THE SOFTWARE DISPATCH

DOS/BOSS

digital

THE SOFTWARE DISPATCH

DOS/BOSS

NOVEMBER 1973

SOFTWARE ENGINEERING AND SERVICES
SOFTWARE COMMUNICATIONS
Digital Equipment Corporation
Maynard, MA 01754

digital

SOFTWARE DISPATCH

DOS/BOSS

November, 1973

TABLE OF CONTENTS

1.Ø INTRODUCTION

2.Ø HOW TO USE YOUR SOFTWARE DISPATCH

2.1 Introductory Section

2.2 General Format of Articles

2.3 Filing

2.3.1 Software Product Components and Descriptions

3.Ø REFERENCE MANUALS

DOS-15 Software Review

Bootstrap

Programming Note Regarding .GET & .PUT MACROS #1

Programming Note on Special Bits Meaning in Reserved Word #2

Card Reader Handlers

Handler for the CR15 Card Reader #1

Problems when conditionalized for the CR15 Card Reader #2

Chain

Programming Notes #1

Chain & Execute

Clarification of obscure points #1

Checkout Package

Functional Description #1

SOFTWARE DISPATCH

DOS/BOSS

November, 1973

DDT

Programming Notes #1

DECtape Handlers

Functional Change #1

Source change to prevent .WAIT X from getting a buffer #2

Support of .TRAN Function on PDP-8/10/11 Tapes #3

Disk Handlers

Programming Note on Use of .USER Macro #1

DOSSAV

Operating Instructions #1

DUMP

Information on .DAT slot Assignments #1

Some Information on QAREA's #2

Patch to correct problem with selective dumps #3

Patch to correct looping problem #4

DUMP described in DOS Keyboard Command Guide #5

FOCAL

The Random Number Generator #1

FORTRAN

Usage Restriction and Precaution #1

Usage of ADSS Versus DOS Magtape Handler MTF #2

Corrected Version of ADJ to Remove Incompatibility with FORTRAN Object Time System #3

SOFTWARE DISPATCH

DOS/BOSS

November, 1973

FORTRAN Compiler

Patch to Correct Improper Imbedded Subscript Calculation #1

Errors not Detected During Compilation #2

Programming to avoid Incorrect Mode Typing of Functions Declared EXTERNAL #3

Incomplete DO Statement Difficulties #4

FORTRAN OTS

Programming Note to Eliminate Carriage Return #1

Programming Note #2

Numerical Restrictions #3

Reading INTEGER Variables Causing IOPS Ø #4

Line Printer Handlers

Source Change to Prevent Return to Mainstream, at API Level 4 on .WAIT #1

Linking Loader

Features of this Version #1

LK35 Keyboard Handler

Functional Description #1

MACRO-15

New Switches in this Version #1

Macros Implemented Specifically for DOS-15 #2

Pseudo-op, .CBD #3

MACRO-15 (Continued)

Note on Binary Output File Extensions #4

Problems Corrected #5

Programming Note on Usage of 'T' Switch #6

Patch to Expand .RTRAN Correctly #7

Patch to Correct Printing of Erroneous Message #8

Solutions to Miscellaneous Problems #9

Leading Commas in Command String Foiled BOSS-15 Operation #10

Problem with Parameter File Name Being Assigned to Listing File #11

Patch to Prevent NEXM System Crash in an X4K Environment #12

Magtape Handlers

Functional Changes #1

Returning Status on a .WAIT #2

Monitor

Operational Note: IOPS 77 After Control Q #1

Operational Note on Overprinting #2

Problem in Restoring UIC's to DAT Slots on LOGOUT #3

Patch to Support LA3Ø at 3ØØ Baud #4

Patch to Restore System Configuration on "LOGOUT" #5

Teletype Handler .INIT Function Limitations #6

Document Number Change Notice #7

Documentation Error #8

Typographical Error in Patch to PIP #9

SOFTWARE DISPATCH

DOS/BOSS

November, 1973

Paper Tape Punch Handlers

Difference in End-of-File Code Between ADSS and DOS #1

Patch

Operation under DOS-15 #1

PIP

Note on Implicit Data Modes #1

Restriction on Diskpack to Diskpack Copies #2

Segmenting Files Under BOSS-15 #3

Patch to Transfer all Files in a 'C' Function #4

Patch to Correct Problem in Transferring Multiple Files from Card Readers #5

Patch to Correct Problem with 'N' Function when 'Snn' Switch Option is Used #6

Problem with 'C' Switch Option in a 'T' Function #7

System

Conversion from ADSS-15: Programs May Not Fit #1

Minimum and Optional Hardware for DOS-15 System #2

Programming Note #3

System Generator

Relative Positions of the Clock and Line Printer Skip IOT's #1

Update

Note on Program File Extensions #1

Update (Continued)

Note on G Option and FREE Command #2

Note on Error Conditions #3

Why V1ØB and not V11A #4

Patch to Correctly Read the EOF/EOM Record #5

VT15 Graphics

Functional Description - Rotate (SRC) #1

Functional Description - Circle (SRC) #2

Writing Tablet Handler

Functional Description #1

BOSS-15 Software Review

Monitor

Patch to Correct Problems in Closing Run Time Files & MICLOG being Active on Exit from BOSS-15 #1

Patch to Correct Problem in Assigning Non-default File Names in $CRT and $ADD Procedure Files #2

Procedure Files

Correction to FOR and ASM to Permit Usage of the Default File Name #1

Correction to JOB Procedure File to Permit Usage of Default UIC #2

SOFTWARE DISPATCH

DOS/BOSS

November, 1973

1.Ø Introduction

The Software Dispatch is a cumulative report (updated monthly) which provides each subscriber with a quick reference library of:

1. Announcements of new software products.

2. Current software status - outstanding problems and problems with solution.

The name Software Dispatch was chosen in accordance with its definition.

> "... a written message; particularly an official communication sent with speed."

With this in mind, it is our intent to assure the subscriber of timeliness and technical quality.

Comments and suggestions from our readers are welcome.

2.Ø How to Use Your Software Dispatch

The Software Dispatch is intended to supplement your software and aid in its maintenance. Updates will be supplied monthly to subscribers.

Changes to relocatable/absolute files/programs will be based on the current version and edit number of the file/program. It is recommended that users make all the published patches. Users possessing source files are suggested to make the indicated source level changes and suitably update their Software System.

2.1 Introductory Section

This introductory material should be filed at the beginning of your notebook and will be referred to as chapter and paragraph numbers. To date, these are the chapters in the Introductory Section:

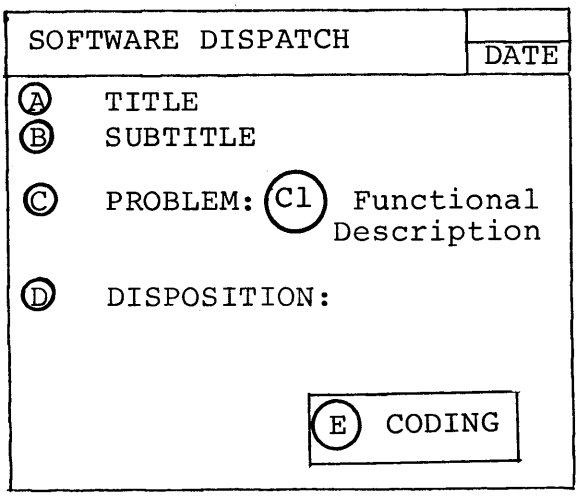
1.Ø - Introduction
2.Ø - How to Use Your Software Dispatch
3.Ø - Software Manuals for DOS/BOSS

For example, if a new paragraph were to be inserted as the fifth paragraph of Chapter 1, the article would have a chapter and paragraph number of 1.5 (or section 1.5).

2.2 General format of the article

Each article is formatted so that you can easily recognize to what the article refers.

This is an overall example of the format. Each part will be explained in detail.



(A) Title

The title of articles pertaining to programs include the latest Version and/or Edit number (whichever applicable). Source level changes will always change the Edit number, e.g., Edit #Ø52 to Edit #Ø53. The Version number will be changed, if applicable, e.g., V6A to V7A. A change to core image system programs (via binary patches) will change the version number, e.g., V6A to V6B, and the Edit number remains the same. Changes to relocatable files will appear only in the form of source code changes.

(B) Subtitle

This brief statement gives the reader a hint about the content of the article. The subtitle is used in the Table of Contents for identifying the problem.

(C) Problem:

A paragraph or two is used to describe the problem in general terms. It may include examples, warning, etc.

The purpose of this paragraph is to make the user aware of an existing problem in the software and its documentation.

(C1) In some cases, the text of the article contains a functional description which may not have been included in the system software document.

(D) Disposition:

This is the section of the article which tells the reader the status of the problem. The disposition of the problem may be one of the following categories:

° No Disposition

An article just states the problem and will not have an answer. This is to inform you that we are aware of the problem, but at this time there is no fix available.

° Solution

If there is a way to avoid the problem, or a patch to fix it, it is stated here.

In some articles you will find that the only solution given is "fixed in the next version".

(E) Coding

This block is used for filing purposes and is further expanded in section 2.3.

2.3 Filing

A system has been devised to help you file each article in its proper place. The key to this system is the block at the bottom of the page.

Below is a close-up view of the coding block for Systems Software.

<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>

Each month, you should take the update and insert the pages in your notebook according to the following instructions.

First, the article is filed by Software Product (1). In this case, all articles will be classified under the major heading.

Secondly, the Software Product is broken down by its components (2). See section 2.3.1 for the list of DOS/BOSS Components.

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

All other information in the coding block is to further clarify the article and not specifically for filing.

Explanation of all the other information included in the coding block follows:

(1A) Version of the entire Software Product.

(2A) Version number of components.

(2B) Edit numbers.

(2C) If more information is necessary to help the user, it will be inserted in this block.

(3A) This block indicates how many pages the article has.

(4) A new article is indicated by an "X" in this block. This article has not been published before.

(5) An article which was previously published and is being published again for reasons of revision or correction is indicated by a number in this block.

The number in the block specifies the number of times the original article has been revised.

For example: the second revision of an article which originally appeared in June, 1973 would be indicated as follows:

New	Replacement Article	Original Date
	2	June, 1973

The customer will find the date of revision #2 in the upper right corner of the article.

(5A) Original date of a revised article is placed here.

2.3.1 COMPONENTS

A list of the components and program names for DOS/BOSS along with relevant comments is found below. The component names always appear in the coding block for filing purposes. The program names, used only if necessary, appear in the coding block under "additional information" and are not used for filing.

Articles concerning changes to manuals will be coded and filed along with changes to programs. For example, an article containing a binary patch 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.

DOS-15:

Components	Additional Information	Comment
BOOTSTRAP	RFBOOT	RF Disk Bootstrap
	RPBOOT	RP Disk Bootstrap
CARD READER HANDLERS	CD.DOS	Non-Batch Card Reader Handler
	DOSBCD	Batch Card Reader Handler
CHAIN		
DDT		
DECTAPE HANDLERS	DTA.	
	DTC.	
	DTD.	
	DTE.	
	DTF.	
DISK HANDLERS	DKA.	RF Disk Handler
	DKB.	RF Disk Handler
	DKC.	RF Disk Handler
	DKL.	
	DPA.	RP Disk Handler
	DPB.	RP Disk Handler
	DPC.	RP Disk Handler
	DPL.	

COMPONENTS (Cont'd)

DOSSAV		System save/restore program
DTCOPY		
DUMP		
EDITOR		
EDITVP		
EDITVT		
EXECUTE		
FOCAL	FOCAL FNEW	
FORTRAN	COMPILER OTS	 object time system utility routines
LINKING LOADER	.LOAD	relocatable file loader
LINE PRINTER HANDLERS	LPA.Ø9 LPA.15 LP.647	PDP-9 Line Printer Handler PDP-15 Line Printer Handler 647 Line Printer Handler
LK35 KEYBOARD HANDLER	LKA.	
MACRO-15	MACRO CREF	 cross reference program
MAGTAPE HANDLERS	MTA. MTC. MTF.	
MONITOR	DOSNRM RESMON TELETYPE HANDLER	Non-Resident MONITOR Resident Monitor
MTDUMP		
PAPERTAPE PUNCH HANDLERS	PPA. PPB. PPC.	

COMPONENTS (Cont'd)

PAPERTAPE READER HANDLERS	PRA. PRB.	
PATCH		
PIP		
QFILE		
SRCCOM		
SYSTEM		all system information that does not fall under any other component will appear here.
SYSTEM GENERATOR	SGEN	
SYSTEM LOADER	.SYSLD	system program loader
UPDATE		
VP15 GRAPHICS	VPA. VPA.S FORT NUVAL VECTOR	
VT15 GRAPHICS	VTA. VTPRIM LTORPB DYLDR TRACK ROTATE CIRCLE	
WRITING TABLET HANDLER	VWA.	
8TRAN		
89TRAN		

BOSS-15:

MONITOR	B.PRE NRBOSS	pre-processor non-resident monitor
PROCEDURE FILES		The additional information may contain any of the procedure files listed in the BOSS-15 USERS MANUAL.

3.Ø DOCUMENTATION FOR DOS-15 SYSTEM

A. Required Documentation

DOS USERS MANUAL	DEC-15-ODUMA-A-D
FORTRAN IV LANGUAGE MANUAL	DEC-15-GFWA-D
FORTRAN IV OPERATING ENVIRONMENT	DEC-15-GFZA-D
PIP DOS MONITOR UTILITY PROGRAM	DEC-15-YWZB-DN13
SGEN DOS UTILITY PROGRAM	DEC-15-YWZB-DN12
DOS KEYBOARD COMMAND GUIDE	DEC-15-NGKA-D
DOS SYSTEM MANUAL	DEC-15-ODFFA-A-D

B. Suggested Additional Manuals

PDP-15 MACRO-15 ASSEMBLER	DEC-15-AMZB-D
FOCAL PROGRAMMING MANUAL	DEC-15-KJZB-D
PDP-15 8-TRAN MANUAL	DEC-15-ENZA-D
PDP-15 DDT UTILITY PROGRAM	DEC-15-YWZA-DN1
PDP-15 CHAIN & EXECUTE UTILITY PROGRAM	DEC-15-YWZB-DN2
PDP-15 MTDUMP UTILITY PROGRAM	DEC-15-YWZB-DN4
PDP-15 PATCH UTILITY PROGRAM	DEC-15-YWZB-DN5
PDP-15 EDIT UTILITY PROGRAM	DEC-15-YWZB-DN6
PDP-15 UPDATE UTILITY PROGRAM	DEC-15-YWZB-DN7
PDP-15 LINKING LOADER UTILITY PROGRAM	DEC-15-YWZB-DN8
PDP-15 SRCCOM UTILITY PROGRAM	DEC-15-YWZB-DN11
VT15 GRAPHICS SYSTEM PROGRAMMING MANUAL	DEC-15-ZFSB-D
VP15A GRAPHICS SOFTWARE MANUAL	DEC-15-UXSB-D

SOFTWARE DISPATCH

DOS-15

PRE-1973

BOOTSTRAP

Programming Note Regarding .GET & .PUT MACROS

In DOS-15, .GET and .PUT MACRO's use the bootstrap for communication with the system device. On DECdisk systems, the bootstrap will ignore unit numbers in .PUT MACRO's. It will assume unit zero.

The source name of the RF DECdisk Bootstrap is RFBOOT Ø11

The source name of the RPØ2 Bootstrap is RPBOOT ØØ3

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
Bootstrap	N/A	
Subprogram or Additional Information	Sequence #	PAGE
	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

BOOTSTRAP

Programming Note on Special Bits Meaning in Reserved Word

There is an important transfer vector in the Bootstrap, located in word 17777 of the highest bank. The high order three bits of this word are reserved for the Monitor, and have the following meanings:

WORD	BIT	MEANING
17777	Ø	1=In batching mode Ø=Not in batching mode
	1	1=$JOB ASCII line or card just read by batch device Ø=Last line or card not $JOB
	2	1=Batch device is card reader Ø=Batch device is paper tape reader

The system loader will refresh the Resident Monitor's patch area only on a bootstrap load or restart. This allows communication between two programs that require a new system configuration.

SOFTWARE DISPATCH

DOS-15

PRE-1973

CD.DOS (EDIT #Ø12) and DOSBCD (EDIT #ØØ3)

Handler for the CR15 Card Reader

The card reader handler (CD.DOS) has undergone the following modifications. It should be noted that none of these changes were implemented in the batch card reader handler (DOSBCD).

1. Defining the parameter "CR15=Ø" for assembly obtains the CR15 version of the card reader handler. The CR15 is a card reader controller which operates on data channel and can handle a variety of card readers up to the 1ØØØ card per minute model.

2. The card reader handlers will recognize two different punches for end-of-file (EOF) cards-

   a. All rows punched in column one (ADSS-15 usage)
   b. The 12, 11, Ø, 1 multi-punch in column one

3. The card reader now returns an EOF code (1ØØ5) instead of an EOM code (1ØØ6).

4. When a hopper empty, stacker full, or reader not ready condition arises, the following expanded message follows the IOPS4:

   'CD NOT READY'

5. If a card is encountered with an illegal card punch the message 'IOPS4 CD-ILLEGAL PUNCH' is printed. At this point the user can punch the card correctly and continue by typing CTRL R.

Restrictions:

DOSBCD ØØ3 may not be used as a batch device handler for tne CR15 device. It may, however, be used as an input device handler.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
CARD READER HANDLERS	N/A	
Subprogram or Additional Information	Sequence #	PAGE
CDB.	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

CD.DOS (EDIT #Ø13)

Problems when conditionalized for the CR15 card reader

1. Problem: The handler did not handle any error conditions properly. On errors like pick failure, block missed, data missed, motion error and photo error the handler moved (offset) the card to the reader station and exited without issuing a message or a .READ MACRO.

   As a result of the above the system would hang up.

   Correction: Since all errors are potentially recoverable the new version of the handler issues an expanded error message "IOPS4 ERR,CHK-CD, RLD-CD" permitting continuation of operation by reloading a fresh image of the bad card.

   The user has the option of aborting the .READ operation by typing ↑P/↑C.

2. Problem: The handler temporarily returned an EOF (in the line buffer header word pair (HWP)) on exit from a .READ MACRO. When the interrupt was received the HWP was set to the appropriate value.

   Although this does not affect the user since he cannot process the data being read until it is complete, it is logically erroneous and time-wise wasteful.

   Correction: Do not alter the HWP.

The following SRC level changes to CD.DOS edit #Ø12, correct these problems.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
CARD READER HANDLERS	N/A	13
Subprogram or Additional Information	Sequence #	PAGE
CDB.	2	1 OF 3
New [X] Replacement Article [ ]	Original Date	

Problems when conditionalized for the CR15 card reader (Cont'd)

PAGE 7 CD.DOS Ø13 .READ

```
2Ø4                                    .TITLE .READ
2Ø5                                /INITIATE READ TO CARD READER.
2Ø6   ØØ1ØØ R 1ØØ214 R        CDREAD  JMS IOUWAY        /ARE WE DONE YET
2Ø7   ØØ1Ø1 R 767ØØØ A                LAW 7ØØØ          /YES, PICK UP CAL

224   ØØ117 R 44Ø755 R                ISZ CDARGP        /BUMP TO RETURN
225   ØØ12Ø R 2ØØØ33 R                LAC CDRBMP+1              /SET RETU
226   ØØ121 R Ø4Ø143 R                DAC CDRRET                /FROM REA
227                                    .IFDEF  NOTGDI
228                                CURRTN CREF               /EOF BUTTON OR HO
```

PAGE 17 CD.DOS Ø13 CDB. CARD READER INTERRUPT HANDLING SECTION

```
6Ø3                                    .TITLE CDB. CARD READER INTERRUPT
6Ø4                                /CARD READER INTERRUPT SECTION.
6Ø5   ØØ46Ø R 741ØØØ A        CDRINT  SKP               /CONTROL COMES HE

647   ØØ475 R Ø4Ø772 R                DAC     CDSTAT    /AND SAVE
648   ØØ476 R 5Ø1Ø2Ø R                AND     (4ØØ      /TROUBLE BIT
649   ØØ477 R 7512ØØ A                SNA!CLA                   /SKIP ON TRO
65Ø   ØØ5ØØ R 6ØØ537 R                JMP     CROONE    /NO ERRORS
651   ØØ5Ø1 R 2ØØ772 R                LAC     CDSTAT    /TROUBLE???????
652   ØØ5Ø2 R 5Ø1Ø21 R                AND (Ø37ØØØ       /WHAT TROUBLE?
653   ØØ5Ø3 R 541ØØ4 R                SAD (2ØØØ         /HOPPER EMPTY?
654   ØØ5Ø4 R 6ØØ537 R                JMP CRDONE        /YES, DONE RETURN
655   ØØ5Ø5 R 14Ø76Ø R                DZM CDIOSW        /TO ENABLE CTL P
```

Problems when conditionalized for the CR15 card reader (Cont'd)

```
PAGE 18       CD.DOS Ø13      CDB. CARD READER INTERRUPT HANDLING SECTION

656    ØØ5Ø6 R 76ØØ2Ø A            LAW 2Ø           /CLEAR INTERRUPT
657    ØØ5Ø7 R 7Ø67Ø4 A            CRLC
658    ØØ51Ø R 76ØØØ4 A            LAW 4            /OR READER; PRIN
659    ØØ511 R 121Ø12 R            JMS* (EXERRS     /IOPS 4 MESSAGE
66Ø    ØØ512 R 6ØØ522 R            JMP CDRST        /TRY AGAIN, REST
661    ØØ513 R 777772 A            LAW -6
662    ØØ514 R Ø52222 A            .SIXBT /ERR CHK-CD, RLD-CD/
       ØØ515 R 4ØØ31Ø A
       ØØ516 R 1355Ø3 A
       ØØ517 R Ø4544Ø A
       ØØ52Ø R 2214Ø4 A
       ØØ521 R 55Ø3Ø4 A
663    ØØ522 R 2Ø1Ø22 R  CDRST      LAC (JMP CDIRET /RESTRORE RE
664    ØØ523 R Ø4Ø143 R             DAC CDRRET      /ADDRESS
665    ØØ524 R 6ØØ123 R            JMP CDOK+1      /RESTART READ
666                                  .ENDC
667                          /---------
668    ØØ525 R Ø6ØØ43 R             DAC* CDIPTR  /INSERT THIS COL
669    ØØ526 R 44ØØ43 R             ISZ CDIPTR   /BUMP BUFFER POI
67Ø                                  .IFDEF NOTGDI
```

CHAIN V7A (EDIT #148)

Programming Notes

CHAIN will accept library indicators (#) on internal and external link components. The name given must correspond to the GLOBAL name of the routine desired in the library. Also note that the .IODEV Pseudo-op should come after the .GLOBL Pseudo-op so that CHAIN will not skip over the .IODEV information when doing a library search.

CHAIN will accept .IODEV information from -18 thru +71 decimal. This is so that it can accommodate the 77 octal .DAT slots allowed DOS-15 by the system.

CHAIN will do a .USER for the "PAG" or "BNK" UIC on .UFDT - 1 if the "PGR" or "BKR" option respectively is given. This is so the correct system library will be searched for the relocation mode in effect.

Since I/O handlers are in the "IOS" UFD as separate files, the user must transfer to his UFD, using PIP, any handlers he may want to include in his overlay system. Also, the user must include the handler's file name (global name) in the link description command string.

CHAIN calculates the number of 4ØØ(8) word blocks needed to store the overlay system, by links, as a core image. This information is stored in the environment indicator in bits Ø through 11 as a right-justified octal number. This total does not include the link table (link 377777) or the resident code (link Ø).

There is an option in CHAIN which allows the user to restrict COMMON areas to bank boundaries. This is useful to the VT15 user who builds display files in COMMON, since the VT15 cannot cross bank boundaries directly (i.e., 13-bit addressing). There are two forms of the option and the giving of one will cancel out the other form if it was given previously.

Programming Notes (Cont'd)

"VTC" option without names restricts all common areas to bank boundaries. The "VTC" option is delimited by being the first option in the command string or by a comma on the left, and by a comma or altmode on the right.

"VTC/NAM1,NAM2,...NAM3/" option with names restricts to bank boundaries only those COMMON areas named (note: blank common is .XX). More than one VTC option with names may be given in the command string and all names specified will be restricted. The option is delimited by being the first option in the command string or by a comma on the left, and a slash on the right. The name field within the option is delimited by a slash right after the "VTC" and the slash that terminates the option. Names in the list are separated by commas.

Notes:

The "VTC" option will not restrict common areas declared in BLOCK DATA SUBPROGRAMS.

The common area will be restricted to bank boundaries even if CHAIN is running in page relocation mode.

CHAIN ignores the Linking Loader code '33'.

CHAIN & EXECUTE (DEC-15-YWZB-DN2)

Clarification of obscure points

Note on operation of CHAIN:

1. Routines/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 to resolve unresolved globals.

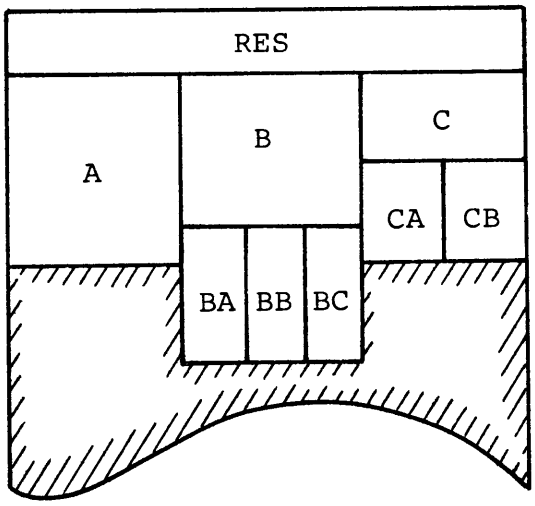
3. CHAIN V1ØA (supplied with DOS-15 V3A forthcoming update release) will resolve globals in a manner similar to the LINKING LOADER.

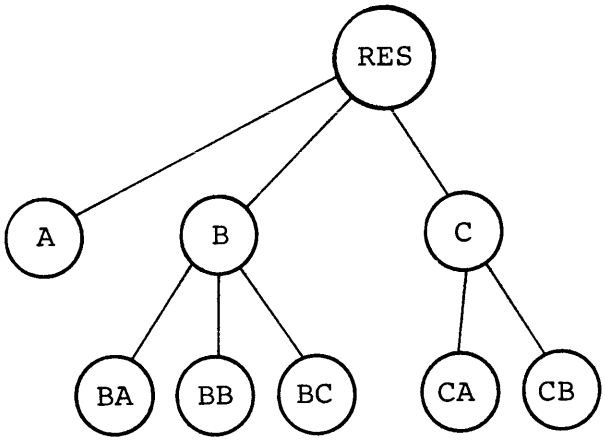
Restrictions in building an overlay structure:

Pages 12 and 13 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.



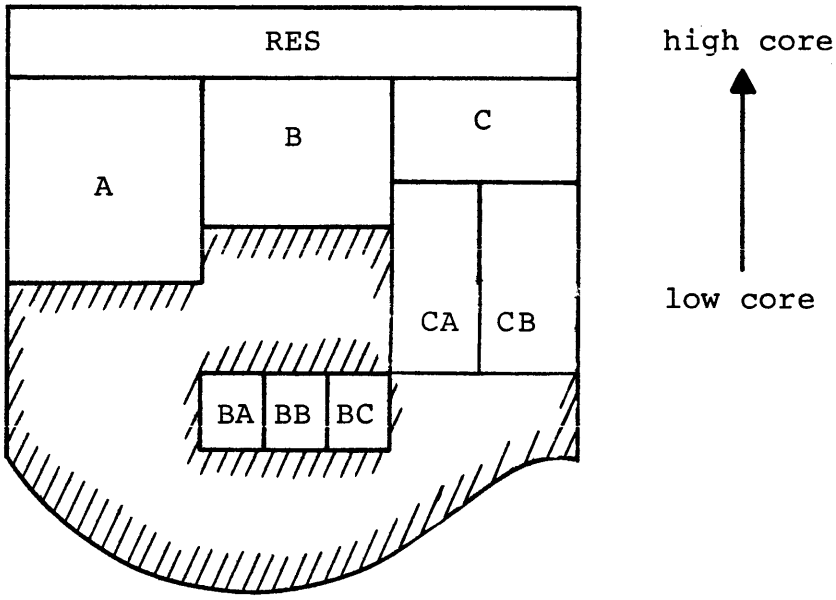


equivalent tree structure

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	
DOS-15	V2A	
Component: CHAIN & EXECUTE	Version	Edit #
Subprogram or Additional Information: DOCUMENTATION CORRECTION	Sequence # 1	PAGE 2 OF 3
New ☐ Replacement Article 1	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	
DOS-15	V2A	
Component: CHAIN & EXECUTE	Version	Edit #
Subprogram or Additional Information: DOCUMENTATION CORRECTION	Sequence # 1	PAGE 3 OF 3
New ☐ Replacement Article ☑ 1	Original Date: May, 1973	

CHECKOUT PACKAGE

Functional Description

The following is a description of the Software Checkout Package. The description and the paper tape are distributed as part of the system. See DOSSAV Operating Instructions for installing the system.

1.0 INTRODUCTION

The purpose of the checkout package is to show that the system has been properly installed onto DECdisk or Disk Pack. It does so by briefly testing all the basic pieces of the system. The following is a list of programs tested:

1. Resident and Nonresident Monitors
2. PIP
3. FORTRAN Compiler and Object Time System
4. Macro Assembler
5. Linking Loader and System Loader
6. Chain and Execute programs
7. System Device Handler (DECdisk or Disk Pack)
8. Paper Tape Reader Handler
9. Teleprinter Handler
10. Batching Mode System Commands
11. DOSSAV system SAVE/RESTORE program

2.0 IDENTIFICATION

The batch paper tapes for the Checkout Package are identified as follows:

RF.CHK	DEC-15-CIDA-PA	(for the RF15 DECdisk System)
RP.CHK	DEC-15-CTAA-PA	(for the RP02 Disk Pack System)

Functional Description (Cont'd)

3.0 CHECKOUT PACKAGE RESULTS

The result of the FORTRAN Object Time System (shortly after the GLOAD Command) should be:

-Ø.1235E+Ø3

Also, the result of the Chain and Execute programs should be:

-Ø.1234E+Ø5

4.0 CHECKOUT PACKAGE OPERATION

The following are the procedures for an RF DECdisk system having two (2) platters or an RPØ2 system.

Take the ASCII paper tape labeled 'RF.CHK' (or RP.CHK, if Disk Pack system), and place it in the high speed paper tape reader. Then type:

BATCH PR)

The Batch commands will then run the checkout package to completion and will so indicate on the teleprinter, before leaving Batch Mode.

4.1 The following are the procedures for an RF DECdisk system having only one (1) platter and DECTAPE:

1. Load the System Software (following the DOSSAV Operating Instructions).

2. When the system monitor announces itself, type in the current date:

Functional Description (Cont'd)

Example:

11/29/71

3. Mount a scratch tape on DT Unit #1 and write enable.

4. Type the underlined commands which follow:

$LOGIN PER↵

$PIP↵

>C DT1← DK↵

>N DK↵

>↑C

5. Now again read in the RF Bootstrap, and enter the date which the monitor requests.

6. Take the ASCII paper tape labeled RF.CHK and place it in the high speed papertape reader, then type:

$BATCH PR↵

The Batch Commands will then run on the Checkout Package to completion and will so indicate on the teleprinter - before leaving Batch Mode.

4.2 SPECIAL PROCEDURES FOR MAGTAPE USERS:

The following are the procedures for an RF DECdisk system having only one (1) platter and MAGTAPE:

1. Load the System Software (following the DOSSAV Operating Instructions).

Functional Description (Cont'd)

2. When the system monitor announces itself, type in the current date: ex: 11/29/71

3. Mount a scratch magtape on MAGTAPE UNIT #1 (have it write enabled)

4. Type the underlined commands which follow:

```
$LOGIN PER↵
$A  MTF  11↵
$PIP↵
>T  MT1  (A)←DK DOSBCD  ØØ3↵
>T  MT1  (A)←DK CD.DOS Ø12↵
>T  MT1  (A)←DK FNEW  ØØ4↵
>T  MT1  (A)←DK LPA.15  Ø42↵
>T  MT1  (B)←DK DYLDR  BIN↵
>T  MT1  (B)←DK TRACK BIN↵
>T  MT1  (B)←DK VTPRIM  BIN↵
>T  MT1  (B)←DK ROTATE  BIN↵
>T  MT1  (B)←DK NUVAL  BIN↵
>T  MT1  (B)←DK FORT  BIN↵
>T  MT1  (B)←DK VECTOR  BIN↵
>T  MT1  (B)←DK CIRCLE  BIN↵
```

Functional Description (Cont'd)

```
>T  MT1  (B)←DK LPA.Ø9  BIN↵
>T  MT1  (B)←DK F4X9  BIN↵
>T  MT1  (B)←DK VPA.S  BIN↵
>T  MT1  (B)←DK LP.647  BIN↵
>T  MT1  (B)←DK LTORPB  BIN↵

>N  DK↵
>↑C
```

5. Now again read in the RF Bootstrap, and enter the date which the monitor requests.

6. Take the ASCII paper tape labeled RF.CHK and place it in the high speed papertape reader, then type:

```
$BATCH PR↵
```

The Batch Commands will then run the Checkout Package to completion and will so indicate on the teleprinter - before leaving Batch Mode.

DDT V9A (EDIT #Ø1Ø)

Programming Notes

DDT identifies itself as DDT Vnn for page mode loads, and BDDT Vnn for bank mode loads.

DDT allows breakpoints on floating point instructions.

After nonrecoverable IOPS errors, DDT awaits a CTRL T before continuing.

DDT uses the tabbing mechanism in the teleprinter handler, instead of multiple spaces, for output. This makes DDT smaller and when using Model 35 teletypes faster on tabbing output.

The routine to print octal numbers has been shortened, and a problem with zero suppression has been corrected.

DDT had a problem in setting .SCOM+2 incorrectly, so that about 5ØØ registers were unusable. This problem has been fixed in version V9A.

DTA. (EDIT #Ø2Ø)

Functional Change

The DTA handler has been modified to use dynamic buffer allocation. The new size is approximately 2750$_8$--this is smaller than the old DTB version. For this reason, DTB has been deleted from the system. DTA. still has only three file capacity. If a user attempts to use more than three files at once on DECtape, an IOPS 17 error will result.

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's buffer. Subsequent .READ's 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:

None of the above changes has been made to the DTC., DTD,. DTE., or DTF., DECtape Handlers.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
DECTAPE HANDLERS	N/A	2Ø
Subprogram or Additional Information	Sequence #	PAGE
DTA.	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

DTA. (EDIT #Ø21)

Source Change to prevent .WAIT X from getting a buffer

Problem:

In Edit #Ø2Ø of the DTA. handler, when a .WAIT X is issued after a .CLOSE X the handler gets a buffer. As a result when there are "B" buffers in a system and when B+1 files are accessed sequentially on B+1 DAT slots (B assigned to DT's and 1 assigned to some other device) the message -

"IOPS 55" (NO BUFFERS AVAILABLE)

is printed.

If all the B+1 slots are assigned to DT's the message

"IOPS 17" (TOO MANY FILES FOR HANDLER)

results.

Solution:

The above restriction is removed by preventing .WAIT from getting a buffer.

Below are the changes in the source code for DTA Ø2Ø.

Source Change to prevent .WAIT X from getting a buffer (Cont'd)

```
LOC     OCTAL     LABEL    INST         COMMENT

 Ø      Ø42644     DTA.    DAC CALP     /CAL POINTER
                             -
                             -
 15     Ø42646             DAC TEMP2    /T STORE DATA MODE OR
                                         /SUBFUNCTION
 16     222642             LAC* ARGP    /CHECK IF
 17     5Ø27ØØ             AND (77      /FUN. IS
 2Ø     5427Ø1             SAD (12      /.WAIT/.WAITR?
 21     6ØØ177             JMP DISPCH   /YES GO TO DISPCH,
                                         /DO NOT GET BUFFER
 22     222644             LAC* CALP    /NO, GET DAT SLOT
                                         /(9-17)
                             -
                             -
```

This problem will be corrected in version DTA Edit #Ø21.

DECTAPE HANDLERS

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

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

In order to realize the above a minor source modification is needed to the handlers that support the .TRAN function. 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 erroneous data in the most significant bits.

Note that since the file structure on these tapes are different from the DOS-15 file structure, only the .TRAN function will work.

This information is provided for user's convenience only and should not be misconstrued as a feature that will be supported by DEC.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
DECTAPE HANDLERS	N/A	
Subprogram or Additional Information	Sequence #	PAGE
DTA., DTD., DTE.	3	1 OF 1
New [X] Replacement Article [ ]	Original Date	

DISK HANDLERS

Programming Note on Use of .USER Macro

The Disk Handlers retain the UIC from the .UFDT in the Busy Table entry for each particular .DAT slot. Thus, changes made in the .UFDT via .USER macros will not become effective until a new Busy Table is established for the affected .UFDT slots.

Since the I/O macros to the disk handlers may or may not establish a new Busy Table entry, depending on a variety of conditions, this article will only give one example of incorrect usage, and then give a recommended procedure.

The following sequence:

```
.USER 1,ABC

.INIT 1

.USER 1,CDE

.SEEK 1,FILE
```

will cause a .SEEK for FILE under the UFD called ABC. If FILE is really under the UFD called CDE, the programmer should have written

```
.USER 1,CDE

.INIT 1

.SEEK 1,FILE
```

Programmers should ensure that all .USER macros are immediately preceded by any one of the following macros:

Programming Note on Use of .USER Macro (Cont'd)

.CLOSE

.RENAM

.MTAPE (rewind)

.DLETE

or immediately followed by a .INIT.

DOSSAV V2A (Edit #Ø43)

Operating Instructions

DOSSAV is the system save/restore program. It resides on a paper tape, which must be HRM loaded at 3772Ø (restart at 345ØØ).

It saves and restores to/from DECdisk, Disk Packs, DECtape and Magtape. A DECdisk system can be saved on and restored from DECtape, Magtape and Disk Pack. A Disk Pack system can use DECtape and Magtape.

Once loaded, it asks for all necessary information, such as input and output devices, unit numbers and, in the case of Magtape - parity and density.

GENERAL INSTRUCTION:

The user must type Carriage Return after all entries, including the character typed to restart after errors.

A. Restoring Systems

The following examples illustrate how to put the systems distributed by Digital on DECtape or Magtape onto a DECdisk or Disk Pack. The user responses are underlined.

1. To restore a DECdisk system from DECtape (on Unit 1)

```
DOSSAV V2A
INPUT DEVICE? DT)
UNIT NO? 1)
OUTPUT DEVICE? DK)
DATE CREATED:  28-SEP-71
TAPE DONE.  MOUNT ANOTHER
```

At this point, mount tape 2 and type any character on the keyboard followed by a Carriage Return.

Operating Instructions (Cont'd)

NOTE: If DK is typed, no unit # is requested.

II. To restore a DECdisk system from Magtape (on Unit Ø)

```
DOSSAV V2A
INPUT DEVICE?  MT↵
UNIT NO?  Ø↵
TRACK (7 OR 9)?  7↵
DENSITY (2,5,8)? 8↵
PARITY (E OR O)? O↵
OUTPUT DEVICE:  DK↵
DATE CREATED:  28-SEP-71
```

NOTE: ALL SYSTEM RESTORE MAGTAPES DISTRIBUTED BY DIGITAL ARE 8ØØ BPI, ODD PARITY.

III. To restore a Disk Pack system from DECtape (on Unit 1)

```
DOSSAV V2A
INPUT DEVICE? DT↵
UNIT NO? 1↵
OUTPUT DEVICE? DP↵
UNIT NO? Ø↵
DATE CREATED:  28-SEP-71
TAPE DONE, MOUNT ANOTHER
```

At this point, mount Tape 2 and type any character on the teleprinter followed by a Carriage Return.

Operating Instructions (Cont'd)

IV. To restore a Disk Pack

```
DOSSAV V2A
INPUT DEVICE?  MT)
UNIT NO?  1)
TRACK (7 OR 9)?  7)
DENSITY (2,5,8)?  8)
PARITY (E OR O)?  O)
OUTPUT DEVICE?  DP)
UNIT NO?  Ø)
DATE CREATED:  28-SEP-71
```

It is possible to restore a software system to the DECdisk which was created for a machine smaller (different # of DECdisk platters) than the one being restored to. DOSSAV does all the necessary adjustments of the SAT's[1]. Therefore, the restore tapes issued by Digital for a 1 platter system can be restored to any system. Note that this should only be done with the Master tape(s), which have block 1775_8 free. That block may be needed during the restore for 5 or more DECdisk platters. Note that it is not possible to restore a software system which is larger than the hardware. (e.g., no restore of a 3-platter to one-platter configuration.)

B. Saving Systems

Once the user has tailored the system to his specific configuration, he will want to save that system for future restorations. To do that, simply reverse the procedure above. To illustrate, consider example 1 above and the changes necessary to it to create a restore tape.

To save a DECdisk system to DECtape (on Unit 1):

[1]SAT's: Storage Allocation Tables - i.e., bit maps.

Operating Instruction (Cont'd)

```
DOSSAV V2A
INPUT DEVICE?  DK)
OUTPUT DEVICE? DT)
UNIT NO.?  1)
TAPE DONE.  MOUNT ANOTHER
```

At this point, mount another tape and type any character on the keyboard followed by a Carriage Return.

Note that DOSSAV allows for as many DECtapes and Magtapes as are necessary to hold the system.

C. Error Conditions and Messages

Recoverable errors during command string decoding: If a question is answered incorrectly, DOSSAV outputs an appropriate error message and then repeats the question. These error messages are:

ILLEGAL DEVICE	An illegal device mnemonic was typed (something other than DP, DK,DT and MT) or an illegal combination of devices was typed (i.e., input = DT and output = MT).
BAD TRACK	Something other than 7 or 9 was typed.
BAD DENSITY	Something other than 2(2ØØ) , 5(556) or 8(8ØØ) was typed.

Operating Instructions (Cont'd)

BAD PARITY	Something other than E(even) or O(odd) was typed.

Recoverable errors during operation: If it is possible to recover from an error, DOSSAV will attempt to do it. The error message will be output to the console. After the problem has been corrected, any character on the keyboard followed by a Carriage Return will resume operation.

TAPE NOT READY	The DECtape or Magtape unit is off line or not write enabled.
DISK NOT READY	DECdisk is write locked.
DISK PACK NOT READY	The disk pack unit is not ready.

Unrecoverable errors[1]: Primarily hardware errors, from which DOSSAV cannot recover. After the error message has been output, DOSSAV restarts.

DECTAPE ERROR MAGTAPE ERROR DISK ERROR DISK PACK ERROR	
ATTEMPT TO RESTORE SYSTEM TO WRONG DISK	To protect users who have access to both a DECdisk and a disk pack and who may have several sets of restore tapes, all restore tapes are created with the mnemonic of the disk type in the first SAT. DOSSAV then checks this code against the output device code. If they differ, this message is output.

[1]DOSSAV retries five times on a parity error before issuing an unrecoverable error message.

Operating Instructions (Cont'd)

BLK 1775 OCCUPIED. NO 2ND SAT CREATED

A DECdisk system created for 4 or fewer platters is restored to a machine with 5 or more platters and block 1775 is already used. Therefore, no second SAT is created. A Master Tape was not used to make the restore.

D. Tape Structure

The restore tapes are structured as follows: The first SAT of the system is the first block put on the tape. This SAT, which is never restored to the disk, has 2 words modified: word 2 contains the creation date (taken from .SCOM+47) and word 376 contains the device mnemonic (.SIXBT, right justified). All the occupied blocks referenced by this SAT are then put sequentially on the tape. The second SAT, if there is one, is then put on, and so on. This structure enables use of Magtape, which is a sequential only device.

E. DOSSAV Restrictions

1. It is not possible to save or restore Magtapes with even parity.

2. DOSSAV fails when two DECtapes are on line with the same unit number. It is necessary to restart under such circumstances.

3. Error checking and recovery is minimal.

DUMP V9A (EDIT #ØØ9)

Information on .DAT slot Assignments

The command string will be output onto whatever device is assigned to .DAT -12.

If .DAT -14 is assigned to any device other than DECtape, Disk Pack or DECdisk, DUMP will print the following message:

.DAT -14 IS NOT ASSIGNED TO DISK OR DECTAPE

and then DUMP will exit to the Nonresident Monitor.

If .DAT -12 is assigned to a mass storage device, DUMP will not require an ALT MODE after an ALL command in order to create a file memory dump, called "MEMORY␣DMP".

DUMP V9A will output only one form feed, instead of two, when the line printer is assigned to .DAT -12.

DUMP V9A (Edit #ØØ9)

Some Information on QAREA's

DUMP will check the device assigned to .DAT -14, in order to determine the first block of the QAREA. If DECtape, the first block is always 1Ø11. If Disk Pack, the first block is always 117Ø3Ø. The QAREA on DECdisk will be in different positions (space allocated during system generation), depending on the contents of .SCOM+66.

In the special case when the system device is Disk Pack, and .DAT -14 is assigned to a DECdisk, DUMP will print the following whenever the user attempts to dump the QAREA.

TYPE 1ST BLOCK # OF QAREA

In such a case, the user should type:

NNNNN#<CR> or ALT MODE

and then repeat the ALL command. This special case is made because .SCOM+66 contains Disk Pack information.

If the user wishes to dump a QAREA that has a size different from the core size current to the system, he should type:

>ALL␣(nn)<CR> or ALT MODE

where nn may be 8, 12, 16, 2Ø, 24, 28 or 32 (The space after the ALL and the right parentheses are optional.)

Note: The first block number and size of the CTRL Q area can be determined by listing SYSBLK with PIP:

L←TT DK (L)

Software Product		Version
DOS-15		V1A
Component	Version	Edit #
DUMP	V9A	9
Subprogram or Additional Information	Sequence #	PAGE
	2	1 OF 1
New [X] Replacement Article [ ]		Original Date

SOFTWARE DISPATCH

DOS-15

PRE-1973

DUMP V9B (EDIT #ØØ9)

Patch to correct problem with selective dumps

PROBLEM:

DUMP V9A outputs incorrect information on selective dumps.

SOLUTION:

Make the following binary corrections to DUMP using the PATCH program.

LOCATION	OLD CONTENTS	NEW CONTENTS	NEW SYMBOLIC	COMMENTS
16256	217406	617472	JMP PATCH	/Patch area
17472	-	116034	JMS DEVICE	/Device check
17473	-	217406	LAC (-1)	/Restore inst.
17474	-	616257	JMP BACK	/Return
17224	106400	206400		/V9B

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
DUMP	V9B	9
Subprogram or Additional Information	Sequence #	PAGE
	3	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

DUMP V9C (EDIT #ØØ9)

Patch to correct looping problem

PROBLEM:

DUMP V9B gets into an endless loop on the command "TYPE 1ST BLOCK OF ↑QAREA".

SOLUTION:

Using PATCH, make the following binary corrections to DUMP.

LOCATION	OLD CONTENTS	NEW CONTENTS	NEW SYMBOLIC
16123	615516	617475	JMP PATCH AREA
17475	-	2174Ø2	LAC (NOP
17476	-	Ø55735	DAC CONTAL
17477	-	615516	JMP TR6
17224	2Ø64ØØ	3Ø64ØØ	V9C

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
DUMP	V9C	9
Subprogram or Additional Information	Sequence #	PAGE
	4	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

December, 1972

DUMP V9C (EDIT #ØØ9)

DUMP described in DOS Keyboard Command Guide

There is no utility manual for the DUMP utility program. This information is instead included as a Chapter in the DOS Keyboard Command Guide (DEC-15-NGKA-D).

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
DUMP	V9C	9
Subprogram or Additional Information	Sequence #	PAGE
DOCUMENTATION	5	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

FNEW (EDIT #ØØ4)

The Random Number Generator

The DOS version of FOCAL has a random number generator (FRND) included in FNEW (user-defined FOCAL functions). This routine is an improvement over the FRAN function in FOCAL.

FRND generates real, floating point numbers in the range of -1.Ø less than or equal N less than or equal +1.Ø. The function can be called in the following ways.

FRND() --- Generate a random number based upon the last number generated.[1]

FRND(Ø) --- Same as above.[1]

FRND(X) --- Start a new sequence of random numbers based on the value of X. FRND will always generate the same number for the same value of X. X can be any valid arithmetic expression.

[1] If this is the first call for the generator, FRND initializes itself. Under the standard output format of FOCAL, this produces a value of 1.ØØØØ.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
FOCAL	N/A	4
Subprogram or Additional Information	Sequence #	PAGE
FNEW	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

FORTRAN, Object Time FORMAT Specification

Usage restriction and precaution

PROBLEM:

Section 6.1.3 of the FORTRAN IV Language Manual (DEC-15-GFWA-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(10),FORM(10)
      DATA FORM(1) /5H(4I10/
      DATA FORM(2) /5H)    /
      DO 10 I = 1,10
10    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(10)
      WRITE (6,3) (FORM(I),I=1,10)
3     FORMAT(1X,10A5)
      WRITE (6,FORM) (IBUF(I),I=1,NSZ)
      RETURN
      END
```

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN	N/A	N/A
Subprogram or Additional Information	Sequence #	PAGE
	1	1 OF 2
New [X] Replacement Article [ ]	Original Date	

Usage restriction and precaution (Cont'd)

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. Actually, the difficulty described here should be flagged as an error when compiling. The restriction will be relieved in the next release of the compiler.

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 above.

FORTRAN

Usage of ADSS versus DOS Magtape handler MTF

PROBLEM:

1. The MTF. handler released with ADSS-15 V5A does not calculate checksums for IOPS binary records. MTF. ØØ6 released with DOS does calculate checksums. Users attempting to read old data tapes under DOS FORTRAN may run into problems.

2. MTF. ØØ6 has a default buffer size of 255 (1Ø). The ADSS version has a fixed buffer size of 56 (1Ø).

3. The buffer size in MTF. ØØ6 may be changed under program control by referencing the global "MTBSIZ". (See DOS Users Manual).

4. MTF. ØØ6 checks for record length errors prior to calculating checksums, and, if the record length is less than 255 (i.e., 56), MTF. ØØ6 returns to the user without calculating a checksum.

   The above inconsistencies create the following possibilities when reading old data tapes under DOS FORTRAN and MTF. ØØ6.

	MTBSIZ	OTS Error	Data Read In
One physical record per logical record	$255_{1Ø}$	None	Good
	$56_{1Ø}$	11	----
More than one physical record per logical record	255	None	Bad
	56	11	---

Usage of ADSS versus DOS Magtape handler MTF (Cont'd)

In order to avoid OTS or data errors when reading old tapes under DOS FORTRAN/MTF. ØØ6, the user must follow this procedure:

1. Call loader with $LOAD (not GLOAD)
2. Load program using P option to get core map
3. When loader types ↑S, depress stop button
4. Deposit 74ØØØØ in MTF. + 423 (8) where MTF. = load map address for MTF.
5. Deposit ØØØØ7Ø in MTF. + 12Ø1 (8)
6. Press continue and type ↑S to start program

Another alternative would be to change the source of MTF. ØØ6 so that it never calculates checksums and then under program control change the contents of "MTBSIZ" with a GLOBAL subroutine.

As a final possibility, the user may copy his old data tapes using the following program.

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

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

Usage of ADSS versus DOS Magtape handler MTF. (Cont'd)

SET56 and SET255 are two macro subroutines:

```
            .GLOBL SET56,MTBSIZ,.FM

SET56       Ø
            LAC   (7Ø)
            DAC*   MTBSIZ
            DAC*   .FM
            JMP*   SET56
            .END

            .GLOBL SET255,MTBSIZ,.FM

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

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

FORTRAN IV OPERATING ENVIRONMENT MANUAL (DEC-15-GFZA-D)

Corrected version of ADJ to remove incompatibility with FORTRAN Object Time System

PROBLEM:

The ADJ subroutine published in DEC-15-GFZA-D, 'FORTRAN IV Operating Environment', Appendix C, is not compatible with FORTRAN OTS libraries using .SS ØØ8. (.SS ØØ8 is edit #8 of .SS, the array element address calculation routine).

Errors will occur for the following case

1. A calling program DIMENSIONS a single subscripted array, e.g., A(1ØØ). This array is passed as a subroutine call parameter, e.g., CALL Z (A).

2. The subroutine adjusts this array to be used as a two dimensional array, e.g.

```
SUBROUTINE Z (A)
DIMENSION A(1Ø,1Ø)
CALL ADJ (A,A(1,1),1Ø,1Ø,Ø)
```

What happens is that .SS ØØ8 is used to calculate the address of A(1,1). This causes a fatal error, since .SS ØØ8 is being used on an array (A(1ØØ) in the main program) that is still single dimensioned until the ADJ call is complete. .SS ØØ8 is not usable for single dimensioned arrays, and will not make a proper return.

SOLUTION:

The following changed ADJ circumvents this problem by re-entering the latter parts of .SS ØØ8 at such a location that the address computed is returned directly to ADJ. This change is compatible only with .SS ØØ8 as assembled without any conditional parameters. The changed version replaces the published version in the aforementioned document.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN	N/A	1
Subprogram or Additional Information	Sequence #	PAGE
ADJ	3	1 OF 4
New [X] Replacement Article [ ]	Original Date	

Corrected version of ADJ to remove incompatibility with FORTRAN Object Time System (Cont'd)

It should be kept in mind that once an array is adjusted in a program this constitutes a change on the array's array descriptor block which remains in effect until another ADJ call is made. Thus, for an array that is adjusted in some program, the adjusted values pertain on return to a calling program from which the array name is passed, or in any called program to which the array name is passed.

```
/     ADJ - FORTRAN ARRAY DESCRIPTOR BLOCK DIMENSION ADJUSTMENT
/
/ EDIT #1 (TAM)
/
/COMPATIBLE ONLY WITH UNCONDITIONALIZED VERSION OF .SS ØØ8
/
/COPYRIGHT 1973, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
/
/CALLING SEQUENCE
/        JMS* ADJ
/        JMP .+6
/        .DSA ADBWD4     /ADDRESS OF WD4 OF ARRAY DESCRIPTOR BLOCK
                         /OF ARRAY PASSED AS A DUMMY PARAMETER
/        .DSA B          /ADDRESS OF LOCATION IN THE ABOVE ARRAY AT
                         /WHICH THE ADJUSTED ARRAY IS TO BEGIN
/        .DSA K1         /ADDRESS OF NEW MAXIMUM FIRST SUBSCRIPT
/        .DSA K2         /ADDRESS OF NEW MAXIMUM SECOND SUBSCRIPT
/        .DSA K3         /ADDRESS OF NEW MAXIMUM THIRD SUBSCRIPT
/
         .GLOBL ADJ,.DA,.AD,.SS
ADJ      Ø
         JMS* .DA        /GET ARGUMENT ADDRESS
         JMP  .+6        /JUMP AROUND PARAMETER LIST
ARRAY    Ø
B        Ø
K1       Ø
K2       Ø
K3       Ø
```

Corrected version of ADJ to remove incompatibility with FORTRAN Object Time System (Cont'd)

```
        LAC* K2         /SPECIAL CONSIDERATION IS MADE FOR CASE OF
        SNA             /ADJUSTING A SINGLE DIMENSIONED ARRAY TO 2 OR
        JMP F+1         /MORE , SINCE .SS ØØ8 WILL THEN FAIL IN COMPUTING
        LAC .SS         /B.  IF K2 IS ZERO, WE'RE OK, AS THERE IS ONLY
        TAD (34         /ONE DIMENSION TO ADJUST.  ELSE THE LAST PART OF
        DAC B           /.SS IS REDONE.  B IS USED TEMPORARILY TO STORE
        LAC (F          /ADDRESS AT WHICH TO ENTER .SS.  AND F IS SET
        DAC* .SS        /AS THE ADDRESS .SS WILL RETURN TO.
        JMP* B          /THUS THIS JMP* IS TO .SS+34.  RETURNED IN THE
F       DAC B           /AC IS THE CORRECT ADDRESS OF THE NEW LOCATION.
/
        LAC (LAC* B     /INITIALIZE SUBSCRIPT POINTERS
        DAC C
        LAC B           /SET NEW STARTING ADDRESS
        DAC* ARRAY
        LAW -3
        DAC CTR#        /COUNT FOR 3 SUBSCRIPTS
        TAD ARRAY       /COMPUTE ADDRESS TO FIRST WORD OF ARRAY DESCRIPTOR
        DAC ARRAY       /BLOCK IN THE PROGRAM IN WHICH THE ARRAY IS
        DAC ARRAYP#     /DEFINED.  THE ARRAY TYPE IS IN BITS 3 & 4 OF THIS
        LAC* ARRAY      /WORD.
        AND (6ØØØØ      /ZERO OUT THE ARRAY SIZE
        DAC* ARRAY      /SAVE CLEAN ARRAY TYPE
        LRSS 15         /RIGHT JUSTIFY THE MODE, AND DETERMINE LENGTH
        TAD (1          /OF THE VARIABLE TYPE.  LENGTH = MODE + 1, EXCEPT
        AND (3          /FOR DOUBLE INTEGER
        SNA
        LAC (2
LOOP    ISZ C           /BUMPS TO LAC* K1, THEN K2, THEN K3
        JMS* .AD        /MULTIPLY INTEGERS - CURRENT AC CONTENTS TIMES
C       XX              /K1, K2, OR K3, WHERE LAC* B WAS DAC'D, THEN
        SNA             /ISZ'D ABOVE.  WHEN THE RESULT IS ZERO, HAVE
        JMP D           /RUN OUT OF SUBSCRIPTS.
        DAC SIZE#       /UPDATE CUMULATIVE SIZE.
        ISZ CTR         /ARE 3 SUBSCRIPTS YET TREATED?
        SKP             /NO - GO STORE SUCCESSIVE SIZES
        JMP E           /YES - GO CLEANUP
```

Corrected version of ADJ to remove incompatibility with FORTRAN Object Time System (Cont'd)

```
/
          ISZ ARRAYP      /STORE SUCCESSIVE SIZES INTO APPROPRIATE WORDS OF
          DAC* ARRAYP     /ARRAY DESCRIPTOR BLOCK
          JMP LOOP
/
D         DZM* ARRAYP     /WHEN HAVE RUN OUT OF SUBSCRIPTS, ZERO THE
          ISZ ARRAYP      /REMAINING WORDS OF THE ADB.
          ISZ CTR         /CHECK FOR FINISH
          JMP LOOP
/
E         LAC SIZE        /CLEANUP - PACK THE SIZE AND THE MODE BITS INTO
          AND (17777      /THE FIRST WORD OF THE ADB.
          XOR* ARRAY
          DAC* ARRAY
          JMP* ADJ
          .END
```

SOFTWARE DISPATCH

DOS-15

August, 1972

FORTRAN COMPILER (EDIT #36) F4X V36B, FPF4X V36B AND F4X9 V36B

Patch to correct improper imbedded subscript calculation

PROBLEM:

F4 Ø36 generates improper subscript calculating code for A(J(I)) with a non-integer array.

SOLUTION:

The following patches, one for each of the three compilers, correct this problem. Use the "LR" command when patching FORTRAN or any relocatable file.

	LOCATION	OLD CONTENTS	NEW CONTENTS	SYMBOLIC
F4X	6546	1Ø64ØØ	2Ø64ØØ	'V36B' 15
	7736	2Ø7724	#+615435	JMP PATCH
	7737	112754	#+21Ø231	LAC TSI
	774Ø	113247	#+Ø55443	DAC TSITMP
	7741	Ø4ØØØ7	#+215443	LAC TSITMP
	7742	11Ø1ØØ	#+Ø5Ø231	DAC TSI
	7747	6Ø7743	#+6Ø7741	JMP .-6
	15435	Ø	#+2Ø7724	LAC EXSBMD
	15436	Ø	#+112754	JMS SETN
	15437	Ø	#+113247	JMS TWOCMA
	1544Ø	Ø	#+Ø4ØØØ7	DAC SSCTR
	15441	Ø	#+11Ø1ØØ	JMS FPPOUT
	15442	Ø	#+6Ø7737	JMP BACK
	15443	Ø	Ø	-

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN COMPILER	V36B	36
Subprogram or Additional Information	Sequence #	PAGE
F4X, FPF4X, F4X9	1	1 OF 2
New [X] Replacement Article [ ]	Original Date	

Patch to correct improper imbedded subscript calculation (Cont'd)

	LOCATION	OLD CONTENTS	NEW CONTENTS	SYMBOLIC
FPF4X	66Ø7	333202	3332Ø4	'V36B' 15
	1ØØØ4	2Ø7772	#+615714	JMP PATCH
	1ØØØ5	113Ø72	#+21Ø344	LAC TSI
	1ØØØ6	113365	#+Ø55722	DAC TSITMP
	1ØØØ7	Ø4ØØØ7	#+215722	LAC TSITMP
	1ØØ1Ø	11Ø213	#+Ø5Ø344	DAC TSI
	1ØØ15	61ØØ11	#+61ØØØ7	JMP .-6
	15714	Ø	#+2Ø7772	LAC EXSBMD
	15715	Ø	#+113Ø72	JMS SETN
	15716	Ø	#+113365	JMS TWOCMA
	15717	Ø	#+Ø4ØØØ7	DAC SSCTR
	1572Ø	Ø	#+11Ø213	JMS FPPOUT
	15721	Ø	#+61ØØØ5	JMP BACK
	15722	Ø	Ø	-
F4X9	6533	64Ø432	641Ø32	'V36B' 15
	7716	2Ø77Ø4	#+615412	JMP PATCH
	7717	112742	#+21Ø211	LAC TSI
	772Ø	113235	#+Ø5542Ø	DAC TSITMP
	7721	Ø4ØØØ7	#+21542Ø	LAC TSITMP
	7722	11ØØ6Ø	#+Ø5Ø211	DAC TSI
	7727	6Ø7723	#+6Ø7721	JMP .-6
	15412	Ø	#+2Ø77Ø4	LAC EXSBMD
	15413	Ø	#+112742	JMS SETN
	15414	Ø	#+113235	JMS TWOCMA
	15415	Ø	#+Ø4ØØØ7	DAC SSCTR
	15416	Ø	#+11ØØ6Ø	JMS FPPOUT
	15417	Ø	#+6Ø7717	JMP BACK
	1542Ø	Ø	Ø	-

FORTRAN COMPILER (EDIT #36) F4X V36B, F4X9 V36B, FPF4X V36B

Errors not detected during compilation

PROBLEM:

The following errors have been found to occur during compilation using either the F4X V36B, F4X9 V36B, or FPF4X V36B compiler. These will be corrected in subsequent releases of the compiler.

1. Unbalanced parentheses in subroutine calls of the form

   CALL LINE ((-MX(I),Ø,1)

   are not trapped as an error, and bad object code is generated.

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

SOLUTION:

1. Solution not yet available.

2. To cure problem 2, use an explicit mode declaration for these variables.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN COMPILER	V36B	36
Subprogram or Additional Information	Sequence #	PAGE
F4X, F4X9, FPF4X	2	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

APRIL 1973

FORTRAN COMPILER (EDIT #36) F4X V36B, FPF4X V36B, F4X9 V36B

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 in the case 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 normally generate an Integer result. No correction is published presently, although a rectification to this will be included in the next compiler release.

RESTRICTION:

For now, 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 invoked in some program, do not declare it in 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	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN COMPILER	V36B	36
Subprogram or Additional Information	Sequence #	PAGE
F4X, FPF4X, F4X9	3	1 OF 1
New [X] Replacement Article [ ]	Original Date	

FORTRAN COMPILER (EDIT #36) F4X V36B, FPF4X V36B, F4X9 V36B

Incomplete DO Statement difficulties

PROBLEM:

Statements of the form

DO 2Ø I = 1

which are in fact an error, the DO statement not being complete, are not flagged by the compiler properly. When the total number of non-blank characters between column 7 and the equal sign does not exceed 6, no error is noted. The compiler, in fact, codes the assignment statement DO2ØI=1, due to its eliminating of all blanks before processing statements.

It also occurs that when the total number of non-blank characters between column 7 and the equal sign exceeds 6, the error indicating a string of more than six characters occurs. For example, the compiler tries to interpret

DO 2ØØØ I = 1

as DO2ØØØI = 1, which is an invalid assignment statement.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN COMPILER	V36B	36
Subprogram or Additional Information	Sequence #	PAGE
F4X, FPF4X, F4X9	4	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

July, 1972

FORTRAN OTS

Programming note to eliminate carriage return

There is a FORTRAN level trick to 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 re-init by FIOPS). Also, the read .DAT slot must have been previously INITed (done for example a REWIND).

Example:

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

response

This results in a FOCAL type read, viz, ENTER NUMBER PLEASE:_____

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN OTS	N/A	N/A
Subprogram or Additional Information	Sequence #	PAGE
	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

July, 1972

FORTRAN OTS ROUTINE .DA (EDIT #Ø11)

Programming note

Because of the multiple entry feature in F4X, the argument address fetch subroutine .DA does double ended double indirection when 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 suddenly cease to function under F4X.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN OTS	N/A	11
Subprogram or Additional Information	Sequence #	PAGE
.DA	2	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

December, 1972

FORTRAN OTS ROUTINES AMOD AND DMOD

Numerical Restrictions

In the existing FORTRAN documentation there is not 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.

This restriction can be relieved somewhat by the following example. (The following 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
```

Explicitly state this program's 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.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN OTS	N/A	
Subprogram or Additional Information	Sequence #	PAGE
AMOD, DMOD	3	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

MARCH 1973

FORTRAN OTS ROUTINE DDIO (Edit #14), FPP VERSION

Reading INTEGER variables causing IOPS Ø

PROBLEM:

The following error has been found to occur during the execution of FORTRAN compiled programs using the FPF4B V36B compiler. Corrections will appear in subsequent releases of this compiler.

1. Reading in INTEGER variables (not arrays) using DDIO, Edit #14 may cause an IOPS Ø, although it cannot in general be predicted when this situation will arise. To avoid the problem, read the element into a DOUBLE INTEGER variable, then convert it to an INTEGER by an assignment statement or use of the ISNGL function. This problem could conceivably show up when using non-FPP versions, but has not been found to occur in test situations.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
FORTRAN OTS	N/A	14
Subprogram or Additional Information	Sequence #	PAGE
DDIO	4	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

LPA.15 (Edit #Ø44)

Source Change to Prevent Return to Mainstream, at API Level 4 on .WAIT

PROBLEM:

The current LPA.15 handler does not do a DBR on return from a successful .WAIT system MACRO. Therefore, the calling program is entered at API level 4.

SOLUTION:

The problem is corrected by editing the source (LPA.15 Edit #Ø43) as indicated below.

Location	Old Contents	New Contents
LPWAT1+1	JMP* LPARGP	JMP LPNEXT

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
LINE PRINTER HANDLERS	N/A	44
Subprogram or Additional Information	Sequence #	PAGE
LPA.15	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

LINKING LOADER V12A (Edit #Ø23)

Features of this Version

The Linking Loader will load programs in whatever addressing mode is current to the system. The BANK and PAGE Nonresident Monitor commands set the addressing mode. This loader identifies itself as LOADER V12A for page mode loads, and BLOADER V12A for bank mode loads.

There is a new Linking Loader code, CODE=33, which indicates the extension of the source file used to create the binary file via MACRO-15. When the "P" switch is used with the Linking Loader, the loader will print both the Object File's name, and the Source file's extension, for each file loaded. This allows programmers to identify different versions of their binary files.

The loader has been modified to allow arbitrary numbers of positive .DAT slots (as determined at system generation, up to 77 octal), and it will compute the position of .DAT from .SCOM+23.

The Linking Loader searches the 'IOS' UIC, the user's library, if one is present, and the system library as many times as is necessary to resolve .GLOBL's.[1] If a complete pass through them yields no new resolution, the Loader tries to match the missing .GLOBL's to Common Blocks. If some missing .GLOBL's still remain, a LOAD 3 is generated. These searches terminate early, if all .GLOBL's are resolved. The effect of this change is to allow backward references in user libraries, and to allow user files to reference the IOS and the system library.

---

[1] IOS, User Library, System Library, IOS;
(Test for resolution on last pass)
User Library, System Library, IOS;
(Test) etc.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
LINKING LOADER	V12A	23
Subprogram or Additional Information	Sequence #	PAGE
	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

LKA. (EDIT #ØØØ)

Functional Description

1. Overall Description

The LK35 Keyboard Device Handler provides an interface between the user and the hardware. In general, it conforms to the conventions of the Disk Operating System, as described in DEC manual, DOS-15 User's Manual, DEC-15-ODUMA-A-D. Since the LK35 keyboard is a send only device, this handler will only handle input functions. The input functions are initiated by standard user program commands and all interrupt management is done automatically by the handler. The handler does not make the LK35 a console control keyboard; it is only an input device.

The primary goals of the handler are to relieve the user from writing his own device-handling subprograms and to centralize all direct communication between the PDP-15 and the LK35 keyboard. This handler will only input IOPS ASCII or IMAGE ASCII into a user's designated buffer. It is up to the user to display the text on the VTØ4, or write the text on any other device. The LK35 will be connected to the LT15 or the LT19D. Only one LK35 under DOS is supported.

Equipment required includes a PDP-15/20 with a

VT15
VTØ4 or VTØ7
LK35
LT19D or LT15

Functional Description (Cont'd)

Legal System Macros	Illegal Functions	Ignored Functions
.INIT	.WRITE	.SEEK
.READ	.DLETE	
.WAIT	.RENAM	
.WAITR	.ENTER	
.CLOSE	.CLEAR	
.FSTAT	.MTAPE	
	.TRAN	

2. Legal Functions

2.1 .INIT (initialize) Macro

The .INIT causes the keyboard to be initialized and must be called before any other I/O Macro to this device is issued.

a. Form .INIT␣[-]ds,dd,restrt

b. Variables

ds = .DAT slot number
dd = ignored - may be any number between Ø and 3
restrt = CTRL P address

c. Expansion

```
LOC+Ø  CAL+1ØØØ*dd␣[-]ds&777
LOC+1  1
LOC+2  restrt
LOC+3  ØØØØØØ /standard size of buffer (34₁Ø) will be
              /returned
```

d. Description

1. Handler will return standard line buffer size (34_{10})
2. .INIT will abort a .READ

Functional Description (Cont'd)

2.2 .READ Macro

The .READ macro is used to transfer data from the LK35 to core memory.

a. Form .READ␣[-]ds,m,bufadd,wc

b. Variables

ds = .DAT slot number
m = Data mode
2 = IOPS ASCII
3 = IMAGE ASCII
bufadd = Line buffer address
wc = Line buffer word count, including the two-word header.

c. Expansion

```
LOC+Ø  CAL+1ØØØ*m␣[-]ds&777
LOC+1  1Ø
LOC+2  bufadd
LOC+3  -wc
```

d. Description: .READ will

1. Allow previous input to terminate.
2. Set input underway indicator.
3. Set up to accept characters from keyboard.
4. Accept data control characters in IOPS ASCII.
   a) Rubout - delete previous character typed.
   b) ↑U - delete entire line typed so far.
5. Carriage Return or ALT MODE terminates an IOPS ASCII read.
6. The word count terminates an IMAGE ASCII read.

Functional Description (Cont'd)

2.3 .WAIT Macro

The .WAIT macro is used to detect the availability of the user's line buffer.

a. Form .WAIT␣[-]ds

b. Variable

ds = .DAT slot number

c. Expansion

```
LOC+Ø  CAL␣[-]ds&777
LOC+1  12
```

d. Description

1. Control is returned to the user immediately after the expansion, if the line buffer is available.
2. If transfer of data has not been completed, control loops on the .WAIT macro.

2.4 .WAITR Macro

The .WAITR macro allows the user program to proceed in line if the previous .READ is complete. If the previous .READ is not complete control is given to the location in the user program specified by the .WAITR call. This allows the user to branch to some other part of his program while waiting for the .READ to finish. The user must continue to check for completion by periodically issuing .WAITR's or by using a .WAIT.

a. Form .WAITR␣[-]ds,waitad

b. Variables

ds = .DAT slot number

Functional Description (Cont'd)

waitad = location in the user program to branch to if input is not completed.

c. Expansion

```
LOC+Ø  CAL+1ØØØ␣[-]ds&777
LOC+1  12
LOC+2  waitad
```

2.5 .CLOSE Macro

a. Form .CLOSE␣[-]ds

b. Variables

ds = .DAT slot number

c. Expansion

```
LOC+Ø  CAL␣[-]ds&777
LOC+1  6
```

d. Description

Same as .WAIT

2.6 .FSTAT Macro

a. Form .FSTAT␣[-]ds,address

b. Variables

ds = .DAT slot number
address = ignored

Functional Description (Cont'd)

c. Expansion

```
LOC+Ø  CAL+3ØØØ␣[-]ds&777
LOC+1  2
LOC+2  address
```

d. Description

1. Will return a zero in the AC, since this since this device is non-file oriented.

3. Legal Control Characters

a. Control C (↑C) does a .EXIT to the Monitor.

b. Control P (↑P) transfers control to the address given in the .INIT CAL.

c. Control D (↑D) gives an End-of Medium Header Word Pair to the user.

MACRO-15 V7A (EDIT #Ø82)

New Switches in this Version

The Omit (O) switch

The O switch will cause MACRO-15 to omit the source extension and the new Linking Loader code (code=33) from the binary file. This should be used when assembling programs to be run on other PDP-15 systems, in particular, BACKGROUND/FOREGROUND, and ADVANCED SOFTWARE SYSTEM.

The Error (E) switch

The E switch allows programmers to have any assembly errors printed on the console teleprinter, in addition to the device assigned to .DAT-12. The E switch is useful only if the L or N switches are also used.

This will help programmers who assign DECtape or Disk to .DAT-12, but want to know where any error lines are. Previously, such a situation would require two assemblies: One to find any errors (e.g.:←FIL↵), that is, no switches), and one other if no errors occurred, this time using the N or L switches.

The Table of Contents (T) switch

If "T" is typed in the command string, a table will be generated during Pass one with the page number and the text of all assembled .TITLE statements in the program. This file will be generated with the program listing name on .DAT-12. Note that it is still necessary to type N, L, etc. if other listing output is required.

An assumption about switches:

If the L and X options are typed, MACRO-15 will assume that the N-option was typed also. This will help the user who typically forgets to type "N" and gets a cross reference that is effectively useless, because the source lines of the listing are not numbered.

SOFTWARE DISPATCH

DOS-15

PRE-1973

MACRO-15 V7A (EDIT #Ø82)

Macros Implemented Specifically for DOS-15

MACRO-15 for DOS implements .RAND, .RTRAN, .USER, .OVRLA, .GTBUF and .GVBUF, as per Appendix B-1 and B-2 of the DOS PRELIMINARY MANUAL (DEC-15-MZDA-D). MACRO accepts .ENTER macros with file protect codes. MACRO-15 has not implemented Macro calls for .GET and .PUT.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
MACRO-15	V7A	82
Subprogram or Additional Information	Sequence #	PAGE
	2	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

MACRO-15 V7A (EDIT #Ø82)

Pseudo-op, .CBD

The Macro Assembler pseudo-op .CBD (Common Block Definition) allows the assembly language programmer to declare a COMMON of an indicated name and size, and to specify a word to be set to its base address.

The .CBD pseudo-op takes a COMMON name and its size as arguments, reserves one word of core, and outputs loader codes and parameters to direct the Linking Loader or CHAIN to set a vector to the first element of the indicated COMMON in the reserved word. For example, the statement

```
BASE .CBD ABCD 6
```

will provide the base address of COMMON/ABCD/ in the word labelled BASE.

Attached is a sample subroutine and an expansion of its assembled output. The Ø7, 1Ø, 14, 15, 16 (octal) loader code sequence is the result of the .CBD pseudo-op. The Ø7-1Ø-14 codes declare a COMMON named ABCD, and the 15-16 codes indicate that a vector to location ØØØØ in the COMMON is to be set in location ØØ12 of the routine being relocated.

NOTE: The .CBD pseudo-op must not be the first line of code.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
MACRO-15	V7A	82
Subprogram or Additional Information	Sequence #	PAGE
	3	1 OF 3
New [X] Replacement Article [ ]	Original Date	

Pseudo-op, .CDB (Cont'd)

```
PAGE   1      SIGMA  SRC

                          /SIGMA -- FORTRAN CALLABLE SUBROUTINE
                          /
                                  .GLOBL  SIGMA
                          /
    00000 R 000000 A      SIGMA   0
    00001 R 200013 R              LAC     (5)
    00002 R 722000 A              PAL
    00003 R 735000 A              CLX
    00004 R 750000 A              CLA
                          /
    00005 R 370012 R      LOOP    TAD*    BASE,X
    00006 R 725001 A              AXS     +1
    00007 R 600005 R              JMP     LOOP
                          /
    00010 R 070012 R              DAC*    BASE,X
    00011 R 620000 R              JMP*    SIGMA
                          /
    00012 R 000000 A      BASE    .CBD ABCD 6     /BASE OF COMMON BLOCK "
                          /
            000000 A              .END
    00013 R 000005 A *L
SIZE=00014      NO ERROR LINES
```

Pseudo-op, .CDB (Cont'd)

```
NAME = SIGMA
BIAS =
   01  000014
   07  474057              SIG
   10  050550              MA@
   12  000000              SIGMA@=00000
   07  474057              SIG
   10  050550              MA@
   33  074623              SRC
   23  400000              SIGMA@=00000
   02  000000
   04  000000      00000   CAL     00000
   03  200013      00001   LAC     00013
   04  722000      00002   IOT     02000
   04  735000      00003   IOT     15000
   04  750000      00004   OPR     10000
   03  370012      00005   TAD*    10012
   04  725001      00006   IOT     05001
   03  600005      00007   JMP     00005
   03  070012      00010   DAC*    10012
   03  620000      00011   JMP*    00000
   07  403223              ABC
   10  014400              D@@
   14  000006
   15  000000
   16  000012
   04  000000      00012   CAL     00000
   02  000013
   04  000005      00013   CAL     00005
 ? 07  406273              BAS
   10  017500              E@@
   23  000012              BASE@@=00012
   07  446547              LOO
   10  062000              P@@
   23  000005              LOOP@@=00005
   07  474057              SIG
   10  050550              MA@
   23  000000              SIGMA@=00000
   27  000000              .END    0000
```

Software Product		Version
DOS-15		V1A
Component	Version	Edit #
MACRO-15	V7A	82
Subprogram or Additional Information	Sequence #	PAGE
	3	3 OF 3
New [X] Replacement Article [ ]		Original Date

MACRO-15 V7A (EDIT #Ø82)

Note on Binary Output File Extensions

MACRO-15 will include the extension of the source file in the binary output file. This means that programmers may use a numeric source file extension and increment it with each edit. (Other extensions are also legal, of course.) Then the P switch in the Linking Loader will produce the edit number of the current core load -- an invaluable identification tool in development programming. UPDATE's L command will also list the extension of the source file for each program in a library.

MACRO-15 accomplishes this by using a new Linking Loader code, code=33, as shown in the example on the next page.

Other PDP-15 systems, in particular, the ADVANCED SOFTWARE SYSTEM and BACKGROUND/FOREGROUND, will not work with the 33 Loader code. If programmers wish to assemble source files for such systems with this version of MACRO-15, they should use the "O" (omit) switch, which will omit the source file extension and the 33 Linking Loader code from the binary file.

Note on Binary Output File Extensions (Cont'd)

EXAMPLE:

PROGRAM LISTING

```
PAGE 1                    J.BOND ØØ7

                                        /SHOW NEW LINK LOADER CODE
                                        /AND WHERE IT IS PLACED
                                        /
                                        /EDIT* ØØ7  9·28·71
                                        /
        ØØØØØ R 2ØØØØ1 R                        LAC A
        ØØØØ1 R 74ØØØØ A                A       NOP
                ØØØØØØ A                        .END

SIZE=ØØØØ2      NO ERROR LINES


DIRECTORY LISTING
-----------------
>L TT←DK

  28-SEP-71
DIRECTORY LISTING
767 FREE BLKS
  1 USER FILES
11Ø SYSTEM BLKS
J.BOND BIN        1         1
```

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
MACRO-15	V7A	82
Subprogram or Additional Information	Sequence #	PAGE
	4	2 OF 3
New [X] Replacement Article [ ]	Original Date	

Note on Binary Output File Extensions (Cont'd)

BINARY FILE DUMP

WORD	CONTENT	
Ø	Ø1Ø71Ø	3 LINK LOADER CODES (SIX BITS EACH)
1	ØØØØØ2	PROGRAM SIZE
2	441342	J.B = FIRST SIX CHARACTERS OF
3	Ø57764	OND = NAME IN RADIX 5Ø
4	332302	3 LINK LOADER CODES
5	226067	EXTENSION NAME IN RADIX 5Ø
6	4ØØØØØ	
7	ØØØØØØ	
1Ø	Ø3Ø4Ø7	
11	2ØØØØ1	FIRST INSTRUCTION
12	74ØØØØ	SECOND INSTRUCTION
13	ØØ31ØØ	

(REST OF GENERATED CODE NOT SHOWN)

MACRO-15 V7A (Edit #Ø82)

Problems corrected

The following MACRO-15 problems that existed in the Advanced Software System version have been corrected:

1. CAL's were incorrectly treated under the following circumstances:

   In page mode, when a programmer wrote a CAL combined with an operator (e.g., CAL+1ØØØØ ADDRES)

   The leftmost six bits were stripped off, so that CAL+ØØØØØ ADDRES was produced.

2. The .LST and .NOLST pseudo-ops did not work correctly where .NOLST occurred without being followed by a .LST before the end of the program. It would put the program in the .NOLST state at the beginning of the second pass.

3. Forward references in literals caused literal phase errors during pass two.

4. The .LOCAL and .NDLOC pseudo-ops caused a symbol to be lost whenever a break in alphabetic order occured in a list of tags.

   e.g.:
   ```
   XXX=1
   YY=2
   Z=3
   X1=4ØØØ
   ```

   The X1 would clobber the YY symbol when stored in the user's symbol table.

5. If the F and X switches were used together, the wrong file name was passed to CREF by MACRO in pass three. This resulted in a fatal IOPS 13 message during Pass three.

Problems corrected (Cont'd)

6. MACRO-15 lost some literals during PASS one in a 28K environment. This caused internal pointers to be out of phase. When these pointers were passed on to CREF (Pass three), CREF would wipe out part of lower core.

7. The .EBREL and .DBREL pseudo-op now change addressing modes in .ABS(P) and .FULL(P) programs.

8. The following sequence assembled correctly, but caused E errors. The error message is no longer produced. e.g.,

```
    .LOC   6ØØØØ
    LAW    -1
    .LOC   7ØØØØ
    LAW    -1
```

SOFTWARE DISPATCH

DOS-15

PRE-1973

MACRO-15 V7A (EDIT #Ø82)

Programming Note on Usage 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	
DOS-15	V1A	
Component	Version	Edit #
MACRO-15	V7A	82
Subprogram or Additional Information	Sequence #	PAGE
	6	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

MACRO-15 V7B (EDIT #Ø82)

Patch to Expand .RTRAN Correctly

PROBLEM:

MACRO-15 V7A expands the .RTRAN system macro with a positive rather than negative word count.

SOLUTION:

The following patch to MACRO corrects this error. The source code notation to the right need not be keyed in; it is shown only for information purposes.

LOCATION	OLD CONTENTS	NEW CONTENTS	SYMBOLIC
3272	426Ø61	426Ø61	.ASCII /EC/<15>
3273	5Ø44Ø2	5Ø4532	<11>/-/
3274	Ø24337	ØØ4121	.ASCII <ØØ1><ØØ5>
3275	7ØØØØØ	5774ØØ	<15><177>
7461	1Ø64ØØ	2Ø64ØØ	V7B

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
MACRO-15	V7B	82
Subprogram or Additional Information	Sequence #	PAGE
	7	1 OF 1
New [X] Replacement Article [ ]	Original Date	

CREF V7C (EDIT #Ø16)

Patch to correct printing of erroneous message

PROBLEM:

When the command string to MACRO ends with a carriage return, CREF prints out the message: "PROGRAM NOT IN COMBLK".

SOLUTION:

The following patch to CREF corrects this problem.

LOCATION	OLD CONTENTS	NEW CONTENTS	NEW SYMBOLIC
17136	1714Ø	17137	.+1

To change the version number patch the following location in MACRO:

LOCATION	OLD CONTENTS	NEW CONTENTS	NEW SYMBOLIC
7461	2Ø64ØØ	3Ø64ØØ	V7C

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
MACRO-15	V7C	16
Subprogram or Additional Information	Sequence #	PAGE
CREF	8	1 OF 1
New ☐ Replacement Article ☑ 1	Original Date	
	Pre-1973	

CREF (EDIT #Ø18)

Solutions to miscellaneous problems

1. Problem - Outputs one line too many for the VP15 storage scope.

   Correction - Output one line less on each page.

2. Problem - Does not detect symbols occurring after an operator in an expression. This was because CREF as soon as it saw a number after the equal sign in an expression terminated processing of current line.

   Correction - Check and make sure that the operand after the operator is a number before processing the next line; in the event the operand is a symbol, update the symbol table.

3. Problem - Did not handle line numbers beyond $9999_{1Ø}$. They were converted modulo $1ØØØØ_{1Ø}$.

   Correction - Handles line numbers beyond $9999_{1Ø}$.

Attached is a copy of the modified portions of the SRC listing. Changes are indicated by underscored and enclosed instructions.

Users have to make note that in order to patch the new CREF (edit #Ø18) into the system, MACRO-15 V7A, edit #Ø82 has to be modified as follows:

change P3ARGS=1761Ø on p. 5 of the SRC listing to
P3ARGS=1762Ø

MACRO has then to be reassembled and patched into the system.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
MACRO-15	N/A	18
Subprogram or Additional Information	Sequence #	PAGE
CREF	9	1 OF 5
New [X] Replacement Article [ ]	Original Date	

Solutions to miscellaneous problems (Cont'd)

The above becomes necessary due to the increase in size of CREF. If the above is not done, users are assured that CREF will not run.

NOTE WELL

The following source code modifications do not encompass the program correction to CREF described as a binary patch in the preceding article. Also, subsequent CREF articles with patches will show corrections to the original version of CREF (edit #Ø16) rather than to this one.

```
PAGE 11       CREF      Ø18      CREF

4Ø6                     Ø17642       LSTLIN=EXPAGE+1
4Ø7                     Ø17643       LSTPGE=LSTLIN+1
4Ø8                     Ø17644       MULTIN=LSTPGE+1
4Ø9          16236      ØØØØØØ       OVRL1    Ø
41Ø          16237      777713       LINCNT   =65
411                                  /
412                                  /CHARACTER CONSTANTS
413                                  /        .OCT
414          1624Ø      ØØØØ4Ø       SPACE    4Ø
```

Solutions to miscellaneous problems (Cont'd)

```
PAGE 15    CREF    Ø18        CREF

558                           /SUBROUTINE TAGPRO
559                           /
56Ø                           /TAGPRO RECEIVES CONTROL IF THE FIRST
561                           /OF A STATEMENT IS AN ALPHA OR DOT
562                           /DIRECT ASSIGNMENTS ARE ALSO HANDLED HERE
563                           /
564    16426    ØØØØØØ        TAGPRO  Ø                          /


588    16456    116746                JMS     FALPHA            /IS IT AL
589    16457    1175Ø3                JMS     ADRPRO            /YES, TRE
59Ø    1646Ø    116764                JMS NUMERC        /NO, CHECK IF IT
591    16461    616463                JMP CHKSYM        /YES, NOW CHECK IT
592    16462    616154                JMP     NXTLIN            /NO
593                           /
594                           /
595    16463    1166Ø7    CHKSYM  JMS GETCHR
596    16464    116764            JMS NUMERC        /CHECK IF CHARACTE
597    16465    616463            JMP CHKSYM        /YES, LOOP
598    16466    1167Ø2            JMS ADRDEL        /NO, CHECK IF NEXT
599    16467    616455            JMP TAGDIR+1      /YES, CHECK FOR S
6ØØ    1647Ø    616154            JMP NXTLIN        /NO, GO TO NEXT L
6Ø1                           /
6Ø2                           /
6Ø3    16471    1166Ø7    TAGEND  JMS GETCHR        /EDIT #16
6Ø4             1166Ø7    /
```

Solutions to miscellaneous problems (Cont'd)

```
PAGE 26    CREF    Ø18        CREF

1Ø18                          /
1Ø19   17174   ØØØØØØ         UNDEC   Ø                          /
1Ø2Ø   17175   157313                 DZM     OINIT              /FLAG LEAD
1Ø21                                  .IFUND DOS15
1Ø22                                  DAC OVRL1
1Ø23                                  JMS DIVIDE
1Ø24                                  LAC OVRL1
1Ø25                                  LAC DTENTH
1Ø26                                  .ENDC
1Ø27                                  .IFDEF DOS15
1Ø28   17176   673323                 IDIV!SHAL
1Ø29   17177   Ø2342Ø          DTENTH  2342Ø              /1ØØØØ DEC.
1Ø3Ø   172ØØ   Ø56236                 DAC OVRL1
1Ø31   172Ø1   641ØØ2                 LACQ
1Ø32                                  .ENDC
1Ø33   172Ø2   11723Ø                 JMS UNSEND
1Ø34                                  .IFUND  DOS15
1Ø35                                  DAC     OVRL1              /
```

```
PAGE 27      CREF    Ø18      CREF

1Ø4Ø                                  .IFDEF  DOS15
1Ø41    172Ø3   216236                LAC OVRL1
1Ø42    172Ø4   673323                IDIV!SHAL
1Ø43    172Ø5   ØØ175Ø                175Ø     /1ØØØ DEC
1Ø44    172Ø6   Ø56236                DAC      OVRL1
```

Solutions to miscellaneous problems (Cont'd)

```
PAGE 28      CREF   Ø18     CREF

1Ø93   17253   516312            AND   L77777
1Ø94   17254   Ø57174            DAC   UNDEC

1113   17276   2173Ø3   MACRO    LAC   MACR   /PRINT 'MACRO' IN
1114   17277   117151            JMS   UNSIX          /OCTAL VA
1115   173ØØ   2173Ø4            LAC   MACR+1
1116   173Ø1   117151            JMS   UNSIX
1117   173Ø2   617273            JMP MACRO-3
1118   173Ø3   4Ø15Ø1   MACR     .SIXBT ' MACRO'
       173Ø4   Ø32217                    -

1122   173Ø5   ØØØØØØ   OINIT    Ø
1123   173Ø6   777773            LAW    -5
1124   173Ø7   Ø56633            DAC    GETPTR
1125   1731Ø   216272            LAC I8 /REINITIALIZE POINTER
1126   17311   Ø57321            DAC OPACK+2
1127   17312   216317            LAC     LINBF%
1128   17313   Ø56367            DAC     CHRCNT
1129   17314   77777Ø            LAW     -1Ø    /8 LINE NUMBERS P
113Ø   17315   Ø5755Ø            DAC     OFLAG
1131   17316   6373Ø5            JMP*    OINIT

PAGE 3Ø      CREF   Ø18     CREF

1198   17411   637365            JMP*    OWRITE
1199   17412   637365            JMP*    OWRITE          /RETURN


1212   17421   117174            JMS     UNDEC
1213   17422   777775            LAW     -3
1214   17423   Ø56746            DAC     ALPHA
1215   17424   21624Ø            LAC     SPACE

1236   17451   ØØØ766            CAL+766
1237   17452   ØØØØ12            12
1238   17453   777713            LAW     -65
1239   17454   Ø56237            DAC     LINCNT
124Ø   17455   637413            JMP*    LINHDR
```

MACRO-15 V7D (EDIT #Ø82)

Leading commas in command string foiled BOSS-15 operation

PROBLEM:

In MACRO V7C, leading commas are not ignored in the input command string. This prevents successful operation under BOSS-15.

SOLUTION:

Using PATCH, make the following binary corrections to MACRO:

LOCATION	OLD CONTENTS	NEW CONTENTS	NEW SYMBOLIC	COMMENTS
7247	154431	JMP 17566	JMP PATCH	/PATCH AREA
17566	Ø	SAD 17572	SAD COMMA	/IS IT A COMMA?
17567	Ø	JMP 7244	JMP CMDSUB-5	/YES-GET CHAR.
1757Ø	Ø	DZM 14431	DZM PRVCHR	/NO, RESTORE
17571	Ø	JMP 725Ø	JMP CMDSUB-1	/AND CONTINUE
17572	Ø	54		/.ASCII FOR COMMA
7461	3Ø64ØØ	4Ø64ØØ		/V7D

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
MACRO-15	V7D	82
Subprogram or Additional Information	Sequence #	PAGE
	1Ø	1 OF 1
New ☐ Replacement Article [1]	Original Date	
	Pre-1973	

MACRO-15 V7E (EDIT #Ø82)

Problem with parameter file name being assigned to listing file

PROBLEM:

MACRO-15 assigns the parameter file name (when present on a file-oriented device) to the listing file on the file-oriented device assigned to DAT-12.

SOLUTION:

Use PATCH to make the following binary corrections to MACRO. Note that the source code notation on the right need not be keyed in. It is shown only for information purposes.

```
PATCH  Vxx
>MACRO
>L 17114
>17114/2Ø7Ø1Ø>2Ø7Ø21       LAC  OUTNAM
>17115/Ø571Ø1>
>17116/2Ø7Ø11>2Ø7Ø22<ALT> LAC  OUTNAM+1
>L 7461
>Ø7461/4Ø64ØØ>5Ø64ØØ<ALT> V7E
>EXIT
```

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
MACRO-15	V7E	82
Subprogram or Additional Information	Sequence #	PAGE
	11	1 OF 1
New ☐ Replacement Article 1	Original Date	
	August, 1973	

CREF V7F (EDIT #Ø16)

Patch to prevent NEXM system crash in an X4K environment

PROBLEM:

When the X-switch option (cross-reference) is used in the MACRO-15 command string input, CREF crashes the system through a NEXM (non-existent memory) reference in systems using 2ØK or 28K of memory.

SOLUTION:

Use PATCH to make the following binary corrections to CREF and to MACRO. The source code notation to the right need not be keyed in; it is shown for the purpose of illustration.

```
PATCH  Vxx
>CREF
>L 15667
>15667/35647Ø>74ØØØØ<ALT>     NOP
>MACRO
>L 7461
>Ø7461/5Ø64ØØ>6Ø64ØØ<ALT>     V7F
>EXIT
```

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
MACRO-15	V7F	16
Subprogram or Additional Information	Sequence #	PAGE
CREF	12	1 OF 1
New ☐ Replacement Article 1	Original Date	
	August, 1973	

MTF. (EDIT #ØØ6)

Functional Changes

MTF has been greatly modified. Details are on subsequent sheets. Standard buffer size is 377_8, as returned on .INIT. The user may dynamically change this size by the following instructions:

```
.GLOBL MTBSIZ
  .
  .
  .
  .
LAC     BUFSIZ    /(any desired buffer size)
DAC*    MTBSIZ
  .
  .
  .
  .
.WRITE
  .
  .
.READ
```

Thus, since FIOPS re-.INIT's a handler every time I/O transfer direction is changed, one can input from magtape drive N with one record size[1], and output to drive M with another record size, bearing in mind that MTBSIZ must be called before transfer direction is changed.

[1] Record size must be less than or equal 256_{10} words for FORTRAN .READ's and .WRITE's. Larger record sizes are possible via MACRO routines.

Functional Changes (Cont'd)

.INIT

A. Return standard buffer size of 255 words. Buffer size can be changed by modifying the .GLOBL MTBSIZ through a call MACRO subroutine.
B. Call .SETUP - API channel register 48(8).
C. Setup transfer direction (input or output).
D. If first .INIT to this device, the assigned default values will be odd parity and 8ØØ BPI. Track count will be specified by bit 6 of SCOM+4 register. (Ø=7 channel, 1=9 channel).
E. Update the reference drive table that this unit is open for I/O transfers.

.OPER

A. .FSTAT is allowed and implemented AC=Ø on return.

.SEEK

A. Undefined. Error return IOPS6.

.ENTER

A. Undefined. Error return IOPS6.

.CLEAR

A. Undefined. Error return IOPS6.

.CLOSE

A. Checks transfer direction. If not output an error return IOPS 6 is issued.
B. Writes an end-of-file mark and returns to caller.

Functional Changes (Cont'd)

.MTAPE

A. Honors the subfunction specification as follows:

ØØ) Rewind: A rewind is issued to the specified drive and control returned to caller.

Ø1) Undefined: Error return IOPS6.

Ø2) Backspace: A backspace is issued to backspace one physical record.

Ø3) Backspace file: Issue backspace until two end-of-file marks have been passed over, then space forward over the last end-of-file mark. If beginning of tape is sensed during the backspacing, the function is terminated and control returned to the caller.

Ø4) Write EOF: Issue a write end-of-file mark.

Ø5) Forward space: A space forward is issued to forward space one physical record. If the end-of-tape is sensed an error return IOPS65 is issued.

Ø6) Forward space file: A space forward is issued to forward space until an end-of-file is sensed. If the end-of-tape is sensed a IOPS65 is issued.

Ø7) Space to logical EOT: A space forward is issued until two consecutive

Functional Changes (Cont'd)

end-of-file marks are passed, then a backspace one physical record is issued. If the end-of-tape is sensed a IOPS65 is issued.

1Ø→17) Describe the tape configuration. The reference drive table will be updated for the unit referenced. Subsequent I/O transfers will be performed in the density, parity and channel count given below:

Subfunction	Channel Count	Parity	Density
1Ø	7	Even	2ØØ BPI
11	7	Even	556
12	7	Even	8ØØ
13	9	Even	8ØØ
14	7	Odd	2ØØ
15	7	Odd	556
16	7	Odd	8ØØ
17	9	Odd	8ØØ

.READ

A. Check if referenced unit is input. Error return IOPS6 issued if not.
B. Check if data mode is legal. Modes Ø or 2; any other mode is illegal. Error return IOPS7 is issued.
C. Initiate data transfer.
D. Read errors:
   1. Parity, checksum and record length incorrect. The appropriate header bits are set and returned to the data buffer.
   2. Bad tape or data late. These are considered unrecoverable tape errors and an error return IOPS65 is issued.
   3. End-of-file. The appropriate header bits are set to a 5 and returned to the data buffer.

Functional Changes (Cont'd)

4. End-of-tape. The appropriate header bits are set to a 6 and returned to the data buffer.
5. Any real read error (parity, data late or bad tape) is given 25 rereads before error action is taken.

.WRITE

A. Check if referenced unit is output. Error return IOPS6 issued if not.
B. Check if data mode is legal. Modes Ø or 2; any other mode is illegal. Error return IOPS7 is issued.
C. Initiate data transfer.
D. Write errors:
   1. Any write error is given 25 rewrites. If unsuccessful, six inches of tape are skipped and record rewritten.
   2. End-of-tape. The appropriate leader bits are set to 6 and returned to the data buffer.

.WAIT, .WAITR

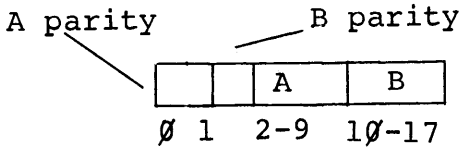
A. Check I/O underway.
   1. Busy: Return to CAL or to address in CAL+2 (.WAITR).
   2. Non-busy: Return to CAL+2 or to CAL+3 (.WAITR).

.TRAN

Allows standard core dump mode (both 7 and 9-track look like 7-track), and true 9-track mode. Bit 9 of LOC+Ø=1 yields true 9-track.

Functional Changes (Cont'd)

9-track data arrives in core with parity bits in bits Ø and 1, and two 8-bit bytes in low-order 16 bits. Thus:



Legal Data Modes

OUTPUT

A. IOPS BINARY (Ø).
IOPS ASCII (2).

The checksum is completed and stored in the second word of the line in the IOPS buffer area. Bits 12-13 of the first header word are set to zero. The count of words to write is taken from the word pair count in the header and transfer from the IOPS area is initiated.

INPUT

A. IOPS BINARY (Ø).
IOPS ASCII (2).

The count of words to transfer is taken from the CAL sequence and input is initiated from the next physical block on tape directly into the IOPS buffer area. When the read is complete, the line validity bits are modified under the following conditions: bits 12-13 of header word Ø are set if buffer overflow occured. A checksum is calculated and compared with the checksum read. If they differ, bits 12-13 are set to 1Ø. Finally, a check is made to assure that the line was transferred without hardware-detected error. If an error occured, bits 12-13 are set to Ø1.

Functional Changes (Cont'd)

Recoverable Errors

A. Transport not ready. IOPS4
Results from:
1. Write request with write lock.
2. 9-channel I/O request to 7-channel transport, and vice-versa.
3. Transport off line or otherwise not ready.

Remedy:
1. Ready the device.
2. Type control R on the teletype.

Unrecoverable Errors

A. Illegal function. IOPS6.
B. Illegal data mode. IOPS7.
C. End-of-tape encountered on a file spacing command. IOPS44.
D. Unrecoverable mag tape error (data late or bad tape). IOPS65.

MTF. (EDIT #ØØ7)

Returning status on a .WAIT

PROBLEM:

Magnetic Tape (MT) handlers which support .WAIT are expected to return MT status in the AC. System programs like MTDUMP utilize this information for proper operation. MTF. ØØ6 did not return the status on a .WAIT.

SOLUTION:

The following SRC modification to MTF. edit #ØØ6 corrects this problem.

Page 18 MTF. ØØ7 .WAIT

```
                        /COME HERE ON IO COMPLETE.  RETURN TO CAL+2 IF WAIT, TO CAL.
Ø1ØØØ R 4ØØ773 R    MTDONE  XCT MTWTSW      /IO FINISHED.  SKIP IF WAIT, NOP IF
Ø1ØØ1 R 441232 R            ISZ MTARGP      /WAITR, BUMP TO NON-BUSY RETURN.
Ø1ØØ2 R 2ØØØ51 R            LAC MTSTAT
Ø1ØØ3 R 6ØØ1Ø7 R    MTJDBR  JMP MTDBR       /THEN RETURN THRU MTARGP.
                        /
Ø1ØØ4 R 741ØØØ A    MTSKIP  SKP
```

Software Product		Version
DOS-15		V1A
Component	Version	Edit #
MAGTAPE HANDLER	N/A	7
Subprogram or Additional Information	Sequence #	PAGE
MTF.	2	1 OF 1
New [X] Replacement Article [ ]		Original Date

SOFTWARE DISPATCH

DOS-15

PRE-1973

RESMON V1A (EDIT #Ø57)

Operational Note: IOPS 77 After Control Q

The SGEN program (DOSGEN) allows users with an RF15 system device to change the CTRL QAREA. Should the user delete the QAREA, any CTRL Q dump would produce disastrous results, by overwriting the disk. The Resident Monitor therefore checks the size of the QAREA before it allows any dumps. If the QAREA is too small for the current core size, the Resident Monitor will give IOPS 77 after any of the following:

A program issues a .GET or .PUT macro

The user tries a manual (CTRL Q) dump

The user tries to restore core, using .SCOM+64

The Resident Monitor will print a number after the IOPS 77. Bits 3-17 of this number will be the desired restart address after the dump or restore which could not be processed.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
MONITOR	V1A	57
Subprogram or Additional Information	Sequence #	PAGE
RESMON	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

TELEPRINTER HANDLER V1A (EDIT #Ø57)

Operational Note on Overprinting

The teleprinter handler will support double spacing at the FORTRAN level. When FORTRAN outputs a line with its first character equal to zero, the FORTRAN OTS replaces it with a 21 code, which requests a double space. If the FORTRAN output line has a first character equal to +, the OTS routines output a 2Ø which requests an overprint (no advance).

The teleprinter handler has been modified such that, if the first character in an IOPS ASCII record is an octal 2Ø, the handler will do an overprint (carriage return, with no line feed). If the first character is octal 21, the handler will do a carriage return and two line feeds.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
MONITOR	V1A	57
Subprogram or Additional Information	Sequence #	PAGE
TELEPRINTER HANDLER	2	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

DOSNRM V2B (EDIT #Ø6Ø)

Problem in restoring UIC's to DAT slots on LOGOUT

Note: The V2A version of DOS-15 was provided as a tape update rather than as an entirely new system. This is described in document DEC-15-QUDA-D, "DOS-15 V1A UPDATE PROCEDURES".

PROBLEM:

The DOS-15 system provides the user with the capability of assigning UIC's to DAT slots at SGEN time. The system will restore the UFDT to the UIC's assigned at the last system generation in the absence of "KEEP ON" commands.

The above capability was not realizable due to a bug in DOSNRM V2A (edit #Ø61).

SOLUTION:

The following patch corrects this problem.

Software Product	Version	
DOS-15	V2B	
Component	Version	Edit #
MONITOR	V2B	6Ø
Subprogram or Additional Information	Sequence #	PAGE
DOSNRM	3	1 OF 3
New [X] Replacement Article [ ]	Original Date	

Problem in restoring UIC's to DAT slots on LOGOUT (Cont'd)

```
DOS-15 V2A
$PATCH

PATCH V1ØA
>DOS15
>NB
>ØØ236/ØØØØ33>34
>L 6774
>Ø6774/Ø76ØØ4>616Ø35          LOC1          JMP   PATCH1
>L 6777
 Ø6777/235567>616Ø44         LOC2          JMP   PATCH2
>L 7ØØ7
 Ø7ØØ7/Ø6ØØ13>616Ø5Ø         LOC3          JMP   PATCH3
>L 7Ø12
 Ø7Ø12/235567>616Ø53         LOC4          JMP   PATCH4
>L 7Ø2Ø
 Ø7Ø2Ø/Ø6ØØ13>616Ø57         LOC5          JMP   PATCH5
>L 7Ø23
 Ø7Ø23/235567>616Ø63         LOC6          JMP   PATCH6
>L 16Ø24
>16Ø24/ØØØØØØ>14             L14           14
 16Ø25/ØØØØØØ>2511Ø3         LUIC          2511Ø3
 16Ø26/ØØØØØØ>141            L141          141
 16Ø27/ØØØØØØ>                UFDUP1        Ø
 16Ø3Ø/ØØØØØØ>22ØØ14                       LAC* 14
 16Ø31/ØØØØØØ>556Ø25                       SAD (2511Ø3
 16Ø32/ØØØØØØ>236Ø26                       LAC*  (141
 16Ø33/ØØØØØØ>Ø6ØØ13                       DAC*  13
 16Ø34/ØØØØØØ>636Ø27                       JMP*  UFDUP1
 16Ø35/ØØØØØØ>Ø76ØØ4          PATCH1       DAC*  (13
 16Ø36/ØØØØØØ>2171Ø2                       LAC NODAT
 16Ø37/ØØØØØØ>744Ø2Ø                       CLL!RAR
 16Ø4Ø/ØØØØØØ>3571Ø1                       TAD  NOPAR
 16Ø41/ØØØØØØ>35ØØØ2                       TAD  .SGNBLK
 16Ø42/ØØØØØØ>Ø76Ø24                       DAC*  (14
 16Ø43/ØØØØØØ>6Ø6775                       JMP  LOC1+1
```

Problem in restoring UIC's to DAT slots on LOGOUT (Cont'd)

```
16Ø44/ØØØØØØ>116Ø27          PATCH2      JMS UFDUP1
16Ø45/ØØØØØØ>44677Ø                      ISZ  CT1
16Ø46/ØØØØØØ>616Ø44                      JMP  PATCH2
16Ø47/ØØØØØØ>6Ø7ØØ3                      JMP  LOC2+4
16Ø5Ø/ØØØØØØ>Ø6ØØ13         PATCH3      DAC*  13
16Ø51/ØØØØØØ>476Ø24                      ISZ*  (14
16Ø52/ØØØØØØ>6Ø7Ø1Ø                      JMP  LOC3+1
16Ø53/ØØØØØØ>116Ø27          PATCH4      JMS  UFDUP1
16Ø54/ØØØØØØ>44677Ø                      ISZ  CT1
16Ø55/ØØØØØØ>616Ø53                      JMP  PATCH4
16Ø56/ØØØØØØ>6Ø7Ø16                      JMP  LOC4+4
16Ø57/ØØØØØØ>Ø6ØØ13         PATCH5      DAC*  13
16Ø6Ø/ØØØØØØ>476Ø24                      ISZ*  (14
16Ø61/ØØØØØØ>476Ø24                      ISZ*  (14
16Ø62/ØØØØØØ>6Ø7Ø21                      JMP  LOC5+1
16Ø63/ØØØØØØ>116Ø27          PATCH6      JMS  UFDUP1
16Ø64/ØØØØØØ>44677Ø                      ISZ  CT1
16Ø65/ØØØØØØ>616Ø63                      JMP  PATCH6
16Ø66/ØØØØØØ>626771                      JMP*  UFDUPD
> LR  1117
> Ø1117/6312Ø2>6312Ø4                    V2B
> EXIT
```

Software Product	Version	
DOS-15	V2B	
Component	Version	Edit #
MONITOR	V2B	6Ø
Subprogram or Additional Information	Sequence #	PAGE
DOSNRM	3	3 OF 3
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

August, 1973

TELEPRINTER HANDLER V2C (EDIT #Ø64)

Patch to support LA3Ø at 3ØØ baud

This patch to support LA3Ø at 3ØØ baud has been revised to suppress echoing of the ALT-MODE (ASCII code 175). The following changes are made by using the "L" command under PATCH.

LOCATION	OLD CONTENTS	NEW CONTENTS		SYMBOLIC	
Ø571	1ØØ722	1Ø2615		JMS	PATCH4
1633	76ØØ15	76Ø212	TTLFO	LAW	212
1635	76Ø212	1Ø26Ø5	TTOCRT	JMS	PATCH
1636	1Ø2452	74ØØØØ		NOP	
2252	6Ø1635	6Ø2577	TTALFO	JMP	PATCH2
2451	741ØØØ	6Ø26Ø2		JMP	PATCH3
25Ø4	76ØØ15	6Ø1635		JMP	TTOCRT
25Ø5	6Ø1636	74ØØØØ		NOP	
2577	-	76Ø212	PATCH2	LAW	212
26ØØ	-	1Ø2452		JMS	TTOSVC
26Ø1	-	6Ø1637		JMP	TTLNFN
26Ø2	-	542547	PATCH3	SAD	TTYØ15
26Ø3	-	6Ø2627		JMP	PATCH5+1
26Ø4	-	6Ø2453		JMP	TTYVT3
26Ø5	-	ØØØØØØ	PATCH	Ø	
26Ø6	-	77777Ø		LAW	-1Ø
26Ø7	-	Ø42625		DAC	TTFLCT
261Ø	-	76ØØ15	TTFLP	LAW	15
2611	-	1Ø2452		JMS	TTOSVC
2612	-	442625		ISZ	TTFLCT
2613	-	6Ø261Ø		JMP	TTFLP
2614	-	6226Ø5		JMP*	PATCH

Software Product	Version	
DOS-15	V2C	
Component	Version	Edit #
MONITOR	V2C	64
Subprogram or Additional Information	Sequence #	PAGE
TELEPRINTER HANDLER	4	1 OF 2
New ☐ Replacement Article 4	Original Date	
	Pre-1973	

Patch to support LA3Ø at 3ØØ baud (Cont'd)

LOCATION	OLD CONTENTS	NEW CONTENTS		SYMBOLIC	
2615	-	ØØØØØØ	PATCH4	Ø	
2616	-	77777Ø		LAW	-1Ø
2617	-	Ø42625		DAC	TTFLCT
262ſ	-	76ØØ15		LAW	15
2621	-	1ØØ722		JMS	TOUT
2622	-	442625		ISZ	TTFLCT
2623	-	6Ø262Ø		JMP	PATCH4+3
2624	-	622615		JMP*	PATCH4
2625	-	ØØØØØØ	TTFLCT	Ø	
2626	-	ØØØØØØ	PATCH5	Ø	
2627	-	2Ø2452		LAC	TTOSVC
263Ø	-	Ø42626		DAC	PATCH5
2631	-	1Ø26Ø5		JMS	PATCH
2632	-	2Ø2626		LAC	PATCH5
2633	-	Ø42452		DAC	TTOSVC
2634	-	622452		JMP*	TTOSVC
24Ø6	6Ø2441	6Ø2435		JMP	TTLNFA-1
1Ø1	-	2635			

The version number is changed from V2B to V2C by making the following changes to the non-resident monitor (DOS-15). Use "LR" command.

1117	6312Ø4	6312Ø6	V2C

Bootstrap to refresh core.

RESMON V2D (EDIT #Ø64)

Patch to restore system configuration on "LOGOUT"

PROBLEM:

In the current version of the DOS monitor 'V2C' on issuing a "LOGOUT" monitor command the system is not restored to the default (as specified in SGEN) configuration, as stated in the DOS systems manual.

SOLUTION:

The above problem is corrected by the following patch:

```
PATCH V1ØA
>DOS15
>LR 5571
>Ø672Ø/2356Ø1<14452>616Ø2Ø (ALT)        LOC    JMP      PATCH
>LR 14671
>16Ø2Ø/ØØØØØØ>2171Ø5                    PATCH  LAC SGNBLK+5
>16Ø21/ØØØØØØ>Ø75563                           DAC* (.SCOM+4
>16Ø22/ØØØØØØ>2356Ø1                           LAC* (152
>16Ø23/ØØØØØØ>6Ø6721(ALT)                      JMP LOC+1
>PS
>ØØ24Ø/Ø15362>15366(ALT)
>EXIT
```

Bootstrap to refresh core. The version number is changed from 'V2C' to 'V2D' by modifying 'location 1117' from '6312Ø6' to '63121Ø', before exiting from PATCH. Use 'LR' command.

Software Product		Version	
DOS-15		V2D	
Component		Version	Edit #
MONITOR		V2D	64
Subprogram or Additional Information		Sequence #	PAGE
RESMON		5	1 OF 1
New	Replacement Article	Original Date	
	1	Pre-1973	

SOFTWARE DISPATCH

DOS-15

October, 1973

RESMON V2D (EDIT #Ø64)

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 bumped, the I/O underway switch (TTIOSW) is tested and, if set, the program loops back to the CAL.

SOLUTION:

.SCOM+35 contains the instruction "DZM TTIOSW". If a program desires to abort teletype I/O, it can merely issue the instruction "XCT* (.SCOM+35)", which will clear out the busy flag. The following is an alternate method for clearing the flag:

```
LAW   -3            /COMPUTE DAT -3
TAD*  (.SCOM+23
DAC   DATM3         /STORE IN TRANSFER VECTOR
LAW   -1            /C(.DAT -3) = TTA
TAD*  DATM3         /TTA. -1 = TTIOSW
DAC   TTIOSW        /STORE IN TV
DZM*  TTIOSW        /CLEAR TTIOSW
```

Software Product	Version	
DOS-15	V2D	
Component	Version	Edit #
MONITOR	V2D	64
Subprogram or Additional Information	Sequence #	PAGE
RESMON	6	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

October, 1973

DOS-15 USERS MANUAL (DEC-15-ODUMA-A-D)

Document Number Change Notice

New #: DEC-15-ODUMA-A-D

Old #: DEC-15-MRDA-D

There has been no change in the contents of the above manual except for a few corrections of minor typographical errors etc., in the manual as listed below:

Page	Change	Comment
1-5	add 'or LA3Ø DEC Writer' in the 5th line of Section 1.2.2	new console terminal supported by DEC.
2-1Ø	delete 'manipulate and' from the 1st line of Section 2.11.	PIP has limited capabilities with MT.
	add 'DEC' in front of 'tape storage' in item 7 in the same section.	-"-
6-25	change all occurrences of 'wrdcnt' to 'wdc' on this page.	for sake of consistency
6-26	change 'nn' to 'ds' under FORMAT.	typographical error
7-5	change '3' to '1' under 'UNIT #?' in the printout depicted in Section 7.3.1.3.	typographical error

Software Product	Version	
DOS-15	V2Ø	
Component	Version	Edit #
MONITOR	N/A	N/A
Subprogram or Additional Information	Sequence #	PAGE
DOCUMENT CORRECTION	7	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

AUGUST 1973

DOS-15 USER'S MANUAL (DEC-15-ODUMA-A-D)

Documentation error

PROBLEM:

In Section 6.7.11 on page 6-23 in the DOS-15 User's Manual it is claimed that "beg" and "wdc" are ignored by the disk pack handler in the .RTRAN system MACRO call. This is not correct. The "beg" and "wdc" are ignored only on output.

SOLUTION:

The manual should read:

ignored for output to disk pack

instead of reading:

ignored for disk pack

Software Product	Version	
DOS-15	V2D	
Component	Version	Edit #
MONITOR	N/A	N/A
Subprogram or Additional Information	Sequence #	PAGE
DOCUMENTATION CORRECTION	8	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

October, 1973

DOS-15 V1A UPDATE PROCEDURE (DEC-15-QUDA-D)

Typographical error in patch to PIP

In the patch published for DOSPIP V6B, the new contents of LOC 14744 should be

```
JMP #+1Ø4Ø
```

and not

```
JMP #1Ø4Ø
```

as stated in the published article.

Software Product	Version	
DOS-15	V2D	
Component	Version	Edit #
MONITOR	N/A	N/A
Subprogram or Additional Information	Sequence #	PAGE
DOCUMENTATION CORRECTION	9	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

PAPER TAPE PUNCH HANDLERS

Difference in End-of-File Code Between ADSS and DOS

The Paper Punch Handlers in the Advanced Software System output the following EOF header on a .CLOSE:

ØØ15Ø5
776273

The Handlers for DOS all output the following EOF header:

ØØ1ØØ5
776773

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
PAPER TAPE PUNCH HANDLERS	N/A	
Subprogram or Additional Information	Sequence #	PAGE
PPA., PPB., PPC.	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

PATCH V1ØA (EDIT #Ø16)

Operation under DOS-15

PATCH V1ØA for DOS release is functionally similar to earlier versions. For patching a disk, PATCH requires the operator to log in under the MIC. If the user fails to do so, PATCH will type:

.DAT-14 NOT PATCHABLE

When patching a disk, PATCH finds the system blocks (SYSBLK and COMBLK) from bits 3-17 of word 2 of the first MFD entry. SYSBLK and COMBLK occupy two contiguous blocks on the DOS-15 system device, and PATCH uses this information when reading or writing these two blocks.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
PATCH	V1ØA	16
Subprogram or Additional Information	Sequence #	PAGE
	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

PIP V6A (EDIT #Ø9Ø)

Note on Implicit Data Modes

If implicit data modes are to be used in PIP command strings, certain rules must be followed.

1. It is not possible to mix data modes in one command.

   a. T DP ,,←DK FILE1,FILE2,FILE3 (B)

      All three files must be binary files with a BIN Extension.

   b. T DP ,←DK FILE1 BIN,FILE2 SRC

      This is not a legal command string.

2. Files with extensions used by PIP to determine data modes should not actually be of another data mode.

   Example:

   FILE1 BIN should be a file written in IOPS Binary mode (Ø in bits 14-17 of the first header word). It should not, for example, have been written in IOPS ASCII (2 in bits 14-17 of the first header word) and then had its extension changed to BIN.

If such a file is transferred, PIP will put the correct data mode code in the header words. If, however, that file is being transferred in a multiple file string and another follows it, PIP cannot handle the next file correctly.

Note on Implicit Data Modes (Cont'd)

PIP uses the following extensions as data mode indicators:

BIN - IOPS Binary

SRC - IOPS ASCII

ABS - Dump

A number in the third location of the extension (e.g., ABL FZ2) makes PIP assume the file is in IOPS ASCII.

SOFTWARE DISPATCH

DOS-15

PRE-1973

PIP V6A (EDIT #Ø9Ø)

Restriction on Diskpack to Diskpack copies

PROBLEM:

One cannot use PIP to COPY the contents of a UFD on one pack to another if the UIC's are the same.

```
>C DP1←DPØ↵
```

for example will not work.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
PIP	V6A	9Ø
Subprogram or Additional Information	Sequence #	PAGE
	2	1 OF 1
New [X] Replacement Article [ ]	Original Date	

PIP V6B (EDIT #ØØ9Ø)

Segmenting files under BOSS-15

PROBLEM:

When using PIP V6A under BOSS-15 to segment files, it was not possible to key in CTRL P as needed. Also, in violation of system convention, PIP would restart itself following an (H) mode COPY command instead of exiting to the monitor.

SOLUTION:

Make the following patches to PIP using the "LR" command to list locations.

LOCATION	OLD CONTENTS	NEW CONTENTS	NEW SYMBOLIC	COMMENTS
1Ø32	6332Ø2	6332Ø4	V6B	
1Ø4Ø	776	775	-3	/OUTPUT TO -3
1Ø37	154473	JMP #+14736	JMP PATCH	/PATCH AREA
14736	Ø	DZM #+14473	DZM YEOFSW	/RESTORE INST.
14737	Ø	LAC* #+14527	LAC* (BOSS	/BOSS REGISTER
1474Ø	Ø	XOR #+14644	XOR (2ØØØØ	/IF BIT 4
14741	Ø	AND #+14644	AND (2ØØØØ	/IS NOT SET.
14742	Ø	XOR* #+14527	XOR* (BOSS	/SET IT FOR ↑P
14743	Ø	DAC* #+14527	DAC* (BOSS	/SAVE IT
14744	Ø	JMP #+1Ø4Ø	JMP RESTRT+2	/RETURN
13537	776	775	-3	/OUTPUT TO -3
13535	7412ØØ	JMP #+14745	JMP PATCH	
14745	Ø	SNA	SNA	/RESTORE INST.
14746	Ø	JMP #+1Ø36	JMP RESTRT	
14747	Ø	LAC* #+14527	LAC* (BOSS	/BOSS REGISTER

Segmenting files under BOSS-15 (Cont'd)

LOCATION	OLD CONTENTS	NEW CONTENTS	NEW SYMBOLIC	COMMENTS
14750	0	XOR #+14644	XOR (20000	/SET BIT 4
14751	0	AND #+14644	AND (20000	/OF .SCOM+52
14752	0	XOR* #+14527	XOR* (BOSS	/FOR ↑P
14753	0	DAC* #+14527	DAC* (BOSS	/
14754	0	JMP #+13537	JMP NUORRE+4	/RETURN
13106	113520	JMP #+14755	JMP PATCH	/PATCH
14755	0	JMS #+13520	JMS ZCLOS	/RESTORE
14756	0	LAC #+14512	LAC XITFLG	/IS EXIT
14757	0	SZA		/FLAG SET?
14760	0	JMP #+13107	JMP PIP-3	/NO, RESTART
14761	0	JMP #+13571	JMP BOSEXT	/YES, EXIT

SOFTWARE DISPATCH

DOS-15

PRE-1973

PIP V6C (EDIT #Ø9Ø)

Patch to Transfer All Files in a 'C' Function

PROBLEM:

In DOSPIP V6B if a COPY command with no switches is typed and there are seventy files to be transferred, only sixty-six files will be transferred.

SOLUTION:

Using the "LR" command to patch PIP, make the following binary corrections.

LOCATION	OLD CONTENTS	NEW CONTENTS	NEW SYMBOLIC
4747	354542	TAD #+7314	TAD LM1
1Ø32	6332Ø4	6332Ø6	V6C

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
PIP	V6C	9Ø
Subprogram or Additional Information	Sequence #	PAGE
	4	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

PIP V6D (EDIT #Ø9Ø)

Patch to correct problem in Transferring Multiple Files from Card Readers

PROBLEM:

PIP V6C will cause the system to hang if an attempt is made to read more than 1 file from the card reader, using the "T" function.

The machine will be in a tight loop.

```
XXX          JMP          XXX
```

The above is due to the fact that the program on reaching an EOF thinks it is reading from a paper tape reader and it waits for the user to type ↑P (indicating presence of new medium) to continue.

SOLUTION:

Use PATCH to make the following binary corrections to PIP. The source code notation to the right is for information only and need not be keyed in.

```
$PATCH
PATCH V1ØA
>PIP
>LR 5574
>1Ø421/61Ø435<Ø561Ø>61Ø551  (ALT)    JMP ZEOF
>LR 1Ø32
>Ø3657/6332Ø6>63321Ø (ALT)                V6D
>EXIT
```

Bootstrap to refresh core.

Software Product	Version	
DOS-15	V2A	
Component: PIP	Version: V6D	Edit #: 9Ø
Subprogram or Additional Information	Sequence #: 5	PAGE: 1 OF 1
New: X / Replacement Article:	Original Date	

PIP V6E (EDIT #Ø9Ø)

Patch to Correct Problem with 'N' Function when 'Snn' Switch Option is Used

PROBLEM:

PIP V6D (Edit #Ø9Ø) does not handle the 'Snn' switch option in an 'N' function properly. This is due to a programming error in the ZS CODE section of the program.

SOLUTION:

The following patch corrects this problem. The source code notation to the right is for information only and need not be keyed in.

```
$PATCH
PATCH V1ØA
>PIP
>LR 7237
>12Ø64/7452ØØ>SNA!CLL!CML
>LR 7243
>12Ø7Ø/612Ø74<Ø7247>JMP #+14762
>LR 14762
>176Ø7/ØØØØØØ>LAC #+14556          LAC (1ØØ
>1761Ø/ØØØØØØ>DAC #+145Ø2          DAC DIRBLK
>17611/ØØØØØØ>JMP #+7247           JMP DTCLER
>LR 1Ø32
>Ø3657/63321Ø>633212               V6E
>EXIT
```

Software Product		Version
DOS-15		V2A
Component	Version	Edit #
PIP	V6E	9Ø
Subprogram or Additional Information	Sequence #	PAGE
	6	1 OF 1
New ☐ / Replacement Article ☑ 1		Original Date: February, 1973

PIP V7A (EDIT #Ø91)

Problem with 'C' switch option in a 'T' function

PROBLEM:

In DOSPIP the user can use the "C" switch option for a "T" function. This provides him with the capability of converting two or more spaces into tabs.

The above capability was not realizable due to improper handling of the "C" switch option, in DOSPIP V6A.

SOLUTION:

The following SRC level changes correct this problem. Changes are indicated by underscored and enclosed instructions. The print-out of an example which explains the operation of the "C" switch option for a "T" function is provided at the end. Note well that these changes do not encompass the corrections to PIP from the preceding articles and that those patches cannot be made as is to PIP V7A because of the shift in code position.

Problem with 'C' switch option in a 'T' function (Cont'd)

```
PAGE 79     DOSPIP Ø91          CECHECK: C,E,T SWITCH PROCESSOR.

3502      Ø6525 R ØØØØØØ A   R1      Ø
3503      Ø6526 R 346620 R           TAD MQ2
3504      Ø6527 R Ø54534 R           DAC TEMPQ          /SAVE TO BUMP (MAYBE)
3505      Ø653Ø R 7412ØØ A           SNA
3506      Ø6531 R 6Ø6542 R           JMP NOTABS                 /Q1=Q2: NO T
3507      Ø6532 R 1Ø66Ø7 R           JMS OUTBUN                 /OUTPUT Q2-Q
3508            ØØ6533 R   LIT11=.
3509      Ø6533 R ØØØØ11 A   HT      11
3510      Ø6534 R 214534 R           LAC TEMPQ          /CONVERT # OF TABS
3511      Ø6535 R 74ØØØ1 A           CMA                /TO A POSITIVE
3512      Ø6536 R 354555 R           TAD (1  /NUMBER
3513      Ø6537 R 746Ø1Ø A           CLL!RTL /MULTIPLY BY
3514      Ø654Ø R 74ØØ1Ø A           RAL                /EIGHT
3515      Ø6541 R Ø54534 R           DAC TEMPQ          /SAVE TO BUMP TYPPOS
3516      Ø6542 R 21112Ø R   NOTABS  LAC CNTSW
3517      Ø6543 R 7412ØØ A           SNA
3518      Ø6544 R 6Ø6547 R           JMP .+3
3519      Ø6545 R 2Ø6657 R           LAC CSET
3520      Ø6546 R 6Ø656Ø R           JMP TAG1
3521      Ø6547 R 146657 R           DZM CSET           /ZERO, IN CASE WE SE
3522      Ø655Ø R 214534 R           LAC TEMPQ          /CHECK IF WE SENT AN
3523      Ø6551 R 74Ø2ØØ A           SZA                /SKIP IF NO
3524      Ø6552 R 6Ø6563 R           JMP NOSPC          /YES, DONT SEND ANY
3525      Ø6553 R 2Ø6525 R           LAC R1
3526      Ø6554 R 74ØØØ1 A            CMA
3527      Ø6555 R 354656 R           TAD (11
3528      Ø6556 R Ø46657 R            DAC CSET
3529      Ø6557 R 354621 R           TAD (-1
3530      Ø656Ø R 74ØØØ1 A   TAG1    CMA
3531      Ø6561 R 1Ø66Ø7 R   ONESP1  JMS OUTBUN                 /OUTPUT 8-R
3532      Ø6562 R ØØØØ4Ø A   LIT4Ø   4Ø
3533      Ø6563 R 2Ø6617 R   NOSPC   LAC TYPPOS
```

Problem with 'C' switch option in a 'T' function (Cont'd)

```
PAGE 8Ø     DOSPIP Ø91          CECHECK: C,E,T SWITCH PROCESSOR.

3534     Ø6564 R 354534 R            TAD TEMPQ
3535     Ø6565 R 346657 R            TAD CSET
3536     Ø6566 R Ø46617 R            DAC TYPPOS          /T=T+S (+1
3537     Ø6567 R 146657 R            DZM CSET            /S=Ø.
3538     Ø657Ø R 6Ø6466 R            JMP TOUTCH          /GO OUTPUT
3539     Ø6571 R 75ØØØ1 A  TONESP    CLC
354Ø     Ø6572 R 6Ø6561 R            JMP ONESP1          /OUTPUT ONLY


DOSPIP V7A

>T TT←DK PIPTST SRC


12345678123456781234567812345678123456781234567812345678123456781234567812345678
X X  XXX X X  X
    XXX X  X
X  X  X  X
X       X       X
X        X          X                    X

DOSPIP V7A

>T TT← DK PIPTST (AC)

12345678123456781234567812345678123456781234567812345678123456781234567812345678
X X     XXX X X X
        XXX X   X
X       X       X       X
X       X       X
X       X       X               X

DOSPIP V7A

>
```

Software Product		Version	
DOS-15		V2A	
Component		Version	Edit #
PIP		V7A	91
Subprogram or Additional Information		Sequence #	PAGE
		7	3 OF 3
New [X]	Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

SYSTEM

Conversion from ADSS-15: Programs may not fit

PROBLEM:

When updating ADSS systems to DOS-15 V1A, sometimes the user's 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-

   - LP ON
   - VT ON
   - Size of monitor patch area
   - Size of buffer pool
   - Size of buffers
   - Size of .DAT table

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

3. The resident monitor itself (RESMON) has grown by approximately ($2ØØ_{1Ø}$) words.

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	
DOS-15	V1A	
Component	Version	Edit #
SYSTEM	N/A	N/A
Subprogram or Additional Information	Sequence #	PAGE
	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SYSTEM HARDWARE

Minimum and Optional Hardware for DOS-15 System

The equipment configuration for the utilization of the DOS-15 software facilities includes a PDP-15/20 computer with 16K of memory and at least one DECdisk or Disk Pack.

A. The minimum hardware required by DOS consists of:

PDP-15 with 16K, 18-bit, 8ØØ-NS core memory

+KSR-35 or KSR-33 Teleprinter

+PC15 High-Speed Paper Tape Reader and Punch

+KE15 Extended Arithmetic Element

+ TC15 DECtape Control with 1 TU56 Dual DECtape Transport or 2 TU55 DECtape transports

or

TC59 Magtape Control with 1 TU1Ø, TU2Ø, or TU3Ø Magtape Transport (7- or 9-track)

+ RF15 DECdisk control with 1 RSØ9 DECdisk Drive (262,144 words)

or

RP15 Disk Pack Control
1 RPØ2 Disk Pack Drive (1Ø.24 million words)
1 RPØ2P Disk Pack

Minimum and Optional Hardware for DOS-15 System (Cont'd)

B. The optional hardware supported by DOS is as follows:

PDP-15 with 32K, 18-bit, core memory

KA15 Automatic Priority Interrupt

KW15 Real Time Clock

FP15 Floating Point Processor

KSR-33 Teleprinter

PC15 High Speed Paper Tape Reader and Punch

TC15 DECtape Control with up to 4 TU56 Dual DECtape Transports or 8 TU55 DECtape Transports

RF15 DECdisk Control with up to 8 RSØ9 DECdisk Drives (262,144 words per drive)

RP15 Disk Pack Control with up to 8 RPØ2 Disk Pack Drives (1Ø,6ØØ,ØØØ words per drive)

TC59 Magtape Control with up to 8 TU1Ø, TU2Ø, or TU3Ø Magnetic Tape Transports (7- or 9-track)

CRØ3B 2ØØ cpm Reader and Control
or
CR15 1ØØØ cpm Reader and Control

VP15 Point Plotting Display
or
VT15 Graphic Display Processor
with
VTØ4 Graphic Display Console

Minimum and Optional Hardware for DOS-15 System (Cont'd)

LK35 Keyboard

Line Printers*

LP15C - 1ØØØ lpm, 132 column line
LP15F - 356 lpm, 8Ø column line

VWA Writing Tablet

*Other line printers supported are the LP15H, LP15J and LP15K, which operate from the LP15F Controller.

SOFTWARE DISPATCH

DOS-15

October, 1973

Radix 50_8 Format

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 A of the Linking Loader Utility Manual. The following Radix 50_8 table should be added to the description in the Linking Loader Manual:

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
SYSTEM	N/A	N/A
Subprogram or Additional Information	Sequence #	PAGE
	3	1 OF 2
New [X] Replacement Article [ ]	Original Date	

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 DISPATCH

DOS-15

PRE-1973

SGEN V1A (EDIT #Ø36)

Relative Positions of the Clock and Line Printer Skip IOT's

PROBLEM:

Disk Pack systems have been supplied with the line printer skip before the clock skip. Under rare circumstances, this will cause the system to hang mysteriously.

SOLUTION:

During System Generation, therefore, the System Manager must ensure that the clock skip, "CLSF", comes before the line printer skip, "LSDF". The rest of the skip chain is in good order.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
SYSTEM GENERATOR	V1A	36
Subprogram or Additional Information	Sequence #	PAGE
	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

SOFTWARE DISPATCH

DOS-15

PRE-1973

UPDATE V1ØA (EDIT #ØØ1)

Note on Program File Extensions

The Linking Loader code (33_8) allows UPDATE to retrieve the extension of the source program for information purposes. UPDATE will print this information when the user specifies the LIST (L) option in the file specification command. If the Linking Loader code 33 is missing from a relocatable binary file, UPDATE will not insert a code 33 or extension.

Software Product	Version	
DOS-15	V1A	
Component	Version	Edit #
UPDATE	V1ØA	1
Subprogram or Additional Information	Sequence #	PAGE
	1	1 OF 1
New [X] Replacement Article [ ]	Original Date	

UPDATE V1ØA (EDIT #ØØ1)

Note on G option and FREE Command

G-Option

The GET (G) option is the file specification command option that allows the free command to be legal. The G-option is legal only with the L and S options, and only the FREE (F), END (E) and CLOSE (C) commands are legal.

FREE (F) Command

Frees the named file from the library file, and makes it a separate file on the output device.

FORMAT:

FREE (F) FILENAME <CR> or ALT MODE

The FREE command can only be used with the G-option. If the user gives the FREE command without specifying the G-option, UPDATE will type:

VALID ONLY IN GET MODE

UPDATE V1ØA (EDIT #ØØ1)

Note on Error Conditions

1. If, after a CLOSE command, the named library file is already on the output device, UPDATE will print the following:

   NAMED FILE ALREADY ON OUTPUT DEVICE

   DO YOU WISH TO CONTINUE (Y/N)

   Any response to the above other than:

   Y<CR> or YES<CR>

   will be interpreted as a "no". If Y or YES, UPDATE will delete the old file and rename the WRK file with the old file's name. If not Y or YES, UPDATE will type:

   COMMAND IGNORED

   and the user may type in a new CLOSE command with a different file name, a normal close, or exit to the nonresident Monitor.

   NOTE: If the user exits back to the monitor, the new or updated file will be present on the output device under a work extension (WRK). If the DOS-15 disk handlers are not being used (e.g., with DECtape), the A-version is required. This is because the new code uses the .FSTAT system macro. If only one file is to be open on DECtape, the "E" or "D" handlers could be used.

2. If, after an INSERT (I) or REPLACE (R) command, the file name (typed) and the program name (in the binary

Note on Error Conditions (Cont'd)

code) are not the same, and the device on .DAT-10 is a directoried (file-oriented) device, UPDATE will print the following:

```
PROGRAM NAME DISCREPANCY
FILE NAME -- (name)
PROGRAM NAME -- (name)
SOURCE EXT -- (extension)
DO YOU WISH TO ACCEPT COMMAND (Y/N)
```

Any response to the above, other than:

Y <CR> or YES <CR>

will be considered a "NO", and UPDATE will type:

```
COMMAND IGNORED
```

If Y or YES, UPDATE will type:

```
DO YOU WISH TO USE FILE NAME (Y/N)
```

In the user's response, UPDATE follows the same conventions in determining a YES or a NO. If NO, UPDATE will use the program's name. In either case, UPDATE will type:

```
SOURCE EXT WANTED
```

The user should then type zero-to-three characters, and terminate with a carriage return. If the user types a carriage return only, then the extension will be omitted from the program unit. The extension must be made up of legal RADIX 5Ø characters.

Note on Error Conditions (Cont'd)

2-b. After an INSERT (I) or REPLACE (R) command, if the file name (typed) and the program name (in the binary code) are not the same, and the device on .DAT-1Ø is not a directoried (i.e., non-file oriented) mass storage device, UPDATE will print:

```
PROGRAM NAME DISCREPANCY
FILE NAME -- (name)
PROGRAM NAME -- (name)
SOURCE EXT -- (extension)
DO YOU WISH TO CHANGE INPUT (Y/N)
```

The convention for yes and no answers is as above.

If yes, UPDATE will type:

```
CHANGE INPUT AND ↑P
```

If no, UPDATE will type:

```
DO YOU WISH TO USE FILE NAME (Y/N)
```

A yes reply will update the library with the file name; a no will cause UPDATE to use the program's name.

In either case, UPDATE will then type:

```
SOURCE EXT WANTED
```

The user should then type zero-to-three characters, and terminate with a carriage return. If the user types a carriage return only, the extension will be omitted from the program unit. The source extension must be made up of legal RADIX 5Ø characters.

Note on Error Conditions (Cont'd)

NOTE: In the case of discrepancy between typed name and binary name, the sense of the questions for directoried devices (DECtape, disk and some Magtape) is opposite to that for non-directoried devices (paper tape reader, and some Magtape). Thus, a NO for a directoried device has the effect of a YES for the non-directoried devices. The user should therefore read the questions carefully.

2-c. More information on REPLACE and INSERT

If the secondary input device (.DAT-1Ø) is a directoried device (that is, file-oriented), UPDATE will now do a .FSTAT for the file name in the command string for REPLACE or INSERT. If the file is not found, UPDATE will type:

```
FILE NOT FOUND
COMMAND IGNORED
```

If the secondary input device is not a directoried device (non-file oriented), and the first buffer of the unit to be inserted contains an End-of-File or End-of-Medium header word pair, UPDATE will type:

```
PGM UNIT NOT FOUND
FILE POSITIONING COMPLETED
COMMAND IGNORED
```

3. If UPDATE discovers that the file named in a FREE command is already on the output device, it types the following message:

```
NAMED FILE ALREADY ON OUTPUT DEVICE
DO YOU WISH TO CONTINUE (Y/N)
```

Note on Error Conditions (Cont'd)

If the answer is Y or YES, UPDATE will delete the file already present on the output device, and write the one derived from the object library. The object library itself is not affected. If the answer is not Y or YES, UPDATE types:

COMMAND IGNORED

4. If, after a FREE command is made, UPDATE cannot find the named file in the library, UPDATE will type:

EOF REACHED BY SEARCH

5. If, after a FREE command is made, UPDATE finds the named file, but cannot find the .END Linking Loader code (27_8) within the named file, UPDATE will print:

END OF FILE FOUND

which indicates the library is no longer valid.

SOFTWARE DISPATCH

DOS-15

October, 1973

UPDATE V1ØB (EDIT #ØØ2)

Why V1ØB and not V11A

PROBLEM:

A new source level version of UPDATE was distributed with the second of the updates to the DOS-15 system as described in DEC-15-ODUDA-B-D, "DOS-15 Update, Number 2". The source file name is UPDATE ØØ2. According to the rules for assigning program version numbers a source file change should have caused the version number to go from V1ØA to V11A; instead, the source was edited to print V1ØB, which indicates erroneously that there has been a patch.

SOLUTION:

This discrepancy, which has no operational effect, will be corrected in the next release of the software.

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
UPDATE	V1ØB	2
Subprogram or Additional Information	Sequence #	PAGE
	4	1 OF 1
New [X] Replacement Article [ ]	Original Date	

UPDATE V1ØC (EDIT #ØØ2)

Patch to correctly read the EOF/EOM record

PROBLEM:

UPDATE V1ØB detects a record with linking loader code 27 as the last record, instead of the record with EOF/EOM. As a result when secondary input is from a papertape, the record with the EOF/EOM is not read.

SOLUTION:

The following patch to UPDATE corrects this problem.

```
DOS-15 V2A
$MICLOG SYS
$PATCH
PATCH V1ØA
>UPDATE
>L 16Ø47
>16Ø47/ØØØ77Ø>114710      JMS READ     /READ. EOF/EOM?
>16Ø5Ø/ØØØØØ6>617573      JMP PATCH    /YES. CLOSE FILE & EXIT
>16Ø51/152546>616ØØØ      JMP WRONGP   /NO. REPORT TO USER
>L 16421
>16421/41Ø32Ø>41432Ø                   /V1ØC
>L 17573
>17573/ØØØØØØ>77Ø  PATCH CAL+77Ø      /CLOSE INPUT
>17574/ØØØØØØ>6           6
>17575/ØØØØØØ>152546      DZM SRCEXT   /RESET FILE EXT
>17576/ØØØØØØ>635744      JMP* INSUNT  /DONE-EXIT
>EXIT
```

Software Product	Version	
DOS-15	V2A	
Component	Version	Edit #
UPDATE	V1ØC	2
Subprogram or Additional Information	Sequence #	PAGE
	5	1 OF 1
New [X] Replacement Article [ ]	Original Date	

ROTATE (SRC)

Functional Description

The Linking Loader allows the VT handler to reside in IOS. There are two new calls to the graphics software: ROTATE and CIRCLE.

ROTATE modifies a user's array so that he may plot a three dimensional figure. A single call to ROTATE will effect a rotation about one or more of the X, Y or Z axes. A rotation about any other axis will require one or two more calls to rotate. Note that ROTATE uses the same left-handed system that the rest of the Graphics Software uses. That is, origin at the center of the screen, X horizontal, positive to the right, Y vertical, positive up, and Z = YxX (positive into the screen). Systems with a different origin will require a translation followed by the rotation(s) followed by a retranslation.

Programmers should use ROTATE carefully. A rotation of a large figure in the X-Y plane about the Z-axis, for example, may cause part or all of the figure to disappear. This will happen whenever one end-point of a line passes off the screen. The cautious programmer will save the original buffer before calling ROTATE, and be prepared to reduce the size of the figure, should a rotation destroy parts of the picture.

Functional Description (Cont'd)

Here are the FORTRAN and MACRO formats for calls to ROTATE:

FORTRAN:

```
CALL ROTATE(ISTR, IA, IB, IC, X, Y, Z, SINA, CSA)
```

MACRO:

```
.GLOBL      ROTATE
JMS*        ROTATE
JMP         .+12
.DSA        ISTR
.DSA        IA
.DSA        IB
.DSA        IC
.DSA        X
.DSA        Y
.DSA        Z
.DSA        SINA
.DSA        CSA
```

The input variables for ROTATE are as follows:

1. ISTR is the array length, in locations.

2. IA specifies whether rotation about the Z axis is desired.
   If IA=1, rotation will occur about the Z -axis.
   If IA=Ø, there will be no rotation about the X-axis.

3. IB specifies whether rotation about the Y-axis is desired.
   IB=1 indicates rotation is desired, as with IA.

4. IC specifies whether rotation about the X-axis is desired.
   IC=1 indicates rotation is desired, as with IA.

Functional Description (Cont'd)

5. X is the name of the X array.

6. Y is the name of the Y array.

7. Z is the name of the Z array.

8. SINA is the sine of the angle of rotation.

9. CSA is the cosine of the angle of rotation.

Restrictions:

The values in the user's array must be in floating point format.

The user must calculate the sine and cosine of the angle of rotation before he calls ROTATE.

It is up to the user to change integers into floating point numbers and make the correct calls for displaying the rotated figure.

CIRCLE (SRC)

Functional Description

The CIRCLE subroutine will construct approximations of arcs and circles as subpictures. (The approximations are really a series of chords. The user may specify how long the chords should be.) The user may subsequently display such subpictures with the proper calls to the handler. The subpicture will be at the array named in the call.

Here is the form for calls to CIRCLE:

FORTRAN:

```
CALL CIRCLE (R, THETA, GAMA, DEG, PNAME)
```

MACRO:

```
.GLOBL      CIRCLE
JMS*        CIRCLE
JMP         .+7
.DSA        R
.DSA        THETA
.DSA        GAMA
.DSA        DEG
.DSA        PNAME
```

The input variables for CIRCLE are as follows:

1. R is the radius of the circle, in Raster units.

2. THETA is the start of the arc in degrees from the X-axis, rotating counter clockwise about the center of the screen.

3. GAMA is the end of the arc in degrees from the X-axis, rotating counter clockwise about the center of the screen.

Functional Description (Cont'd)

4. DEG is the chord length, in degrees.

5. PNAME is the name of the location at which CIRCLE will start the new array.

NOTE:

1. Circles will only be produced if GAMA-THETA=360°.

2. The array at PNAME will be as long as there are chords in the constructed arc. That is, the length at the array PNAME will equal the number of chords required to construct the arc--one chord per element.

VWA. (EDIT #ØØØ)

Functional Description

The device handler for the VWØ1 Sonic Digitizer Writing Tablet provides an interface between the user and the hardware. In general it conforms to the conventions of the Disk Operating System* as described in Disk Operating System, DEC-15-MZDA-D (prelim). Initialize and input functions are initiated by standard user program commands (system macros). The primary goal of the device handler is to relieve the user from writing his own device handling subprograms.

1. .INIT (INITIALIZE) MACRO

The macro .INIT causes the Writing Tablet to be initialized and must be given prior to any other I/O command referencing this device.

The action taken by the .INIT software is the clearing of one software and two hardware flags. These flags are:

1) Handler busy flag /Software
2) Data Ready flag /Hardware
3) Pen Data flag /Hardware

The form is: .INIT␣[-]ds

where: ds = .DAT slot number

The expansion is:

```
LOC+Ø  CAL␣[-]ds&777
LOC+1  1                /Function code for .INIT
LOC+2  ØØØØØØ
LOC+3  ØØØØØØ
```

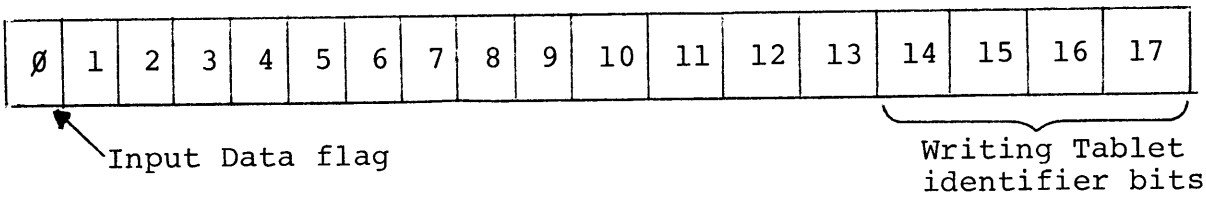
*VWØ1 will work in an Advanced Monitor Software System, as well.

Functional Description (Cont'd)

2. .READ MACRO

The .READ macro is used for input to the user from the Writing Tablet. The input always consists of one status word and two words containing the X and Y coordinates respectively.

The status word has the following format:




```
Bit Ø  = Ø     Input from "DATA READY"
       = 1     Input from "PEN DATA"
Bit 14 = 1     Input from Writing Tablet 1
    15 = 1     Input from Writing Tablet 2
    16 = 1     Input from Writing Tablet 3
    17 = 1     Input from Writing Tablet 4

The form is:   .READ␣[-]ds,m,bufadd,w

    Where:     ds = .DAT slot number
               m = Data Mode
                   Ø = Single Point
                   1 = Single point multiplexed
                   2 = Data input (not scan)
                   3 = Data input multiplexed (scan)
               bufadd = Line buffer address
                        points to a three word data buffer.
               w = Writing tablet to be selected (1 - 4)
```

Functional Description (Cont'd)

The expansion is:

```
LOC+Ø     CAL+1ØØØ*m␣[-]ds&777
LOC+1     1Ø                /Function code for .READ
LOC+2     bufadd
LOC+3     -w
```

3. .WAIT MACRO

The .WAIT macro is only used with respect to the .READ. If a .WAIT is given the user program waits until the .READ has completed--that is, when the line buffer is filled and again available for the user program. If the line buffer is available, control is returned to the user immediately after the .WAIT macro expansion (LOC+2). If the input of data has not yet been completed, control loops on the .WAIT macro.

The form is: .WAIT␣[-]ds

where ds = .DAT slot number

The expansion is:

```
LOC+Ø     CAL␣[-]ds&777
LOC+1     12                /Function code for .WAIT
```

4. .WAITR MACRO

The .WAITR macro is only used with respect to the .READ. If the previous .READ is completed, control is returned to the user immediately after the .WAIT in order to proceed in line. If the input of data has not yet been completed, however, control is given to the location in the user program specified by the .WAITR call.

The form is: .WAITR␣[-]ds,waitad

Functional Description (Cont'd)

where: ds = .DAT slot number
waitad = location in the user program to which control must be transferred if input is not completed.

The expansion is:

```
LOC+Ø     CAL+1ØØØ␣[-]ds&777
LOC+1     12              /Function code for .WAITR
LOC+2     waitad
```

5. .FSTAT MACRO

The .FSTAT macro has meaning only for directoried (i.e., file oriented) devices. On return, the AC will contain zero and bits Ø - 2 of LOC+2 will also be zero, stating that the device was non-file oriented.

The form is: .FSTAT␣[-]ds,namptr

where: ds = .DAT slot number
namptr = address of the first of three words containing the .SIXBT representation of the name and extension of a file.

The expansion is:

```
LOC+Ø     CAL+3ØØØ␣[-]ds&777
LOC+1     2               /Function code for .FSTAT
LOC+2     namptr
```

6. .CLOSE MACRO

When action has been initiated (.INIT and .READ), it must be terminated via the .CLOSE macro. The hardware flags (Data, Ready and Pen Data) will be cleared and the writing tablet(s) will be disabled in order to prevent illegal interrupts.

Functional Description (Cont'd)

The form is: .CLOSE␣[-]ds

where: ds = .DAT slot number

The expansion is:

```
LOC+Ø     CAL␣[-]ds&777
LOC+1     6              /Function code for .CLOSE
```

7. IGNORED FUNCTIONS

The following macros are ignored by the device handler for the VWØ1 Writing Tablet:

1. .SEEK
2. .ENTER
3. .CLEAR
4. .MTAPE
5. .WRITE
6. .TRAN
7. .DLETE
8. .RENAM

8. INTERACTIVE ROUTINES FOR VWØ1 AND VT15

In the Writing Tablet handler no tests are made on the incoming X and Y coordinates. All coordinates are directly transferred to the user. This means that if the pen stays on the same spot (Data Input mode) or is pushed on the same spot for more than once (Single point mode) the same X and Y coordinates are transferred to the user. He must not transfer these X and Y coordinates directly to the VT-handler because a hole could be burned in the display screen. For this reason, it is the user's responsibility to test for consecutive X and Y coordinates on one and the same spot. The number of times the same coordinates could be accepted depends on the intensity.

SOFTWARE DISPATCH

BOSS-15

PRE-1973

NRBOSS V1B (EDIT #Ø52)

Patch to correct problems in closing Run Time Files & MICLOG being active on exit from BOSS-15

PROBLEM:

On exit from BOSS via an EOF card the system was restored under the MICLOG regardless of the state of the system when BOSS was entered. As a result of this the system was unprotected (any user could modify the system, delete files, etc.).

A $ card which was supposed to close all Run Time Files (RTF) did not do so. This was due to the mixup between the card image expected by the routine that decodes the card and the routine that reads the card in NRBOSS. A card with just the $ sign in column 1 gets mapped into $ sign followed by a "carriage return". The routine that decodes the card was erroneously expecting to see a card image of $ sign followed by a "space".

SOLUTION:

Use PATCH as shown below to make binary corrections to BOSS-15. This ensures that the system is restored to the user mode under the UIC that was current when BOSS was entered. Hence, it is not necessary for users to logout on exit from BOSS in case they entered it under the MICLOG. Users have to make note that they have to do a fresh login on exit from BOSS if they want to do privileged operations.

Software Product	Version	
BOSS-15	V1B	
Component	Version	Edit #
MONITOR	V1B	52
Subprogram or Additional Information	Sequence #	PAGE
NRBOSS	1	1 OF 2
New [X] Replacement Article [ ]	Original Date	

Patch to correct problems in closing Run Time Files & MICLOG being active on exit from BOSS-15 (Cont'd)

```
$PATCH
PATCH V10A
>BOSS15
>L 3273                                  PATCH2
>03273/541043 > 541003    (ALT)                SAD   CR
>L 5732
>05732/777762 > 206271    (ALT)  PATCH1   LAC   PLIT1
>L 6271
>06271000000000 > 377762  (ALT)  PLIT1   377762
>L 1703
>01703/106400   206400 (ALT)              V1B
 EXIT
```

Bootstrap to refresh core.

Software Product	Version
BOSS-15	V1B
Component: MONITOR	Version: V1B / Edit #: 52
Subprogram or Additional Information: NRBOSS	Sequence #: 1 / PAGE: 2 OF 2
New: X / Replacement Article:	Original Date:

NRBOSS V1C (EDIT #Ø52)

Patch to correct problem in assigning non-default file names in $CRT and $ADD procedure files

PROBLEM:

NRBOSS V1B does not permit usage of non-default file names in $CRT and $ADD procedure files.

SOLUTION:

The following patch corrects this problem:

```
$PATCH

>BOSS15
>L 26Ø2
>Ø26Ø2/2ØØ664>
 Ø26Ø3/7412ØØ>
 Ø26Ø4/6Ø2622>
 Ø26Ø5/777766>
 Ø26Ø6/Ø4Ø442>
 Ø26Ø7/2ØØ557>1Ø6Ø35                  JMS   GT.CHR
 Ø261Ø/1Ø574Ø>5Ø1Ø35    CR.LOP        AND   L77
 Ø2611/541Ø51>541ØØ3                  SAD   CR
 Ø2612/6Ø2624>
 Ø2613/7412ØØ>541Ø43                  SAD   SPACE
 Ø2614/1Ø277Ø>741ØØØ                  SKP
 Ø2615/1Ø5776>
 Ø2616/1Ø5747>1Ø6Ø35                  JMS   GT.CHR
 Ø2617/44Ø442>
 Ø262Ø/6Ø2611>6Ø2746<ALT>             JMP   CR.LOP
>L 2743
 Ø2743/777766>
 Ø2744/Ø4Ø442>
 Ø2745/2ØØ557>1Ø6Ø35                  JMS   GT.CHR
```

Software Product	Version	
BOSS-15	V1C	
Component	Version	Edit #
MONITOR	V1C	52
Subprogram or Additional Information	Sequence #	PAGE
NRBOSS	2	1 OF 2
New [X] Replacement Article [ ]	Original Date	

Patch to correct problem in assigning non-default file names in $CRT and $ADD procedure files (Cont'd)

```
Ø2746/1Ø574Ø>5Ø1Ø35     EN.LOP     AND   L77
Ø2747/541Ø51>541ØØ3                SAD   CR
Ø275Ø/6Ø2762>
Ø2751/7412ØØ>541Ø43                SAD   SPACE
Ø2752/1Ø277Ø>741ØØØ                SKP
Ø2753/1Ø5776>
Ø2754/1Ø5747>1Ø6Ø35                JMS   GT.CHR
Ø2755/44Ø442>
Ø2756/6Ø2747>6Ø2746<ALT>           JMP   EN.LOP
>L 17Ø3
>Ø17Ø3/2Ø64ØØ>3Ø64ØØ<ALT>          V1C
>EXIT
```

BOSS-15 V1A PROCEDURE FILES

Correction to FOR and ASM to permit usage of the default file name

PROBLEM:

BOSS-15 permits users to use the default file name 'FILTMP' in the $FOR and $ASM cards. This is not possible due to a bug in both these procedure files.

SOLUTION:

The following edit change corrects this problem.

```
$MICLOG SYS
$PIP
DOSPIP V6C
>T {DP,DK},←{DP,DK}     <CTP>     FOR PRC,ASM PRC(A)
DOS-15 V2A
$EDIT
EDITOR V18A
>OPEN FOR PRC
EDIT
>L @A
@AØØ()@
>C/(/(FILTMP
@AØØ(FILTMP)@
>C
EDITOR V18A
>OPEN ASM PRC
EDIT
>L @A
@AØØ()@
>C/(/FILTMP
@AØØFILTMP)@
>E
```

Software Product		Version
BOSS-15		V1A
Component	Version	Edit #
PROCEDURE FILES	N/A	
Subprogram or Additional Information	Sequence #	PAGE
FOR PRC, ASM PRC	1	1 OF 2
New [X] Replacement Article [ ]		Original Date

Correction to FOR and ASM to permit usage of the default file name (Cont'd)

```
DOS-15 V2A
$PIP
DOS PIP V6C
>T {DP} <CTP>,← {DP}   FOR PRC,ASM PRC(A)
   {DK}         {DK}
```

Software Product	Version	
BOSS-15	V1A	
Component: PROCEDURE FILES	Version: N/A	Edit #:
Subprogram or Additional Information: FOR PRC, ASM PRC	Sequence #: 1	PAGE 2 OF 2
New: X / Replacement Article:	Original Date:	

BOSS-15 V1A PROCEDURE FILES

Correction to JOB procedure file to permit usage of default UIC

PROBLEM:

BOSS-15 permits users to use the default 'SRC' UIC in the $JOB command card. This is not possible due to an error in the $JOB procedure file.

SOLUTION:

The following edit change corrects this problem.

```
$A {DK}  <CTP> -14
   {DP}
$EDIT
EDITOR V18A
>OPEN    JOB  PRC
EDIT
>F  N
N @D11()@  <SCR> @D14()@
>C /SCR/@AØ2(SRC)@
N @D11()@  <@AØ2(SRC)@>D14()@
>E
```

Software Product	Version	
BOSS-15	V1A	
Component: PROCEDURE FILES	Version: N/A	Edit #
Subprogram or Additional Information: JOB PRC	Sequence #: 2	PAGE: 1 OF 1
New: X / Replacement Article:	Original Date	

Printed in U.S.A. DEC-15-XSPSA-A-D