERMINE
        TAD     .FC6    /(RKB-043) DEVICE TYPE
        SMA             /(RKB-043) SKIP IF NOT DT, MT, RK, RP, DK
        JMP     FC11.1  /(RKB-043) ITS A DISK OR TAPE UNIT
        AAC     375-62  /(RKB-043) IF BUF SIZE IS 62, ITS PP OR PR
        SNA!CLA         /(RKB-043) SKIP IF NOT PP OR PR
FC11.1  LAC     (400000) /(RKB-043) SET THE 'NON-PRINTER' BIT
        DAC     010ITS  /(RKB-043) (I KNOW NO OTHER BITS ARE SET)
FC11.2  LAC     .DSKF   /(RKB-043) RECOVER OLD DISK FLAG
        JMP* .DSK
/DSK TABLE 200000 OR 300000 ENTRY INDICATES SLOT .FSTAT'D AND
/DISK PRESENT.  777777 ENTRY INDICATES SLOT .FSTAT'D AND NO
/DISK PRESENT
DSKTBL=.
        .REPT DKTBSZ
        0
*R
*R
```

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
FORTRAN OTS	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE / PAGE OF
FIOPS	4 / 3 3
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE
	March 1976

Programming Note

Radix 50_8 is a technique used by the MACRO Assembler and the FORTRAN Compiler to condense the binary representation of symbolic names in symbol tables. It is described in Appendix C of the Linking Loader Utility Manual.

The Radix 50_8 table on the following page should be added to the description in the Linking Loader Manual.

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
LINKING LOADER		N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION		SEQUENCE	PAGE OF
LINKING LOADER XVM UTILITY PROGRAM DEC-XV-ULLUS-A-D		1	1 2
NEW	REPLACEMENT ARTICLE	ORIGINAL DATE	
	1	October 1973	

RADIX 50_8 VALUES

X--		-X-		--X	
A	003100	A	000050	A	000001
B	006200	B	000120	B	000002
C	011300	C	000170	C	000003
D	014400	D	000240	D	000004
E	017500	E	000310	E	000005
F	022600	F	000360	F	000006
G	025700	G	000430	G	000007
H	031000	H	000500	H	000010
I	034100	I	000550	I	000011
J	037200	J	000620	J	000012
K	042300	K	000670	K	000013
L	045400	L	000740	L	000014
M	050500	M	001010	M	000015
N	053600	N	001060	N	000016
O	056700	O	001130	O	000017
P	062000	P	001200	P	000020
Q	065100	Q	001250	Q	000021
R	070200	R	001320	R	000022
S	073300	S	001370	S	000023
T	076400	T	001440	T	000024
U	101500	U	001510	U	000025
V	104600	V	001560	V	000026
W	107700	W	001630	W	000027
X	113000	X	001700	X	000030
Y	116100	Y	001750	Y	000031
Z	121200	Z	002020	Z	000032
%	124300	%	002070	%	000033
.	127400	.	002140	.	000034
0	132500	0	002210	0	000035
1	135600	1	002260	1	000036
2	140700	2	002330	2	000037
3	144000	3	002400	3	000040
4	147100	4	002450	4	000041
5	152200	5	002520	5	000042
6	155300	6	002570	6	000043
7	160400	7	002640	7	000044
8	163500	8	002710	8	000045
9	166600	9	002760	9	000046
#	171700	#	003030	#	000047

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
LINKING LOADER	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION LINKING LOADER XVM UTILITY PROGRAM DEC-XV-ULLUS-A-D	SEQUENCE 1	PAGE OF 2 2
NEW ☐ REPLACEMENT ARTICLE 1	ORIGINAL DATE October 1973	

Incomplete Loading of Subroutines with Multiple Entry Points

PROBLEM:

In subroutines with multiple entry points, if the mutiple entry point .GLOBAL definitions are interspersed with executable code (unavoidable in FORTRAN routines with multiple entry points), the following situation exists.

If the routine calling the subroutine does not reference the first entry point, the Loader will incompletely load the subroutine. The Loader loads from the first referenced entry point to the end of the program unit.

RESTRICTION:

A dummy reference to the first entry point must be included to cause complete loading by the Loader.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A000	
COMPONENT	VERSION	EDIT
LINKING LOADER	V1A000	144
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	2	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

BATCH .DAT Slots

PROBLEM:

When the BATCH device is used by a user program but is not assigned to any of the Loader .DAT slots (-1, -4 or 5), two copies of the BATCH device handler will be loaded. The first reference to the second of these handlers will result in a fatal system error.

SOLUTION:

This situation can be avoided by either assigning to BATCH a device not used by the user program or by assigning the BATCH device to one of the Loader .DAT slots.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
LINKING LOADER	V1A000	
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	3	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

BATCH I/O Device Use

PROBLEM:

When a user program that is loaded via GLOAD within a BATCH stream uses the locations between those pointed to by .SCOM+2 and .SCOM+3 but does not use the BATCH device, i.e., no .IODEV was issued, a fatal error can occur upon return to BATCH.

SOLUTION:

This error can be avoided by loading, along with the user programs, a dummy program containing the instructions:

```
.IODEV n
.END
```

where n is a .DAT slot previously assigned to the BATCH device.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
LINKING LOADER	V1A000	144
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	4	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

System Build Leaves Erroneous .DAT Assignment

PROBLEM:

Because XVM/DOS systems generated by the BUILD program are left with .DAT-4 ASSIGNed to a default UIC of SYS, attempts to load programs are directed to either the PAG or BNK UFDs. ince user programs are normally not kept in either the NAME or BNK UFDs, attempted program loading results in an IOPS13 error.

SOLUTION:

SGEN the system to modify .DAT-4 as follows:

```
>A SY <UIC> -4
```

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	VERSION
LINKING LOADER	V1A000
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE: 5 / PAGE OF: 1 1
NEW: X / REPLACEMENT ARTICLE:	ORIGINAL DATE: March 1976

Programming Note on Use of T Switch

PROBLEM:

When using the T-switch, a discrepancy in the page count may occur where the user has a .LTORG in his program with many forward referencing literals. The forward referencing literals waste space and should be removed--one location is reserved per forward reference. The page count discrepancy occurs because the page count is adjusted during pass one to reflect the total literal count. If the count is smaller after pass two (forward references have been defined), the page count is likely to be inaccurate.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
MACRO XVM	V1A	118
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	1	1 OF 1
NEW ☐ REPLACEMENT ARTICLE 2	ORIGINAL DATE	
	Pre-1973	

MACRO E Switch Does Not Give Correct Lines in Console Device Under BOSS

PROBLEM:

When using the MACRO E switch and running under BOSS, the error lines printed on the console terminal are not correct. They are most often a combination of several legitimate error lines.

SOLUTION:

Avoid using the MACRO E switch while operating under BOSS. Since the error lines are correctly listed on the printer, this should only be an inconvenience.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A000	
COMPONENT	VERSION	EDIT
MACRO XVM	V1A000	118
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	2	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

Edits to Bring MACRO to XVM/RSX Release Level

PROBLEM:

Because of the time between the releases of XVM/DOS and XVM/RSX, the sources of MACRO released with each system no longer match.

SOLUTION:

The following source changes to MACR15 118, which pertain only to the RSX version of MACRO, should nevertheless be made to the DOS version to keep both versions identical. These edits produce MACR15 120.

```
/ 116   25/NOV/75 (JMW)        .WAIT AFTER 'E' OUTPUT.
/ 117   26/NOV/75 (JMW)        CORRECT A RSX ERROR MESSAGE.
/ 118   04/DEC/75 (JMW)        ANOTHER GARBAGE COLLECTOR FIX!
/ 119   17/DEC/75 (JMW)        2 RSX CORRECTIONS.
/ 120   30/DEC/75 (JMW)        ANOTHER RSX CORRECTION, TO ALLOW BATCHING, AS BEFORE.
/
/
/
/THERE ARE SEVERAL ASSEMBLY PARAMETERS FOR MACRO:
/       1. NONE YIELDS THE STANDARD SYSTEM VERSION!
/       2. %PATCH CAN BE DEFINED TO A DESIRED PATCH SIZE (DEFAULT = 40)!
/       3. %BIN YIELDS A VERSION TO BE RUN WITH DDT (CREF BECOMES A SUBROUTINE)!
/       4. %DEBUG INCLUDES DEBUGGING CODE FOR VERIFICATION OF MACRO DEFINITION TABLE.
/       5. %DMACRO WILL INCLUDE THE ABS LOADER WHICH LOADS AT 17700 RATHER THAN 17720.
/       6. %RSX PRODUCES THE RSX VERSION OF MACRO.  NOTE THAT THERE ARE SEVERAL
```

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
MACRO		N/A	119
SUBPROGRAM OR ADDITIONAL INFORMATION		SEQUENCE	PAGE OF
		3	1 4
NEW	REPLACEMENT ARTICLE	ORIGINAL DATE	
X		March 1976	

Edits to Bring MACRO to XVM/RSX Release Level

```
NOTIT=.
        .ENDC
        LAC YSWCH       /IF Y AND L ARE USED FORCE 'N'.
        DZM SAVEF
        SNA
        JMP .+4
        LAC ALAPTN
        SZA
        SET NSWCH
        .IFDEF %RSY
        DZM POTSW       /SET NO EXIT ON CR = ALTMODE SWITCH.
        LAC TSWCH       /SEE IF NEED TO ATTACH.
        SNA
        JMP .+3         /NO = DON'T ATTACH.
        LAC (DAT12      /YES = ATTACH OUTPUT LISTING DEVICE.
        JMS ATACH
        .ENDC
        .IFUND %RSY
        LAW -3
        DAC FLDSW
        .ENDC
(C)---> .IFDEF %RSY             /(JMW:120) THE % HAD BEEN LEFT OUT AND BATCHING DIDN'T WORK.
        DZM FLDSW       /CLEAR THE FILES COUNT.
        LAC PSWCH       /PARAMETER FILES?
        SNA
        JMP NPSW        /NO = DON'T COUNT IT.
        LAC (DAT10      /DO A HINF ON THIS LUN TO SEE IF IT IS
        JMS HINFRQ      / FILE STRUCTURED.
        SZA             /0= NON DIRECTORIED.
        IDX FLDSW       /FILE STRUCTURED = COUNT THE FILE.
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
MACRO	N/A	119
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	3	2 4
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

Edits to Bring MACRO to XVM/RSX Release Level

```
NXTCHR  JMS GETCMD
        SAD L40
        JMP NXTCHR              /IGNORE LEADING BLANKS/
                                /(JMW:119) SINCE RSX CAN BATCH ASSEMBLIES,
(I)---->.IFUND XPSW             /V7C,V7D PATCH
        SAD     COMMA           /IGNORE COMMAS IN BOSS  MODE
        JMP     NXTCHR          /(JMW:119)  NEED THE COMMAS.
(I)---->.ENDC
        DZM PRVCHR          /PSEUDO COUNTER FOR SPACES.
        SKP
CMDSUB  JMS GETCMD          /GET A CHAR AND RETURN IF NOT DELIMITED.
        SAD L40
        JMP CMDEXT
        SAD KOMMA
        JMP NAMDUN
        JMS SIXP            /DONT PACK AFTER 9 CHARS ARE IN.
        JMP CMDSUB          /GET NEXT CHAR
CMDEXT  SAD PRVCHR
        JMP PARALL          /LINE DELIMITED FIND CR OR ALT.
        DAC PRVCHR
        LAW -3
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
MACRO	N/A	119
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	3	3 4
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

Edits to Bring MACRO to XVM/RSX Release Level

```
CREF    .SIXBT /CREF/
        .ENDC
        .IFDEF %RIN     /(JMW:106) MAKE CREF A SUBROUTINE FOR MACRO UNDER DDT.
        JMS* CREF       /(JMW:106)
        JMP PASS1       /)JMW:106)
        .ENDC
        .ENDC
        .IFDEF %RSX
        JMS WAIT12      /WAIT FOR LISTING DEVICE.
        LAC FILNMS      /MOVE THE FILE NAMES FOR CREF INTO THE RESIDENT AREA.
        AAC -1
        DAC* L11
        LAC .SIXBT      /MOVE BOTH NAMES.
        JMS SET10
        LAW -6
        JMS MOVAUT
        LAC (FSWCH      /MOVE THE REST OF THE PARAMETERS.
        JMS SET10
        LAW -12         /(JMW:113)
        JMS MOVAUT
        JMP* MACR1S     /RETURN TO RESIDENT CODE TO DISPATCH TO CREF.
        .ENDC
        .TITLE  .END MESSAGES AND UTILITIES
        .IFDEF %RSX
(C) ->  LRCR-ERRLIN+2/1000+2+2  /(JMW:119)
        0
        .ENDC
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
MACRO	N/A	119
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
	3	4 4
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

Teletype Handler .INIT Function Limitations

PROBLEM:

A .INIT issued to the teletype handler will not cancel a .READ/.WRITE which is in progress.

The reason for this problem is that after the CAL pointer has been saved and the argument pointer incremented, the I/O underway switch is tested and, if set, the program loops back to the CAL.

SOLUTION:

.SCOM location SC.CTT contains an instruction to clear the teletype busy switch. IF a program desires to abort the teletype I/O, it should execute the following:

```
SC.CTT=135
XCT* (SC.CTT)
```

FOLLOWING this perform a .INIT to the handler.

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
MONITOR		V1A	213
SUBPROGRAM OR ADDITIONAL INFORMATION		SEQUENCE	PAGE OF
RESMON		1	1 1
NEW	REPLACEMENT ARTICLE	ORIGINAL DATE	
	1	October 1973	

.DAT -12 Set Up

PROBLEM:

While exiting a CUSP and returning to the nonresident Monitor there exists a period of time when typing ↑C will cause .DAT -12 to be set up incorrectly. This problem is seen as a NON-EXISTENT HANDLER NUMBER message when a REQUEST command is issued. If a program tries to use .DAT -12 while in this condition, the following situations arise.

If LP OFF, references to .DAT -12 go to the teletype. If LP ON, unpredictable results occur, such as IOPSØ. The critical period exists from the issuance of the ↑C or .EXIT until the nonresident Monitor starts typing the XVM/DOS V1AØØØ message.

RESTRICTION:

This problem in the resident monitor will not be fixed. To clear up the problem with .DAT -12, do anything which restores the default .DAT settings. This occurs whenever exiting from a program with KEEP OFF or after typing LOGIN or LOGOUT commands.

For example:

```
$K OFF ↵
$PIP   ↵          /Type any Cusp name
↑C                /Pause one second, then type ↑C
XVM/DOS V1AØØØ    /Now .DAT -12 is set up correctly
$
```

The problem can also be corrected by performing an ASSIGN to .DAT-12.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A000	
COMPONENT	VERSION	EDIT
MONITOR	V1A	213
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
RESMON	2	1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	March 1976	

MICLOG Before Patching on Disk

PROBLEM:

The PATCH manual does not mention the need under XVM/DOS to MICLOG before patching on disk.

SOLUTION:

Make the following changes to the manual.

1. Replace the first sentence of Section 6.2.1 ".DAT Slot Assignments" on page 6-1 with:

   When operating in XVM/DOS and about to make patches to a disk, first log in under the Monitor ID Code (MICLOG). Then, for all operating systems, make the .DAT slot assignments in Table 6-1 prior to loading the PATCH program.

2. Add the following error message to Table 6-4 on page 6-6:

   .DAT-14 NOT PATCHABLE

   This means that the user failed to MICLOG prior to loading PATCH in a XVM/DOS system.

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
PATCH	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION PATCH XVM UTILITY PROGRAM DEC-XV-UPUMA-A-D	SEQUENCE 1	PAGE OF 1 1
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE August 1974	

FORTRAN Programs Cannot be Made System Programs

PROBLEM:

Section 3 of the PATCH manual describes how relocatable user programs (those normally loaded and executed by the Linking Loader) can be converted into SYS (System) programs. This is a useful capability because it allows programs to be called and executed by a direct command to the Monitor, which decreases loading time and teletype interaction. However, the manual does not explicitly say that all FORTRAN programs (or ALGOL for that matter) are excluded. This fact is implicit in the restriction of Section 3.2.2. that the binary not contain external .GLOBL references.

SOLUTION:

1. Add text to line 1 of Section 3.1 so that the line reads:

   A PATCH load function enables the user to convert a stand-alone relocatable file (which excludes FORTRAN and ALGOL programs as explained in Section 3.2.2).

2. Add text to Section 3.2.2, item 2 on the sixth line from the bottom of page 3-2 so that it reads:

   not contain external .GLOBL references. This excludes all programs which must be linked to other binary routines. FORTRAN and ALGOL language programs require linkage to external routines and thus cannot be made into SYS files.

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
PATCH		N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION PATCH XVM UTILITY PROGRAM DEC-XV-UPUMA-A-D		SEQUENCE 2	PAGE 1 OF 1
NEW [ ]	REPLACEMENT ARTICLE [1]	ORIGINAL DATE June 1974	

IOPSØ after (H) Copy to System Disk

PROBLEM:

When an H mode copy is put onto the system disk (DOØ or RKØ), an IOPSØ will sometimes occur at the end of the copy. This occurs only when the new system copied onto the disk is different than the one that originally resided on the disk.

SOLUTION:

The copy has been successfully completed by the time the error occurs. Therefore the user need only reboot the system to continue.

This problem has been fixed in the PIP in the XVM/DOS V1A system by forcing an .EXIT to the monitor after an H mode copy to the system disk. Because of the complexity of a binary patch (the patch is in the part of PIP overlaid during the copy), prior versions of PIP will not be modified.

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	**VERSION**
PIP	V1A
SUBPROGRAM OR ADDITIONAL INFORMATION	**SEQUENCE** 1 / **PAGE** 1 **OF** 1
NEW [X] **REPLACEMENT ARTICLE** [ ]	**ORIGINAL DATE** February 1976

Conversion from ADSS-15: Programs May Not Fit

PROBLEM:

When updating ADSS-15 systems to XVM/DOS, sometimes the user programs do not fit. This symptom has several possible causes:

1. The FORTRAN OTS library has grown in average size of routine by about 1Ø-15%.

2 There are a number of core-consuming monitor environmental considerations

- VT ON
- Size of monitor patch area
- Size of buffer pool
- Size of buffers
- Size of .DAT table
- UNICHANNEL TCB Buffers

which are not present in ADSS-15 but can help to exhaust core quickly if not handled properly.

3. The resident monitor itself (RESMON) has grown.

Sometimes this size increase is just enough to force a device handler to be loaded in the second memory field (4K page) instead of the first, leaving a non-obvious hole in low core.

SOLUTION:

If this occurs, enable BANK mode operation since it ignores 4K boundaries and loads to 8K bounds.

SOFTWARE PRODUCT		VERSION	
XVM/DOS		V1A	
COMPONENT		VERSION	EDIT
SYSTEM		N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION		SEQUENCE	PAGE OF
		1	1 1
NEW	REPLACEMENT ARTICLE	ORIGINAL DATE	
	1	Pre-1973	

Using API Level 4

PROBLEM:

There is a potential timing problem when more than one routine in the system is using API software level 4. Since the standard XVM/DOS disk handler uses level 4, users must be careful to code any of their routines which use level 4 to avoid such timing problems.

SOLUTION:

The following code illustrates how to set up level 4 requests. The essential point is that no interrupts be allowed while the set up is being done.

SETUP AND USE OF API LEVEL 4

```
/
API.R4=4Ø4ØØØ
SC.LV4=.SCOM+12
.INH  =7Ø5522
.ENB  =7Ø5521
/
/MAKE THE LEVEL 4 REQUEST
/
    .INH            /ALLOW NO INTERRUPTS
    RPL
    AND  (API.R4)
    DAC  SAVREQ     /SAVE ANY PREVIOUS LEVEL 4 REQUEST
    LAC* (SC.LV4)
    DAC  SAVVCT
    LAC  (API.R4)
    ISA             /MAKE LEVEL 4 REQUEST
    LAC  (LV4INT)   /ADDRESS OF THIS PROGRAM'S LEVEL 4
    .ENB
    DAC* (SC.LV4)   /PROCESSOR GOES INTO .SCOM+12
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
SYSTEM	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
API LEVEL 4 SOFTWARE	2	1 2
NEW [x] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	October 1975	

Using API Level 4

```
    :
/INTERRUPT PROCESSOR
/
LV4INT  Ø
    .INH
    DAC SAVAC
    LAC SAVREQ      /MAKE PREVIOUS LEVEL 4 REQUEST
    ISA
    LAC SAVVCT
    .ENB
    DAC* (SC.LV4)   /RESET.SCOM+12
    :
    LAC SAVAC       /RESTORE THE AC
    DBR             /AND DEBREAK
    JMP* LV4INT
/
SAVREQ  Ø           /SAVE ANY PREVIOUS LEVEL 4 REQUEST
SAVVCT  Ø           /AND VECTOR ADDRESS
SAVAC   Ø           /SAVE AC ON LEVEL 4 INTERRUPT
```

SOFTWARE PRODUCT	VERSION	
XVM/DOS	V1A	
COMPONENT	VERSION	EDIT
SYSTEM	N/A	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	SEQUENCE	PAGE OF
API LEVEL 4 SOFTWARE	2	2 2
NEW [X] REPLACEMENT ARTICLE [ ]	ORIGINAL DATE	
	October 1975	

System Manual Not Complete

PROBLEM:

The XVM/DOS System Manual (DEC-XV-ODSAA-A-D) is incomplete in that Chapter 9 is not in the manual although it is referenced within the manual.

SOLUTION:

Chapter 9 will be released at a later date as an addendum to the manual.

SOFTWARE PRODUCT	VERSION
XVM/DOS	V1A
COMPONENT	**VERSION**
SYSTEM	N/A
SUBPROGRAM OR ADDITIONAL INFORMATION	**SEQUENCE** / **PAGE OF**
XVM/DOS SYSTEM MANUAL DEC-XV-ODSAA-A-D	3 / 1 1
NEW [X] **REPLACEMENT ARTICLE** [ ]	**ORIGINAL DATE** March 1976

SOFTWARE PROBLEMS OR ENHANCEMENTS

Questions, problems, and enhancements to Digital software should be reported on a Software Performance Report (SPR) form and mailed to the SPR Center at one of the following Digital Offices: (SPR forms are available from the SPR Center.)

Areas Covered	SPR Center
Australia/New Zealand	Digital Equipment Australia Pty. Ltd. 123-125 Willoughby Road, P.O. Box 491 Crows Nest New South Wales, Australia 2065
Brazil	Digital Equipment Comercio E Industria LTDA Rua Batatais, 429 (Esq. Al. Campinas) 01423-Jardim Paulista São Paulo-SP-Brazil
Canada	Digital Equipment of Canada, Ltd. Software Services P.O. Box 11500, K2H 8K8 Ottawa, Ontario, Canada
Caribbean	Digital Equipment Latin America, Inc. 407 del Parque Street Santurce, Puerto Rico 00912
France	Digital Equipment France 18, rue Saarinen Centre Silic - CIDEX L225 F-94533 Rungis, France
Israel	DEC-sys Computers Ltd. 7 Habakuk Street IL-Tel Aviv 63505, Israel
Italy	Digital Equipment S.P.A. Corso Garibaldi 49 I-20121 Milano, Italy
Japan	Digital Equipment Corp. Intl. Kowa Building 25 (3d Floor) 8-7 Sunban-Cho Chiyoda-ku, Tokyo 102, Japan
Mexico	Equipo Digital, S.A. de C.V. 109 Concepcion Beistegui Mexico 12, D.F.
The Netherlands	Digital Equipment B.V. Kaap Hoorndreef 38, P.O. Box 9064 NL-Utrecht - Overvecht, The Netherlands
Nordic	Digital Equipment AB Englundavägen 7 S-17141 Solna Sweden
Switzerland Spain Portugal Greece Bulgaria Romania Yugoslavia	Digital Equipment Corp. SA 20, Quai Ernest Ansermet Case Postale 23, CH-1211 Geneva 8 Switzerland
United Kingdom	Digital Equipment Co. Ltd. Fountain House, Butts Centre GB-Reading RG1 7QN, England
West Germany Austria East Germany Russia Hungary Poland Czechoslovakia	Digital Equipment GmbH D-8000 München 40 Wallensteinplatz 2 West Germany
United States; remainder of Far East, Middle East, Africa, Latin America	Software Communications P.O. Box F Maynard, MA 01754

DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754
European Headquarters: 81 route de l'Aire, 1211 Geneva 26. Switzerland
Digital Equipment of Canada Ltd., P.O. Box 11500 Ottawa, Ontario K2H8K8.

DIGITAL EQUIPMENT CORPORATION, Component Group Headquarters: 1 Iron Way, Marlborough, Mass. 01752, Telephone: (617) 481-7400

DIGITAL EQUIPMENT CORPORATION, Corporate Headquarters: Maynard, Massachusetts 01754, Telephone: (617) 897-5111
SALES AND SERVICE OFFICES
DOMESTIC — ARIZONA, Phoenix and Tucson • CALIFORNIA, Los Angeles, Monrovia, Oakland, Ridgecrest, San Diego, San Francisco (Mountain View), Santa Ana, Sunnyvale, and Woodland Hills • COLORADO, Englewood • CONNECTICUT, Fairfield and Meriden • DISTRICT OF COLUMBIA, Washington (Latham, Md.) • FLORIDA, Orlando • GEORGIA, Atlanta • ILLINOIS, Chicago (Rolling Meadows) • INDIANA, Indianapolis • IOWA, Bettendorf • KENTUCKY, Louisville • LOUISIANA, Metairie (New Orleans) • MASSACHUSETTS, Marlborough and Waltham • MICHIGAN, Detroit (Farmington Hills) • MINNESOTA, Minneapolis • MISSOURI, Kansas City and St. Louis • NEW HAMPSHIRE, Manchester • NEW JERSEY, Fairfield, Metuchen and Princeton • NEW MEXICO, Albuquerque • NEW YORK, Albany, Huntington Station, Manhattan, Rochester and Syracuse • NORTH CAROLINA, Durham/Chapel Hill • OHIO, Cleveland, Columbus and Dayton • OKLAHOMA, Tulsa • OREGON, Portland • PENNSYLVANIA, Philadelphia (Bluebell) and Pittsburgh • TENNESSEE, Knoxville • TEXAS, Austin, Dallas and Houston • UTAH, Salt Lake City • WASHINGTON, Bellevue • WISCONSIN, Milwaukee (Brookfield) •
INTERNATIONAL — ARGENTINA, Buenos Aires • AUSTRALIA, Adelaide, Brisbane, Canberra, Melbourne, Perth and Sydney • AUSTRIA, Vienna • BELGIUM, Brussels • BOLIVIA, La Paz • BRAZIL, Puerto Alegre, Rio de Janeiro and São Paulo • CANADA, Calgary, Halifax, Montreal, Ottawa, Toronto and Vancouver • CHILE, Santiago • DENMARK, Copenhagen • FINLAND, Helsinki • FRANCE, Grenoble and Paris • GERMANY, Berlin, Cologne, Hannover, Hamburg, Frankfurt, Munich and Stuttgart • HONG KONG • INDIA, Bombay • INDONESIA, Djakarta • ISRAEL, Tel Aviv • ITALY, Milan and Turin • JAPAN, Osaka and Tokyo • MALAYSIA, Kuala Lumpur • MEXICO, Mexico City • NETHERLANDS, Utrecht • NEW ZEALAND, Auckland • NORWAY, Oslo • PHILIPPINES, Manila • PUERTO RICO, Santurce • SINGAPORE • SPAIN, Barcelona and Madrid • SWEDEN, Gothenburg and Stockholm • SWITZERLAND, Geneva and Zurich • TAIWAN, Taipei and Taoyuan • UNITED KINGDOM, Birmingham, Bristol, Dublin, Edinburgh, Leeds, London, Manchester and Reading • VENEZUELA, Caracas • YUGOSLAVIA, Ljubljana •

printed in U.S.A.