<u>C.S.I.R.O.</u>

COMPUTING RESEARCH SECTION

CANBERRA, A.C.T.



<u>i</u>

MEMORANDUM NO. 5

THE DAD SYSTEM FOR THE FORTRAN USER

by

T. Pearcey

C.R.S. CANBERRA

S

1.40

21.7.66 Edition 1.

. X.

λ.

• л. А:

.

;

1.

THE DAD SYSTEM FOR THE FORTRAN USER

Preface

1

This document is a brief description of the DAD (Drums and Displays) system for the C.S.I.R.O. 3600 which replaces the earlier SCOPE system and should be sufficiently explicit, and cover the areas of interest, to FORTRAN oriented users.

It will be noted that the differences between DAD and SCOPE are generally small and that the user will have to make only very small changes to familiar conventions and existing programs will continue to run. Items requiring particular attention have been underlined. However some change and fluctuation in cost will be noticed due to accounting being based mainly upon the actual time which the central processor takes to process the job and partly upon the input and output operations upon the program and data.

This Memorandum is supplied in a loose-leaf form to facilitate the insertion of revision pages.

<u>T. Pearcey</u> 21/7/66 2.

NOTICE OF REVISIONS

-

| Date | Page No. | Comment |
|------|----------|---------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| I | | |
| | | |

.

CONTENTS

- 1. INTRODUCTION
- 2. THE DAD PHILOSOPHY
 - Main Document List (MDL) 2.1
 - Execution List (EL) 2.2
 - 2.3 Output List (OL)
- 3. LOGICAL UNITS AND MNEMONICS
- SERIAL AND RANDOM ACCESS 4.
- 5. DOCUMENT NAMES
- 6. TAPE NAMES
- SAVING DOCUMENTS AND TAPES 7.
- 8. CHARGE-CODE CHECKS
- 9. CONTROL STATEMENTS
 - 9.1 JOB Statement
 - 9.2 DOC Statement
 - 9.3 EQUIP Statement
 - 9.4 Entry Point Name Statement 9.5 LOAD Statement

 - 9.6 RUN Statement
 - 9.7 EOD Statement
 - SEQUENCE Statement 9.8
 - 9.9 FILE Statement
 - 9.10 FILE END Statement
 - 9.11 EXECUTE Statement
 - 9.12 ENDREEL Statement
 - 9.13 DELETE Statement
 - 9.14 PRINT Statement
 - 9.15 PUNCH Statement
 - 9.16 PLOT Statement
- 10. OVERLAYS
 - 10.1 MAIN Statement
 - 10.2 OVERLAY Statement
 - 10.3 SEGMENT Statement
 - 10.4 LOADMAIN Statement
 - Special Overlay Statements 10.5
 - 10.6 Calling Overlays
 - . 10.7 Overlay Rules
- 11. EOF MARKERS
- 12. PAPER TAPE
- DEBUGGING AIDS 13.
 - 13.1 SNAP Dump
 - Memory Map 13.2
 - 13.3 Recovery
- 14. ACCOUNTING

THE DAD SYSTEM FOR THE FORTRAN USER

4.

1. INTRODUCTION

The DAD (Drums and Displays) system designed and implemented by the cooperative efforts of the Computing Research Section of C.S.I.R.O. and Control Data Australia allows more efficient utilisation of the Section's 3600 and its peripheral units. It also is generally compatible with SCOPE, the earlier operating system, so that little effort will be required of users in order that they make effective use of DAD. It also provides additional facilities not available under SCOPE. DAD provides at least the same storage volume for program execution so that established programs may continue to run unaffected.

2. THE DAD PHILOSOPHY

DAD has been made possible by the addition to the 3600 system of a set of magnetic drums with transfer rates of about 250,000 48-bit words per second. Among other functions they are used to perform system operations which were, under SCOPE, provided by the 8 magnetic tape units. Although the tapes may be used at execution time in the usual way DAD provides facilities allowing the user access to many more logical units than was earlier possible since the drums are used as a reservoir of programs and data which are called into core by the DAD monitor at appropriate times. As a consequence users may supply programs and data to the system and have these held for a more or less arbitrary length of time during which they may call upon their programs for execution at any time by supplying a single appropriate statement.

As an extension of the system allowance is made for the on-line use of a number of keyboard display consoles, the facilities for which will be detailed elsewhere.

DAD operates input/output peripherals in a timeshared manner together with execution of user programs. All input and output is controlled by DAD via a routine called BACKGROUND, part of DAD resident.

The main unit of data recognised by DAD is called a DOCUMENT. A document is always headed by, and referred to, by a name, and may consist of more than one file, each of any number of logical records. BACKGROUND reads documents from all available input units simultaneously and loads them in blocks of 256 words, onto the drums. At the same time it unloads documents requested for output, including those created by user program executions onto output units handling the appropriate media. Loading and unloading documents occurs at a high priority and will interrupt the user's program when any required peripheral unit becomes free to provide or accept a record. (Operations called for by the consoles also interrupt the user program but do not halt BACKGROUND).

Documents may be retained within the system, by appropriate declaration, for later use by special single statement calls (see Sections 7 and 9). However users should avoid internal retention of documents for long periods to avoid unnecessary congestion of the mass store.

During operation three lists are maintained as follows:-

2.1 Main Document List (MDL)

This consists of a list of all documents and indicates their names, activity conditions, date of last activity, length, location of access etc. MDL has new entries supplied as document headers are seen by BACKGROUND or documents are created by execution of user programs. Entries are deleted from MDL on special request or, if not declared to be saved, as output is completed and at the termination of the user's execution.

2.2 Execution List (EL)

As programs of suitable kind are entered into the system or requests are received for execution of jobs the names of the documents to be executed are entered into an execution list. This list consists of two parts, EL1 and EL2, and an entry is made into one or other list depending upon the declared expected duration of the execution.

Those expected to run only a few minutes, the upper limit being a system parameter, are entered into EL1, the rest are entered into EL2. Executions in EL1 take priority over those in EL2 whenever they are acceptable. At the termination of an execution EL1 is scanned and the named document accessed via MDL to find the names of their subject documents. The first document in EL1 which has all its subject input documents available, already complete in the system and are not active in any way (accessed by consoles or being input, output or under execution etc.), is accepted for execution. Only when there is no executable entry in EL1 is EL2 treated in a like manner.

When a document is accepted by DAD for execution its entry in EL is erased and execution proceeds.

2.3 Output List (OL)

The names of documents to be output are placed on an output list (OL) as they are released at termination of an execution or as they are specifically called for output.

This list consists of two parts OL1 and OL2. OL1 contains all those documents which are only a few segments long, the upper limit being a system parameter. The remainder are listed in OL2.

As a peripheral completes the output of a document it is allocated to any document in OL which is to be output on the medium handled by that peripheral, preference being given to the first such document in OL1 before proceeding to deal with entries in OL2.

On completion of output the document is deleted from the system unless it has been declared to be saved (standard output, OUT, lun 61 cannot be saved).

Note that input, execution and output are now separate functions and priority is given to short durations and short outputs. Delays may occur for large durations and large volumes of output.

3. LOGICAL UNITS AND MNEMONICS

The following logical units are either available to the user or are system units:

| <u>Lun/M</u> | nemonic | Comment |
|---------------|---------|--|
| 1 - 49 | | Programmer units; may be saved. |
| 50-59 | SO-S9 | User scratch units; <u>cannot</u> be saved. |
| 60 | INP | Standard input; document accepted and allocated to 60. May be saved. |
| 61 | OUT | Released to OL for printing on termination. May <u>not</u> be saved. |
| 62 | PUN | Released to OL for punching on termination. May be saved. |
| 64 | OCM | Output comment medium, the typewriter, avail- able but <u>use very specialised and undesirable.</u> |
| 66 | RAN | Random access of 32K words. Available to programmer but receives dumps at abnormal termination. |
| 67 | TVU | Currently allocated to DD250 display. |
| 68 | REC | Scratch unit used by LOADER and recovery dump program: may be saved. |

| 69 | LGO | Compiled | program | is | allocated | to | 69. | May |
|----|-----|-----------|---------|----|-----------|----|-----|-----|
| | | be saved. | 0 | | | | | |

7.

- 70 LIB Library (71-79, are auxilliary units).
- 80 SCR System scratch.
 - OIL Operator's information log, permanently assigned to the 150 lpm printer.

All units are held on the drum unless otherwise stated.

A user's program may call upon any or all of the programmer units 1-49 and 50-59 thus providing for simultaneous use of more logical units than can be provided by the 8 tape units via SCOPE.

Logical units used by the programmer, 1-49 and 50-59, are allotted to drums unless declared otherwise.

4. SERIAL AND RANDOM ACCESS

Documents are allotted to drum areas unless specifically declared otherwise, e.g. as magnetic tape, MT. They may be of two kinds, serial or random access.

Serial documents do <u>not</u> need special declaration and behave in a manner analogous to magnetic tapes. To position his document at a particular record the user must specifically pass across all intervening records. Serial documents may be created in the usual way by writing each record in a specific sequence. Space for the successive records in the drums is allocated by DAD as needed.

Random access documents <u>must</u> be specifically declared (see Section 9.3) and may be of any number of blocks of 256 words up to a total of 128 blocks i.e. a total of 32768 words. Any number of consecutive words of such a document may be transferred to or from core. Random access documents are scratch-type documents and cannot be output or saved.

FORTRAN users have access to the random access mode via a special subroutine, the calling sequences of which are:

For reading from drum to core -

CALL RDUM(u,fwa,wct,fda)

and for writing to drum from core .

CALL WDRUM(u,fwa,wct,fda)

where u is the logical unit number of the document handled, fwa is the first word in core of the random access block of wct words transferred from the first drum address fda.

Completion of transfer should be checked by the user's program via a statement:

where n is a label corresponding to a 'not ready' status and n_2^{-1} to a 'ready' status.

5. DOCUMENT NAMES

All documents are named. A name consists of an alpha-numeric group of up to 16 characters consisting of the 8 characters of the charge code and a title of up to 8 characters. If the user provides more than 8 characters the last 8 will be accepted. An additional two characters gives the edition number from 1 to 99 and may be used in special circumstances. If an edition number is not specifically stated the document is given one a unit larger than the largest number existing for the same name. If this is 99 the existing document is deleted and the new document is given the number 99. A document of a new name is given unity edition number.

The user must always supply a name.

The system will always supply the following information immediately after the document name

,,HHnn,dd/mm/yy

where HHnn gives the termination time in hours and minutes of the program which created the document, and where dd, mm and yy are the day, month and year respectively.

This enables the user to differentiate between several outputs from the same document. Any output document in punched tape or punched card form will necessarily contain this timing information and this document is in a form suitable for re-input.

The user, if he wishes, may also add this timing information to an input header.

6. TAPE NAMES

Tape names will consist of up to 14 or the first 14 characters of a name. It is intended that the back-up system to DAD will be SCOPE 6 and SCOPE 6 tape labels will become standard. However in the initial stages SCOPE 5 will be used and SCOPE 5 tape labels will be accepted and produced (see SCOPE Manual CDC publication 60053300, Sept. 1964 and June 1965).

7. SAVING DOCUMENTS AND TAPES

On drums each document is allocated a unique logical unit number at execution time and documents may be saved if so declared in their relevant EQUIP statements. <u>It is the user's responsibility to ensure that documents</u> to be retained are suitably declared. Note that a document may save itself, e.g. lun 60 may be saved. Unit number 61, OUT, <u>cannot</u> be saved.

User tapes may be declared as saved. Both drum units and tapes may be preserved from damage by declaring them to be Read Only $(R\emptyset)$. In the case of tapes, <u>input</u> <u>tapes must be declared $R\emptyset$ </u> to avoid the allocation of an undesired tape as the input tape is called.

<u>Users should avoid retaining documents on the drums</u> for extended periods to avoid congestion of drum space. Periodically unused documents will be unloaded from drums onto tape by consolidation procedures and finally will be dropped out of the system.

8. CHARGE CODE CHECKS

Charge codes are checked for format, i.e. eight non-blank characters, by BACKGROUND as a document enters the system. The charge code of a document accepted from EL for execution is checked against a list of valid codes and the execution is terminated and the document deleted if the document charge code is not listed. <u>Fresh users</u> <u>must make sure that their charge codes are on the legal</u> list of charge codes before using the system.

9. CONTROL STATEMENTS

DAD Control Statements possess an asterisk, *, in column 1: For SCOPE compatibility the $\frac{7}{9}$ in column 1 is also accepted. On input a $\frac{7}{9}$ is converted to * for storage. A* is used on displays or on paper tape.

9.1 JOB Statement

This statement heads a program to be executed. Multiple executions are possible. The format is:

10.

*JOB,c,i,t

where c is the user's 8-character charge code after embedded blanks have been removed.

- i is the program title of up to 8 characters (or the last 8 given).
- t is the total expected duration, in minutes, of all execution functions.

c and i comprise the document's name and <u>must</u> be given. c is checked against a list of valid charge codes at start of execution and is rejected if invalid.

The document will not be executed until all referenced input documents are available.

The statement may be followed by the time and date (see Section 5).

9.2 DOC Statement

A DOC statement heads a document which is to be input but not executed. It may be a program and/or data and is held until called for operation or deleted. The format is:

*DOC,c,i,CODE

where c, and i are as in the JOB statement and must be supplied. CODE is a mnemonic which sets the input device to an appropriate conversion mode for acceptance of following records. If CODE is missing it will be taken as standard for the medium (for equipment mnemonic see Section 9.3.1). The mnemonics are:

- CR; BCD (except for LOADER and COSY cards).
- MT; For job stack tapes (in odd parity). Cards or card images from tape with only $\frac{7}{9}$ in column 1 are translated to BCD with an * in column 1.

TR; Typetronics code and assembly mode (8 ch. per word) and parity checked.

Non standard codes may be selected as follows:

- for CR; BIN will select binary reading of all cards except DAD control cards.
 - MT; as for CR.

| TR; | PA | will | select | Par | ity and | 1 Assembly mode. | |
|-----|----|------|--------|----------|---------|------------------------------------|---|
| · | PC | 11 | 11 | | - 11 | Character " (1 ch/word). | • |
| | NA | 11 | 11 | Non | parity | y and Assembly mode. | |
| | NC | 11 | 11 | <u> </u> | - 11 ° | " \overline{C} haracter " (1 ch/ | |
| | | | | | | word). | • |

The statement may be followed by the time and date (see Section 5). The mode of input <u>cannot</u> be changed during input. BCD cards have trailing blanks removed and are each 10 words or less except for * or $\frac{7}{9}$ cards which are accepted as binary in 20 word records. Data in BCD or BIN require separate documents for each type of deck.

9.3 EQUIP Statement

All EQUIP statements should immediately follow their JOB or DOC statements. If not, an execution may waste time and be 'hung' up awaiting an absent document. Such time will be accounted against the user. EQUIP statements are required to declare programmer and scratch units. The format is:

*EQUIP, $u=d_1, d_2, \ldots d_n, u= \ldots$

where u is a logical unit number or mnemonic, d is a declaration.

Up to five statements may be made under the one EQUIP code except that <u>equivalence statements must be one</u> to an EQUIP code. The system checks the consistency of EQUIP statements relating to each logical unit and their equivalence relations and in case of error a diagnostic is provided on OIL.

9.3.1 Hardware Declarations

The format is:

*EQUIP,u=hhn

where hh specifies the hardware type by a mnemonic such as:

CR card reader CP card punch \mathbf{TP} paper tape punch TR paper tape reader \mathbf{LP} line printer MT magnetic tape DR serial access drum ΤV DD250 display PL,PB little and big plotters respectively RA random access drum

n is a decimal integer of up to 3 digits giving the number of drum segments allowed for the logical unit u to an upper limit of 999 segments. If n is absent or is less than 256, 256 is assumed (a segment is 256 48-bit words). In case of RA the upper limit is 128 segments (32K words) and will be assumed to be 128 if n is absent. Unit number 66 is always available as a 32K work RA document. RA is made available to FORTRAN users via a special library subroutine (see Section 4). It is important to remember that magnetic tapes must be declared at MT. Undeclared units will be assumed to be DR.

9.3.2 Usage Declarations

The format is:

*EQUIP,u=hh

where hh specifies the usage type for unit:-

RW read/write, all operations allowed,

BY bypass,

RØ read only, operations affecting the control of the unit are rejected. Units essentially of the RØ type e.g. CR,TR, need not be declared.

All input tapes must be declared as RØ

9.3.3 Mode Declarations

The format is:

*EQUIP,u=hh

where hh is the mode designator and depends upon the hardware type.

| Hardware* <u>Type</u> | <u>hh</u> | Meaning |
|--------------------------|--------------------------|--|
| МТ | LO HI HY | low density (200 bpi) high density (556 bpi) hyper density (800 bpi) |
| CR,CP | BIN BCD | binary read/punch BCD read/punch |
| LP | 1 P 2 P 4 P S P | 1-part paper needed 2-part paper needed 4-part paper needed Special paper needed (notify operator of type) |
| TR,TP | PA PC NA NC | nonstandard code, parity and assembly mode nonstandard code, parity and character mode nonstandard code, nonparity and assembly mode nonstandard code, nonparity and character mode |
| т II I | | |

In the absence of any declaration the standard mode will be set:

| МТ | HI,BIN,RW | | | | |
|-------|------------|------|-----|----------|-------|
| CR,CP | BIN | | | | |
| Ĺ₽ | 1 P | | | | |
| TR,TP | typetronic | code | and | assembly | mode. |

TV, PL and PB operate in one mode only and need not be declared. BIN or BCD mode for magnetic tape is set by a COMPASS mode request or by FORTRAN usage.

9.3.4 Save Declarations

S 4.

The format is:

*EQUIP,u=SV

This is applicable to all logical units except 61 and 50-59, and the user must take care to supply this to documents and tapes he requires to retain for later use. An executed program may be saved by its containing *EQUIP 60=SV, or its equivalent. All drum documents to be SAVED (units 1-49, 60, 62, 68, 69) must be declared so by this type of declaration.

*LO, HI, HY, BCD and BIN are legal on DR and are checked on reading but have no effect.

9.3.5 Equivalence Declarations

The format is:

*EQUIP,u=u'

and is valid if both u and u' are in the range 1-59 or if u is in the range and u' is either of 60,61,62,64,66 or 69. DAD checks to ensure against loops of equivalence or chains of 80 or more.

Separate equip statements must be made for each pair equivalenced. The master unit is the last of a string.

9.3.6 Label Declarations

The identification depends on whether the unit is a magnetic tape or not.

1. Magnetic Tapes

The format_is:

*EQUIP,u=(name,edition,reel)

and is valid for units 1-49 and 69. <u>The name follows that</u> of SCOPE 6 and limits tape names to 14 characters. It is intended that SCOPE 6 will be the back-up system and until then the SCOPE 5 labels will be accepted and written. 'Edition' and 'reel' are integers in the range 1 to 99 or may be omitted. If reel is omitted the edition number may be terminated with), or, if both are omitted,) may follow the name. *n is invalid.

2. Non-tape Units

The format is:

*EQUIP,u=(c,i,edition)

where c is the user's charge code, of 8 non blank characters and i is the title (of up to 8 characters, or the last 8). 'c' need not be identical to that of the execution being processed in which case SV and $R\emptyset$ safeguards are provided.

The edition number may be from 1 to 99 or may be absent in which case the edition number allotted will be unity or one higher than that of any document of the same name already existing or, if this is 99, will be given the number 99 and the original edition 99 deleted.

If the user is referencing one of his own documents u may be 1-49,69, otherwise u must be 50-59 only.

3. Unlabelled

Unlabelled tapes should not be used but are allowed.

The format is:

*EQUIP,u=**,MT.

and applies only to those units declared to be tapes, the MT should be included. Any acceptable tape may be declared to be an unlabelled tape.

- (i) <u>Read:</u> The user must inform the operator, on his JOB request form, which physical tape is to be mounted and the JOB name of the program in which it is to be mounted.
- (ii) <u>Write:</u> A blank label is written on the tape. The user may backspace (BSPR) over this label before writing on the tape if an unlabelled tape is required.

When unlabelled tapes are used the first read request reads the first record on the tape otherwise the label is read and the first user read request reads the second record.

9.4 Entry Point Name Statement

۲.

Library programs are referenced by statements which name the entry point to a library program and direct the named program to be loaded when the statement is processed after execution of the program has been accepted and compilation and assembly begin. Control is then given to the named program which processes those parts of the document which follow until a special terminating statement is reached.

The format is:

*entry point name p₁, p₂ ··· P_m

where p. are parameters interpreted by the library program and may occur in any order. Usually parameters may be followed by =n, indicating a non standard unit is to be used.

9.4.1 COMPASS

The COMPASS statement options correspond to those of COMPASS 5 (see COMPASS Reference Manual, Control Data 60052500, May 1965). The parameters are in free field form, and separated by commas and may be in any order:

*COMPASS, I, Y, P, X, B=n, L, R, M, C

If no options are present, only lines with error flags and the basic assembler headings are printed. Most options may be followed by =n, where n is the logical unit number for that option. If =n is absent, COMPASS will make a standard assignment for the option. Unrecognised options and extraneous characters are ignored. On a card a $\frac{7}{9}$ punch in the first column is acceptable. The options are as follows:

| <u>Option</u> | <u>n absent or zero</u> | <u>n not zero</u> |
|-------------------|---|--|
| <u>INPUT(BCD)</u> | Input on unit 60 | Input on unit n $(1, 40, 60)$ |
| Y INPUT | COSY input on unit 60 | COSY input on n (1-49,60) |
| PUNCH | Punch relocatable binary deck on unit 62. | Punch relocatable binary deck on unit n (1-49,62) |
| <u>x</u> ecute | No load and go produced | Produce binary out- put for load and go on unit n (1-49,69) |
| BCD OUTPUT | n must be specified. Produ n (1-49,62). If C option is option is ignored. | ice BCD output on s specified this. |
| <u>L</u> IST | List assembled programs on unit 61. | List assembled programs on unit n (1-49.61) |
| <u>R</u> EFERENCE | n not relevant. List cross table, if L is specified, of by L. Undefined and doubly will appear whether or not | s reference symbol on unit designated v defined symbols R is specified. |
| MACRO LIST | n not relevant. List the e and macro calls, and list 1 following conditionals. If fied, skipped lines are not macro calls and ECHO protot | expansion of ECHO's ines skipped M is not speci- i listed and only cypes are listed. |

COSY COSY output provided.

9.4.2 FORTRAN

The FORTRAN statement options correspond closely to those of FORTRAN 5 for the 3400-3600 systems (see 3400-3600 FORTRAN Manual Control Data Publication No. 60032900). The full format would be:

FTN,L,P,X,A,I,C,B=n,,R.

but unlike SCOPE the parameter * is redundant in DAD. *FORTRAN,L,P,X etc. is also accepted.

An option may be followed by =n, where n represents the logical unit number for the option. The options are as follows:

| <u>Options</u> | <u>n absent or zero</u> | <u>n not zero</u> |
|-------------------|--|--|
| LIST | List source program on unit 61. | List source program on unit n (1-59 or 61). |
| PUNCH | Punch relocatable binary deck on unit 62. | Punch relocatable binary decks on unit n (1-59 or 62). |
| <u>X</u> ECUTE | Write load and go on unit 61. | Write load and go on unit n (1-59 or 69). |
| <u>A</u> SSEMBLY | List assembled program on unit used, list on unit m. If L is stated unit if n is non zero. | it 61 or, if L=m is is absent list on • |
| <u>I</u> NPUT | Input source deck from unit 60 (also 60 if option not present). | Input source deck from unit n (1-59 or 60). |
| <u>C</u> 0SY | Punch a COSY deck on unit 62. | Punch a COSY deck on unit n (1-59). |
| <u>B</u> CD | n <u>must</u> be designated. | Punch generated cards on unit n (1-59 or 62). |
| × | Compile cards for one bank always. * is redundant and need <u>not</u> be entered. | not relevant |
| <u>R</u> EFERENCE | List COMPASS reference table for ASSEMBLY, or unit assigned ASSEMBLY is absent. If neith entered list on unit 61. | on unit assigned ed to LIST if her option is |

The options may be omitted or placed in any order. If n is entered in any option it must have been suitably assigned by an EQUIP statement placed ahead of the FTN statement. A $\frac{7}{\alpha}$ punch in column 1 is acceptable.

The FTN statement is followed by the source language subprogram. The program may contain assembly language subprograms (COMPASS) and FORTRAN subprograms in any order (see COMPASS/FORTRAN Mixed-deck, Control Data publication No. 60137000). COMPASS subprograms must not contain macros or be COSY decks.

FORTRAN source subprograms must begin with a **PROGRAM SUBROUTINE or FUNCTION statement and terminate** with an END statement. All COMPASS programs must begin with an IDENT statement and end with an END statement (starting in column 10 if on cards, or after 9 spaces following a new line on paper tape).

A SCOPE statement is required to indicate the end of the source subprograms. It is not a DAD control statement but is recognised by the FORTRAN compiler. The word SCOPE begins in column 10 on cards or after 9 spaces on paper tape. No embedded blanks are allowed.

The COMPASS statement and option are as for COMPASS (see COMPASS Reference Manual, Control Data Publication No. 60052500, May 1965).

9.4.3 Terminating Program Statements

To maintain compatibility with SCOPE and to avoid having to adjust compilers and assemblers, the terminating statement treated by a library program will be:

9 spaces SCOPE.

This will return control to DAD which expects a control statement to follow. This format is acceptable on cards and paper tape.

If the next statement is not a DAD control card the execution is terminated and a diagnostic sent to OUT.

9.5 LOAD Statement

Relocatable binary subprograms can be loaded into storage from programmer and scratch units 1-59 or for 69. The format is:

*LOAD,u

where u is the logical unit number. If u is absent 69 is assumed. If u is a drum unit the subprogram will be loaded directly. If u is a tape it will be backspaced one file and the subprogram loaded until EOF or two transfer records or another control statement is encountered.

If the unit cannot be backspaced DAD loads the subprograms directly. A second transfer record is taken as a loading termination but is not required. If binary subprograms are transferred from INP and produced by a system program e.g. a compiler or assembler, they are stored on the same logical unit during execution, only one EOF will be present and it will follow the last subprogram stored on the unit.

9.6 RUN Statement

This RUN statement causes the loaded user program to be executed. The format is:

*RUN,t',p,r,m

- where t' is the execution time in minutes and should not be greater than t on the JOB (or EXECUTE) statement to which it corresponds otherwise the execution will be terminated on reaching t. If t' is zero or blank it is assumed to be one minute.
 - p is the maximum number of print lines on OUT and does <u>not</u> include dumps. The execution is terminated if this is exceeded. <u>If p is blank</u> a nominal value of 100 lines is given.
 - r is the recovery indicator specifying an area to be dumped, in case of abnormal termination, as follows:

| <u>r</u> | area dumped on OUT |
|------------|---|
| 0 or blank | console, |
| 1 | program and console, |
| 2 | labelled common and console, |
| 3 | program and labelled common and console, |
| 4 | numbered common and console, |
| 5 | program and numbered common and console, |
| 6 | labelled and numbered common and console, |
| 7 | console and all store. |

m is ignored by DAD but is allowed for SCOPE compatibility.

9.7 EOD Statement

This has the format:

*EOD

and indicates the end of a document. <u>Users must terminate</u> their documents with this statement. Documents output on media which can be used as input automatically have an EOD statement supplied by the system.

Documents not terminated by an EOD (or immediately followed by a SEQUENCE statement) cause a diagnostic to be listed on OIL and the document deleted.

If the terminated document is headed by a JOB statement its MDL entry is completed and its name is entered into EL1 or EL2 according to the value of its t parameter.

EOD statements immediately following an EOD are ignored.

9.8 SEQUENCE Statement

For SCOPE compatibility the sequence statement may take the place of an EOD on job tapes written by subsidiary systems for running on the 3600 under DAD. The format is:

*SEQUENCE, n

and may terminate a document. n is an integer but has no specific meaning for DAD. Other comments are as for the EOD statement.

9.9 FILE Statement

The format is:

*FILE,u

and causes all records which follow up to FILE END to be transferred to logical unit u, where u may be in the range 1-59 or 69. Only binary records are written and a diagnostic is provided on OUT if there is inconsistent usage and the execution is terminated.

Records between FILE and FILE END must not contain JOB, DOC, EOD, ENDREEL, or EXECUTE, DELETE, PRINT, PUNCH or PLOT.

Ň

nounder 1 14 11 66

C.R.S. Memo 5/Ed.1 (Revised 26.9.66)

Table Runke ('n'.) unt

9.10 FILE END Statement

The format is:

*FILE END

4 NUV .J05 and causes an end of file mark or its drum equivalent to be written and backspaced over it. Any number of FILE-END MSL-LIBRARY-R.P. sequences may be directed to the same or different logical unit.

9.11 EXECUTE Statement (++)

This statement causes a named document to be listed for execution. Its format is:

21.

*EXECUTE, c, i, t

where c and i are as in the JOB or DOC statement and t is the time limit for execution and corresponds to the t in a JOB statement. t may not exceed 2236 minutes. In case of documents with the same name that with the largest edition number is accepted. Execution will not proceed until all referenced documents are available.

The charge code c is checked against the list of legal codes and if not legal the document is deleted from the system.

If no document of the required name is listed in MDL when the EL entry is tested for execution a diagnostic is printed on OIL and the entry is deleted.

9.12 ENDREEL Statement

The format is:

*ENDREEL

and is ignored on card or paper tape readers. It causes the reel of a job stack tape e.g. a 3200 job stack tape, to be unloaded and releases the hardware items from use by the I/O program.

9.13 <u>DELETE Statement</u>(++)

In the first implementation this is available through the DD210 display system (see DAD programmers Reference Manual Pt. II), and has the format:

*DELETE, c, i, ed

where c and i give the name and ed the edition number of the document to be deleted.

If ed is blank all editions of the named document are deleted. If the document is being used, at the time the DELETE statement is received the deletion will be delayed until the activity ceases and the document becomes dormant.

The effect of this statement may be obtained by a 'dummy' program made up as follows:

9.14 PRINT Statement (++)

In the first implementation this is available through the display system under DAVE and has the format:

*PRINT, c, i, ed, SV

where c,i,ed is the document identification and SV implies that the referred document is to be saved after the activity. The statement will cause the identified document to be placed on the OL list according to its size. Short documents are placed on OL1 and have priority over those in OL2. If SV is absent the document is deleted after output. A document to be printed must have a control character in the first character of each record, having been created in a suitable format by the run generating it.

If ed is omitted the highest edition is assumed and if also SV is supplied the commas must be present.

The effect of this statement may be achieved by supplying the following job:

*JOB,c,i',t *EQUIP, n=(c,i,ed),RØ,LP,SV *EOD

in which SV is optional, deletion of the document c,i occurring if SV is absent.

9.15 <u>PUNCH Statement</u>(++)

This is initially available via the displays (provided under DAVE) and has the format:

*PUNCH, c, i, ed, SV

It causes the identified document to be output onto cards in BCD code.

Other comments are as for PRINT except that <u>no</u> initial control character is needed. The effect of this statement may be achieved by supplying the following job:

> *JOB,c,i',t *EQUIP, n=(c,i,ed),RØ,CP,SV *EOD

9.16 PLOT Statement (++)

This is available initially in displays (provided under DAVE) and has the format:

*PLOT, c, i, ed, SV

and causes the identified document to be output on the big plotter and the document must consist of plot type records.

Other comments are as for PUNCH. The effect of this statement may be achieved by supplying the job:

*JOB,c,i',t *EQUIP, n=(c,i,ed),RØ,PB,SV *EOD

in which SV is optional and PB may be replaced by PL if it is known that the document will fit onto PL.

Statements marked by (++) are not implemented in the initial version of DAD.

10. OVERLAYS (++)

The following statements deal with the formats and use of overlay statements. They may have an asterisk or a 7,9 punch in column 7. Note that the SCOPE 11,0,7,9 punch is not acceptable. Records following a control statement may consist of binary subprograms, assembly or compiler language sub-programs. DAD control statements may be included if compilation and assembly is to be performed prior to execution of the binary overlay document. Each MAIN OVERLAY, or SEGMENT must contain one transfer address. MAIN may contain two transfer addresses. (OVERLAY facilities are not provided in the first implementation).

10,1 MAIN Statement

The format is:

*MAIN,u

where u is the logical unit number of the assembled overlay document and may be in the range 1-49 and <u>must</u> be provided.

This statement <u>must</u> be provided and <u>must</u> precede object subprograms comprising the main section. The main subprogram must precede the first overlay and may be declared SV and MT if needed by suitable EQUIP statements heading the whole document.

24.

10.2 OVERLAY Statement

The format is:

*OVERLAY,u,n

where u is the logical number, 1-49, on which the OVERLAY is to be written in absolute binary, and <u>must</u> be provided. n is the particular overlay identifier, a decimal integer. This statement precedes object subprograms comprising the overlay.

10.3 SEGMENT Statement

The format is:

*SEGMENT,u,n

where u is the logical unit number, 1-49, of the overlay file on which the segment is written in absolute binary and <u>must</u> be provided. n is a segment identifier, a decimal integer. The statement precedes object subprograms which comprise the segment.

Relocatable binary subprograms may be included in overlay card decks but are not allowed on paper tape documents which should be supplied in typetronic code.

10.4 LOADMAIN Statement

The format is:

*LOADMAIN,u

which causes the main section to be loaded into store from the overlay unit u and initiated. If this statement is used a RUN statement need not be used.

10.5 Special Overlay Statements

Although overlay statements are compatible with SCOPE and need FILE and FILE END statements, three special statements are provided:

> *MAIN,u,u' *OVERLAY,u,n,u' *SEGMENT,u,n,u'

- where u is the logical unit number on which the overlay (segment) is written in absolute binary (range 1-49),
 - n is the overlay (segment) identifier,
 - u' is the logical unit number of the final unit, usually 69, and that which would be placed on a FILE statement. FILE and FILE END are not required.

10.6 Calling Overlays

Overlays and segments may be called as FORTRAN subprograms by the calling sequence:

where SEGMENT or OVERLAY is entered as relevant and

- n is the overlay number,
- s is the segment number, or blank if an overlay,
- u is the logical unit number,
- d is a dummy parameter which must be present if control parameters appear,
- p_i are the actual parameters passed to the overlay or segment. No more than 59 may appear.

If n,s,u or d is blank the comma must appear, the order is fixed. One subprogram in each overlay/segment must begin with the statement:

PROGRAM (name) $(p_1 \dots p_m)$

where 'name' is the transfer address of the overlay/segment, and $p_1 \cdots p_m$ are formal parameters corresponding to those in the CALL.

10.7 Overlay Rules

- 1. Numbered and labelled common and all entry points declared in the main subprogram may be referred to by any overlay and any segment.
- 2. Numbered and labelled common and all entry points declared in any overlay may be referred to by that overlay and its associated segments but not by the main subprogram, with overlay, or segment contained in another overlay.
- 3. Numbered and labelled common and all entry points declared in a segment may be referred to by that segment only.
- 4. The first overlay record <u>must</u> be preceded by the subprogram.
- 5. Overlay identifiers <u>must</u> be monotonically increasing.
- 6. Segment identifiers must be monotonically increasing.
- 7. Only 4 different overlay logical units may be declared (this may be relaxed later) and if a tape, cannot exceed one reel.
- 8. Each overlay and segment has a single named transfer point.
- 9. All segments for a particular overlay <u>must</u> immediately follow an overlay on its unit.

For use of FILE and FILE END statements see the SCOPE Manual.

11. EOF MARKERS

Documents may consist of more than one file and each must be separated by an EOF marker. An EOF marker need not be placed immediately before an EOD statement.

Card files are terminated by a card with a $\frac{7}{8}$ in column 1. Paper tape files are terminated by a 'master stop', a $\frac{7}{5}$ code.

12. PAPER TAPE

The statement code and character set is described in CRS Memo No. 3. The following points are to be noted:

- (i) All control statements are in standard code and are preceded by *.
- (ii) All paper tape documents must be provided with a header and in standard code in 1 in. or $\frac{7}{8}$ in. wide tape and must consist of:

*JOB,c,i,t or *DOC,c,i,CODE

to which may be added, HHnn, dd/mm/yy the time and date of operation. The statement is terminated by a new-line character.

- (iii) In the absence of a CODE entry it is assumed that the document is in typetronic code and will be converted to internal BCD on reading.
 - (iv) The modes specified by CODE are:
 - (a) NA (Non Parity and Assembly). Level 7 defines the word marker. If it is missing then expected frames are ignored until the next word mark appears.
 - (b) PA (Parity and Assembly). Leading blanks are ignored, and if encountered within records will either be ignored, wrongly read as 00 and/or cause missed data. 6-bit bytes are packed 8 to a word.
 - (c) NC (<u>Nonparity and Character</u>). This allows 8,7 or 5 channels to be read. All codes are valid and data is stored one frame per word, right justified.
 - (d) PC (<u>Parity and Character</u>). Leading blanks are ignored and blanks encountered in records are treated logically like PA mode. The internal format of one frame per word, right justified, is used.
 - (e) Typetronics code maps the 7 channel standard code to its equivalent internal BCD. No distinction is made between upper and lower cases and typewriter functions are ignored. No line imaging is attempted so that 'backspace' is ignored and 'tabulate' converts to 'space, space'.

- (f) A paper tape reader becoming not ready, due to 'physical end of tape' being reached (by intention or by tape breakage), causes a switch from the previously set reading mode, to the typetronics mode, and unless EOD follows, an 'abnormal end of document' condition will be indicated to the operator.
- (g) All paper tape <u>documents</u> must be terminated by *EOD in typetronics code.
- (h) All paper tape <u>files</u> will be terminated by a 'master stop' in typetronics code, i.e. $\frac{7}{5}$ code.
- (i) In typetronics code NEWLINE defines 'end of logical record' but no such record is to exceed 20 words, a packing of 8 characters per computer word being automatic. In other modes for which the 'end of logical record' is not defined, e.g. character mode with or without parity, the system will assume 20 words per record, i.e. 20 paper tape frames (one frame per computer word being the mode of input). In typetronics code attempts by programmers to read in more than 20 words will cause loss of data. In other than standard mode, attempts to read other than 20 words or multiples of 20 words will cause loss of data e.g. if, say 5 words are requested the 15 following words will be lost. Similarly if 20n+5 words are requested. Documents coded in non-typetronics code and for which end of logocal record is defined, e.g. assembly mode, will be limited to lengths of 20 words, as for typetronic coded documents.

It is therefore to be noted that a paper tape document which may be on other than one inch wide tape or is in other than an 8-channel code and is to be read in a binary fashion must be handled in a peculiar manner. A special header in typetronics code must be provided and read. It will terminate on a 'newline' code. If CODE is specified the following 'binary' document will consist of one file on a tape and the physical end of tape will terminate the file and a *EOD may be then supplied in typetronics code since the end of file resets the unit to standard code. No binary paper tape can consist of more than one file. New files must be input as new documents and the programmer must allow for this.

13. DEBUGGING AIDS

At job termination the whole of core is dumped onto RAN(66). Dump formats are identical to those in SCOPE but a wider range of diagnostics is provided.

13.1 <u>SNAP Dump</u>(++)

The programmer specifies the instruction address where the request is executed, the frequency and areas to be dumped. The contents of the areas dumped are placed on OUT,61, at execution time.

More than one SNAP dump may be specified for an address and any number of addresses may produce dumps. After each SNAP is produced normal program operation resumes. The only restrictions on the number of SNAP dumps are the maximum time request limit in the RUN statement (see Section 12.5) and the amount of available storage remaining after the program is loaded.

The format is:

(as for standard SCOPE, see SCOPE Reference Manual 60053300).

a is the program address where the dump is to be initiated, a program name, or entry point name, plus or minus an octal displacement, 'name' +n where n is the octal displacement. If 'a' is blank a snap is taken only if abnormal termination occurs.

fwa, lwa are the first word and last word addresses respectively of the area to be dumped and may be:

- (1) an octal location of up to 6 digits (normally the leftmost of the 6 digits will be a bank indicator and for DAD 5 digits will be the maximum size of the number);
- (2) a common block name, an entry point name or a program name (p), plus or minus an octal displacement (n), i.e. p + n. If a common-block name is used it is enclosed in slashes, viz., /p/. If the block name is blank, two slashes are given, viz., //;
- (3) an octal displacement relative to the previously declared entry point, common block, a program name on the SNAP record;
- (4) a 'blank' implies no area snapped; the console will be snapped if C is suffixed to the mode

| | designator (see below). Where fields are omitted commas must be retained, unless no non-blank fields follow. |
|--|--|
| : | If fwa equals lwa, a console scoop will be given. |
| f speci | fies the format of the dump on unit 61:- |
| O or bla M I S D B C | ank, octal dump, octal dump with mnemonic operation codes, fixed decimal dump, i.e. decimal integers, floating decimal dump, single precision, floating decimal dump, double precision, BCD dump, is affixed to the designator, if a snap of the console is to be included; |
| ^d 1, ^d 2, ^d 3 | control the start, stop and frequency of the dump. |
| | A dump will be produced at the d_1 -th encounter of the address 'a' and at every d_3 - th encounter ther after until the d_2 -th encounter is made. If these |

30.

C.R.S. Memo 5/Ed.1

e –

encounter between d₁ and d₂; is an optional identification of each dump, <u>It may</u> be up to 5 alphanumeric characters.

parameters are blank, a dump is produced at every

SNAP records are placed immediately before the RUN statement in the document. No TRACE statement, nor any equivalent, has been implemented in DAD.

13.2 Memory Map

id

Memory maps are always provided. The indicator 'm' on the RUN record becomes unnecessary but may be included for SCOPE format compatibility.

Locations are given as 6 octal digits (of which only 5 are significant to DAD, being a 1 bank system). The memory map has the absolute locations of the following items:

> Subprograms labelled common numbered common entry points number of card images read in remainder of memory available memory map if store overrun (as much as DAD will allow).

The format differs from the SCOPE map format in that the bank term is omitted from location values (see SCOPE Reference Manual 60053300 App. C).

If available storage is overrun on loading, the map also shows the amount of additional storage required.

Programs which fit into core only when displays are disabled and their core area made available must be submitted with directions to the operator for retention for running when displays are disabled.

13.3 Recovery Dump

On the RUN statement a recovery dump may be specified in case of abnormal termination of a program. Recovery dumps are in octal with mnemonics and have the following form:

- (i) Console scoop, if requested;
- (ii) print lines containing an absolute octal address, an octal address relative to the beginning of the subprogram or common block and the content of 10 words above and below the point causing the dump. When a new subprogram or common block is encountered, its name is printed and the relative address reset to zero. One or more identical words are omitted.

Details of functions of the recovery options are given under the RUN statement (see Section 12.5). For recovery dump diagnostics, and mnemonics and formats, see Appendix III of the main DAD Programmers Manual and SCOPE Ref. Man. 60053300 App. C.

14. ACCOUNTING

Accounting is performed by statistical sampling of the type of program running at the occurrence of standard time interrupts (at intervals of 1/10 second) and the total for the same execution may be subject to small fluctuations.

Some nominal charge may be made for input and output of documents according to their size.





N.B. *EXECUTE statement not implemented in the initial version of DAD for CR input. It is available via the Keyboard displays.



1

17

N.B. *EXECUTE statement not implemented in the initial version of DAD for CR input. It is available via the Keyboard displays.