INTERNATIONAL ATOMIC ENERGY AGENCY

NUCLEAR DATA SERVICES

DOCUMENTATION SERIES OF THE IAEA NUCLEAR DATA SECTION

IAEA-NDS-29
Rev. 7, November 1995

ENDF Utility Codes Release 6.10

C.L. Dunford

Description and Operating Instructions

Abstract: Description and operating instructions are given for a package of utility codes operating on evaluated nuclear data files in the formats ENDF-6 (and ENDF-5). Included are the data checking codes CHECKER, FIZCON, PSYCHE; the code INTER for retrieving thermal cross-sections and some other data; graphical plotting subroutines PLOTEF, GRALIB, INTLIB; and the file maintenance and retrieval codes LISTEF, SETMDC, GETMAT, STANEF. This program package which is designed for CDC, IBM, DEC and PC computers, can be obtained on magnetic tape or floppy diskette, free of charge, from the IAEA Nuclear Data Section.

Nuclear Data Section
International Atomic Energy Agency
P.O. Box 100
A-1400 Vienna
Austria

e-mail, INTERNET: SERVICES@IAEAND.IAEA.OR.AT
e-mail, BITNET: RNDS@IAEA1
fax: (43-1) 20607
cable: INATOM VIENNA
telex: 1-12645 atom a
telephone: (43-1) 2060-21710

online: TELNET or FTP: IAEAND.IAEA.OR.AT
username: IAEANDS for interactive Nuclear Data Information System
username: NDSOPEN for FTP file transfer

Note:

The IAEA-NDS-documents should not be considered as publications or reports. When a nuclear data library is sent out by the IAEA Nuclear Data Section, it will be accompanied by an IAEA-NDS-document which should give the data user all necessary information on contents, format and origin of the data library.

IAEA-NDS-documents are updated whenever there is additional information of relevance to the users of the data library.

For citations care should be taken that credit is given to the author of the data library and/or to the data center which issued the data library. The editor of the IAEA-NDS-document is usually not the author of the data library.

Neither the originator of the data libraries nor the IAEA assume any liability for their correctness or for any damages resulting from their use.

94/11

Citation guideline:

The codes should be cited as follows:

> C.L. Dunford, "ENDF utility codes release 6.10", report IAEA-NDS-29 Rev. 7 (Nov. 1995); codes received on tape (or diskettes) from the IAEA Nuclear Data Section (date).

Table of Contents

Introduction

The ENDF-6 Utility Codes Release 6.10 by C.L. Dunford are distributed by the IAEA Nuclear Data Section and the National Nuclear Data Center, Brookhaven, U.S.A. These codes operate on evaluated nuclear data files in the formats ENDF-6 and ENDF-5. They are up-to-date as of 30 November 1995. Users of these codes are asked to verify that they are using the most up-to-date versions of these codes.

ENDF, the internationally agreed format for coding evaluated nuclear data, is documented as follows:

ENDF-5, see document IAEA-NDS-75 Rev.1
ENDF-6, see document IAEA-NDS-76 Rev. 4 (1992)

The program package has a size of 3.15 Mbytes. It is available on a small magnetic tape or 3 High Density floppy diskettes.

Condition for use of the codes

The codes are distributed free of charge under the following conditions.

- If any result obtained from these codes are used or referenced in a publication, a reprint should be sent to the IAEA Nuclear Data Section

- Any comments on the use of the codes, including difficulties encountered or any suggestions, should be sent to

C.L. Dunford
National Nuclear Data Center
Brookhaven National Laboratory
P.O. Box 5000
Upton, N.Y. 11973-5000
U.S.A.

Available ENDF Computer Codes

The ENDF 1994 Pre-Processing Codes by D.E. Cullen
See document IAEA-NDS-39 Rev. 8

ENDF Utility Codes Release 6.10. See the present document.

PLOTC4: Plots ENDF formatted data with related experimental data.
See document IAEA-NDS-79 Rev. 1

CONV45/CONV56: converting ENDF-4 to ENDF-5 and ENDF-5 to ENDF-6 format.
See document IAEA-NDS-78

INDXENDF: A PC code by R. Paviotti Corcuera et al., which indexes ENDF-6 formatted data files that are on the hard disk.
See document IAEA-NDS-131

Not available from IAEA:

NJOY: A system for processing ENDF formatted data files. For a summary see document IAEA-NDS-119. This code package must be requested from the

Radiation Shielding Information Centre (RSIC)
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN, USA-37831

List of codes in the ENDF 6.10 Utility Tape

SETMDC: Convert source code between batch and interactive modes
(3 versions included: for CDC, IBM, DEC).
Note: This module is not required for PC implementation.

GETMAT: Retrieve materials from an ENDF format file.

STANEF: Creates directory, adds tape label and converts numeric fields.
Converts to binary format.

CHECKR: To check the structure, consistency and legal formats of ENDF data files.

FIZCON: To check ENDF data for physics consistency and to see that recommended procedures are followed.

PSYCHE: Physics checks of ENDF data files. (More complicated checking)

INTER: To calculate selected cross-sections and integrals:

a) the thermal Maxwellian averaged cross section;
b) the thermal cross section (0.0253 eV);
c) the g-factor, i.e. the ratio between a) and b);
d) the resonance integrals;
e) 14 MeV cross sections.

LISTEF: To generate a file summary and to produce interpreted data listings.

PLOTEF: To generate simple data plots of ENDF data according to material number and file number.

GRALIB: Graphic subroutine package used by PLOTEF.

INTLIB: Graphical interface routines for various plotting devices: Tektronix, PostScript and HPGL.

ENDF Utility Codes Release 6.10

C. L. Dunford

November 30, 1995

The present release of the ENDF utility codes, version 6.10, supersedes all previous versions of the codes and this document supersede all previous documentation. The programs are written to process ENDF-6 formatted files including all formats approved up to and including the October 1995 CSEWG meeting except for the Generalized R-Matrix resonance region format and the generalized format for covariances (file 30). This release contains mainly corrections to version 6.9.

The standard distributed version of these programs is written in ANSI-77 Fortran and should operate on any computer with sufficient memory and a Fortran compiler conforming to this standard. The codes are designed for batch mode operation as were previous released versions. Input records consist of free format data fields with each field except the last delimited by a comma. Each program can process more than one input file (ENDF tape) per run with the user supplying the input file specifications appropriate to his computer. Standard processing options can be selected by specifying a completely blank option specification record.

In addition to the standard batch mode versions of the programs, there is an interactive version. This version prompts the user for input data. It uses a few non-standard VAX Fortran features such as the $ in a format statement to suppress a carriage-return linefeed following output of a record to a terminal and non-standard OPEN statement parameters which can easily be modified to conform to a non VAX Fortran compiler. Both versions are contained in the single Fortran source code for each program. A small computer program, SETMDC, is distributed with the program tape which can convert the source code from one mode to the other. The standard batch mode version is indicated by "ANS" code and the interactive version by "VAX" code.

List of Codes on the ENDF 6.10 Utility Tape

SETMDC	—	Convert source code between batch and interactive modes
GETMAT	—	Retrieve materials from an ENDF format file
STANEF	—	Creates directory, adds tape label and converts numeric fields Converts to binary format
CHECKR	—	Format checking program
FIZCON	—	Procedures and simple physics checking program
PSYCHE	—	More complicated physics checking program
INTER	—	Calculates selected cross sections and integrals
LISTEF	—	Generates file summary and annotated data listing
PLOTEF	—	Generates simple data plots
GRALIB	—	Graphic subroutine package used by PLOTEF
INTLIB	—	Graphic interfaces for Tektronix, PostScript and HPGL

SETMDC

SETMDC is a small Fortran program written to ANSI-77 standards which is designed to convert a computer program source for use on different computers. Within the source code are specially formatted comment cards which flag sections of machine dependent code. The code is sandwiched between a comment card `C+++MDC+++` and a card `C---MDC---`. Code for a specific machine is initiated by a comment card `C...machine_1, machine_2, etc.`

Input Requirements:

The user must supply the following information repeated for each input file to be processed.

RECORD 1 — Input file specification
RECORD 2 — Output file specification
RECORD 3 — Machine selection
- ANS - batch mode program, standard distribution
- VAX - interactive mode program

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a blank record.

Sample Input:

This sample input will convert the STANEF source code as distributed to the interactive version.

	Record Type
`STANEF.BAT`	(1)
`STANEF.FOR`	(2)
`VAX`	(3)
(blank record)	end of run

This page intentionally blank

GETMAT

GETMAT is designed to retrieve one or more materials from an ENDF formatted data file. The output file will contain only the selected materials.

Fortran Logical Units Used:

5	—	Input
6	—	Output
20	—	Input data file, ENDF format
21	—	Output data file, ENDF format

Input Requirements:

In batch mode operation, the user must supply the following control information repeated for each input file to be processed.

RECORD 1	—	Input file specification
RECORD 2	—	Output file specification
RECORD 3	—	Tape label number (up to 4 digits). ≤ 0 means tape has no label >0 means the value is used as the tape label number
RECORD 4	—	Up to 66 character text for output file ENDF tape label (use only if tape label number is greater than zero)

The materials to be extracted are specified on one or more records with the following contents. They need not be given in material order number although the output file will be ordered that way. Material specification is terminated by a blank record.

RECORD(S) 5	—	One or two material numbers in integer format One number to give the material number of the evaluation needed. Two numbers to specify a range of material numbers.

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a record containing the word DONE.

In interactive mode operation, the above data is supplied in response to the appropriate query.

Sample Batch-mode Input:

	Record Type
`TAPE.603`	(1)
`NEW.20`	(2)
`20`	(3)
`Sample material extraction from TAPE 603`	(4)
`2800,2865`	(5) range
`2925`	(5) single
`325,330`	(5) range
(blank record)	(5) end
`TAPE.702`	(1)
`NEW.40`	(2)
`0`	(3)
`9440`	(5) single
(blank record)	(5) end
`DONE`	end of run

STANEF

STANEF is designed to perform bookkeeping operations on a data file containing one or more material evaluations in ENDF format. These operations include:

1. Creation or modification of a "tape ID" record,
2. Creation or update of the directory in MT=451,
3. Create or modify special hollerith ID records in MT=451 (ENDF-6 only),
4. Resequencing,
5. Conversion of integer and floating point fields to standard format,
6. Creation of a binary (ENDF alternate format) file.

Fortran Logical Units Used:

5	—	Input
6	—	Output
20	—	Input data file, ENDF format
21	—	Output data file, ENDF format
22	—	Temporary storage

Input Requirements:

In batch mode operation, the user must supply the following control information repeated for each input file to be processed.

RECORD 1	—	Input file specification
RECORD 2	—	Output file specification
RECORD 3	—	Tape label number (up to 4 digits). <0 means tape has no label =0 means copy old label >0 means the value is used as the tape label number
RECORD 4	—	Up to 66 character text for output file ENDF tape label (use only if tape label number is greater than zero)

RECORD 5 — Options selection (3 fields)

FIELD 1 - Mode of output file (integer)
- 0 - Character format output
- 1 - Binary format (fields 2 and 3 ignored)

FIELD 2 - Update directory and special hollerith (character)
- Y - execute this option
- N - do not execute this option

FIELD 3 - Numeric field standardization (character)
- Y - execute this option
- N - do not execute this option

If RECORD 5 is left entirely blank, then the "standard" options are executed. Those are character format output file, update MT=451 and standardize numeric fields.

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a record containing the word DONE.

In interactive mode operation, the above data is supplied in response to the appropriate query.

Sample Batch-mode Input:

	Record Type
`RAW.603`	(1)
`INDEX.603`	(2)
`0`	(3)
`0,Y,N`	(5)
`RAW.610`	(1)
`TAPE.610`	(2)
`610`	(3)
`ENDF/B-VI FISSION PRODUCT TAPE 2`	(4)
(blank record)	(5) default
`DONE`	end of run

VAX/VMS Command Mode Operation:

If the VAX/VMS foreign command, `STANEF :== $STANEF.EXE`, has been defined, then some of the input data may be given on the command line as follows

`STANEF` *input_file/output_file/standard_options*`[Y or N]`

If no input data is given on the command line, the program operates in the normal interactive mode. If no output file is given, the output file specification is assumed to be the same as the input file. Any tape label is copied unchanged. If non-standard options are selected, then the program prompts for the non-standard option input in the usual way.

This page intentionally blank

CHECKR

CHECKR is a program for checking that an evaluated data file conforms to the ENDF format. It can recognize the difference between ENDF-6 and ENDF-5 formats and performs its tests accordingly. Integer control fields are checked to see that ENDF/B procedural limits on those fields are not violated. To the extent possible, fatal format errors are trapped to prevent unwanted termination of the program. Any file which passes through CHECKR without error messages fully conforms to the ENDF format.

Fortran Logical Units Used:

5	—	Input
6	—	Output
20	—	Input data file, ENDF format
21	—	Message file for program checking results

Input Requirements:

In batch mode operation, the user must supply the following control information repeated for each input file to be processed.

RECORD 1 — Input file specification
RECORD 2 — Message file specification
(if blank, messages go to standard output file on unit 6)
RECORD 3 — Options selection (2 fields)
FIELD 1 - Material number where processing starts (integer)
(If zero, then checking begins with the first material)
FIELD 2 - Material number where processing ends (integer)
(If zero, then checking continues to end of the file)

If RECORD 3 is left entirely blank, then the entire input file is processed.

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a record containing the word DONE.

In interactive mode operation, the above data is supplied in response to the appropriate query.

Sample Batch-mode Input:

	Record Type
`TAPE.603`	(1)
(blank record)	(2) output to unit 6
`5200,5642`	(3)
`TAPE.610`	(1)
`CHECKR.610`	(2)
(blank record)	(3) default
`DONE`	end of run

VAX/VMS Command Mode Operation:

If the VAX/VMS foreign command, `CHECKR :== $CHECKR.EXE`, has been defined, then some of the input data may be given on the command line as follows

`CHECKR` *input_file/output_file/standard_options*`[Y or N]`

If no input data is given on the command line, the program operates in the normal interactive mode. If non-standard options are selected, then the program prompts for the non-standard option input in the usual way.

FIZCON

FIZCON is a program for checking that an evaluated data file has valid data and conforms to recommended procedures. It can recognize the difference between ENDF-6 and ENDF-5 formats and performs its tests accordingly. Some of the tests performed include

1. data arrays are in increasing energy order,
2. resonance parameter widths add up to the total,
3. Q-values are reasonable and consistent,
4. no required sections are missing and all cover the proper energy range,
5. secondary distributions are normalized to 1.0,
6. energy conservation in decay spectra.

Optional tests can be performed to check that redundant cross sections such as the inelastic cross section has an energy grid which is the union of all its components and the the cross section values are the sum of the component values at each energy (SUMUP test). Also optionally, algorithms are used to check for possible incorrect entry of data values (Deviant Point test). It is assumed the the file being checked has passed the CHECKR program without any errors being detected.

Fortran Logical Units Used:

5	—	Input
6	—	Output
20	—	Input data file, ENDF format
21	—	Message file for program checking results
22 & 23	—	Temporary paging files for large data arrays
24 & 25	—	Temporary files for the SUMUP tests

Input Requirements:

In batch mode operation, the user must supply the following control information repeated for each input file to be processed.

RECORD 1	—	Input file specification
RECORD 2	—	Message file specification (if blank, messages go to standard output file on unit 6)

RECORD 3 — Options selection (5 fields)

FIELD 1 - Material number where processing starts (integer)
(If zero, then checking begins with the first material)

FIELD 2 - Material number where processing ends (integer)
(If zero, then checking continues to end of the file)

FIELD 3 - Deviant point test control (character)
Y - Do the test
N - Do not do the test

FIELD 4 - SUMUP test control (character)
Y - Do the test
N - Do not do the test

FIELD 5 - Fractional acceptable difference (real)
The floating point number entered here represents the maximum fractional difference tolerated in an equality test such as a SUMUP test. The default value if none is entered is .001 (1/10 of a percent).

If RECORD 3 is left entirely blank, then the"standard" options are executed. Those are to process the entire input file, to omit the SUMUP and Deviant Point tests and to assume a allowed fractional error of .001.

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a record containing the word DONE.

In interactive mode operation, the above data is supplied in response to the appropriate query.

Sample Batch-mode Input:

	Record Type
`TAPE.603`	(1)
(blank record)	(2) output to unit 6
`5200,5642,N,Y,.002`	(3)
`TAPE.610`	(1)
`FIZCON.610`	(2)
(blank record)	(3) default
`DONE`	end of run

VAX/VMS Command Mode Operation:

If the VAX/VMS foreign command, `FIZCON :== $FIZCON.EXE`, has been defined, then some of the input data may be given on the command line as follows

`FIZCON` *input_file/output_file/standard_options*`[Y or N]`

If no input data is given on the command line, the program operates in the normal interactive mode. If non-standard options are selected, then the program prompts for the non-standard option input in the usual way.

This page intentionally blank

PSYCHE

PSYCHE is a program for checking the physics content of an evaluated data file. It can recognize the difference between ENDF-6 and ENDF-5 formats and performs its tests accordingly. The present version checks for energy conservation for emitted neutrons and photons, checks Wick's limit for elastic scattering, analyzes resonance parameter statistics, calculates thermal cross sections and resonance integrals, examines continuity across resonance region boundaries and checks "Q" values against mass tables. It is assumed the the file being checked has passed the CHECKR program without any errors being detected.

Fortran Logical Units Used:

5	—	Input
6	—	Output
20	—	Input data file, ENDF format
21	—	Message file for program checking results
22 - 25	—	Temporary files for energy conservation tests

Input Requirements:

In batch mode operation, the user must supply the following control information repeated for each input file to be processed.

RECORD 1 — Input file specification
RECORD 2 — Message file specification
(if blank, messages go to standard output file on unit 6)
RECORD 3 — Options selection (2 fields)
FIELD 1 - Material number where processing starts (integer)
(If zero, then checking begins with the first material)
FIELD 2 - Material number where processing ends (integer)
(If zero, then checking continues to end of the file)

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a record containing the word DONE.

In interactive mode operation, the above data is supplied in response to the appropriate query.

Sample Batch-mode Input:

	Record Type
`TAPE.603`	(1)
(blank record)	(2) output to unit 6
`5200,5642,`	(3)
`TAPE.610`	(1)
`PSYCHE.610`	(2)
(blank record)	(3) default
`DONE`	end of run

VAX/VMS Command Mode Operation:

If the VAX/VMS foreign command, `PSYCHE :== $PSYCHE.EXE`, has been defined, then some of the input data may be given on the command line as follows

`PSYCHE` *input_file/output_file/standard_options*`[Y or N]`

If no input data is given on the command line, the program operates in the normal interactive mode. If non-standard options are selected, then the program prompts for the non-standard option input in the usual way.

INTER

INTER is a program for calculating thermal cross sections, g-factors, resonance integrals, fission spectrum averaged cross sections and 14.0 Mev (or other energy) cross sections for major reactions in an ENDF-6 or ENDF-5 format data file. To operate properly, the cross sections must be given pointwise in File 3 and have been linearized. Therefore evaluations containing resonance parameters must first be processed by a code such as RECENT to produce a complete pointwise data file and interpolation codes which are not constant or linear-linear must be processed by a code such as LINEAR.

Fortran Logical Units Used:

5 — Input
6 — Output
20 — Input data file, ENDF format
21 — Output file for program results

Input Requirements:

In batch mode operation, the user must supply the following control information repeated for each input file to be processed.

RECORD 1 — Input file specification
RECORD 2 — Message file specification
(if blank, messages go to standard output file on unit 6)
RECORD 3 — Options selection (7 fields)
FIELD 1 - Material number where processing starts (integer)
(If zero, then checking begins with the first material)
FIELD 2 - Material number where processing ends (integer)
(If zero, then checking continues to end of the file)
FIELD 3 - Thermal calculation control (character)
Y - Calculate thermal cross sections and g-factors
N - Do not do the calculations
FIELD 4 - Resonance integral calculation control (character)
Y - Calculate resonance integrals
N - Do not do the calculations
FIELD 5 - Fission spectrum calculation control (character)
Y - Calculate fission spectrum averages
N - Do not do the calculation

FIELD 6 - Single energy (real)
The energy entered here in eV is used to calculate and list the cross sections at any single required energy, for example, at 14.0 Mev.

FIELD 7 - Fractional error (real)
The floating point number entered here is the fractional error which will be tolerated when integrating between two points. The default value is .001.

If RECORD 3 is left entirely blank, then defaults for all values are assumed which are to process all materials, calculate both thermal cross sections and resonance integrals, calculate 14.0 Mev cross sections and allow a fractional error of .001. Neither of the following two records are required and the default parameters described below are assumed.

RECORD 4 — Thermal calculation parameters (3 fields)
(If thermal calculation is not done, omit this record)

FIELD 1 - Thermal energy (real)
The energy entered here in eV is used to calculate and list the Maxwellian averaged cross sections and g-factors, for example, at 0.0253 eV.

FIELD 2 - Lower limit for Maxwellian integral(eV) (real)

FIELD 3 - Upper limit for Maxwellian integral(eV) (real)

If RECORD 4 is left entirely blank, then defaults for all values are assumed which are a thermal energy of 0.0253 eV and integration between .01 mV and 10.0 eV.

RECORD 5 — Resonance integral calculation parameters (2 fields)
(If resonance calculation is not done, omit this record)

FIELD 1 - Lower limit for Resonance integral(eV) (real)

FIELD 2 - Upper limit for Resonance integral(eV) (real)

If RECORD 5 is left entirely blank, then defaults for all values are assumed which are integration between .5 eV and 100.0 keV.

RECORD 6 — Fission spectrum calculation parameters (3 fields)
(If fission spectrum calculation is not done, omit this record)

FIELD 1 - Fission spectrum temperature(real)
The temperature entered here is in eV and is used as the temperature in a Maxwellian distribution, for example, 1.3E+6 eV.

FIELD 2 - Lower limit for fission integral(eV) (real)

FIELD 3 - Upper limit for fission integral(eV) (real)

If RECORD 6 is left entirely blank, then defaults for all values are assumed which are a temperature of 1.02E+6 eV and integration between 1.0 keV and 20.0 Mev.

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a record containing the word DONE.

In interactive mode operation, the above data is supplied in response to the appropriate query.

Sample Batch-mode Input:

	Record Type
`TAPE.639`	(1)
`VTOCU.INT`	(2)
`2300,2931,Y,Y,Y,14.+6,.001`	(3)
(blank record)	(4) default
`1.,100.+3`	(5)
`1.12E+6,1.E+3,20.E+6`	(6)
`TAPE.610`	(1)
`FISSPR.610`	(2)
(blank record)	(3) default
`DONE`	end of run

VAX/VMS Command Mode Operation:

If the VAX/VMS foreign command, `INTER :== $INTER.EXE`, has been defined, then some of the input data may be given on the command line as follows

`INTER` *input_file/output_file/standard_options*`[Y or N]`

If no input data is given on the command line, the program operates in the normal interactive mode. If non-standard options are selected, then the program prompts for the non-standard option input in the usual way.

This page intentionally blank

LISTEF

LISTEF is a program designed to produce summary and annotated listings of a data file in either ENDF-6 or ENDF-5 format.

Fortran Logical Units Used:

5	—	Input
6	—	Output
20	—	Input data file, ENDF format
21	—	Summary file
22	—	Annotated listing file
23	—	Temporary file for paging large data arrays

Input Requirements:

In batch mode operation, the user must supply the following control information repeated for each input file to be processed.

RECORD 1 — Input file specification

RECORD 2 — Summary file specification
(if blank, summary goes to standard output file on unit 6)

RECORD 3 — Annotated listing file specification
(if blank, no annotated listing is generated)

RECORD 4 — Options selection (2 fields)

FIELD 1 - Output paging control for annotated listing (character)
- Y - New page for each section (MT)
- N - No new page for each section (MT)

FIELD 2 - Material processing control (character)
- Y - List the entire file
- N - List only selected portions of the file

If RECORD 4 is left entirely blank, then default values are assumed which are to process all materials and start each section on a new page of the annotated listing. If only selected materials and files are to be processed, they are specified on one or more records with the following contents. They need not be given in material order number although the output file will be ordered that way. Material and file specification are terminated by a blank record.

RECORD(S) 5 — Selection of materials and files to be processed

FIELD 1 - Material number or ZA number

Material number must be 1 to 4 digits with no period.

ZA number must be a 4 or 5 digit number with a trailing period having the value 1000.*Z + A for the desired material.

FIELD(S) 2 thru 20 - File selections for the material (integer)

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a record containing the word DONE.

In interactive mode operation, the above data is supplied in response to the appropriate query.

Sample Batch-mode Input:

	Record Type
`TAPE.654`	(1)
(blank record)	(2) default
(blank record)	(3) no annotated list
(blank record)	(4) default
`TAPE.633`	(1)
`IRON.SUM`	(2)
`IRON.FULL`	(3)
`N,N`	(4)
`26054.,1,3,8,9,10`	(5) ZA no.
`2634,1,2`	(5) MAT no.
(blank record)	(5) end
`DONE`	end of run

VAX/VMS Command Mode Operation:

If the VAX/VMS foreign command, `LISTEF :== $LISTEF.EXE`, has been defined, then some of the input data may be given on the command line as follows

`LISTEF` *input_file/summary_file/output_file/standard_options*`[Y or N]`

If no input data is given on the command line, the program operates in the normal interactive mode. If no summary or output file is given, then the file is not produced. If non-standard options are selected, then the program prompts for the non-standard option input in the usual way.

PLOTEF

PLOTEF is a program designed to produce graphical displays of a data file in either ENDF-6 or ENDF-5 format. The form of graphical output depends on the graphical devices available at the installation where this code will be used.

Fortran Logical Units Used:

5 — Input
6 — Output (other than graphical)
20 — Input data file, ENDF format
21 — Temporary file for plot selection data
22 — Temporary file for paging large data arrays

Input Requirements:

In batch mode operation, the user must first supply a record to specify the format and the name of the graphics output file.

RECORD 1 — Selection of graphic output file format and name
FIELD 1 - Graphic format selection
1 — Tektronix format instructions
2 — PostScript format instructions
3 — HPGL format instructions

See "Special Requirements" below for more detail.

Then the user must supply the following control information repeated for each input file to be processed.

RECORD 2 — Input file specification
RECORD 3 — Material processing control (character)
Y — Plot the entire file
N — Plot only selected portions of the file

If RECORD 3 is left blank, then all materials, all files and all sections will be plotted. If only selected materials, files and sections are to be processed, the selections, one material per record up to a maximum of 30, are given with the following contents. They need not be given in material number order although the graphics output will be ordered that way. Plot specification records are terminated by a blank record.

RECORD(S) 4 — Selection of files and sections for a material

FIELD 1 - Material number or ZA number

Material number must be 1 to 4 digits with no period.

ZA number must be a 4 or 5 digit number with a trailing period having the value 1000.*Z + A for the desired material.

FIELD(S) 2 thru 13 - Plot selections for the material (integer)

These records can be up to 250 characters long but are read by the program 80 characters at a time, so use a dash(-) to indicate that input for the material continues. Plot selections for a given material are specified in the form

```
MF-specification(MT-specification;MT-specification;...)
```

where

MF-specification format

- Blank - all values of MF for the material
- MF-value - a single selected MF for the material
- MF/MF - a range of MF-values

MT-specification format

- Blank - all values of MT for the selected MF's
- MT-value - a single selected MT for the selected MF's
- MT/MT - a range of MT-values for the selected MF's

Examples:

1. 9440,3,12/14
   Files 3, 12, 13 and 14 of material 9440 will be plotted.

2. 9440
   All of material 9440 will be plotted.

3. 9440,(51/90)
   All inelastic reaction data in all files of 9440 will be plotted.

4. 94240.,3(18/21;38)
   All fission cross section of Plutonium-240 will be plotted.

5. 94241.,1(452;455/456),3/5(18/21;38),-
   8(457),12/15
   Complex selection with continuation(-).

Multiple input files can be processed to produce multiple output files by repeating the above input data sequence. The program execution is terminated by a record containing the word DONE.

In interactive mode operation, the above data is supplied in response to the appropriate query.

Special Requirements:

In addition to the PLOTEF source code, a library of graphics subroutines called GRALIB produced by NNDC is required. Documentation for the GRALIB subroutine library is available separately. This library contains routines to generate and manipulate graphic data which are independent of the output graphics device. In order to produce output for a given graphics device, a device driver must be written by the user for his device.

Both the PLOTEF and GRALIB software have small amounts of clearly labeled code which should be modified to meet the requirements of the user's computer facility and output graphics device.

Graphic device drivers interact with the GRALIB subroutine package using the "DISSPLA" conventions. A library of graphic device interface subroutines for Tektronix, PostScript or HPGL driven laser printers is included in the INTLIB subroutine library. Documentation for the GRALIB subroutine library is available separately.

Sample Batch-mode Input:

	Record Type
`1,FE.LN3`	(1)
`TAPE.654`	(2)
(blank record)	(3) default
`TAPE.633`	(2)
`N`	(3)
`26054.,1,3/5(4;51/90),8,-`	(4) ZA no.
`11/15`	(4) continuation
`2634`	(4) MAT no.
(blank record)	(4) end
`DONE`	end of run

VAX/VMS Command Mode Operation:

If the VAX/VMS foreign command, `PLOTEF :== $PLOTEF.EXE`, has been defined, then some of the input data may be given on the command line as follows

`PLOTEF` *input_file/ output_file/ graphic_format/ standard_options*`[Y or N]`

If no input data is given on the command line, the program operates in the normal

interactive mode. The graphic format choices are 1(TEKTRONIX), 2(PostScript), or 3(HPGL). The default format is PostScript. If non-standard options are selected, then the program prompts for the non-standard option input in the usual way.

GRALIB – Graphic Subroutine Library

C. L. Dunford

December 14, 1990

Abstract

An alternative graphic subroutine package compatible with DISSPLA device driver subroutines has been developed. This package contains only a small portion of the DISSPLA software capability. However, it contains sufficient capability for developing graphics programs for general distribution to users who do not have a DISSPLA software license. Additional subroutines can be added as the need arises.

The library is designed to be used in conjunction with the NNDC graphic interface library, INTLIB. The output devices supported are laser printers which support Tektronix or HPGL graphic instructions, or PostScript and graphic terminals which support Tektronix or Regis graphics instructions, or PostScript.

Contents

1 Introduction

The graphics subroutines can be conveniently grouped into four categories listed below.

1. CONTROL
   - HWSCAL
   - HWROT
   - BGNPLT
   - ENDPLT
   - DONEPL
   - ORIGIN
   - SCALE
   - WHERE
   - SETMES

2. LINE DRAWING
   - THLINE
   - DRLINE
   - DRAWTO
   - MOVETO

3. TEXT GENERATION
   - THITE
   - TANGLE
   - LETTER
   - EXTENT
   - CIPHER
   - DIGITS
   - NUMLEN

4. GRAPHS
   - DECIDE
   - AXLOG
   - AXLIN
   - XRULER
   - YRULER
   - STICKX
   - STICKY
   - SCALEX
   - SCALEY
   - DCURVE
   - PLOTSY

All position and dimension parameters are given in inches. To use the metric system (centimeters) call the subroutine `SCALE(2.54)`.

The library is available on the NNDC VAX-cluster as `OBJ:GRALIB.OLB`. The graphic device nomination routines supported in this package are described in the INTLIB documentation. These routines are FR80, NNDCTK, NNDCPS, NNDCRG, and NNDCHP. One of these nomination routines MUST be called before any subroutine from this library. A procedure for linking a program with F77 GRALIB, `GRALIB.COM` and a procedure for linking with F66 GRALIB, `GRALIB66.COM` will be found on `COM:`.

2 CONTROL SUBROUTINES

2.1 HWSCAL

CALL HWSCAL(IOPT) — Subroutine HWSCAL may be used to set the plot scaling option with relation to the physical dimensions of the output plotting device.

IOPT - character variable indicating the scaling option selected.

- DOWN - All plots exceeding one or more dimensions of the output device will be scaled down to fit the output device.
- SCREEN - All plots are scaled such that at least one dimension is equal to the maximum in that dimension allowed on the output device.
- NONE - No scaling is done. Vectors partly or completely outside the available plotting surface are handled according to to output device specifications.

2.2 HWROT

CALL HWROT(IOPT) — Subroutine HWROT may be used to set the plot rotation option with relation to the physical orientation of the output plotting device.

IOPT - character variable indicating the rotation option selected.

- COMIC - The x-dimension of the plot is aligned with the x-axis of the output device.
- MOVIE - The x-dimension of the plot is aligned with the y-axis of the output device.
- AUTO - The longest dimension of the plot is aligned with the longest axis of the output device.

2.3 BGNPLT

CALL BGNPLT(XDIM,YDIM) — Subroutine BGNPLT must be called at the beginning of each new plot. The routine provides for plot parameter initialization and sets the page size for the plot.

XDIM - Real variable giving the x-dimension of the plot page.

YDIM - Real variable giving the y-dimension of the plot page.

2.4 ENDPLT

CALL ENDPLT — Subroutine ENDPLT must be called on completion of a plotted page.

2.5 DONEPL

CALL DONEPL — Subroutine DONEPL must be called when all plotting has been completed and before the program has been terminated.

2.6 ORIGIN

CALL ORIGIN(XORIG,YORIG) — Subroutine ORIGIN may be used to provide a user origin which is different from the output device origin.

XORIG - Real variable giving the user origin x-coordinate relative to the output device origin.

YORIG - Real variable giving the user origin y-coordinate relative to the output device origin.

2.7 SCALE

CALL SCALE(PSCALE) — Subroutine SCALE may be used to scale all coordinate values by a factor.

PSCALE - real variable giving the scale factor.

2.8 WHERE

CALL WHERE(XPOS,YPOS) — Subroutine WHERE may be used to provide a user with the current plotting coordinates relative to the current user plotting origin.

XPOS - Real variable giving the current plotting x-coordinate relative to the user defined origin.

YPOS - Real variable giving the current plotting y-coordinate relative to the user defined origin.

2.9 SETMES

CALL SETMES(IMES) — Subroutine SETMES is used to set the output message unit if it is to be different from the default unit, 6.

IMES - Integer variable giving the output message unit. If a value of zero is given, messages are suppressed.

3 LINE DRAWING

3.1 THLINE

CALL THLINE(ITHICK) — Subroutine THLINE may be used to set the relative thickness of all lines drawn.

ITHICK - An integer variable between 1 and 5. For ITHICK=1, a line of normal thickness is drawn; for ITHICK=n, lines will be n-times as thick as the normal line.

3.2 DRLINE

CALL DRLINE(XFROM,YFROM,XTO,YTO) — Subroutine DRLINE will draw a line from location (XFROM,YFROM) to (XTO,YTO). The thickness of the line can be set with THLINE.

XFROM - A floating point variable containing the x-coordinate of the starting point of the line.

YFROM - A floating point variable containing the y-coordinate of the starting point of the line.

XTO - A floating point variable containing the x-coordinate of the end point of the line.

YTO - A floating point variable containing the y-coordinate of the end point of the line.

3.3 DRAWTO

CALL DRAWTO(XTO,YTO) — Subroutine DRAWTO will draw a line from the current position to the location (XTO,YTO). The thickness of the line can be set with THLINE.

XTO - A floating point variable containing the x-coordinate of the end point of the line drawn from the current position.

YTO - A floating point variable containing the y-coordinate of the end point of the line drawn from the current position.

3.4 MOVETO

CALL MOVETO(XTO,YTO) — Subroutine MOVETO will change the current position to the location (XTO,YTO) without drawing a line.

XTO - A floating point variable containing the x-coordinate of the new current position.

YTO - A floating point variable containing the y-coordinate of the new current position.

4 TEXT GENERATION

4.1 THITE

CALL THITE(HEIGHT) — Subroutine THITE may be used to set the height of characters displayed by the LETTER and DIGITS subroutines.

HEIGHT - A floating point variable giving the character height. The default value is 0.14 inches.

4.2 TANGLE

CALL TANGLE(ANGLE) — Subroutine TANGLE may be used to set the angle relative to horizontal at which text from LETTER or DIGITS will be displayed.

ANGLE - A floating point number giving the text rotation from horizontal in degrees. The default value is 0.0 degrees.

4.3 LETTER

CALL LETTER(TEXT,NTEXT,X,Y) — Subroutine LETTER will display a character string. All six-bit ASCII characters are recognized plus all seven-bit lower case letters. Four characters are interpreted as control characters so that more complex text strings can be displayed. The display position of the string is the lower left-hand corner of the first character.

TEXT - A character variable containing character string to be displayed.

NTEXT - An integer variable containing the number of characters in the text string.

X - A floating point variable containing the x-coordinate of the position where the character string will be displayed.

Y - A floating point variable containing the y-coordinate of the position where the character string will be displayed.

Table 1
Control Characters

$	Interpret following character from alternate set
;	Backspace one character
<	Begin superscript string or terminate subscript
>	Begin subscript string or terminate superscript

Table 2
Alternate Character Set

Input	Output	Input	Output
$!	↑	$0	α
$"	←	$1	β
$(	[	$2	γ
$)	]	$3	η
$*	×	$4	$\bar{\nu}$
$+	±	$5	μ
$,	'	$6	π
$.	°	$7	σ
$/	%	$8	Γ
$;	;	$9	θ
$:	:	$-	ϵ
$<	<	$?	Δ
$>	>	$@	Σ
$=	~		

4.4 EXTENT

CALL EXTENT(TEXT,NTEXT,TLEN) — Subroutine EXTENT will calculate the displayed length of a text string.

TEXT - A character variable containing character string to be displayed.

NTEXT - An integer variable containing the number of characters in the text string.

TLEN - A floating point variable in which the length of displayed text is returned.

4.5 CIPHER

CALL CIPHER(VALUE,DELTA,X,Y) — Subroutine CIPHER is used to display a number as text. The number will be displayed in either floating point or exponential form depending on the size of the number. The number of significant figures to be displayed is determined by DELTA.

VALUE - A floating point variable containing the number to be displayed.

DELTA - The numeric value of the last significant digit to be displayed.

X - A floating point variable containing the x-coordinate of the position where the number text will be displayed.

Y - A floating point variable containing the y-coordinate of the position where the number text will be displayed.

4.6 DIGITS

CALL DIGITS(FPNUM,NDEC,X,Y) — The subroutine digits will display a floating point number as a character string in floating or integer form.

FPNUM - A floating point variable containing the number to be displayed.

NDEC - An integer giving the number of digits displayed to the right of the decimal point. It can take on values between -9 and +9. Use NDEC=0 to plot a number as an integer without a trailing decimal point.

X - A floating point variable containing the x-coordinate of the position where the number text will be displayed.

Y - A floating point variable containing the y-coordinate of the position where the number text will be displayed.

4.7 NUMLEN

CALL NUMLEN(FPNUM,NDEC,TLEN) — The subroutine will calculate the length of the text string which would be generated by the subroutine DIGITS.

FPNUM - A floating point variable containing the number to be displayed.

NDEC - An integer giving the number of digits displayed to the right of the decimal point. It can take on values between -9 and +9. Use NDEC=0 to plot a number as an integer without a trailing decimal point.

TLEN - A floating point variable in which the length of displayed text is returned.

5 GRAPHS

5.1 DECIDE

CALL DECIDE(ZMIN,ZMAX,NDIV,LOGLIN,AMIN,AMAX,ASTEP,
NTICK) — Subroutine DECIDE is used to determine the parameters for an axis of a graph for use in one of the RULER routines. These parameters are determined from the minimum and maximum values of the data plotted and the maximum number of annotated divisions desired. See Table 3 below for meaning of arguments.

Table 3
Arguments for DECIDE, AXLOG and AXLIN

Input Parameters

ZMIN	Minimum value of the data to be plotted.
ZMAX	Maximum value of the data to be plotted.
NDIV	Maximum number of annotated scale divisions permitted.

Output Parameters

LOGLIN	Chosen axis type: 0-linear, 1-logarithmic.
AMIN	Minimum value for axis scale in units of the data.
AMAX	Maximum value for axis scale in units of the data.
ASTEP	Linear scale - distance between annotated scale divisions in units of the data. Logarithmic scale - number of annotated scale divisions per decade.
NTICK	Linear scale - number of divisions between annotated scale divisions. Logarithmic scale - number of tenth decade divisions to be marked.

5.2 AXLOG

CALL AXLOG(ZMIN,ZMAX,NDIV,AMIN,AMAX,ASTEP,NTICK) — Subroutine AXLOG is used to determine the parameters for a logarithmic axis of a graph for use in one of the RULER routines. These parameters are determined from the minimum and maximum values of the data plotted and the maximum number of annotated divisions desired. If one data limit is negative or zero, then parameters for a linear axis are returned and NTICK is set to a negative value. See Table 3 above for meaning of arguments.

5.3 AXLIN

CALL AXLIN(ZMIN,ZMAX,NDIV,AMIN,AMAX,ASTEP,NTICK) — Subroutine AXLIN is used to determine the parameters for a linear axis of a graph for use in one of the RULER routines. These parameters are determined from the minimum and maximum values of the data plotted and the maximum number of annotated divisions desired. See Table 3 below for meaning of arguments.

5.4 XRULER

CALL XRULER(XOR,YOR,AXLEN,LOGLIN,AMIN,AMAX,STEP,
NTICK,LTICK,LABEL,NC) — Subroutine XRULER is used to generate an x-axis for a data plot. The subroutine also sets the scaling parameters used by the data plotting routine. For meaning of the subroutine arguments, see the Table 3 below.

5.5 YRULER

CALL YRULER(XOR,YOR,AXLEN,LOGLIN,AMIN,AMAX,STEP,
NTICK,LTICK,LABEL,NC) — Subroutine YRULER is used to generate an x-axis for a data plot. The subroutine also sets the scaling parameters used by the data plotting routine. For meaning of the subroutine arguments, see the Table 4 below.

5.6 STICKX

CALL STICKX(ISX) — The subroutine is used to specify the x-axis tick mark option. All tick marks are the same length, only the location relative to the axis can be changed. The subroutine must be called before XRULER to have any effect.

ISX - An integer variable containing the option value selected.

- = 1 - ticks are located on the top side of the axis.
- 0 - ticks are cross of the axis. (Default)
- -1 - ticks are located on the bottom side of the axis.

Table 4

Arguments for XRULER and YRULER

XOR	x-coordinate for the origin of the axis in inches.
YOR	y-coordinate for the origin of the axis in inches.
AXLEN	Length of the axis in inches.
LOGLIN	AXIS type: 0-linear, 1-logarithmic.
AMIN	Minimum value for axis scale in units of the data.
AMAX	Maximum value for axis scale in units of the data.
ASTEP	Linear scale - distance between annotated scale divisions in units of the data. Logarithmic scale - number of annotated scale divisions per decade.
NTICK	Linear scale - number of divisions between annotated scale divisions. Logarithmic scale - number of tenth decade divisions to be marked.
LTICK	Grid division annotation control. = 1 - annotation displayed in normal position. 0 - suppress annotation. −1 - annotation displayed on opposite side of the axis.
LABEL	Axis label text string.
NC	Number of characters in the axis label text.

5.7 STICKY

CALL STICKY(ISY) — The subroutine is used to specify the y-axis tick mark option. All tick marks are the same length, only the location relative to the axis can be changed. The subroutine must be called before YRULER to have any effect.

ISY - An integer variable containing the option value selected.

- = 1 - ticks are located on the right side of the axis.
- 0 - ticks are cross of the axis. (Default)
- −1 - ticks are located on the left side of the axis.

5.8 SCALEX

X = FUNCTION SCALEX(XVALUE) — The function SCALEX will return the x-distance from the graph's origin of the data x-coordinate XVALUE.

5.9 SCALEY

X = FUNCTION SCALEY(YVALUE) — The function SCALEY will return the y-distance from the graph's origin of the data y-coordinate YVALUE.

5.10 DCURVE

CALL DCURVE(XDATA,YDATA,NP) — The subroutine will draw a curve on a graph between the points given.

XDATA - A floating point array containing the x-coordinates of the data to be displayed as a curve.

YDATA - A floating point array containing the y-coordinates of the data to be displayed as a curve.

NP - A integer variable giving the number of points supplied.

5.11 PLOTSY

CALL PLOTSY(X,Y,NS,SIZE) — The subroutine is used to plot a point on a graph as a symbol.

X - A floating point number containing the x-coordinate of the point to be plotted.

Y - A floating point number containing the y-coordinate of the point to be plotted.

NS - An integer variable containing the plotting symbol selected.

= 1 - symbol is an X.

2 - symbol is a box.

3 - symbol is a diamond.

INTLIB – Graphic Device Interface Subroutine Library

C. L. Dunford

January 24, 1991

Abstract

Graphic subroutine libraries in use at NNDC, DISSPLA and GRALIB, generally produce output which is independent of the output graphic device. A set of device dependent interface routines is required to translate the device independent output to the form required for each graphic device available to NNDC. The interface library described herein provides interface routines for the following output formats.

TEKTRONIX	-	LN03 PLUS
	-	video display terminal
POSTSCRIPT	-	LN03 PLUS with PostScript
	-	LaserJet III in PostScript mode
	-	video display terminal
REGIS	-	VT240 and VT1200
HPGL	-	LaserJet III in HPGL mode
FR80	-	COMP80 film, fiche and hard copy

Contents

1 Introduction

The library is available on the NNDC VAX computer as OBJ:INTLIB.OLB. The DCL linking command should include the following strings.

```
DISSPLA  -  OBJ:INTLIB/LIB,DISLIB/LIB,INTLIB_CA11/LIB,
               DISLIB/LIB,NNDCLIB/LIB
GRALIB   -  OBJ:INTLIB/LIB,GRALIB/LIB,NNDCLIB/LIB
```

Note that the NNDCLIB library should be searched last!

The interface routines for the Versatec, Vector Automation and the NNDC universal format which were supported at one time, are no longer included in current version of this library. The subroutine names for the Tektronix, Postscript and Regis interface calls have been changed but the old names still exist as entry points. Also the names of some auxiliary subroutines have been changed to meet ANSI Fortran 77 conventions with the old names remaining as entry points. The COMP80 interface and associated routines are not included in the external distribution of this library.

There is machine dependent code in the interface subroutine package source code for all but the COMP80 interface supporting both a VAX version and and ANSI standard F77 version for export to other computers. The ANSI standard version does not support asynchronous interrupt handling described is Section 4.

2 Basic Device Interfaces

The following device initiation routines are available for use in NNDC graphics programs. Output to only one graphics device at a time is supported by DISSPLA and the GRALIB packages. Therefore only one of these routines should be called at the beginning of a program before any graphics subroutines are executed. If a plotting termination routine (ie. DONEPL) is called, then a new device initiation routine may be called within a program.

2.1 TEKTRONIX Instruction Devices

The program signals to the graphics software that the output will go to an output device which can interpret Tektronix graphic instructions by calling the following Fortran subroutine.

```
CALL NNDCTK(IMODEL,IOPTN)
```

where	IMODEL=0, IOPTN=0	LN03 PLUS laser printer
	IMODEL=1, IOPTN=0	video terminal where interface controls interrupts
	IOPTN=1	video terminal where user program controls interrupts

The LN03 PLUS laser printer/plotter operates in a TEKTRONIX emulation mode. The graphic instruction output from a user's program is stored in a disk file on logical unit 46. The default specification for this disk file is GRAF:GRDATA.LN03. If the program is to have control over the output file specification, then use subroutine SETNAM described in Section 3.1 below. The output file should be submitted to one of the LN03 output queues for plotting as shown below.

```
$PRINT/NOTIFY/NOFEED/QUEUE=LAZQ GRAF:GRDATA.LN03
                or
$PLASER GRAF:GRDATA.LN03
```

If IMODEL = 1, then output is assumed to go to a TEKTRONIX mode video display terminal. In this case, the output pauses after each frame until the user enters any character on his terminal. If IOPTN=0, then the device interface controls the action taken when a control C or a control Y is entered during the graphic display generation. The control C will terminate the current plot softly and the control Y terminates the program. If IOPTN=1, then the user controls the action taken for these two asynchronous interrupts. See Section 4.

The device hardware characteristics are:

IMODEL	-	0	1
Line width	-	.004 inches	.004 inches
Resolution	-	.004 inches	.004 inches
Max. X-dimension	-	10.24 inches	6.4 inches
Max. Y-dimension	-	7.68 inches	4.8 inches
Scaling	-	Down	Down
Rotation	-	Auto	Comic

2.2 PostScript Instruction Devices

A user program signals to the graphics software that the output will go to an output device which can interpret PostScript graphic instructions by calling the following Fortran subroutine.

```
CALL NNDCPS(IMODEL,IOPTN)
```

where	IMODEL=0, IOPTN=0	POSTSCRIPT laser printer
	IMODEL=1, IOPTN=0	video terminal where interface controls interrupts
	IOPTN=1	video terminal where user program controls interrupts

One of the NNDC LN03 PLUS laser printers is a PostScript printer. The graphic instruction output from a user's program is stored in a disk file on Fortran logical unit 49. The default specification for this disk file is GRAF:GRDATA.PSCR. If the program is to have control over the output file specification, then use subroutine SETNAM described in Section 3.1 below. The output file should be submitted to the LN03 PostScript output queue for plotting as shown below.

```
$PRINT/NOTIFY/QUEUE=POST GRAF:GRDATA.PSCR
```

If IMODEL = 1, then output is assumed to go to a PostScript video display terminal. In this case, the output pauses after each frame until the user enters any character on his terminal. If IOPTN=0, then the device interface controls the action taken when a control C or a control Y is entered during the graphic display generation. The control C will terminate the current plot softly and the control Y terminates the program. If IOPTN=1, then the user controls the action taken for these two asynchronous interrupts. See Section 4.

The device hardware characteristics are:

IMODEL	-	0	1
Line width	-	.00463 inches	.00463 inches
Resolution	-	.004 inches	.004 inches
Max. X-dimension	-	7.68 inches	7.99 inches
Max. Y-dimension	-	10.24 inches	4.8 inches
Scaling	-	Down	Down
Rotation	-	Auto	Comic

2.3 Regis Instruction Devices

A user program signals to the graphics software that the output will go to a video output device which can interpret Regis graphic instructions by calling the following Fortran subroutine.

```
CALL NNDCRG(IMODEL,IOPTN)
```

where	IMODEL=0, IOPTN=0	video terminal where interface controls interrupts
	IOPTN=1	video terminal where user program controls interrupts

The DEC graphic terminals (VT240 and VT1200) take graphic instructions in REGIS mode. The graphic instruction output from a user's program is sent directly to the user's terminal which has been OPENed on logical unit 47. The output pauses after each frame until the user enters any character on his terminal. If IOPTN=0, then the device interface controls the action taken when a control C or a control Y is entered during the graphic display generation. The control C will terminate the current plot softly and the control Y terminates the program. If IOPTN=1, then the user controls the action taken for these two asynchronous interrupts. See Section 4.

The device hardware characteristics are:

Line width	-	.01 inches
Resolution	-	.01 inches
Max. X-dimension	-	7.99 inches
Max. Y-dimension	-	4.79 inches
Scaling	-	Down
Rotation	-	Comic

2.4 HPGL Instruction Devices

A user program signals to the graphics software that the output will go to an output device which can interpret HPGL graphic instructions by calling the following Fortran subroutine.

```
CALL NNDCHP(IMODEL,IOPTN)
```

where IMODEL=0, IOPTN=0 HP LASERJET III laser printer

The Hewlett-Packard LaserJet III printers can process instructions in the HPGL graphic language. The graphic instruction output from a user's program is stored in a disk file on Fortran logical unit 48. The default specification for this disk file is GRAF:GRDATA.HPGL. If the program is to have control over the output file specification, then use subroutine SETNAM described in Section 3.1 below. The output file should be submitted to the LaserJet output queue for plotting as shown below.

```
$PRINT/NOTIFY/QUEUE=1HP3D GRAF:GRDATA.HPGL
                or
$PLJET GRAF:GRDATA.LN03
```

The device hardware characteristics are:

Line width	-	.00167 inches
Resolution	-	.00167 inches
Max. X-dimension	-	7.68 inches
Max. Y-dimension	-	10.24 inches
Scaling	-	Down
Rotation	-	Auto

2.5 Comp80 High Resolution Cathode-ray Tube Plotter

A user program signals to the graphics software that the output will go to FR80 cathode-ray tube plotter by calling the following Fortran subroutine.

```
CALL FR80(IMODEL,IOPTN)
```

where	IMODEL=1, IOPTN=0	unsprocketed 35mm film
	IMODEL=3, IOPTN=0	sprocketed 35mm film
	IMODEL=9, IOPTN=1	8.5 x 11 inch fiche with 42 magnification
	IOPTN=2	8.5 x 14 inch fiche with 42 magnification
	IOPTN=3	8.5 x 11 inch fiche with 48 magnification
	IOPTN=4	8.5 x 14 inch fiche with 48 magnification
	IMODEL=16, IOPTN=0	full size positive film at BNL GRAPHIC ARTS
	IMODEL=21, IOPTN=0	full size positive film at INFO CONVERSION

The FR80 plotter at BNL contains a PDP-15 with a 9-track, 1600 bpi tape drive. The graphic instruction output from a user's program is stored in a disk file on Fortran logical unit 45. The default specification for this disk file is `GRAF:GRDATA.F80`. If the program is to have control over the output file specification, then use subroutine SETNAM described in Section 3.1 below.

The output file should be copied to magnetic tape with the DCL commands shown below. This tape can then be taken either to GRAPHIC ARTS or mailed to INFO CONVERSION.

```
$ALLOCATE tape_unit
$MOUNT/NOLABEL/DEN=1600/BLOCK=1200/REC=300 tape_unit
$COPY GRAF:GRDATA.F80 tape_unit
```

The hardware device characteristics are stored in the file `COMDAT:GRDEV.INI` because of the need to modify them when a new CRT tube is installed at GRAPHIC ARTS. Its format and contents are described in Table 1. Any program using these graphic interface routines will search first a GRDEV.INI file in its default directory and then one in COMDAT: for a record with the appropriate device identifier, model and option. Therefore the NNDC defaults for any device described above can be overridden at execution time with a GRDEV.INI file in the program's default directory. In addition, the routine SETPAR described in Section 5.1 can be used on parameter by parameter basis within the program itself. The file format is a free format with fields separated by commas.

Table 1

Format of GRDEV.INI

Field	Parameter No.	Meaning
1		Device code
2		Model number
3		Option
4		Version number
5		Number of parameters to follow
6	1	Thickness of a line in inches
7	2	Diameter of a dot in inches
8	3	X-resolution in inches
9	4	Y-resolution in inches
10	5	X page size in inches
11	6	Y page size in inches
12	7	X page size in raster units
13	8	Y page size in raster units
14	9	Scaling control (1.0 - 3.0) 1.0 - Down 2.0 - Screen 3.0 - None
15	10	Rotation control (1.0 - 3.0) 1.0 - Comic 2.0 - Movie 3.0 - Auto
16	11	Default beam intensity 0.0 to 7.0
17	12	Default number of strokes 0.0 to 7.0
18	13	Default spot size 0.0 to 7.0
19	14	Number of film units/inch of movement

The default device hardware parameters are as follows.

1. CALL FR80(1,0) Unsprocketed 35mm film

Line width	-	.005	inches
Resolution	-	.0007198	inches
Max. X-dimension	-	14.0	inches
Max. Y-dimension	-	11.05	inches
Scaling	-	Screen	
Rotation	-	Auto	
Beam intensity	-	7	
Beam strokes	-	1	
Beam spot size	-	0	

2. CALL FR80(3,0) Sprocketed 35mm film

Line width	-	.005	inches
Resolution	-	.0007198	inches
Max. X-dimension	-	11.0	inches
Max. Y-dimension	-	14.42	inches
Scaling	-	Screen	
Rotation	-	Auto	
Beam intensity	-	7	
Beam strokes	-	1	
Beam spot size	-	0	

3. CALL FR80(9,IOPTN) Microfiche

Line width	-	.005	inches
Resolution	-	.0007198	inches
Scaling	-	Screen	
Rotation	-	Comic	
Beam intensity	-	5	
Beam strokes	-	2	
Beam spot size	-	0	

IOPTN		1	2	3	4	
Max. X-dimension	-	8.64	14.0	8.8	14.0	inches
Max. Y-dimension	-	11.00	11.0	11.0	11.3	inches

4. `CALL FR80(16,0)` Full size positive file (BNL GRAPHIC ARTS)

Line width	–	.005	inches
Resolution	–	.0007198	inches
Max. X-dimension	–	8.5	inches
Max. Y-dimension	–	11.75	inches
Scaling	–	Down	
Rotation	–	Auto	
Beam intensity	–	7	
Beam strokes	–	9	
Beam spot size	–	0	
Film movement	–	360.15 units/inch	

5. `CALL FR80(21,0)` Full size positive file (INFO CONVERSION)

Line width	–	.005	inches
Resolution	–	.0007198	inches
Max. X-dimension	–	8.5	inches
Max. Y-dimension	–	11.55	inches
Scaling	–	Down	
Rotation	–	Auto	
Beam intensity	–	5	
Beam strokes	–	1	
Beam spot size	–	0	
Film movement	–	500.00 units/inch	

2.6 F80DMP - Utility Program

This program is designed to produce the content of an FR80 format graphic instruction file in a user intelligible format. The program run is initiated by the DCL command `RUN EXE:F80DMP`. The user must supply the input FR80 graphic instruction file specification and the output interpreted file specification when prompted. If the output file specification is `TT:`, then the output will appear on the user's terminal. The output consists of one record per graphic instruction containing

1. file block number
2. word number within the block
3. frame(or plot) number
4. instruction in octal
5. instruction in English

Table 2

Sample Run of F80DMP

```
$RUN EXE:F80DMP


        PROGRAM TO DUMP AN FR80 GRAPHICS FILE

    ENTER INPUT FILE SPECS - GRAF:TEST.F80

    ENTER OUTPUT FILE SPECS - TT:


     DUMP OF FR80 GRAPHICS FILE ON 29-DEC-87
     FILE = SA3:[GRAF.DUNFORD]TEST.F80;1


   1     1    1   20000  START JOB
   1     4    1  221020  VERIFY CAMERA AND SELECT ROTATION
   1     7    1  205007  SET INTENSITY
   1    10    1  214000  SELECT COLOR MODE
   1    13    1  600014  NUMBER OF STROKES
   1    16    1  206000  SET SPOT SIZE
   1    19    1  223063  SET X AND Y OFFSETS
   1    22    1  104333  X OFFSET   2267
   1    25    1  141014  Y OFFSET    524
   1    28    1  214000  SELECT COLOR MODE
   1    31    1  600003  NUMBER OF STROKES
   1    34    1  100000  BEAM OFF   ABSOLUTE X COORDINATE =     0
   1    37    1   40000  BEAM OFF   ABSOLUTE Y COORDINATE =     0
   1    40    1  527112  BEAM ON    ABSOLUTE X COORDINATE = 11850
   1    43    1  575747  BEAM ON    ABSOLUTE Y COORDINATE = 15335
   1    46    1  100000  BEAM OFF   ABSOLUTE X COORDINATE =     0
   1    49    1  540000  BEAM ON    ABSOLUTE Y COORDINATE =     0
```

3 General Utility Routines

3.1 SETNAM

CALL SETNAM(FNAME) — Subroutine SETNAM can be used to override the default graphic output file specification described above for each output device. This routine must be called before a device nomination routine to have any effect. If the user supplied file specification is incomplete (ie. no file type is included) then that component of the default file specification is substituted.

FNAME - a character variable containing output graphic file specification.

3.2 GETRSP

CALL GETRSP(RCHAR) — Subroutine GETRSP will return the character entered by the user at the last time the user requested a new frame to be displayed in an interactive graphic display mode.

RCHAR - a character variable containing the returned character.

3.3 THPEN

CALL THPEN(TPEN) — Subroutine THPEN(TPEN) is used to set the relative thickness of all vectors drawn by the plotting routine. This function is valid on all graphic output devices.

TPEN - floating point number to set the thickness of any line drawn to be TPEN times the basic thickness. The default value is 1.0. Any integral value greater than 0.0 may be given.

4 Asynchronous Interrupts

Special handling of asynchronous interrupts (control C and control Y) is available for use with programs running on VAX computers and displaying plots interactively on graphic terminals. The interactive interfaces allow two options. If IOPTN=0, the interface subroutines perform all of the control for the user. When a control C is entered, the plot is discontinued and control returns to the user's program. However, when a control Y is entered, a soft termination of the program execution occurs.

If IOPTN=1 then the user controls the interrupt handling. In this case the user must disable VMS asynchronous interrupt handling with the routine LIB$DISABLE_CTRL before beginning plotting. Use the routine LIB$ENABLE_CTRL to resume the VMS trapping after plotting has been completed. By default control C's are ignored and control Y's terminate the current plot. The following simple subroutine controls this action.

```
      SUBROUTINE CY_TROL(I)
C
C     ROUTINE TO BE CALLED ASYNCHRONOUSLY WHEN A ^C OR ^Y
C        IS TYPED.
C
      COMMON/NDCGRC/ITPEN,CYTYPD
      INTEGER CYTYPD
C
C     A ^Y IS FLAGGED, A ^C ONLY IF THERE HAS BEEN NO ^Y
C
      IF(I.EQ.-1) THEN
         CYTYPD = -1                    ! control Y
      ELSE
         IF(CYTYPD.EQ.0) CYTYPD = 1     ! control C
      ENDIF
C
C     RESET THE TRAPS
C
      CALL CYTRAP(1)
C
    RETURN
    END
```

The user may supply a modified version of this subroutine and change the settings of the control variable CYTYPD which the graphics interface routines use to control whether or not graphic instructions are sent to the user's video terminal.

5 FR80 Utility Routines

5.1 SETPAR

CALL SETPAR(IPAR,PARVAL,IERR) — Subroutine SETPAR can be used to override one or more device initialization parameters. This routine must be called after device initialization and before plot initialization. The routine must be called once for each parameter to be revised. The meaning of each parameter is described in Table 1.

IPAR - integer containing the parameter number of the parameter to be changed.

PARVAL - floating point number containing the new value of the modified parameter.

IERR - error return indicator.

- 0 - no error in executing the subroutine.
- 1 - error encountered during execution of the subroutine.

5.2 OPRMES

CALL OPRMES(IFA,NC) — Subroutine OPRMES can be used to display a message on the FR80 operator's console.

IFA - character string containing the message to be displayed.

NC - integer containing the number of characters in the string.

5.3 STROKR

CALL STROKR(ISTOK,IFUNT) — Subroutine STROKR can be used to modify or return the current number of strokes drawn for each vector.

ISTOK - integer containing the stroking number.

IFUNT - integer indicating function to be performed by the routine.

- -3 - return offset stroking number.
- -2 - return default stroking number.
- -1 - return current stroking number.
- 0 - set current stroking number to default value.
- 1 - set current stroking number to ISTOK.
- 2 - set default stroking number to ISTOK(initially 1).
- 3 - set offset stroking number to ISTOK.

5.4 SPOTSZ

CALL SPOTSZ(ISPOT,IFUNT) — Subroutine SPOTSZ can be used to modify or return the current FR80 beam spot size setting.

ISPOT - integer containing the beam spot size value.

IFUNT - integer indicating function to be performed by the routine.

- -3 - return offset spot size value.
- -2 - return default spot size value.
- -1 - return current spot size value.
- 0 - set current spot size to default value.
- 1 - set current spot size value to ISPOT.
- 2 - set default spot size value to ISPOT(initially 1).
- 3 - set offset spot size value to ISPOT.

5.5 INTEN8

CALL INTEN8(ITEN,IFUNT) — Subroutine INTEN8 can be used to modify or return the current FR80 beam intensity setting.

ITEN - integer containing the beam intensity value.

IFUNT - integer indicating function to be performed by the routine.

- -3 - return offset intensity value.
- -2 - return default intensity value.
- -1 - return current intensity value.
- 0 - set current intensity to default value.
- 1 - set current intensity value to ITEN.
- 2 - set default intensity value to ITEN (initially 3).
- 3 - set offset intensity value to ITEN.

5.6 FTITLE

CALL FTITLE(IOPT,IARR,NCCC,NL,NNDCN,NCDCN) — Subroutine FTITLE can be used to create a fiche header.

IOPT - integer indicating fiche header option.

- 0 - use default title.
- 1 - user defines title.
- 2 - user defines entire header.

IARR - character string containing user supplied title/header.

NCCC - integer containing the number of characters in the string.

NL - integer containing the number of fiche rows occupied by the header. (Default is 2).

NNDCN - string containing user supplied identification number. (Ignored if IOPT=2).

NCDCN - containing the number of characters in the identification number string. (Ignored if IOPT=2).

5.7 FLMADV

CALL FLMADV(XIN) — Subroutine FLMADV can be used to advance the film in the FR80 camera.

XIN - floating point number giving the film movement distance in inches.

5.8 SETFRA

CALL SETFRA(XIN,IFUNT) — Subroutine SETFRA is be used to control film advance between frames.

XIN - floating point number giving the film advance.

IFUNT - integer indicating function to be performed by the routine.

- -100 - disable frame advance
- -3 - return offset frame advance in inches.
- -2 - return default frame advance in inches.
- -1 - return current frame advance in inches.
- 0 - set current frame advance to default value.
- 1 - set current frame advance to XIN inches.
- 2 - set default frame advance to XIN inches.
- 3 - set offset frame advance to XIN inches.
- 100 - enable frame advance.
- 1000 - set frame advance based on camera selected.

5.9 QQXFR

CALL QQXFR(ISPECS,IFLU,NPLOTN,IERR) — Subroutine QQXFR can be used to extract a frame from an FR80 formatted plotting file and include it in the current plotting file.

ISPECS - character string containing input file specification.

IFLU - integer containing the Fortran logical unit number for the input file.

NPLOTN - integer containing the frame number in the input file to be copied to the current plot file.

IERR - integer indicating error status after executing the subroutine.

- 0 - no error encountered.
- 1 - error encountered

6 Alternate Device Interfaces

Two ISSCO supplied interfaces are included in the NNDC version of the graphic device interface library. They are included primarily to give access to the DISSPLA hardware character feature for the LN03 PLUS laser printer/plotter and for PostScript.

6.1 Alternate LN03 Interface

The ISSCO supplied interface library for DISSPLA contains an interface for the LN03 PLUS printer/plotter operating in TEKTRONIX mode. The source code was modified in the following ways:

a. two arguments were added to the call to LN03TK,

b. some parameters are initialized from the GRDEV.INI disk file,

c. ISSCO spooler system calls added to control output vector file name, unit number and mode of access.

The modified ISSCO interface can be activated by the Fortran subroutine call statement

```
CALL LN03TK(0,0)
```

The following is an intercomparison of the two LN03 PLUS interface routines NNDCTK and LN03TK.

NNDC interface NNDCTK

- Works with all versions of DISSPLA on the VAX.
- Works with NNDC graphics package GRALIB.
- Supports boldfacing.
- Does not support hardware character generation.
- Will create a new output file version if user requested output file already exists.

- ISSCO interface LN03TK
  - Does not work with NNDC graphics package GRALIB.
  - Does not support boldfacing.
  - Supports hardware character generation.
  - Will append to the output file if user requested output file already exists.

6.2 Alternate POSTSCRIPT Interface

The ISSCO supplied interface library for DISSPLA contains an interface for POSTSCRIPT. The source code was modified in the following ways:

a. two arguments were added to the call to PSCRPT,

b. some parameters are initialized from the GRDEV.INI disk file,

c. ISSCO spooler system calls added to control output vector file name, unit number and mode of access.

The modified ISSCO interface can be activated by the Fortran subroutine call statement

```
CALL PSCRPT(0,0)
```

The following is an intercomparison of the two PostScript interface routines, NNDCPS and PSCRPT.

NNDC interface NNDCPS

- Works with all versions of DISSPLA on the VAX.
- Works with NNDC graphics package GRALIB.
- Supports boldfacing.
- Does not support hardware character generation.
- Will create a new output file version if user requested output file already exists.

- ISSCO interface PSCRPT

  - Does not work with NNDC graphics package GRALIB.
  - Does not support boldfacing.
  - Supports hardware character generation.
  - Will append to the output file if user requested output file already exists.