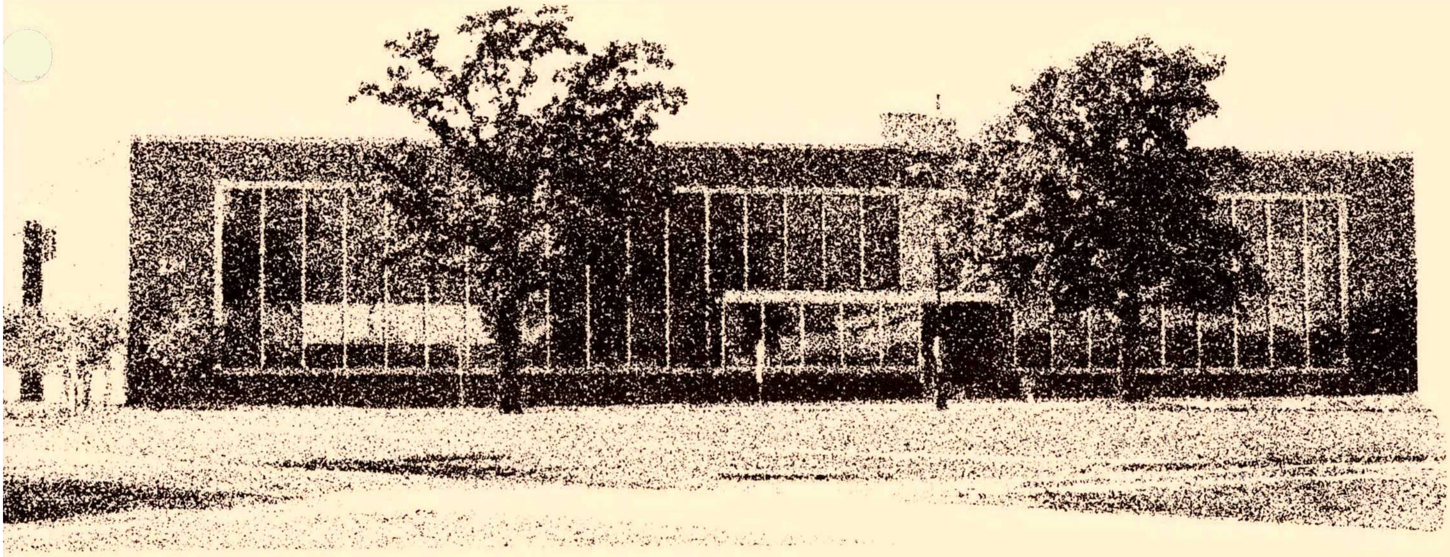
10
4/24/89

M.S.L.

①

ANL/TM 467
MARCH 1989

ANL Supplement to the CA-Disspla User's Manual



Argonne National Laboratory
Computing and Telecommunications Division
Operated by The University of Chicago for the U. S. Department of Energy under Contract W-31-109-Eng-38

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ANL/TM--467

DE89 010290

ANL/TM 467
COMPUTING AND
TELECOMMUNICATIONS
MARCH 1989

ANL Supplement to the CA-Disspla User's Manual

By

Michael M. Thommes

Edited by

Elizabeth M. Larson

(Intended primarily for internal distribution)

ACKNOWLEDGMENTS

We are grateful to those who reviewed drafts of the *ANL Supplement to the CA-DISSPLA USER'S MANUAL*: Pete Bertoncini, Clifford Caruthers, Chuck Harrison, Barry Miller, Fred Moszur, and Rich Raffenetti. Some of the information in this *Supplement* appeared originally in *Guide to Using Cuechart, Tellagraf, and Disspla at ANL* (ANL/TM 433).

We have prepared the text with University of Waterloo Generalized Markup Language (GML) and Script (Version 86.1) for printing on a Linotype L300P typesetter.

CONTENTS

TABLES

CHAPTER 1

GENERAL INFORMATION ON DISSPLA

The *ANL Supplement to the CA-DISSPLA USER'S MANUAL* (ANL/TM 467) summarizes installation-dependent options and features of Disspla; this *Supplement* supersedes *Using Cuechart, Tellegraf, and Disspla at ANL* (ANL/TM 433).

The information in this *Supplement* applies to version 10.5 of Disspla (which is currently installed in CMS, in MVS batch, and in several Argonne VAX/VMS systems), to Disspla 11.0 on the VAX 8700, and to version 10.0 of Disspla (which is currently installed on the Cray X-MP/14 under UNICOS). Unless this *Supplement* states otherwise, you should write Disspla programs according to instructions in the *CA-Disspla User's Manual.*

This chapter contains information common to Disspla as installed in CMS, MVS, VAX/VMS, and UNICOS. (Chapter Two contains information specific to using Disspla in each of these computer systems.)

PRECISION OF DISSPLA SUBROUTINES

You should be aware that all Disspla subroutines are written in single precision. Any variables that you pass to Disspla should be single precision.

FORTRAN LOGICAL UNIT NUMBERS RESERVED BY DISSPLA

Users should be aware that on each operating system, Disspla reserves the use of various logical unit numbers for scratch files and optional features. A complete list of Fortran logical unit numbers reserved by Disspla appears in Appendix A. Programs calling Disspla subroutines should not use these logical unit numbers except for their stated purpose. The unit number assignments are identified for each operating system. Note that unit numbers can vary by operating system. The dataset names and file characteristics corresponding to these unit numbers also appear.

Unlike previous versions, Disspla 10.5 will dynamically allocate many of the files that it requires. You no longer need to supply CMS file definitions, MVS JCL DD cards, or VAX/VMS ASSIGN statements for these files. If you do supply CMS file definitions, DD cards, or VAX/VMS ASSIGN statements for these files, Disspla will use the information *you* supply to allocate the files. Those files with an entry of "No" in the "Dynamic Allocation" column of the tables appearing in Appendix A indicate that you should issue the appropriate CMS file definitions, code MVS JCL DD cards, or supply VAX/VMS ASSIGN statements to use the corresponding Disspla options.

You may change the unit numbers Disspla uses for its DISSPOP metafiles by using the Disspla subroutine SETCPR. For example, in VMS the Fortran code

```
....
CALL COMPRS
CALL SETCPR(30,0,0,0)
...
```

will change the DISSPOP metafile output default logical unit number (19) to logical unit 30.

For CGM (Computer Graphics Metafiles) metafiles, you may change the default output name Disspla uses for CGM metafile output when you invoke the CGM generator subroutine CGMBO.[1] For example, in CMS the Fortran code

```
....
CALL CGMBO ('MYCGM OUTPUT',12,0,)
....
```

will route the CGM binary metafile to the file "MYCGM OUTPUT" rather than the default file, "CGMBOUT DATA." Section F of the *CA-Disspla User's Manual* for versions 10.0 and 10.5, and the "Metafile" section in the *CA-Disspla User's Manual,* Volume 2, Version 11.0 (RG99DS1102S) describe specific use of these Disspla subroutines.

Whether a file is dynamically allocated or not, you cannot use a particular unit number for your own purposes if you intend to use any feature of Disspla that uses that unit number. If the Fortran logical unit numbers that Disspla reserves conflict with the logical unit numbers used by your programs, you may (in MVS batch) call a locally written Fortran-callable subroutine, ANLDIS, to change the unit numbers reserved by Disspla. For more information about ANLDIS, see "Using ANLDIS to Change Disspla Unit Numbers" in Chapter Two.

[1] The CGMBO subroutine produces CGM metafiles written in binary format; other CGM formats can be produced. See the *CA-Disspla User's Manual* for details.

AVAILABLE DISSPLA OPTIONS

Table 1 lists the Disspla options currently available on the Argonne computer systems.

Table 1

Disspla Options Available on Argonne Central Computers

Option	*System*			
	CMS	MVS	8700 VAX/VMS	Cray UNICOS
Version	10.5	10.5	11.0	10.0
DISSPOP	Yes	Yes	Yes	*
GKS	Yes	Yes	Yes	Yes
Dynamics	Yes	Yes	Yes	Yes
Mapping	Yes	Yes	Yes	Yes
Page Layout (Tabletting)	Yes	Yes	Yes	Yes
Business Features	Yes	Yes	Yes	Yes
Contouring	Yes	Yes	Yes	Yes
Shaded Fonts	Yes	Yes	Yes	Yes
Codebook	Yes	Yes	Yes	

* CGM metafiles will be available in Disspla Version 11.0 under UNICOS

USING COMPUTER ASSOCIATES DEVICE DRIVERS WITH DISSPLA

To use Computer Associates (CA) graphics device drivers with your Disspla graphics programs, you must specifically call the CA device driver nomination routines. Table 8 lists common device drivers that are in use here at the Laboratory. For other available device drivers or for more explicit information on a particular driver, see the subsection "Obtaining Information on the CA Graphics Drivers" in Chapter Two within the section on your particular operating system ("Using Disspla in CMS," "Using Disspla in MVS Batch," "Using Disspla in VAX/VMS," or "Using Disspla in Cray UNICOS").

Your graphics program may include calls for one or more CA device driver nomination routines. You can call only one specific device driver or, optionally, branch to a device driver based on an answer to a previous prompt (see the example on page 13). If you decide to add a new device to your application program, you must add a call to the appropriate CA device driver nomination routine and recompile your program.

By specifically calling most CA device driver nomination routines, your graphics programs will become device-dependent. To retain device independence in your graphics programs, we recommend that you use the CA metafile driver routines to prepare CA metafiles for postprocessing with the CA DISSPOP postprocessor. See "Producing CA Graphics Metafiles with Disspla" below for specific details on using CA metafiles at Argonne. CA metafiles are both device-independent and machine-independent. You can send a CA metafile created on one computer to be postprocessed by the DISSPOP postprocessor or by Disspla or Tellagraf on another computer.

For a discussion of the advantages of writing device-independent graphics programs and for a general description of how to create and postprocess a CA metafile, see Section F of the *CA-Disspla User's Manual* for versions 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual,* Volume 2, Version 11.0 (RG99DS1102S).

You should call the local device driver nomination routines (STRTPL, STRTP2) only if you continue to use the locally written (obsolete) graphics device drivers (see "Using Locally Written Device Drivers with Disspla" below). No maintenance is applied to these obsolete drivers; for new programs or old programs where the source is still available, we encourage you to use the CA graphics drivers. In addition, the CA device drivers will ignore any calls to PLOTUTL routines.

PRODUCING CA GRAPHICS METAFILES WITH DISSPLA

Currently, two different types of graphics metafiles can be produced by CA device drivers: DISSPOP (or POP) and CGM (Computer Graphics Metafile). Each is device-independent and system-independent. Since each metafile is distinctly different, your particular needs should dictate which to use. A discussion of each appears below.

Producing DISSPOP or POP Metafiles

The DISSPOP (or POP) graphics metafile, written in binary format, was the first to be produced by CA. Although it is device-independent and system-independent, it is produced only by CA software products and can be read only by them. If you are passing metafiles between Tellagraf and Disspla or are postprocessing metafiles to the central computing graphics output devices (e.g., the Matrix camera, the CalComp 5835 electrostatic plotter), choose the DISSPOP metafile (CALL COMPRS). DISSPOP is the most compact of the metafiles available.

To create a DISSPOP metafile with Disspla, call the COMPRS subroutine by using the following structure in writing your graphics programs:

```
CALL COMPRS
        .
        .
        .

 (The rest of your Disspla program)

        .
        .
        .
CALL DONEPL
```

The call to COMPRS instructs Disspla to write graphics output into a CA DISSPOP metafile. In CMS, VAX/VMS, and Cray UNICOS, Disspla will dynamically allocate the CA DISSPOP metafile (for CMS and VAX/VMS, see Table 4 and Table 6 in Appendix A.

The CA DISSPOP metafile will be named POPFIL DATA A in CMS, POPFIL.DAT in the VAX/VMS system, and POPFIL in the Cray UNICOS operating system. In MVS batch, Disspla will write the metafile into the file specified by the DD card for Fortran logical unit 19. You must write the DD statement for this metafile; Disspla will not dynamically allocate the metafile. Use a DD card similar to the following in MVS batch when creating a DISSPOP metafile:

```
//GO.FT19F001 DD DSN=dsn,UNIT=unit,DISP=(NEW,CATLG),
//   SPACE=(TRK,(primary,secondary),RLSE)
```

where "dsn" is the name of the dataset to be used to store the metafile (e.g., B12345.POPFIL.DATA), "unit" is the class of output device used to store the metafile (e.g., PERM), and "primary" and "secondary" are the number of tracks used to store the metafile. Do not include DCB (Dataset Control Block) information. Disspla will build the file with the appropriate attributes.

Producing CGM Metafiles

The Computer Graphics Metafile (CGM) is a newer type of graphics metafile. It is an American National Standards Institute (ANSI) standard and an International Standards Organization (ISO) standard. It can be produced by both CA software and other vendor software. Because many products can both accept and produce the CGM metafile format, this metafile is appropriate if you want to modify or enhance your Disspla drawing with personal computer graphics packages like SuperImage (Computer Associates), Pixie (Zenographics, Inc.) or Harvard Graphics (Software Publishing Corp.). Because of CGM's transportability, you can also bring symbols or drawings from other personal computer graphics packages into your Disspla drawing.

CA CGM metafiles can be generated in three different encoding formats (bindings): binary, character, and clear text. Part F of the *CA-Disspla User's Manual* states the following reasons for selecting a particular coding:

- If your computer processing resources are limited, or if you require optimal speed generating and translating metafile, use binary encoding.
- If you require an ASCII file that can be efficiently transferred between computer systems, use character encoding.
- If you require a file in "English" to read and edit with a standard text editor, use a "clear text file."

Disk space requirements can also be an important consideration. While the binary and character (ASCII) output files produced from the same Disspla source code each occupy approximately the same amount of space, the clear text output file can occupy 3.5 times as much.

The most important consideration, however, is the usability of these CGM metafiles at ANL. The Computing and Telecommunications Division (CTD) has adopted the binary encoding format for CGM metafiles as its local standard. This form is produced and accepted by SuperImage, produced by the Tellagraf software, and expected as input by the CGM postprocessing utilities (HARDCOPY and CGMPOP). The binary encoding format should be chosen in most cases.

To create a CA CGM binary metafile with Disspla (available with Fortran 77 only), use the following structure in writing your graphics program:

```
CALL CGMBO ('name',length,prec)
      .
      .
      .
      .

 (The rest of your Disspla program)

      .
      .
      .
      .
CALL DONEPL
```

where "name" is the name of the CGM output file, "length" is the number of characters in "name," and "prec" is precision (0 for low precision storage and 1 for high precision storage). If "name" is 0 or if "length" = 0, the default name CGMBOUT (e.g., in VMS, CGMBOUT.DAT) is used.

As with the DISSPOP metafile, dynamic allocation occurs for the CGM metafile on the CMS and VAX/VMS systems. On MVS batch, no dynamic allocation occurs and you are required to code a DD statement when producing CGM metafiles. Use the structure shown below for coding this DD statement:

```
//GO.FT61F001 DD DSN=dsn,UNIT=unit,DISP=(NEW,CATLG),
// SPACE=(TRK,(primary,secondary),RLSE)
```

See Section F of the *CA-Disspla User's Manual* for versions 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual*, Volume 2, Version 11.0 (RG99DS1102S). for a discussion of the advantages of writing device-independent graphic programs and for a detailed description of the Disspla routines that you can use to create and postprocess CA metafiles.

To learn how to produce graphics output from CA graphics metafiles, see Appendix B.

USING LOCALLY WRITTEN DEVICE DRIVERS WITH DISSPLA

Although doing so was opposite to the current trend in CTD graphics, CTD wrote a PostScript graphics driver in 1986 for Disspla out of necessity, since (though PostScript printers were available), Computer Associates had not yet written a PostScript driver. This locally-written driver is available on all CTD operating systems, and the respective operating system sections below describe its use.

With the exception of the PostScript driver mentioned above, CTD has written no graphics device drivers for *new* graphics devices. However, so that previously developed graphics programs will continue to run without change, CTD still makes the locally written graphics device drivers available for use by your programs (see Appendix B of *Guide to Graphics at ANL* (ANL/TM 335). Specify which local graphics device driver to use by means of the PLOTTER exec in CMS, the PLOTTER keyword in MVS batch, and the PLOTTER logical name in VAX/VMS. To use the locally written device drivers, call the STRTP2 device nomination routine[2] to initialize the graphics environment in CMS and MVS batch. Call

[2] We have extended the ANL device nomination routine STRTPL by adding an additional entry point, STRTP2. The STRTPL routine previously used for this purpose is still available to insure compatibility with previously written graphics programs. We recommend that any new

the STRTPL device nomination routine in VAX/VMS. These locally written device drivers are not available in Cray UNICOS.

STRTP2 differs from STRTPL in that, after the device is determined from the PLOTTER parameters, the STRTP2 routine provides information to Disspla about the capabilities of the device. In addition, if you select G1DATA as the PLOTTER (on IBM systems only), then the STRTP2 routine can provide Disspla with the characteristics of the batch device on which the plot will ultimately be drawn. These characteristics consist of:

- The capability to honor color calls (all our drivers do).
- The capability to do hardware area fill (only the MATRIX driver does).
- The capability to display hardware characters (none of our drivers has this capability).
- The device maximum page size.
- The device pen width.
- The device X and Y resolutions.
- The device foreground and background colors.
- The device FILL/INCH factors (paper devices use INCH, metafiles use none, and all other devices use FILL).

To use STRTP2, use the following call:

```
CALL STRTP2(bdrvr,icode,ioption)
```

where "bdrvr" (a REAL*8 variable containing Hollerith characters) is the name of a batch driver (e.g., MATRIX), a 0, or a blank. Its use provides Disspla with device information when the PLOTTER is G1DATA. Disspla ignores "bdrvr" at other times. The variable "icode" (INTEGER*4) is equivalent to the Tellagraf DEVICE CODE. Currently, for our drivers, a value of 55 will produce a trace on FT06F001 of calls to our CA interface routines. The variable "ioption" (INTEGER*4) is equivalent to the Tellagraf DEVICE OPTION. Currently, only the Matrix camera uses it. An "ioption" value of 1 gives low resolution; 2 gives high resolution. All other values give low resolution.

graphics applications that use the locally written graphics device drivers use the new STRTP2 routine to initialize the graphics environment.

CHAPTER 2

SYSTEM-SPECIFIC INFORMATION ON DISSPLA

This chapter contains information specific to using Disspla as installed at Argonne in CMS, MVS, VAX/VMS, and UNICOS.

USING DISSPLA IN CMS

Disspla 10.5 resides in a variety of text libraries as object modules on the Graphics 001 minidisk. You must LINK and ACCESS this minidisk (or just enter GRAPHICS) before you can run a program using Disspla 10.5, and you must include the Disspla text libraries in a GLOBAL TXTLIB command before you run any Disspla program. Which text libraries you use are determined by the Fortran compiler and Disspla functions you want to use. Table 2 lists the libraries currently available. Examples of particular applications are in Table 3, which also lists auxiliary libraries that are not part of Disspla but may be necessary for your particular application.

The order in which the libraries appear in the GLOBAL TXTLIB statement is important! Be sure that you reference the INTLIB15 text library (if needed) first.

In addition, you should make sure your "blip" (the character CMS uses to show that CPU is being used) is set off (SET BLIP OFF). This action prevents spurious lines from being drawn on your terminal. Also, if you are using the CA graphics terminal drivers, you must issue the command

```
cp term linesize 255
```

before you run your Disspla program.

Table 2

Disspla Text Libraries in CMS

Library Name	*Function*
D105VSA TXTLIB	standard Disspla library for use with VS Fortran compiler
D105VSB TXTLIB	standard Disspla library for use with VS Fortran compiler (continued)
DISLIBVS TXTLIB	needed if you're using VS Fortran compiler
D105HXA TXTLIB	standard Disspla library for use with Fortran H extended compiler
D105HXB TXTLIB	standard Disspla library for use with Fortran H extended compiler (continued)
D105MOD TXTLIB	Disspla interface file library
INTLIB15 TXTLIB	contains CA graphics driver subroutines
PLOTUTL TXTLIB	library of routines for ANL-written graphics drivers and utility functions
GKSLIBA TXTLIB	Graphics Kernel Standard (GKS) library
GKSLIBB TXTLIB	Graphics Kernel Standard (GKS) library (continued)

Table 3

Required Text Libraries for CMS Fortan Compilers

Compiler Used	*Device Driver Wanted*	*Text Libraries Needed*
FORTVS	CA drivers	INTLIB15, D105MOD, D105VSA, D105VSB, VFORTLIB, CMSLIB*
FORTVS	local drivers	PLOTUTL, INTLIB15, D105MOD, D105VSA, D105VSB, VFORTLIB, CMSLIB*
FORTHX	CA drivers**	INTLIB15, D105MOD, D105HXA, D105HXB, FORTMOD2, CMSLIB
FORTHX	local drivers	PLOTUTL, INTLIB15, D105MOD, D105HXA, D105HXB, FORTMOD2, CMSLIB
FORTG1	CA drivers**	INTLIB15, D105MOD, DISLIBGI, D105HXA, D105HXB, FORTMOD2, CMSLIB

* In addition, the load library VFLODLIB LOADLIB must be made available with the GLOBAL LOADLIB VFLODLIB command when using the FORTVS compiler.

** The CGM Metafile driver is available only through the Fortran VS (FORTVS) compiler.

The following is an example of a short Disspla program that creates a POP metafile. The example shows the subsequent compilation of the program with VS Fortran and its execution in CMS. The resulting metafile, POPFIL DATA, is then postprocessed by the HARDCOPY command:

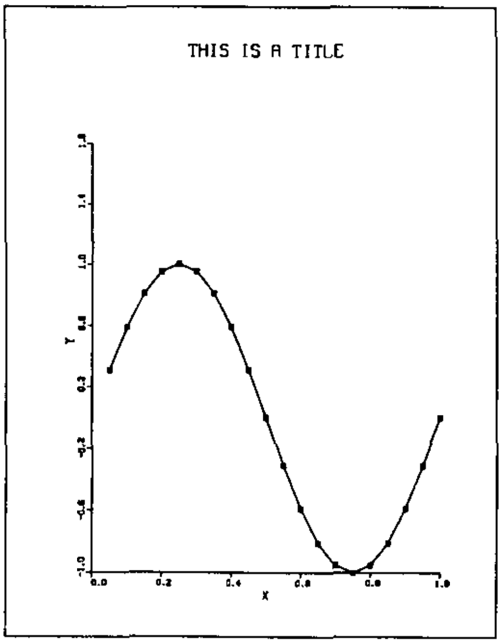
```
R; type example fortran

      DIMENSION X(20), SINX(20)
      DO 10 I=1,20
      X(I) = I/20.0
   10 SINX(I) = SIN(X(I)*2.0*3.14159)
      CALL COMPRS
      CALL AREA2D (6.0, 8.0)
      CALL HEADIN ('THIS IS A TITLE$', 100, 1.5, 1)
      CALL XNAME ('X$',100)
      CALL YNAME ('Y$',100)
      CALL GRAF (0.0, 0.2, 1.0, -1.0, 0.4, 2.0)
      CALL CURVE (X, SINX, 20, 1)
      CALL ENDPL (0)
      CALL DONEPL
      STOP
      END

R; set ldrtbls 10
R; global txtlib intlib15 d105mod d105vsa d105vsb vfortlib cmslib
R; global loadlib vflodlib
R; fortvs example
R; load example (start clear)

DMSLIO740I Execution begins...

END OF DISSPLA 10.5 -- 606 VECTORS IN 1 PLOTS.
RUN ON 1/26/89 USING SERIAL NUMBER 1 AT ARGONNE NATIONAL
PROPRIETARY SOFTWARE PRODUCT OF ISSCO, SAN DIEGO, CALIF.
269 VIRTUAL STORAGE REFERENCES; 5 READS; 0 WRITES.
R;
R; hardcopy
```



The following Fortran example shows you how you can choose from a variety of graphics devices at run time:

```
      DIMENSION X(20), SINX(20)
      DO 10 I=1,20
      X(I) = I/20.0
   10 SINX(I) = SIN(X(I)*2.0*3.14159)
C
C---->SELECT DEVICE TYPE
      PRINT *, '1 = TEK 4105, 2 = POP METAFILE, 3 = CGM METAFILE'
C
      READ *, IDEV
      IF (IDEV .EQ. 1) CALL TK41 (4105)
      IF (IDEV .EQ. 2) CALL COMPRS
      IF (IDEV .EQ. 3) CALL CGMBO (' ',0,0)
C
      CALL AREA2D (6.0, 8.0)
      CALL HEADIN ('THIS IS A TITLE$', 100, 1.5, 1)
      CALL XNAME ('X$',100)
      CALL YNAME ('Y$',100)
      CALL GRAF (0.0, 0.2, 1.0, -1.0, 0.4, 2.0)
      CALL CURVE (X, SINX, 20, 1)
      CALL ENDPL (0)
      CALL DONEPL
      STOP
      END
```

PL/I programmers can call the Fortran-written Disspla subroutines as they would any other Fortran subroutines. For information on how to call these subroutines, see Chapter 19 ("Interlanguage Communication Facilities") of *OS PL/I Checkout and Optimizing Compilers: Language Reference Manual* (GC33-0009).

Postprocessing CA Metafiles in CMS

The type of CA metafile you create (i.e., POP, CGM) and the kind of graphical output you want (e.g., screen, hardcopy) will determine how you should postprocess it. With POP metafiles, if you want to draw to your terminal, you have two choices: the ANYPOP utility or the DISSPOP exec. The ANYPOP utility (created by CA), requires a POP metafile with the name META DATA A4 as input (you will have to rename the Disspla default output POPFIL DATA A4 to META DATA A4). The system will prompt you for a device from the many (including a CGM metafile) available to you. Enter ANYPOP to invoke it. The DISSPOP exec uses the primary device specified in your Tellagraf profile to avoid prompting for a device. Enter HELP DISSPOP CMS for more information.

With binary CGM metafiles, if you want to draw to your terminal or create a POP metafile, use the CGMPOP utility. Enter CGMPOP to invoke it. The input file for this postprocessor should be named CGMBIN DATA A1; the system will prompt you for a device type.

For hardcopy output of either POP or CGM metafiles, use the HARDCOPY command. The system will prompt for the name of the hardcopy device (and its options) on which you want to draw your plots.

When you use the ANYPOP or CGMPOP utilities or the HARDCOPY command, the system will prompt you for postprocessor directives that allow you to choose and modify the plots you want to draw. See Part F ("DISSPOP") in the *CA-Disspla User's Manual* for versions 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual*, Volume 2, Version 11.0 (RG99DS1102S) for more information on these postprocessor directives.

Getting the ANL Logo

The ANL logo is available in CMS for your use as both a DISSPOP (or POP) metafile and as a binary CGM metafile. To draw the logo, use the following structure within the Disspla code that describes your plots (just before the call to ENDPL is a good place to draw it). This example places the logo in a 2 x 2 inch box at the lower left hand corner of the plot (1 inch from the bottom and left sides).

Using the POP format of the ANL logo:

```
CALL METNAM ('POPLOGO DATA *$',100)
CALL GETMET (0.,0.,0.,0.,1)
CALL PUTMET (1.0,1.0,2.0,2.0,0.,'AUTO')
```

Using the CGM format of the ANL logo:

```
CALL CGMBIN ('CGMLOGO DATA *$',100)
CALL GETMET (0.,0.,0.,0.,1)
CALL PUTMET (1.0,1.0,2.0,2.0,0.,'AUTO')
```

Note that when you are using metafiles for either input or output, you must have access to the INTLIB15 text library.

For additional information on the use of the above metafile routines, see Section F in the *CA-Disspla User's Manual* for versions 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual*, Volume 2, Version 11.0 (RG99DS1102S).

Using the ANL PS PostScript Driver

In CMS you can invoke the locally written PostScript driver, PS, by using the following subroutine call as your driver nomination routine:

```
CALL PS (ioption, icode, iunit)
```

where "ioption" is

0 - black an white output (default)
1 - gray scale with white drawn as black
2 - gray scale

"icode" is

0 - 8.5 x 11 page or portrait orientation (default)
1 - 11 x 8.5 page or landscape orientation

"iunit" (the Fortran logical unit number for the output file) is

0 - defaults to 32

The default output file created has the name FILE FT32F001 A1. Hardcopy output can be obtained through the HARDCOPY command if you first rename this file to have the filetype LISTPS before invoking HARDCOPY. Alternately, you can build a properly named file for the HARDCOPY command if you issue an appropriate file definition command for your PostSript file before running your Disspla program:

```
FILEDEF nn DISK filename LISTPS A
```

where "nn" is the Fortran logical unit number (usually 32), and "filename" is the output filename of the PostScript graphics file.

Obtaining Information on the CA Graphics Drivers

When using the CA graphics drivers with Disspla, you may find it necessary to get more information about a particular graphics driver and the capabilities of your hardware. Information on the graphics devices compatible with Disspla is available online within each of the operating systems that has Disspla software available.

In CMS, this information resides in a macro library (INTDOCnn MACLIB) on the graphics minidisk. To view the names of the members (graphic devices) in this library, enter:

```
MACLIB MAP INTDOCnn
```

where "nn" is the interface document number (currently "15").

Issuing this command will produce a file (INTDOCnn MAP A1) that contains the macro library member names (the names of the graphics drivers). Once you've found the name of the driver you're interested in (e.g., TK41 for the Tektronix 4100 series graphics terminals), you can view information about that particular driver by either editing or printing the macro library member as shown below:

```
XEDIT INTDOCnn MACLIB (MEMBER TK41)

OSPRINT INTDOCnn MACLIB * 3800 (MEMBER TK41)
```

The "OSPRINT" example sends the particular member of the macro library to the IBM 3800 laser printer in Building 221.

USING DISSPLA IN MVS BATCH

Disspla 10.5 in MVS batch resides in the following libraries: SYS1.DISLIBVS (for use with VS Fortran compiled programs) and SYS1.DISLIBHX (for use with the Fortran H extended compiler). These libraries also include the CA device driver nomination routines. Specify the appropriate library as the PRELIB or POSTLIB symbolic parameter within our local cataloged procedures (e.g., FORTVCLG). This specification will insure that the Disspla subroutines are available to the linkage editor or loader as one of the libraries searched in the SYSLIB chain. For example, to use Disspla with the Fortran H-extended compiler, code the following job contol language as part of your MVS batch job:

```
//stepname EXEC FTXCLG,PRELIB='SYS1.DISLIBHX'
```

To use Disspla with the VS Fortran compiler, code:

```
//stepname EXEC FORTVCLG,PRELIB='SYS1.DISLIBVS'
```

Because IBM no longer maintains the Fortran G1 compiler, the Disspla library for Fortran G1 is not available.

With Disspla 10.5 in MVS, most logical units are now dynamically allocated. Users should not code DD statements for logical units 11 (land blanking and contouring), 13 (tabletting), 76 or 77 (scratch area). (These unit numbers were used in Version 10.0.) The only logical units for which you might need a DD statement are 18 (for POP or CGM metafile input), 19 (for CA POP metafile output), or 61 (for CGM metafile output). For a complete description of MVS unit assignments used by Disspla, see Table 5 on page 29.

The following example shows the job control language necessary to compile, load, and run the EXAMPLE Disspla program (provided in "Using Disspla in CMS" above) on the MVS operating system. In Step 2, the resulting POP metafile is postprocessed to the Matrix camera for hardcopy:[3]

```
//EXAMPLE JOB
//STEP1 EXEC FORTVCLG,PRELIB='SYS1.DISLIBVS'
//FORTV.SYSIN DD *
      DIMENSION X(20), SINX(20)
      DO 10 I=1,20
      X(I) = I/20.0
   10 SINX(I) = SIN(X(I)*2.0*3.14159)
      CALL COMPRS
      CALL AREA2D (6.0, 8.0)
      CALL HEADIN ('THIS IS A TITLE$', 100, 1.5, 1)
      CALL XNAME ('X$', 100)
      CALL YNAME ('Y$', 100)
      CALL GRAF (0.0, 0.2, 1.0, -1.0, 0.4, 2.0)
      CALL CURVE (X, SINX, 20, 1)
      CALL ENDPL (0)
      CALL DONEPL
      STOP
      END
//GO.FT19F001 DD DSN=B12345.METAFILE.OUTPUT,
//    DISP=(NEW,CATLG),UNIT=PERM,SPACE=(TRK,(10,10),RLSE)
//STEP2 EXEC PMATRIX,INDSN='*.GO.FT19F001'
```

Note that your Disspla program could just have easily been read from an online disk instead of instream. Make sure that your program exists on disk in 80-column card format so the Fortran compiler can read it. An example of this method follows:

```
//EXAMPLE JOB
//STEP1 EXEC FORTVCLG,PRELIB='SYS1.DISLIBVS'
//FORTV.SYSIN DD DSN=B12345.DISSPLA.PGM,DISP=SHR
//GO.FT19F001 DD DSN=B12345.METAFILE.OUTPUT,
//    DISP=(NEW,CATLG),UNIT=PERM,SPACE=(TRK,(10,10),RLSE)
//STEP2 EXEC PMATRIX,INDSN='*.GO.FT19F001
```

[3] See Table 7 on page 30 for detailed information on available postprocessing cataloged procedures.

Postprocessing CA Metafiles in MVS Batch

Both the POP and CGM (binary) metafiles can be postprocessed in MVS batch with the suite of postprocessing cataloged procedures made available by CTD. Through these procedures (e.g., PMATRIX) you can produce output on all of the graphics hardcopy devices available. See "Using the MVS Batch Postprocessing Cataloged Procedures" beginning on page 26 for a list of available cataloged procedures and their options. The following job control language typifies use of one of these procedures:

```
//STEP1  EXEC PMATRIX,INDSN='B12345.MY.METAFILE'
```

Modification of plots in the metafile is possible through the postprocessing cataloged procedures with postprocessor directives. Some of the modifications available include selection of drawings, omission of drawings, scaling of drawings, and enabling of hardware features (e.g., rotation or hardware characters).

For example, the job control language given above might have included instructions for only drawing the first three plots in the metafile and scaling them down to half (0.5) size:

```
//STEP2  EXEC PMATRIX,INDSN='B12345.MY.METAFILE'
//FT05F001 DD
//         DD DDNAME=SYSIN1
//SYSIN1   DD *
 *
 DRAW=1-3*MODIFY=1-3(SCALE=0.5)
```

Note that the asterisk (*) in the second to last line (which starts the directives) and the directives themselves cannot start in column one.

See the DISSPOP section "F" in the *CA-Disspla User's Manual* for versions 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual,* Volume 2, Version 11.0 (RG99DS1102S) for more information on postprocessor directives.

Getting the ANL Logo

The ANL logo is available in MVS for your use as both a DISSPOP (or POP) metafile and as a binary CGM metafile. To draw the logo, use the following structure within your Disspla code that describes your plots (just before the call to ENDPL is a good place to draw it). This example places the logo in a 2 x 2 inch box at the lower left hand corner of the plot (1inch from the bottom and left sides).

Using the POP format of the ANL logo:

```
CALL METNAM ('SYS1.POPLOGO$',100)
CALL GETMET (0.,0.,0.,0.,1)
CALL PUTMET (1.0,1.0,2.0,2.0,0.,'AUTO')
```

In addition, you should include the following statement in your JCL:

```
//GO.FT18F001 DD DSN=SYS1.POPLOGO,DISP=SHR,LABEL=(,,,IN)
```

Using the CGM format of the ANL logo:

```
CALL CGMBIN ('SYS1.CGMLOGO$',100)
CALL GETMET (0.,0.,0.,0.,1)
CALL PUTMET (1.0,1.0,2.0,2.0,0.,'AUTO')
```

In addition, you should include the following statement in your JCL:

```
//GO.FT18F001 DD DSN=SYS1.CGMLOGO,DISP=SHR,LABEL=(,,,IN)
```

For additional information on the use of the above metafile routines, see Section F in the *CA-Disspla User's Manual* for versions 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual,* Volume 2, Version 11.0 (RG99DS1102S).

Using the ANL PS PostScript Driver

In MVS, to invoke the locally-written PostScript driver, PS, use the following subroutine call as your driver nomination routine:

```
CALL PS (ioption, icode, iunit)
```

where "ioption" is

0 - black and white output (default)

1 - gray scale with white drawn as black

2 - gray scale

"icode" is

0 - 8.5 x 11 page or portrait orientation (default)

1 - 11 x 8.5 page or landscape orientation

"iunit" (the Fortran logical unit number for the output file) is

0 - defaults to 32

MVS Disspla users should include the following JCL statements in the execution step of this program:

```
//PSOUTPUT OUTPUT DEST=node.printer,UCS=PS
//GO.FTnnF001 DD SYSOUT=A,OUTPUT=*.PSOUTPUT,
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000)
```

where "node" is the node that has the PostScript printer attached, "printer" is the name of the printer, and "nn" is the Fortran logical unit number (usually 32).

Using the CC5835 CalComp Driver

In MVS, you can create output for the CalComp 5835 Electrostatic wide-bed plotter with your Disspla program. To call this driver, include the following call in you Disspla program:

```
CALL CC58X5 (5835,-idunit)
```

where "-idunit" is the negative of the Fortran unit number (e.g., -31) of the DD statement for your graphics output. You must also include a FORMAT statement and DD statement pertaining to the Fortran output unit number as in the following example:

```
//jobname JOB
//*MAIN LINES=50
//*FORMAT PR,DDNAME=,DEST=LOCAL
//*FORMAT PR,DDNAME=GO.FTnnF001,
//*   DEST=ANLOS.CC5835,UCS=CC58
//    EXEC FORTVCLG,PRELIB='SYS2.DISMOD.LOAD',
//    PRELIB2='SYS1.DISLIBVS'
      .
      .
      .
      .
      .
      .
//GO.FTnnF001 DD SYSOUT=A
```

Because this driver program is still in beta test, you need to add the parameter PRELIB='SYS2.DISMOD.LOAD' to the cataloged procedure that you use to linkedit or load your Disspla program.

Obtaining Information on the CA Graphics Drivers

When using the CA graphics drivers with Disspla, you may find it necessary to get more information about a particular graphics driver and the capabilities of your hardware. Information on the graphics devices compatible with Disspla is available online within each of the respective operating systems that has Disspla software available.

In MVS, you can discover which graphics drivers are available by issuing the following command (in Wylbur):

```
SHOW DIR IN $SYS1.DISLIB.INTDOC
```

Next, submit job control language similar to the following to get a printout of a particular member (i.e., TK41):

```
//jobname JOB
// EXEC SDSKLIST,INDSN='SYS1.DISLIB.INTDOC(TK41)'
```

Using ANLDIS to Change Disspla Unit Numbers

ANLDIS is a locally written Fortran-callable subroutine--for use in MVS batch--that changes unit numbers used for Disspla files. If your program uses the standard Disspla Fortran unit numbers for purpose other than graphics input and output, you need to use ANLDIS. If you use ANLDIS to reassign any of the default Disspla units (76, 77, 89, 89, and 89) at run time, you will have to modify the ddnames used in the DD statements to match your unit assignments.

With ANLDIS, you can change logical unit assignments used by Disspla in MVS batch: add the following statement to your program before calling any of the graphics routines:

```
CALL ANLDIS (IFILE1,IFILE2,IFILE3,IFILE4,IFILE5)
```

where IFILE1, IFILE2, IFILE3, IFILE4, and IFILE5 are all integers. Each argument specifies the Fortran unit number to be assigned to a reserved Disspla file:

IFILE1	Disspla font file for Disspla 10.5 (optional) (Default = 76)
IFILE2	Scratch file for Disspla 10.5 (optional) (Default = 77)
IFILE3	Scratch file for contouring and land blanking (optional) (Default = 89)
IFILE4	Scratch file for contouring and land blanking (optional) (Default = 89)
IFILE5	Scratch file for contouring and land blanking (optional) (Default = 89)

A zero argument means that you do not want the Fortran unit number changed from the Disspla default. You can shorten argument lists, so long as you do not omit any preceding argument in the sequence. See the examples below.

ANLDIS Calling Example

ANLDIS, which resides in SYS1.DISLIBHX and SYS1.DISLIBVS, must be called before any device nomination calls.

Example 1: `CALL ANLDIS (2,3)`

Change Fortran unit assignments for Disspla 10.5 font file and scratch file:

```
//GO.FT02F001 DD DISP=SHR,DSN=SYS1.DISSPLA.DATA
//GO.FT03F001 DD UNIT=SASCR,SPACE=(TRK,(10,10)
```

ANLDIS Limitations

The following are guidelines to ANLDIS:

- You cannot call ANLDIS more than once.
- You cannot call ANLDIS after STRTPL.
- You cannot use Fortran unit numbers 5 or 6, nor can the unit numbers be outside the range of 1 to 99.
- You cannot call ANLDIS with no arguments or with more than five arguments.

USING DISSPLA IN VAX/VMS

Once you've created your Disspla program with one of the many editors available on the VAX/VMS system, it must be compiled and linked before it can be run. Initially, enter SETUP DISSPLA to establish the proper environment. When you link your Disspla program on the VAX 8700, use the DISLIB graphics library. (Disspla users on other VAX/VMS systems should consult their system manager for further details on using Disspla on those systems.) Note the following typical VMS session in which the EXAMPLE Fortran Disspla program (in "Using Disspla in CMS" above) is run:

```
$ FORTRAN EXAMPLE
$ LINK EXAMPLE,DISLIB/OPT
$ RUN EXAMPLE
```

The POP metafile, created by the COMPRS subroutine in the EXAMPLE Disspla program, is placed in your current directory under the name POPFIL.DAT.

If you have a graphics terminal and wish to plot directly at the terminal, you may replace the COMPRS subroutine with one of the many terminal graphics drivers written by CA. For more information on the graphics drivers available, see below.

Postprocessing CA Metafiles in VAX/VMS

The type of CA metafile you create (POP, CGM) and the kind of graphical output you want (screen, hardcopy) will determine how you should postprocess it. With POP metafiles, if you want to draw at your terminal, use the interactive ANYPOP utility. The ANYPOP utility, invoked by entering ANYPOP at the DCL level, requires as input a POP metafile with the name META.DAT (you will have to rename the Disspla default POP output POPFIL.DAT to META.DAT). The program will prompt you for a device from the many (including a CGM metafile) available to you.

With binary CGM metafiles, if you want to draw to your terminal or create a POP metafile, use the CGMPOP utility. This utility, invoked by entering CGMPOP at the DCL level, expects a binary CGM metafile named CGMBIN.DAT as input (you will have to rename the Disspla default CGM output file, CGMBOUT.DAT, to CGMBIN.DAT). The system will prompt you for a device from the many (including a POP metafile) available to you.

For hardcopy output with either POP or CGM metafiles, use the HARDCOPY command. The system will prompt you for the name of the hardcopy device (and its options) on which you want to draw your plots. You can use HARDCOPY either interactively or in batch. Enter HELP HARDCOPY for more information.

When you are using the ANYPOP or CGMPOP utilities or the HARDCOPY command, the system will prompt you for postprocessor directives that allow you to choose or modify the plots you want to draw. See Part F, "DISSPOP," in the *CA-Disspla User's Manual* for versions 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual,* Volume 2, Version 11.0 (RG99DS1102S) for more information on these postprocessor directives.

Getting the ANL Logo

The ANL logo is available in VMS for your use as both a DISSPOP (or POP) metafile and as a binary CGM metafile. To draw the logo, use the following structure within your Disspla code that describes your plots (just before the call to ENDPL is a good place to draw it). This example places the logo in a 2 x 2 inch box at the lower left hand corner of the plot (1 inch from the bottom and left sides).

Using the POP format of the ANL logo:

```
CALL METNAM ('SYS_ANLDATA:POPLOGO.DATA$',100)
CALL GETMET (0.,0.,0.,0.,1)
CALL PUTMET (1.0,1.0,2.0,2.0,0.,'AUTO')
```

Using the CGM format of the ANL logo:

```
CALL CGMBIN ('SYS_ANLDATA:CGMLOGO.DATA$',100)
CALL GETMET (0.,0.,0.,0.,1)
CALL PUTMET (1.0,1.0,2.0,2.0,0.,'AUTO')
```

For additional information on the use of the above metafile routines, see Section F in the *CA-Disspla User's Manual* for versions 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual*, Volume 2, Version 11.0 (RG99DS1102S).

Using the ANL PS PostScript Driver

In VMS, to invoke the locally-written PostScript driver, PS, use the following subroutine call as your driver nomination routine:

```
CALL PS (ioption, icode, iunit)
```

where "iopion" is

0 - black and white output (default)

1 - gray scale with white drawn as black

2 - gray scale

"icode" is

0 - 8.5 x 11 page or portrait orientation (default)

1 - 11 x 8.5 page or landscape orientation

"iunit" is

0 - defaults to 32

The default output file has the name FOR032.DAT. You can send this file to a PostScript printer with the PSLIST command:

```
PSLIST FOR032.DAT node::printer
```

where "node" is the node where the printer is located (if not supplied, the default is ANLOS), and "printer" is the name of the PostScript printer.

Using the Computer Associates PSCRPT PostScript Driver

In VMS, under Disspla 11.0 you can invoke the Computer Associates PostScript driver, PSCRPT. Prior to nominating this routine, you must establish your graphics environment with calls to the IOMGR subroutine as shown in the following example:

```
C  SET OUTPUT FOR STANDARD SPOOL FILER
   CALL IOMGR (5,-102)
C  GIVE OUTPUT FILE A NAME
   CALL IOMER ('OUTPUT.LISTPS',-103)
C  SET THE FILE MODE;1=NEW,2=OVERWRITE OLD
   CALL IOMER (1,-104)
C  NOW CALL THE POSTSCRIPT DRIVER
   CALL PSCRPT (xdim, ydim, width)
  .
  .
  .
```

"xdim" is the X dimension:

0 - 7.99 inches (default)

"ydim" is the Y dimension:

0 - 10.78 inches (default)

"width" is the line width in inches:

0 - .0139 inches (default)

Currently, CTD believes the Computer Associates PostScript driver to be inferior to the ANL PostScript driver. The CA driver creates much larger and less efficient PostScript output files than does our local driver. For these reasons, we recommend that you use the local PostScript driver until the vendor's driver is deemed as good as or better than our local driver.

Obtaining Information on the CA Graphics Drivers

When using the CA graphics drivers with Disspla, you may find it necessary to get more information about a particular graphics driver and the capabilities of your hardware. Information on graphics devices compatible with CA is available online within each of the respective operating systems that have Disspla software available.

In VAX/VMS on the 8700, the graphics information for each driver is available as a separate file that you may type or print out. To see the names of different files (all have the extension ".DOC"), issue the following command:

```
$ DIR DISLIB_INTDOC
```

Once you have found the appropriate file, you can view the information in it by typing or printing it, as in the following examples:

```
$ TYPE DISLIB_INTDOC:TK41.DOC

$ OSPRINT DISLIB_INTDOC:TK41.DOC 3800
```

The "OSPRINT" example sends the documentation for the TK41 device driver to the IBM 3800 laser printer in Building 221.

USING DISSPLA IN CRAY UNICOS

Some general information on using Disspla in Unicos follows; however, user's should see *Guide to Unicos at ANL* (ANL/TM 460) for detailed information.

There are different versions of Disspla graphics on different computing systems at ANL. The following sections document the Fortran unit numbers reserved by Disspla 10.0 in UNICOS, and the limitations of Disspla 10.0 when processing character string arguments in Cray Fortran. These details affect the portability of Disspla graphics applications among the various computing systems at ANL. We expect both of these topics to change with the release of Disspla 11.0 for UNICOS.

Fortran Logical Unit Numbers Reserved by Disspla in UNICOS

Disspla uses many Fortran logical unit numbers for its data and work files. Some of these files are used by all Disspla applications; others are used only when specific Disspla options (e.g., contouring) are chosen. All files used by Disspla are automatically allocated and deallocated as needed. The scratch files needed by Disspla are created and deleted automatically. You should not use these reserved logical unit numbers for your own files in Fortran programs that use Disspla. (You can use the unit numbers Disspla reserves for options which you are not using.)

The following table describes the files and associated Fortran logical unit numbers used by Disspla, grouped by Disspla options. Interpret the *TYPE/name* column as follows: (1) files labeled *PERMANENT* are automatically allocated Disspla data files; (2) files labeled *SCRATCH* are automatically allocated and deleted Disspla scratch files; and (3) files with lowercase labels are permanent user files identified by their Disspla assigned default filenames. For example, **popfil** is the file name Disspla assigns to the CA/ISSCO graphics metafile you create when you call the **COMPRS** subroutine.

OPTION	*UNIT*	*TYPE/name*	*Description of File*
All	21	PERMANENT	Character set tables
All	23	SCRATCH	Workspace & data areas
All	34	PERMANENT	Security data file
DISPOP	95	**metfil**	Disspla input metafile
DISPOP	96	**popfil**	Disspla output metafile
Map Data	11	PERMANENT	Geographic map data
Landblanking	12	PERMANENT	Geographic outlines
Landblanking	14	SCRATCH	Scratch work file
Contouring	14	SCRATCH	Scratch work file
Tabletting	13	SCRATCH	Scratch work file
Dynamics	1	PERMANENT	Available device types
Dynamics	3	PERMANENT	Device capabilities
Dynamics	97	PERMANENT	Device commands
Dynamics	98	**savsst**	Software segmentation

Disspla reserves different Fortran logical unit numbers in different computing systems. This is an impediment to writing portable Disspla graphics applications. See the appendices of *Guide to Using Cuechart, Tellagraf, and Disspla at ANL* (ANL/TM 433) for tables of Disspla's reserved logical unit numbers in other computing systems at ANL. The locally written **ANLDIS** subroutine, which allows MVS users to reassign several of Disspla's reserved logical unit numbers, is not available in UNICOS. You can, however, reassign the metafile output unit with the Disspla **SETCPR** subroutine.

Limitations on Character Arguments to Disspla 10.0 Subroutines

Disspla subroutines take character arguments to annotate plots and name files. Disspla 10.0 is fully compatible with Fortran 66 style Hollerith arguments, but not with all Fortran 77 CHARACTER type arguments. Disspla 10.0 requires that Cray Fortran CHARACTER arguments begin on Cray word boundaries. Disspla does not have this limitation on character (byte) addressable machines, so Disspla applications that work in IBM and VAX Fortran, may misbehave in Cray Fortran.

In Cray Fortran, CHARACTER constants, variables, and arrays are stored eight characters per Cray word, beginning on Cray word boundaries. But, CHARACTER substrings and individual elements of CHARACTER arrays may begin anywhere within a word. Consider the following Fortran fragment:

```
      CHARACTER*4 FNAME(3)
      DATA FNAME /'POP1','POP2','POP3'/
      ...
      DO 100 I=1,3
      CALL POPNAM (FNAME(I),4)
      ...
   100 CONTINUE
```

The first element of the FNAME array begins on a word boundary. The second element of FNAME does not begin on a word boundary; it begins at the fifth character position of the word containing FNAME(1). Since Disspla 10.0 assumes all character arguments begin on word boundaries, this loop will produce incorrect results for I=2. To avoid this problem, pad all CHARACTER array elements out to multiples of eight characters:

```
      CHARACTER*8 FNAME(3)
      DATA FNAME /'POP1    ','POP2    ','POP3    '/
      ...
      DO 100 I=1,3
      CALL POPNAM (FNAME(I),4)
      ...
   100 CONTINUE
```

Now each array element occupies exactly one Cray word, so each element begins on a word boundary. This restriction will be removed in Disspla 11.0.

APPENDIX A

DISSPLA LOGICAL UNIT ASSIGNMENTS IN CMS, MVS BATCH, AND VAX/VMS SYSTEMS

Disspla reserves certain Fortran logical unit numbers for itself. These logical unit assignments differ in CMS, MVS batch, and VAX/VMS. Table 4, Table 5, and Table 6 list the Fortran logical unit assignments Disspla uses in each of these environments. Logical unit assignments Disspla uses in the Cray UNICOS operating system can be found in the section describing Disspla usage on the Cray. Included in the tables are the default dataset names of each file and the characteristics of each file (e.g., record format, block size, and whether Disspla will dynamically allocate the file). If Disspla dynamically allocates a file, you need not use a CMS filedef, MVS batch DD cards, or a VAX/VMS ASSIGN statement for that file. If you do supply a CMS filedef, MVS batch DD card, or a VAX/VMS ASSIGN statement for a file, it will override the default supplied by Disspla.

Table 4

Disspla CMS Fortran Logical Unit Assignments

Dataset Description and Dataset Name	*Logical Unit*	*Record Format*	*Record Length*	*Block Size*	*Dynamic Allocation*
Landblanking, contouring, tabletting CSCR TMP A (Scratch)	89	F	2400	2400	Yes
Disspla Metafile Input(DISSPOP or POP) META DATA A4	18	VS	724	728	Yes
Disspla Metafile Input (CGM) CGMBIN DATA A1 (binary)	18	F*	512	512	Yes
CGMCIN DATA A1 (character, ASCII)	18	F	80	80	Yes
CGMTIN DATA A1 (clear text)	18	F	80	80	Yes
Disspla Metafile Output (DISSPOP or POP) POPFIL DATA A4	19	VS	724	728	Yes
Disspla Metafile Output (CGM) CGMBOUT DATA A1 (binary)	61	F	512	512	Yes
CGMCOUT DATA A1 (character, ASCII)	61	F	80	80	Yes
CGMTOUT DATA A1 (clear text)	61	F	80	80	Yes
Virtual Storage VSCR TMP A1 (Scratch)	23	F	2000	2000	Yes

* The Disspla CGM translators can accept CGM metafiles with other record formats.

Table 5

Disspla 10.5 MVS Batch Fortran Logical Unit Assignments

Dataset Description and Dataset Name	*Logical Unit*	*Record Format*	*Record Length*	*Block Size*	*Dynamic Allocation*
Contouring &&LBSCR (Scratch)	89	F	2000	2000	Yes
Disspla Device Commands SYS1.DEVCMD.DATA	47	FB	2000	2000	Yes
Disspla Save Session &&SAVSST	46	VBS	2004	10024	Yes
Disspla Metafile Input (DISSPOP or POP) &&META	18	VS	-32768	800	No
Disspla Metafile Output (DISSPOP or POP) &&POPFIL	19	VS	-32768	800	No
Disspla CGM Metafile Input					
&&CGMBIN (binary)	18	F*	512	6144	No
&&CGMCIN (character, ASCII)	18	F	80	6320	No
&&CGMTIN (clear text)	18	F	80	6320	No
Disspla CGM Metafile Output					
&&CGMBOUT (binary)	61	F	512	6144	No
&&CGMCOUT (character, ASCII)	61	F	80	6320	No
&&CGMTOUT (clear text)	61	F	80	6320	No
Virtual Storage SYS1.DISSPLA.DATA (Permanent)	76	F	2000	2000	Yes
Virtual Storage &&VSSCR (Scratch)	77	F	2000	2000	Yes

* The CGM translators can accept CGM metafiles with other record formats.

The filenames META.DAT and POPFIL.DAT in Table 6 reflect the default filenames for Disspla metafile input and output files as provided by the vendor. Check with your VAX manager to ensure that these defaults have been preserved on your system.

Table 6

Disspla VAX/VMS Fortran Logical Unit Assignments

Dataset Description and Dataset Name	*Logical Unit*	*File Org*	*Record Format*	*Record Length*	*Dynamic Allocation*
Landblanking, contouring, tabletting CSCR.TMP (Scratch)	89	Seq	Fixed	2000	Yes
Disspla Metafile Input (POP) META.DAT	18	Seq	Variable	722	Yes
Disspla Metafile Output (POP) POPFIL.DAT	19	Seq	Variable	722	Yes
Disspla Metafile Input (CGM) CGMBIN.DAT (binary)	18	Seq	Fixed	512	Yes
CGMCIN.DAT (character, ASCII)	18	Seq	Fixed	80	Yes
CGMTIN.DAT (clear text)	18	Seq	Fixed	80	Yes
Disspla Metafile Output (CGM) CGMBOUT.DAT (binary)	61	Seq	Fixed	512	Yes
CGMCOUT.DAT (character, ASCII)	61	Seq	Fixed	80	Yes
CGMTOUT.DAT (clear text)	61	Seq	Fixed	80	Yes
Virtual Storage DIS105DATA:DVSD.DAT (Permanent)	21	Seq	Fixed	2000	Yes
Virtual Storage DISSCRAT.TMP (Scratch)	23	Seq	Fixed	2000	Yes

APPENDIX B

USING THE MVS BATCH POSTPROCESSING CATALOGED PROCEDURES

CA metafiles--either the DISSPOP (POP) type or the CGM binary type--can be used as input to the MVS batch postprocessing cataloged procedures to obtain hardcopy. A list of available procedures appears in Table 7, in addition to optional features on each device that you may choose.

The following example uses one of the cataloged postprocessing procedures (PMATRIX) and specifies options available (high resolution, 8"x10" viewgraph format) with an already existing CA metafile:

```
//HARDCOPY JOB
//STEP1  EXEC PMATRIX,POPOPT=POPOPT2,PPARM='8x10',
//    INDSN='B12345.MY.METAFILE'
```

Besides the hardcopy device options available to you, you also have the option of using postprocessor directives to modify your CA metafile (each cataloged postprocessing procedure has its own set of default directives [member &DIR] specified in the dataset ANL1.PROCDATA). Note that any directives you specify are added to the default directives. The post processor will resolve any conflicts and will honor the last given specification. Some of the modifications available include plot selection, plot scaling, plot positioning, and plot rotation. The example below highlights the use of postprocessor directives:

```
//HARDCOPY  JOB
//STEP1  EXEC PMATRIX,POPOPT=POPOPT2,PPARM='8x10',
//    INDSN='B12345.MY.METAFILE'
//FT05F001 DD
//          DD DDNAME=SYSIN1
//SYSIN1    DD *
 *
 DRAW=1,3*SCALE=0.5
```

Note that postprocessor directives cannot start in column one.

A detailed discussion on the use of postprocessor directives appears in Appendix F ("DISSPOP") of the *CA-Disspla User's Manual* for version 10.0 and 10.5 or the "Metafile" section in the *CA-Disspla User's Manual,* Volume 2, Version 11.0 (RG99DS1102S).

The HARDCOPY commands available on the CMS, VMS, and UNICOS operating systems build and submit (to the MVS batch operating system) job control language similar to that above, depending on your choice of output device, device options and postprocessor directives.

Table 7

Available MVS Batch Postprocessing Cataloged Procedures

Cataloged Procedure: PCC5835
Device: CC5835
Description: Calcomp 5835 wide bed Electrostatic Color Plotter
Options: DIR=POPHSHC use hardware shading and hardware characters (default)
DIR=BLOWUP default + scales by a factor of 3
DIR=STACK3 default + stacks plots 3 high (12"x12" plot max)
INDSN=name of input metafile (default=&META)

Cataloged Procedure: PDS35BW
Device: DS35BW
Description: Double frame sprocketed 35mm black and white FR80 film
Options: INDSN=name of input metafile (default=&META)

Cataloged Procedure: PDS35CLR
Device: DS35CLR
Description: Double frame sprocketed 35mm color FR80 film
Options: INDSN=name of input metafile (default=&META)

Cataloged Procedure: PMATRIX
Device: MATRIX
Description: Matrix 35mm rasterizing color camera
Options: INDSN=name of input metafile (default=&META)
POPOPT=POPOPT1 low resolution, 2048 lines/35mm (default)
POPOPT=POPOPT2 high resolution, 4096 lines/35mm
PPARM= mounted positive side (default)
PPARM=NEG unmounted negative film
PPARM='8x10' 8"x10" color acetate viewgraph

Cataloged Procedure: PMETAFIL
Device: METAFILE
Description: Conversion of Computer Associates metafile to old 31-bit encoded Argonne-developed metafile
Options: INDSN=name of input metafile (default=&META)

Cataloged Procedure: PMETASHR
Device: METASHRT
Description: Conversion of Computer Associates metafile to old 15-bit encoded Argonne-developed metafile
Options: INDSN=Name of input metafile (default=&META)

Cataloged Procedure: PPRINTER
Device: PRINTER
Description: any printing device (locally-written driver)
Options: INDSN=name of input metafile (default=&META)

Cataloged Procedure: PPRT
Device: PRT
Description: any printing device (CA-written driver)

Cataloged Procedure: PPS
Device: PS
Description: Any printer that supports the Adobe Postscript page description language
Options: CODE=0 portrait format, 8 1/2"x11" (default)
CODE=1 landscape format, 11"x8 1/2"
INDSN= name of input metafile (default=&META)
OPTION=0 black and white output (default)
OPTION=1 gray scale with white drawn as black
OPTION=2 gray scale on black and white devices else color

Table 7

Available MVS Batch Postprocessing Cataloged Procedures
(Continued)

Cataloged Procedure: PSPU35BW
Device: SPU35BW
Description: spooled version of PU335BW
Options: INDSN= name of input metafile (default=&META)

Cataloged Procedure: PS16BW
Device: S16BW
Description: sprocketed 16mm black and white FR80 film
Options: INDSN= name of input metafile (default=&META)

Cataloged Procedure: PS16CLR
Device: S16CLR
Description: sprocketed 16mm color FR80 film
Options: INDSN= name of input metafile (default=&META)

Cataloged Procedure: PS35BW
Device: S35BW
Description: sprocketed 35mm black and white FR80 film
Options: INDSN= name of input metafile (default=&META)

Cataloged Procedure: PS35CLR
Device: S35CLR
Description: sprocketed 35mm color FR80 film
Options: INDSN = name of input metafile (default=&META)

Cataloged Procedure: PU105BW
Device: U105BW
Description: unsprocketed 105mm black and white FR80 film
Options: INDSN= name of input metafile (default=&META)

Cataloged Procedure: PU35BW
Device: U35BW
Description: unsprocketed 35mm black and white FR80 film
Options: INDSN= name of input metafile (default=&META)

Cataloged Procedure: PWYL4012
Device: WYL4012
Description: Conversion of Computer Associates metafile to Wylbur
file for Tektronix 4012 terminal
Options: BAUD= pads file with nulls for faster baud rates.
Acceptable values are 300, 1200, 2400, 4800, 9600.
INDSN= name of input metafile (default=&META)

Cataloged Procedure: PWYL4014
Device: PWYL4014
Description: Conversion of Computer Associates metafile to Wylbur
file for Tektronix 4014 terminal
Options: BAUD= pads file with nulls for faster baud rates.
Acceptable values are 300, 1280, 2400, 4800, 9600.
INDSN= name of input metafile (default=&META)

APPENDIX C

USING DISSPLA ROUTINES TO NOMINATE CA DEVICE DRIVERS

CA has supplied drivers for approximately 350 different types of graphics output devices. Your program can produce output for any of these graphics devices if you include a subroutine call to the proper device nomination routine. If your program might produce output for one of several different graphics devices, you must include a subroutine call for each potential graphics output device. Table 8 below lists the CA device driver nomination routine calls and calling parameters for a specific device driver nomination routine. Note that if you choose to create only a CA metafile and postprocess the metafile later, you need not bother using the specific device driver nomination routine calls given here.

Table 8

CA Device Driver Nomination Routines

Device Manufacturer and *Model Number*	*Subroutine Call*	Calling Parameters (See Table 9, "Parameter *Descriptions"*)
Advanced Electronics Design		
Model 512	CALL AED512	
Model 767	CALL AED767	
California Computer Products (Calcomp)		
Model 81	CALL CAL81 (IDEVC,IOPTIN)	1
Model 84	CALL CAL84 (IDEVC,IOPTIN)	1
All other models	CALL CALCMP (0,0,IDUNIT)	2
Chromatics		
Models 1398, 1399, 1598, 1599, 1998, 1999, and 3999	CALL CHROM (IMODEL,ICPS, IDEVC,IOPTIN)	3, 4
Model 7900	CALL CG7900	
Computer Devices Miniterm		
Model 1203	CALL MINTRM	
Model 2000	CALL CDI (IRES)	5, 6
C. Itoh		
Model 261	CALL CIT261 (ICPS)	4
Model 414	CALL CIT414 (ICPS)	4
Datagraph VTC	CALL VTC	
Datamedia Colorscan 10	CALL DATMED (0)	
Data Terminal Corporation	CALL DTC	
Digital Engineering		
GEN II Board	CALL DEGII (IMODEL)	7
Retrographics VT-640	CALL DEVT (ICPS)	4
Digital Equipment Corporation REGIS terminals		
Model VT-125	CALL REGIS (1,ICOLOR)	8
Model VK-100	CALL REGIS (2,ICOLOR)	8
Model VT-240 or VT-241	CALL REGIS (3,ICOLOR)	8
Model 350PC	CALL REGIS (4,ICOLOR)	8
Digital Equipment Corporation		
Models VT240 or VT241	CALL VT240	
Envision		
Model 215	CALL ENV230 (ICPS,3)	4
Model 220 with graphics	CALL ENV230 (ICPS,2)	4
Model 230	CALL ENV230 (ICPS,2)	4
Model 220 w/o graphics	CALL ENV230 (ICPS,1)	4

Table 8: CA Device Driver Nomination Routines (Continued)

Florida Computer Graphics		
Beacon	CALL FCG (1, 1)	
Hewlett Packard		
Model 2623	CALL HP2623	
Model 2627	CALL HP2627 (0)	
Model 2647A	CALL HP2647	
Model 2648A	CALL HP2648	
Model 2703A	HP27(IRES)	5
Model 7220A	CALL H7220A	
Model 7220B	CALL H7220B	
Model 7220C	CALL H7220C	
Model 7220S	CALL H7220S	
Model 7220T	CALL H7220T	
Model 7221A	CALL H7221A	
Model 7221B	CALL H7221B	
Model 7221C	CALL H7221C	
Model 7221S	CALL H7221S	
Model 7221T	CALL H7221T	
Model 7470A	CALL HP7470	
Model 7475A	CALL HP7475 (IOPTIN)	11
Model 7550A	CALL HP7550 (IOPTIN)	11
Model 7580A	CALL HP7580	
Model 7580B	CALL H7580B	
Model 7585B	CALL HP7585	
Model 7586B	CALL HP7586	
Models 9872A, B, C, S, T	CALL HP98 (IOPTIN,IDEVC,IADRS)	12
Houston Instruments HIPLOT Series		
Models 3, 4, 6, 7, and 29	CALL HIPLOT (IMODEL,IDUNIT, IOPTIN)	9
Model 5	CALL HUSTN5 (5,IDUNIT,IRES, IPWDTH,IOPTIN,IMODES)	5, 10
Model 6	CALL HUSTN6 (5,IDUNIT,IRES, IPWDTH,IOPTIN,IMODES)	5, 10
IBM GDDM Versions		
Model 3268	CALL GDDM3 (2,IDEVC,IPWDTH, IOPTIN)	13
Model 3279	CALL GDDM3 (1,IDEVC,IPWDTH, IOPTIN)	13
Model 3287	CALL GDDM3 (2,IDEVC,IPWDTH, IOPTIN)	13
IBM 32xx Series		
Model 3277	CALL IBM32 (3277,0,0)	
Model 3278	CALL IBM32 (3278,0,0)	
Model 3290	CALL IBM32 (3290,0,0)	
IBM 4250 Printer	CALL IBM42 (IDEVC,IOPTIN, IPWDTH,0,0)	14
IBM Instruments		
Model XY/749	CALL IBM749 (IDEVC,IOPTIN)	1
Model XY/750	CALL IBM750 (IDEVC,IOPTIN)	1
IBM Personal Computer	CALL IBMPC (IDEVC,IFILL)	15
IBM Personal Computer 2.0	CALL IBMPC2 (IOPTIN,IDEVC)	16
ID Systems ID-100V2		
Model GA	CALL ID100V (ICPS)	4
Model GB	CALL ID100V (ICPS)	4
Image Resources Samurai	CALL SAMRAI (IDEVC,IOPTIN)	17
Intelligent Systems Corporation		
Model 8001 I	CALL ISC (2)	
Model 8001 R	CALL ISC (2)	

Table 8: CA Device Driver Nomination Routines (Continued)

Jupiter Systems Jupiter 7	CALL JUPITER (0,0)	
Modgraph GX100	CALL GX100 (ICPS,IRES)	4, 5
Quality Microsystems Lasergrafix Models 800, 1200, and 2400	CALL QMS	
Ramtek (Compact Vector Firmware)		
Model 6210	CALL RAMCMP (6210)	
Model 6211	CALL RAMCMP (6211)	
Model 6212	CALL RAMCMP (6212)	
Model 6412	CALL RAMCMP (6412)	
Ramtek (TPLOT Firmware)		
Model 6210	CALL RAMTEK (6210)	
Model 6211	CALL RAMTEK (6211)	
Model 6212	CALL RAMTEK (6212)	
Model 6412	CALL RAMTEK (6412)	
Raster Technologies		
Model ONE/10	CALL RASTEC (10)	
Model ONE/25	CALL RASTEC (25)	
Model ONE/40	CALL RASTEC (40)	
Model ONE/60	CALL RASTEC (60)	
Model ONE/80	CALL RASTEC (80)	
Seiko D-Scan		
Model GR-1104	CALL DSCAN (1104)	
Model GR-2412	CALL DSCAN (2412)	
Model GR-2414	CALL DSCAN (2414)	
Selanar Graphics 100	CALL SLNRVT (ICPS)	4
Soltec Servegor Model 281 Digital Plotter	CALL SOLTEC (IDEVC,IOPTIN)	1
Symbolics LGP-1	CALL LGP1	
TAB Computer Products 132/15	CALL TAB	
Tektronix		
Model CX4106	CALL CX41 (4106)	
Model CX4107	CALL CX41 (4107)	
Model CX4109	CALL CX41 (4109)	
Models 410x	CALL TK41 (IMODEL)	18
Model 4115B	CALL TK4115 (IDEVC)	19
Model 4510	CALL TK4510 (IDUNIT,IDEVC, IOPTIN)	20
All other models	CALL TEKALL (IMODEL,ICPS,O, IRES,O)	5, 21
Visual Technology Corporation Model 500	CALL VISUAL (ICPS,0)	4
Nicolet ZETA Models 8, 1453, 3600, 3610, 3620, 3653, and 5400	CALL ZETA (IMODEL,IPWDTH, IDUNIT)	22

Table 9

Parameter Descriptions

1.	IDEVC	=	0	-	No automatic paper advance option
		=	1	-	Paper advance option in use
	IOPTIN	=	0	-	No automatic pen select option
		=	8	-	Pen select option wwth 8 pens
2.	IDUNIT	=	nn	-	Used to direct output to Fortran logical unit number
		=	nn>0	-	signifies off-line mode and corresponds to the logical unit number of the tape drive.
		=	nn<0	-	signifies on-line drect connect mode and directs output to the absolute value of the logical unit number specified.
3.	IMODEL	=	nnnn	-	The model number of the Chromatics graphics terminal
	IDEVC	=	0	-	No Hardware Vector Generator option
		=	1	-	Hardware Vector Generator option
	IOPTIN	=	0	-	No hardware shading
		=	1	-	Hardware shading, history blanking OFF
		=	-1	-	Hardware shading, history blanking ON
4.	ICPS	=	nnn	-	Integer characters/second.ICPS is approximately equal to the baud rate divided by ten
5.	IRES	=	0	-	Low resolution
		=	1	-	High resolution.
6.					Use IRES=0 with 80 column/line device setup Use IRES=1 with 132 column/line device setup
7.	IMODEL	=	4010	-	Tektronix 4010 emulation
		=	4027	-	Tektronix 4027 emulation
8.	ICOLOR	=	0	-	Monochrome display
		=	1	-	Color display
9.	IMODEL	=	nn	-	Model number (3, 4, 6, 7, or 29)
10.	IDUNIT	=	nn	-	Output unit number
	IRES	=	nnn	-	Device resolution. Increments/inch. E.g. 500
	IPWDTH	=	nn	-	Paper Y page size. Maximum 21 inches.
	IOPTIN	=	n	-	Number of pens (1-4)
	IMODES	=	0	-	Immediate operation
			1	-	XON/XOFF operation
			2	-	deferred plotting
11.	IOPTIN	=	1	-	Paper size of 8-1/2" by 11 inches
		=	2	-	Paper ssze of 11 by 17 inches
12.	IOPTIN	=	1	-	Model HP9872A
		=	2	-	Model HP9872B
		=	3	-	Model HP9872C
		=	4	-	Model HP9872S
		=	5	-	Model HP9872T
	IDEVC	=	nnnn	-	Support terminal (2647 or 2648)
	IADRS	=	n	-	Address switch setting (0-7). Normally set to 5
13.	IDEVC	=	nn	-	Number of rows (1-80)
	IPWDTH	=	nnn	-	Number of columns (1-132)
	IOPTIN	=	nn	-	Number of copies (1-99)

Table 9: Parameter Descriptions (Continued)

14.	IDEVC	=	0	-	Graphics only
		=	1	-	Graphics merged with text using IBM Composed Document Composition Facility
	IOPTIN	=	0	-	Monochrome
		=	1	-	Subtractive color masters (Black, cyan, magenta, yellow)
		=	2	-	Additive color masters (Red, green, blue)
	IPWDTH	=	nnn	-	Paper width in tenths of an inch
15.	IDEVC	=	0	-	Color palette 0 (Black, red, green, yellow) Resolution = 320 by 200 dots
		=	1	-	Color palette 1 (Black, cyan, magenta, white) Resolution = 320 by 200
		=	2	-	Monochrome. Resolution = 640 by 200
	IFILL	=	0	-	No box fill
		=	1	-	Box fill
16.	IOPTIN	=	1	-	Low resolution (IBM Color Graphics Adapter)
		=	2	-	Medium resolution
		=	3	-	High resolution
	IDEVC	=	0	-	Monochrome or more than four colors
		=	1	-	Color palette 1 (brown, red, green)
		=	2	-	Color palette 2 (yellow, light red, light green)
		=	3	-	Color palette 3 (light gray, magenta, cyan)
		=	4	-	Color palette 4 (white, light magenta, light cyan)
17.	IDEVC	=	0	-	Output ot Samurai
		=	1	-	Output to file 1
		=	2	-	Output to file 2
		=	3	-	Output to file 3
	IOPTIN	=	0	-	To file n
		=	1	-	To disk files
		=	2	-	To screen
18.	IMODEL	=	nnn	-	Model number (4105, 4106, 4107, 4109, 4112, 4113, 4114, 4115, or 4116)
19.	IDEVC	=	1	-	Resolution of 4,096 by 4,096
		=	2	-	Resolution of 1,024 by 1,024
		=	3	-	Resolution of 65,536 by 65,536
		=	4	-	Resolution of 4,194,304 by 4,194,304
		=	5	-	Resolution of 268,435,456 by 268,435,456
20.	IDUNIT	=	0	-	Primary output
		=	1-3	-	To write an interface scratch file
	IDEVC	=	0	-	Normal hardcopies
		=	1	-	Black/white inversion
	IOPTIN	=	0	-	Paper size 8-1/2 by 11 inches
		=	1	-	Paper ssze 11 by 17 inches
21.	IMODEL	=	nnnn	-	Model number (4006, 4010, 4012, 4013, 4014, 4015 4016, 4025, 4027, 4051, 4052, 4054, 4081, 4662, 4663, or 8888)
	IRES	=	0	-	Low resolution - 1024 addressable points
		=	1	-	High resolution - 4096 addressable points
22.	IMODEL	=	nnnn	-	Model number (8, 1453, 3600, 3610, 3620, 3653, 5400)
	IPWDTH	=	nn	-	Paper width in inches. A negative number defaults the paper width to the plotter maximum. An IPWDTH = 0 sets tte page to 8-1/2 by 11 inches.
	IDUNIT	=	nn	-	Fortran output logical unit number. A positive number signifies offline mode; output is written to tape. A negative number signifies on-line operation.