

TRAC Support Software

*R. P. Jenks
V. Martinez*

*Manuscript submitted: April 1988
Date published: June 1988*

*Prepared for
Division of Reactor Accident Analysis
Office of Nuclear Regulatory Research
US Nuclear Regulatory Commission
Washington, DC 20555*

NRC FIN No. A7016

8808120057 880630
PDR NUREG
CR-5071 R PDR

Prepared by Wendy Rowley, Group N-9

An Affirmative Action/Equal Opportunity Employer

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, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

CONTENTS

LIST OF FIGURES	vi
LIST OF TABLES	vi
ACKNOWLEDGMENTS	vii
ABSTRACT	ix
I. EXCON	I-1
A. Introduction	I-1
B. Basic Concepts And Capabilities	I-1
C. User Information	I-3
D. Programming Details	I-6
REFERENCES	I-6
II. TRAP	II-1
A. Introduction	II-2
B. Basic Concepts and Capabilities	II-3
C. Using TRAP	II-3
D. Syllabus of TRAP Commands	II-6
E. Slice Options	II-26
F. Extended Function Capability	II-26
G. Superposition of Auxiliary Data	II-30
H. TRAC Dependent Variables Names and Descriptions	II-33
I. Programming Guidelines	II-36
J. Examples	II-37
1. Examples of Code Execution	II-37
2. Examples of Option Use	II-54
REFERENCES	II-59
III. EXTRACT	III-1
A. Introduction	III-1
B. Basic Concepts and Capabilities	III-3
C. Using EXTRACT	III-3
D. TRAC Programming Guidelines to Preserve Independence of EXTRACT	III-4
REFERENCES	III-21
IV. CVRT DESCRIPTION	IV-1
A. CVRT Input and Execution	IV-1
Conversion Tasks Currently Performed By CVRT	IV-2
C. Conversion Tasks Not Performed By CVRT	IV-2

FIGURES

I-1.	Structure of COMPij files	I-7
I-2.	Structure of P block	I-8
I-3.	Structure of Q block	I-9
I-4.	Structure of LCDATA block	I-10
I-5.	Output from flow program	I-11
II-1.	TRAP program data flow	II-4
II-2.	Program subroutine function entry	II-38

TABLES

II-1.	LISTING OF EXTFNC	II-31
II-2.	AUXFIL DATA	II-52
II-3.	OUTPUT FROM USING DVNAMES COMMAND	II-57
II-4.	OUTPUT FROM USING SUMMARY COMMAND	II-58
III-1.	DUMP-FILE OVERALL STRUCTURE	III-2
III-2.	EXAMPLES FOR EQUIVALENCING THE NAMELIST AND FLT VARIABLES TO THE DUMMY DVLT ARRAY IN ROUTINE DMPIT	III-6
III-3.	EXAMPLES FOR READING THE DUMMY DVLT ARRAY FROM A DUMP FILE AND REASSIGNING THE VARIABLE TO THE NAMELIST AND FLT TABLES CORRECTLY IN THE RDREST ROUTINE	III-9
III-4.	EXAMPLES FOR EQUIVALENCING THE VLT AND PTL VARIABLES TO THE DUMMY DVLT ARRAY FOR A PIPE COMPONENT IN THE DPIPE ROUTINE	III-12
III-5.	EXAMPLES FOR READING THE DUMMY DVLT ARRAY FROM THE DUMP FILE AND REASSIGNING THE VARIABLES BACK TO THE VLT ARRAY FOR A PIPE COMPONENT	III-19

AUTHORS AND ACKNOWLEDGMENTS

Many people contributed to the TRAC Support Software development and to this report. Because it was a team effort, there was considerable overlap in responsibilities and contributions. The participants are listed according to their primary activity. Those with the prime responsibility are listed first.

PRINCIPAL INVESTIGATORS: Richard P. Jenks and Jay W. Spore
EXCON: Victor Martinez and Mike Turner
TRAP: Ann Barker-Wohletz, Victor Martinez, and Mike Turner
EXTRACT: Victor Martinez
CVRT: Jay W. Spore
REPORT COMPILATION: Wendy M. Rowley, Victor Martinez, and Richard P. Jenks
REPORT TYPING: Wendy M. Rowley and Gloria E. Mirabal

In addition to those listed above, we acknowledge all others who contributed to earlier versions of the software. We also thank all Los Alamos users and external users, both domestic and foreign, who supplied comments and suggestions for this manual.

TRAC SUPPORT SOFTWARE

by

R. P. Jenks and V. Martinez

ABSTRACT

This manual provides users of the Transient Reactor Analysis Code (TRAC) with information about computer codes that can be used to support their analysis efforts. These codes are collectively referred to as TRAC support software.

Specifically, documentation is provided to allow users to implement, apply, and interpret the output of the support software that is available. The TRAC code is an advanced best-estimate code for analyzing transients in thermal-hydraulic systems. Considerable input data are required for modeling a large thermal-hydraulic system and considerable output data are produced from the computer calculation for analysis; the support software assists users in their management of both the input and the output data.

Initially, this document contains sections that describe EXCON, TRAP, EXTRACT, and CVRT, major support software used routinely at Los Alamos National Laboratory. Later sections will be provided as other support codes are documented or as additional support software is developed. In this context, the document is dynamic, growing with time, need, and resource availability.

The first four codes described are central to the efficient and effective use of the TRAC code. EXCON is a TRAC graphics file postprocessor that converts the TRAC graphics output files into component-oriented input data files for use with TRAP. TRAP is an extensive plotting package that allows users to produce a range of plots from quick draft-quality graphs for fast examination of results to final presentation-quality graphics for more formal documentation requirements. EXTRACT is a powerful support code that reads a TRAC dump file and generates "extracted information" for building TRAC input decks with the latest values for system conditions. EXTRACT is particularly useful and cost effective for analyses that require multiple parametric calculation. CVRT converts TRAC-PF1/MOD1 and early-version MOD2 input decks into current TRAC-PF1/MOD2 input decks.

I. EXCON

The EXCON computer code is a TRAC postprocessor which reads the graphics files produced by TRAC and writes individual COMPij files, that a user selects. The COMPij file contains all the geometry and time edit data for just the ij component, where ij corresponds to the user assigned component number. The COMPij files are then used as input to TRAP for a variety of TRAC-graphics tasks.

A. Introduction

The Extract Component Number Code (EXCON) directly supports the Transient Reactor Analysis Code (TRAC) (Ref. I-1) program at Los Alamos National Laboratory under the sponsorship of the US Nuclear Regulatory Commission (NRC). It was developed to extract selected data from the TRAC graphics files (TRCGRF) for further postprocessing by TRAP. Some early versions of the EXCON code shared large blocks of coding with TRAC. Both TRAC and EXCON were maintained by the CDC-UPDATE program, which is similar to HISTORIAN, and the shared blocks of coding were grouped into comdecks. The comdecks would first be extracted from the TRAC program library and then inserted into the EXCON program library, which meant that both codes had identical coding. Most of the comdecks contained the common blocks of variables that were used to write and read data to the TRCGRF files. By sharing the same common blocks it was guaranteed that EXCON would read the variables in the same order that TRAC wrote them, but code sharing also meant that EXCON was version dependent on TRAC. Every time a new version of TRAC was generated, a new version of EXCON was also required.

The current version of EXCON, which includes added capabilities, is not version dependent on TRAC. That means that EXCON needs to be updated only when new components are added to TRAC, when improvements are made, or when errors discovered in the present code are fixed. This drastically reduces the amount of effort required to maintain the code. Some of the added capabilities include the ability to create a separate component file for all system-dependent variable data and the signal and control-block data. The current version of EXCON will also rename all the signal and control variables by suffixing the numerical value to the end of the variable name, thus creating a unique name that can then be requested in TRAP for plotting.

B. Basic Concepts And Capabilities

The following list describes the different operations EXCON will perform on TRCGRF files:

Create a history file called HIST that documents the names of the update error correction files used to produce the version of TRAC that in turn produced the TRCGRF file. This file may be plotted along with the other component files created at the time. Creation of the HIST file is automatic, although it may be destroyed if desired. TRAP will automatically plot the data in this file if it is not destroyed, unless the NCOMP option is enabled in TRAP.

1

Merge the data from as many as 30 TRAC graphics files. EXCON will also merge/extend an existing component file with the component data from a new TRCGRF file. This feature saves the user the inconvenience of retrieving all the TRCGRF files that were used to create such a component file.

Select specified time domains from any graphics file containing more than one time domain (e.g., a graphics file produced during a PWR initialization). This option can also be used to override the default when multiple graphics files have overlapping time domains.

Time shift all time values from any input file by any specified amount.

Input an EXCON data file, which contains all the input data that is requested during a normal interactive EXCON session. This file is not to be confused with TRAC input graphics files. Its purpose is to instruct EXCON which graphics files to use, and to provide corresponding time shifts, time domains, edit frequencies, overall start and stop times, and component information. This is an ideal way to run EXCON in a batch mode.

Delete rod data from VESSEL components, unless rod data are desired, in which case only rod data are included. If your VESSEL component corresponds to component number 1 and you wanted only the rod data, you would request component 1R. If you want both vessel and rod data you would request components 1 and 1R. The VESSEL component is the only component that is split into two separate files. This is done to reduce the size of each file.

Assigns unique names to the signal and control block variables written to the COMP00 file. Since TRAC assigns the variable name SV to all the signal variables and CB to all the control block variables, TRAP would be unable to determine which signal or control block variable a user was requesting. EXCON therefore renames all the SV and CB by suffixing a numerical value beginning with 1 to the variable name and incrementing the value by 1 in the order that the SV or CB variables are encountered. For example, if five signal and four control block variables exist (SV, SV, SV, SV, SV, CB, CB, CB, CB), EXCON would rename them as follows (SV1, SV2, SV3, SV4, SV5, CB1, CB2, CB3, CB4). If the user wanted to plot the second signal variable using TRAP, he would request dependent variable SV2 on the DE command line. This variable name would be unique to all the other signal variables. The label portion of each signal or control block variable would furnish the correspondence

between the user-assigned number used in TRAC and the renamed SV and CB variables.

Writes all dependent variables that do not identify with a given component into the COMP00 file. The variables in this file are known as system parameters or global variables. The file is requested within EXCON and TRAP just like any other component file.

C. User Information

EXCON is a TRAC graphics file postprocessor that converts the TRAC graphics output files (TRCGRF1, TRCGRF2, . . .) into component-oriented input data files for TRAP (COMP01, COMP02, . . .).

To understand the purpose of EXCON, first consider the natural order in which graphics data are output from TRAC:

<u>Time-Point</u>	<u>Data</u>
T ₁	Component 1 Component 2 Component 3
.	.
T ₁	Component 1 Component 2 Component 3
.	.

Thus, at every time step, we find the data for all the variables for each component. It is necessary to rearrange the data into the following format:

<u>Data</u>	<u>Time-Point</u>
Component 1	T ₁ T ₂ T ₃
.	.
Component 2	T ₁ T ₂ T ₃
.	.

This is (albeit oversimplified) the format of the component files (COMPij) produced by EXCON. Each component is described in a separate file containing data pertinent only to that component. Data are also arranged chronologically.

EXCON normally expects the following information:

1. Names of all TRAC graphics files (default is TRCGRF)
2. Time shifts for each file (default is 0.0)
3. Beginning and end times (default is full range)
4. Edit frequency (default is every point)
5. Component numbers (mandatory)
6. Merge time at end of existing component (default is normal end).

If the user runs EXCON in an interactive mode, a choice may be made between an abbreviated input session or a fully detailed session. In the abbreviated session, only the component numbers are required, and EXCON defaults everything else. In the detailed session, EXCON prompts the user for all five categories, allowing a null return to signify a default value.

To initiate the abbreviated session, enter: EXCON GO / t v

To initiate the fully detailed session, enter: EXCON / t v

To initiate EXCON using an input file, enter: EXCON fname / t v

If the user decides to create an input command file, a minimum of five lines is required, corresponding to the first five lines above. The start/stop time and frequency lines may be blank, as can the merge time line. The component name/number line must have entries. All other lines may have a single "E" or "END" to get the default values. In addition, the input file can be any name except GO.

To initiate EXCON in the command file mode, enter:

EXCON fname / t v .

EXCON will create an input command file "EXLIST" (analogous to the TRAP-created command file "LANGFL") from either interactive or command file input. The input command file format allows for multiple-line entries. There is an example of an input file on the following page. Note that comments may be included for clarity and convenience.

The automatically created EXLIST file will have the EXCON internally generated numbers for time domains and shift values. These values may be input using the input file "fname," but a single blank line as in the example will cause the default values to be generated.

Note that if one desires to merge an existing component file, the name of the file must be entered, and not just the number. In the example, components COMP02 and COMP03 would be created from TRCGRF7 only, and an existing COMP22 would be merged with the new component 22 data in TRCGRF7.

If a component is to be merged in an interactive session, the user will be prompted for additional merge information. If desired, all the time points in the last two records of the component can be printed, and the user can choose any one of the values as the beginning merge time for the new data. Normally the same time would be used for all components, but EXCON will prompt the user to indicate whether he or she wants individual file-merging times, in which case EXCON repeats the merge information prompts for each component file.

One word of warning is in order concerning printing out the time points of the last two records of the component file. TRAC, TRAP, and EXCON all use the BFIN/BFOUT triple-buffering subroutines, which have record sizes of 2048 words (including two pointer words). Thus, the last two records of a component file could contain as many as 4094 usable words. If the component involved has short time blocks (such as a one-cell break), then a printout of several hundred lines might follow. Normally, a printout of from 6 to 50 lines might be expected.

NOTE: EXCON will accept and create component files with a three-digit identifier.

All COMP files are written in binary format and structured according to Fig. I-1. A further breakdown of the P and Q blocks is included in Figs. I-2 and I-3. The LCDATA block on Fig. I-4 contains data that are unique to a given type of component and do not exist for most one-dimensional components.

```

* SAMPLE EXCON INPUT COMMAND FILE
*
* NOTE THAT AN ASTERISK IN COLUMN 1 FOLLOWED
* BY A BLANK INDICATES A COMMENT LINE.
*
* OUTPUT GRAPHICS FILE(S) FROM TRAC

TRCGRF

7
E
* TIME-SHIFT VALUES
E
* TIME DOMAIN NUMBERS
E
* START TIME, STOP TIME, AND EDIT FREQUENCY
-1.0E+99 1.0E+99 1
* COMPONENT NUMBERS OR NAMES
2 3 COMP22
E
* COMPONENT MERGE TIME (OVERRIDES THE DEFAULT LAST
TIME)
280.0

```

The following file is functionally equivalent to the above:

```

TRCGRF

7 E
E
E

2 3 COMP22 E
280.0

```

NOTE: If the last entry (280.0) were missing, then EXCON would begin merging COMP22 after the last time found (in the existing COMP22), and not after 280 s.

D. Programming Details

The EXCON code is maintained using HISTORIAN, which is like the CDC UPDATE utility. HISTORIAN will also allow us to export the program library to outside sites in a machine-independent form. The receiving site is also required to process HISTORIAN to convert the ASCII file back to a program library.

We use a subset of the *DEFINE options defined in the TRAC manual. EXCON has not been fully tested on other systems but its relatively small size (~2510 lines) does not present major problems for users who want to implement the code at their sites.

Figure I-5 outlines the calling sequence of the EXCON code.

REFERENCES

1. Safety Code Development Group, "TRAC-PF1/MOD1: An Advanced Best-Estimate Computer Program for Pressurized Water Reactor Thermal-Hydraulic Analysis," Los Alamos National Laboratory report NUREG/CR-3058, LA-10157-MS (July 1986).

BOCBOC	(1 word)	Beginning-of-component-delimiter.
P Block Header	(19 words)	Fixed header (contains start and end times plus title).
Q Block Header	(2 words for COMP00)	Varies for the other type of components.
EOHEOH		End-of-header symbol.
First Time Edit		Packed data for first TIME EDIT.
EOTEOT		Time-edit separator (end-of-time).
.		
.		
.		
Last Time Edit		Packed data for last TIME EDIT.
EOTEOT		
EOCEOC		End-of-component delimiter.

Fig. I-1.
Structure of COMPij files.

(19 words)		
File Name	(1 word)	Name of component COMP00, COMP01, etc.
Start Time	(1 word)	Starting time of COMPIj file.
End Time	(1 word)	Ending time of COMPIj file.
Title 1		Title 16 words long.
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
Title 16		

Fig. I-2.
Structure of P block.

This section does not exist for the COMPO0 file

NDEPV	Number of dependent variables on file.
LTIME	Packed length of dependent variable data in a time edit block. (The EOTEOT symbol not included.)
LDLEN	Zero or the packed length of the vessel level data.
INUM	User assigned component ID number.
TYPE	Component type (vessel, pipe, etc.) hollerith variable.
ID	Computer code assigned ID number
NCELLS	Zero or total number of cells in component.
LCDATA BLOCK	Additional data needed when type = (VESSEL, STGEN, TEE, PLENUM, CORE, ROD, SLAB).
Geometry Block	Geometry data, varies or does not exist depending on component type.
NAME - NAME _{NDEPV}	Dependent variable names.
IPTR - IPTR _{NDEPV}	Dependent variable pointers.
NWRDS - NWRDS _{NDEPV}	Length of dependent variable.
LABEL _{1,1} - LABEL _{3,1}	Three word label for first dependent variable.
.	
.	
.	
LABEL _{1,NDEPV} - LABEL _{3,NDEPV}	Three word label for last dependent variable.

Fig. I-3
Structure of Q block.

Vessel	
NWORD	Number of words in block.
NCLX	Number of cells per axial level.
NASX	Number of axial levels.
NRST	Number of radial rings.
NTSX	Number of azimuthal segments.
NCSR	Number of cell source connection.
NODES	Number of rod radial heat transfer nodes.
NZMAX	Maximum number of axial fine mesh levels.
STGEN	
NWORD	Number of words in block.
NCELL1	Number of cells in primary side.
NCELL2	Number of cells in secondary component.
NSCMP	Number of components that makeup secondary flow path.
NODMX	Number of radial heat transfer nodes.
NCLP	Number of cells in side leg (if tee) of primary component.
STYPE ₁ - STYPE ₁₀	Secondary component types (pipe, tee).
NCLS ₁ - NCLS ₁₀	Number of cells in main tube of secondary component.
NCLT ₁ - NCLT ₁₀	Number of cells in side leg of secondary component.
TEE	
NWORD	Number of words in block.
NCELL1	Number of cells in primary side.
NCELL2	Number of cells in secondary side.
PLENUM	
NWORD	Number of words in block.
NPLJN	Number of junctions.
CORE	
NWORD	Number of words in block.
NDRDS	Number of radial heat transfer nodes.
NRODS	Total number of fuel rods.
NZMAX	Maximum number of axial fine mesh levels.
ROD OR SLAB	
NWORD	Number of words in block.
NODES	Number of radial heat transfer nodes.
NZMAX	Maximum number of axial fine mesh levels.
NRODS	Total number of fuel rods.

Fig. I-4.
Structure of LCDATA block.

```

1  excon      calls
           qgetsym , fexist , rename , exit , iword
           gname  , ljust  , getuff , qassign , bfaloc
           ready  , bfin   , bfclos , close  , chklun
           grfget , qqsort , find  , reorder , qdstroy
           ucopy  , runit  , getlen , famsiz , skipf
           bfouts , bfout  , repl  , compact ,
2  bfaloc    called by
           excon
3  bfclos    called by
           excon
4  bfin      calls
           error
           called by
           excon , grfrd
5  bfout     calls
           error
           called by
           excon
6  bfouts    calls
           error
           called by
           excon
7  grfget    called by
           excon
8  setlcm    calls
           memadj
9  error     calls
           exita
           called by
           bfin , bfout , bfouts , find
10 find      calls
           error
           called by
           excon
11 gname     calls
           rjust , z2b
           called by
           excon
12 grfrd     calls
           bfin , exita
           called by
           ready
13 iword     called by
           excon
14 qqsort    called by
           excon
15 ready     calls
           grfrd
           called by
           excon
16 reorder   called by
           excon
17 repl      calls
           runjs , runjc , ljust , runjf
           called by
           excon
18 z2b       called by
           gname
           library subroutines

```

Fig. I-5.
Output from flow program.

1	qgetsym	called by excon	
2	fexist	called by excon	
3	rename	called by excon	
4	exit	called by excon	
5	ljust	called by excon	
6	getuff	called by excon	repl
7	qassign	called by excon	
8	close	called by excon	
9	chkln	called by excon	
10	qdstroy	called by excon	
11	ucopy	called by excon	
12	runit	called by excon	
13	getlen	called by excon	
14	famsiz	called by excon	
15	skipf	called by excon	
16	compact	called by excon	
17	memadj	called by setlcm	
18	exita	called by error	grfrd
19	rjust	called by gname	
20	runjs	called by repl	
21	runjc	called by repl	
22	runjf	called by repl	

Fig. 1-5 (cont).

II. TRAP

The TRAP computer program is a versatile postprocessor routine that performs a variety of TRAC-graphics tasks. TRAP (TRAC Plot) can graph curves, surfaces, and component structures, and can be extended easily to more complex applications. Users communicate with TRAP by a sequence of command language statements that allow a high degree of flexibility in defining plot requests. Output media for the resulting plots include 16-mm movies, 35-mm film (black and white or color), 105-mm (fiche) film, 11-in. and 36-in. electrostatic plotters, and direct imaging on a Tektronix or similar CRT terminal. Items and commands in italics are of special interest to Los Alamos users only.

A. Introduction

The TRAC (Ref. II-1) computer program outputs nuclear reactor parameters and fluid flow variables, which we call dependent variables, at selected times in the calculation. Because these variables also may change with respect to position within the reactor components, we represent them by either $F(p,t)$ or $\underline{F}(p,t)$. Here, F denotes a scalar-valued function (such as liquid density), while \underline{F} denotes a vector-valued function (such as liquid velocity). The symbols p and t denote, respectively, spatial and time independent variables, where, for example, $p = (r,\theta,z)$ represents a point in cylindrical coordinates within a VESSEL component.

The TRAP graphics program basically produces computer-generated plots of dependent vs independent variables in a variety of useful combinations, including curve and surface displays. On option, curves can be drawn smoothly (that is, with continuous slopes). Also, several curves can be drawn on a single frame. Surface displays erase hidden lines, and users can vary viewing position for such purposes as stereo pair plotting. Examples of the plotting capabilities are given in Sec. J.

Users communicate their plot requests to TRAP by employing a simple yet powerful set of TRAP graphics commands. They either can give these statements to TRAP interactively, thereby responding to program queries, or they can run the program in a noninteractive mode by supplying the name of a previously prepared file of commands.

Interesting features of TRAP include a MOVIE option, MACRO-component displays in which individual components are drawn horizontally and vertically VESSEL SLICE and SLAB displays, and a USRDRW option analogous to the SLICE options (in which users define a loop or system diagram by supplying cell boundary coordinates). Also, there exist EXTENDED FUNCTION capability, in which users define specific dependent variables as mathematical and logical combinations of those supplied by TRAC; superposition of auxiliary data, in which data in a peripheral file are co-plotted on the current frame for data comparison applications; direct on-line creation and manipulation of auxiliary files; and formatted data file preparation. Another useful feature is the FORMAL command, which puts TRAP into a mode that produces publication-quality plots, character styles, and variable descriptions.

One of the guidelines in developing TRAP has been to construct a program capable of handling basic graphics as well as special plotting applications. For example, it is relatively easy to add a particle-pushing movie program (Ref. II-2), where certain combinations of density, velocity, and void fraction dependent variables are used to depict two-phase fluid flow in a reactor vessel. Such extensions provide more realism and drama in the graphics.

B. Basic Concepts and Capabilities

Recall the function representation of dependent variables, $F(p,t)$, mentioned previously. Many tasks given to TRAP can be organized conceptually by considering the extent of the domain variable p . For instance, consider the following cases:

1. p is not defined (for example, component-valued functions or global variables such as reactor power), where
POWER = $f(t)$.
2. p is one-dimensional (for example, a PIPE component with geometric coordinate x), where
PRESSURE = $f(x,t)$.
3. p is two dimensional. For example, a rod with
TEMPERATURE = $f(r,z,t)$
where
 r = node radial coordinate and
 z = axial coordinate.
4. p is three dimensional; for example, a VESSEL component with
VOID FRACTION = $f(r,\theta,z,t)$,
where
 θ = azimuthal coordinate,
 r = vessel radial coordinate, and
 z = vessel axial coordinate.

The current TRAP implementation includes functions that are cell-center valued (such as pressure), cell-edge valued (such as velocity), component valued (such as reactor power), or rod valued (such as average rod temperature), and system-valued parameters such as the signal and control block variables.

For one-dimensional components such as PIPES, the specific plotting combinations are:

Function	Description
f vs t	curve
f vs x	curve/USRDRW graphic diagram
f vs (x,t)	surface
$f(x)$ vs $g(x)$	correlation curve
$f(t)$ vs $g(t)$	correlation curve

Here, we specify only the free independent and the dependent variables. The fixed independent variable is not explicitly named because it is easily inferred. Independent variables are either fixed or free, where free indicates the variable that traverses a curve or surface; that is, it appears as an abscissa coordinate. The fixed independent variable serves as a general index in a set of plotted curves. For each such fixed value, a separate curve is generated. The curves can be members of a family all appearing on one frame; or the individual curves can be plotted on distinct frames. For example, plot $y = f(x,t)$, where t is free and x is fixed at the first through fifth cells of the component. This produces five separate curves, one for each cell in the range one through five. The abscissa variable for each of these curves is time.

Correlation curves are actually parametric curves with the free variable acting as the defining parameter for the points. These curves are not necessarily one-to-one off either axis, and in fact may self-intersect. An example is to cross-plot vapor temperature (TV) vs liquid temperature (TL) over the range 0.0 to 1.0 at intervals of 0.2 s. Here, six values of TL (TL_1, \dots, TL_6) define the abscissa coordinates in a two-dimensional plot and six values of TV (TV_1, \dots, TV_6) define the ordinate coordinates. The pairs (TL_i, TV_i) define points on the correlation curve, where (TL_i, TV_i) corresponds to the time value = $0.2 \cdot (i-1)$, for $i = 1, 2, \dots, 6$.

For three-dimensional components the following combinations are available:

Function	Description
f vs r	curve
f vs θ	curve
f vs z	curve
f vs t	curve
f vs (r, z)	surface/slice/USRDRW graphic diagram
f vs (r, θ)	surface/slice/USRDRW graphic diagram
f vs (θ, z)	surface/USRDRW graphic diagram
$f(r)$ vs $g(r)$	correlation curve
$f(\theta)$ vs $g(\theta)$	correlation curve
$f(z)$ vs $g(z)$	correlation curve
$f(t)$ vs $g(t)$	correlation curve

The surface displays are three-dimensional perspective views of the surface with hidden lines eliminated, or horizontal and vertical cross-sections (slices) of the vessel geometry, shaded or colored in quantized levels of the dependent variable.

C. Using TRAP

TRAC produces a graphics file, TRCGRF, which is converted into a set of reorganized component data files by program EXCON (Sec. I). These ("COMPij" files) are indexed by integers between 1 and 999, corresponding to the number defined by the user. There are three exceptions to the naming convention of the COMPij files: if vessel rod data are to be plotted, then the component file name must have an "R" as the right-most character (for example, COMP01R). Also, EXCON automatically creates a file named HIST, which contains information about the updates (if any) used to create the version of TRAC, which was used to produce the TRCGRF file. TRAP will automatically output (if the HIST file is local) this information as the first plot produced. If the file is not local, TRAP will forget about it, and continue. Also, EXCON will generate a COMP00 file if a component 0 (zero) is requested. This file will contain all the system global variables such as CPUTOT and all the signal and control block variables. Reserving component 0 (zero) does not create any problems since users are not allowed to define a component number 0 (zero) in the TRAC input. Program TRAP employs the COMPij files as a data base for its work in that it extracts appropriate geometry and physical variables from them in preparation for producing plots (Fig. II-1). TRAP also may use an LBLIST file of X and Y axes labels that can be modified by the user.

The user communicates with TRAP either in an interactive mode or by a prepared file of plot commands. If the commands are given in a file, then they appear exactly as they would in an interactive terminal session, except for the prompt symbol "?". See Sec. J for examples.

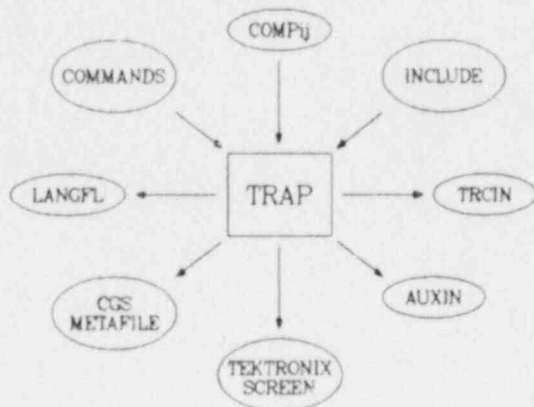


Fig. II-1.
TRAP program data flow.

Comments can also be inserted in the command file by beginning the line with an astric (*). In either mode, a user gives the program a set of language statements forming a plot request. A language statement, or command, consists of an ordered set of space-delimited arguments, such as

FIXED RADIUS 1 3

The first argument is a command primary keyword. It may be followed in the string by other, secondary entries that define arguments to the command. Here, the keyword FIXED tells TRAP that a fixed independent variable is being defined. Also, the phrase

FREE RADIUS 1 5

tells TRAP the independent variable is the VESSEL radius that ranges from the first through the fifth radial cell. At Los Alamos National Laboratory, several commands can be entered on one line if they are separated by an "escape" control character. Command syntax varies according to command type. A list of available TRAP commands is given in Sec. II.D, where the specific format of each is defined.

The command entries are used by the program to assemble a definition of the plot task for dependent and independent variable specifications, component ID, etc. The order of the language statements is flexible, because the program accumulates a plot request (or plot problem definition) by analyzing entries independently. When the user is satisfied with the definition, the DISPLAY command is entered to terminate the plot request. TRAP then checks the current set of commands supplied by the user for completeness and consistency, and if

satisfied, TRAP extracts all pertinent data from the component files and creates the necessary plot frames.

Execution line arguments (order-independent) may be specified in addition to FILE FILE-NAME. These include device release codes, an execution time flag, plot title option, and two formatted output file options.

The device and film choices are:

- 11-in. electrostatic plotter
- 36-in. electrostatic plotter
- 35-mm black and white slides
- 35-mm color slides
- 16-mm black-and-white movie
- 16-mm color movie
- 105-mm microfiche

To enable these options, enter 11, 36, 35, 35c, 35cs, 16, 16c, or 105, respectively; or specify TEK on the execution line if direct Tektronix screen output is required. If KEEP is specified, then the graphics routines define a plot file (specifically, a *CGS metafile*)* that can be viewed at a CRT terminal using the *PSCAN* (Ref. II-3) or *PTEKT* (Ref. II-4) utility upon termination of TRAP. This file can later be released to the system for film production by employing the *PFILM* (Ref. II-5) utility

PFILM / t v

Eleven-inch or 36-in. electrostatic plotter output can be obtained by employing the *PESP* (Ref. II-6) utility.

PESP FORMAT=11 ORIENT=-90 IMAGE=(LIMITS 0. 9. 0.5 7.4545)

If *GIVE* is specified on the TRAP execution line, then the film file is automatically released, and is no longer available to the user. *PSCAN*, *PESP*, and *PFILM* options such as *IMAGE*, *MODEL*, *ORIENT*, etc., can be entered by means of the command input file for use by the *GIVE* option. Parameters are required, although detailed output formatting for these options is automatic. If intermediate execution-time information is desired, then type *TIME* on the execution line. If *TITLE* is entered on the execution line, then TRAP uses the title supplied in the just line in the command language file as a plot title for each frame.

Two keywords, *TRACIN* and *AUXFILE*, may also appear in the execution line options. Each of these keywords causes TRAP to create a formatted output file (named *TRACIN* or *AUXIN*, respectively) instead of graphic output. When the keyword *TRACIN* is specified, the data requested by the user are written to the *TRACIN* file in the TRAC input load format. Keyword *AUXFILE* causes the data to be written in the TRAP INCLUDE input format for plotting as data points. The two keywords are mutually exclusive and TRAP will act on the last one specified if both are accidentally included. Use of these keywords limits the specification of data to a single independent and a single dependent variable, as well as to

* *CGS metafiles* may be output on any of numerous graphics devices. In particular, 16-mm, 35-mm, and 105-mm FR80, 11-in. or 36-in. electrostatic plotters, the Tektronics 40XX series terminals, and similar graphics devices. The metafiles are also machine independent, and may be viewed or output using a CDC 7600, Cray, or VAX computer.

single values of all fixed variables. The TRACIN keyword produces both independent and dependent variable values if the independent variable is spatial. Thus, the output can be included in a TRACIN file directly as either boundary conditions (by a time-dependent FILL or BREAK) or as initial conditions for a renoded component. The AUXFILE keyword always produces both independent and dependent variable values, with corresponding error bar half-widths set to zero.

The defaults for the execution line arguments are

- (a) enter language statements in an interactive mode.
- (b) KEEP release code.
- (c) do not create an intermediate timing printout.
- (d) use the title from the COMPij file as a plot title, and
- (e) create a CGS (graphics) metafile

An example of the use of execution line arguments follows:

```
TRAP 35CS FILE TRAPIN GIVE TITLE / 1 1.2 .
```

File "TRAPIN" is structured as follows:

```
TEST CASE
```

```
etc.
```

Here,

- (1) each frame has TEST CASE as a main title.
- (2) 35mm color slides are generated.
- (3) TRAP processes a previously prepared file of commands, and
- (4) the CGS metafile produced by TRAP is automatically given to the system.

As TRAP processes a set of command statements, it writes a peripheral file, named LANGFL, which contains every user entry. If the user submits commands one per line, then TRAP does not write given commands that are in error. If the user submits multiple commands per line, then any command in the line that is in error causes TRAP to omit writing all of the commands in this line to LANGFL. This provides the opportunity to edit a command file created in the query mode and reuse it directly. However, if a user wishes to reuse LANGFL, the name must be changed because a file name conflict would occur when a new LANGFL was automatically created.

D. Syllabus of TRAP Commands

The following is a complete list of currently implemented TRAP commands. The commands, independent variable names, or secondary keywords can usually be abbreviated to the first two letters in their names (this is indicated by underlining). In the following list, all keywords, abbreviations of keywords, parameters, etc., enclosed in { . . . } are mandatory, and the remaining keywords and/or parameters are optional. TRAP has the property of "file

memory," remembering the user commands from one plot request to another. All previous entries are retained except for the one replaced unless the CLEAR command is given (see description of clear command below). Comments can be placed after the last numerical entry of any line. Comments must be separated from the last numerical entry of the line by at least one space.

ADJUNCT {N}

Sets up TRAP internal logic to handle the adjunct components of a steam generator (as opposed to the main PIPE or main TEE component). These components are PIPES or TEEs, depending on the problem input deck. The N' component will be selected by the above command. This component, if it is a TEE, will have a main tube and a side tube which may be selected by MAIN (default if SIDE is not enabled) or SIDE. The DVNAMES command will cause detailed information about the component types and cell structure to be printed out. CLEAR ADJUNCT or CL AD will cause TRAP to revert to the main PIPE or main TEE of the steam generator.

AVERAGE

Averages a fixed dependent variable over the cell range N1-N2. The averaged dependent variable replaces the original dependent variable in the plot buffers. This option is limited in that only one independent variable may be fixed. (For averaging VESSEL dependent variables, the WRAUX and OPAUX commands could be used to output and average auxiliary file data).

For example:

```
COM M
FREE TIME 25
DEF DV
FIXED X N1 N2
AVERAGE
DISPLAY
```

will compute each point dv_i of DV as follows:

$$dv_i = \frac{1}{N} \sum dv_i(k)$$

where k (the k 'th cell value) ranges from cell N1 through cell N2, and $N = N2 - N1 + 1$ is the total number of cells to be averaged.

AXES (See argument description below).

Defines the number of decimal places plotted at the tick marks on the X and/or Y axes. There are three formats:

```
AXES {NXSTEPS NXDEC NYSTEPS NYDEC} HEIGHT
AXES {X NXSTEPS NXDEC} HEIGHT
AXES {Y NYSTEPS NYDEC} HEIGHT
```


The first format modifies both axes, while the second and third formats modify just the X and Y axes respectively. Parameters are:

NXSTEPS = number of divisions on the X-axis
NXDEC = number of decimal places (X-axis)
NYSTEPS = number of divisions on the Y-axis
NYDEC = number of decimal places (Y axis)
HEIGHT = character height of tick-mark numbering.
(Default value is 0.175 inches.)

NOTE: If NXDEC or NYDEC is negative, scientific notation is implied. Also, FORMAL is enabled by this command, if not already enabled.

BORDER

Draws a border around the overall page. This option is useful for PESP output and the SLICE and USRDRW options. Also, the appearance of color presentation slides is enhanced by this option.

CLEAR {X} Y Z etc.

Clears options or variables X, Y, and Z, etc. Also, "CL ALL" can be used to clear all options and variables previously set with the following exceptions which are not affected by CLEAR ALL (but can be individually cleared): BORDER, CLIP, DUALAXIS, FNUM, FORMAL, NBOX, NLOGO, NUMBER, TRANS, and XTRANS.

CLIP {OPT} DVMIN DVMAX

Forces data within the specified limits. OPT must be provided, and may have one of the following forms:

"ALL" (Clip all TRAC data and auxiliary file data)
"AUX" (Clip only auxiliary file data).
"TRAC" (Clip only TRAC data.)

If no DVMIN and DVMAX are provided, the data will be scissored between the plot area frame boundaries. If either DVMIN or DVMAX is desired, then they must both be provided. This feature is particularly useful when producing plots containing spikes or spurious data which would, if scaled to fit the spikes, cause a loss of resolution in the data surrounding the spikes. Using the CLIP feature, the data may be scaled to any value desired without having the curve running off the plot area.

NOTE: The abbreviation for CLIP is CP, not CL. TRAP allows a 3/4-inch "grace" runoff margin for unclipped data. This option should not be confused with the SCALE option.

COMPID {N}

Defines component no. N as the current component from which plots are to be generated.

NOTE: The abbreviated command for COMPID is COM, not CO.

COPIES {N}

COPIES is used in the TRAP GIVE mode to define the total number of complete output copies desired from the TRAP run.

NOTE: There is no abbreviation for this command.

DCURVE

Deletes connecting lines between auxiliary curve points. This feature is useful when plotting calculated data versus scattered empirical data, or when the auxiliary file has only a few points.

DEP {DV1} DV2 etc.

Defines dependent variables DV1, DV2, etc. No more than two may be defined, unless EXTEND is enabled, in which case five may be defined. If two variables are defined and if EXTEND is not enabled, then TRAP produces correlation plots. Each use of DEP clears the dependent variables from previous definitions.

DISPLAY

Computes and outputs the current plot. This option may also be used to change movie highlighting, components, variables, ΔT , etc., during a movie run by making separate DISPLAY commands during the movie, each command defining a different label, component, variable, ΔT , etc.

DUALAXIS YMULT YSHIFT

Sets up two separate systems of units for the Y-axis scales on the left and right borders. The left border is in SI units, and the right border is in English (sometimes called Engineering) units. User-defined stand-alone units are possible, but normally the user would allow TRAP to draw the plots automatically using the entries found in the LBLIST file. If a stand-alone entry is desired, the right label may be loaded using an OV LD command followed by the desired variable description. YMULT and YSHIFT (optional) are used to scale and shift from the left unit to the right unit:

$$Y_{NEW} = Y_{MULT} * Y_{OLD} + Y_{SHIFT}$$

If YMULT is entered, YSHIFT will be assigned a value of 0.0 if defaulted, and TRAP will ignore all scale-related values from LBLIST until another DUALAXIS command (with no additional entries) is encountered. DUALAXIS and TRANS are incompatible options and TRAP will not allow them to be simultaneously enabled.

DV NAMES BRIEF

Lists the dependent variable names and their corresponding definitions as found in the current component defined by the COMPID command. These names are the only dependent variable names that TRAP will recognize, using the DEP command.

If BRIEF (or BR) is included with the command, only the geometric information relating to the component (type, number of cells, etc.) will be given.

NOTE: The COMPID option must be enabled before to the DV NAMES command.

EJUMP

Terminates the branching over TRAP command file statements which was initiated by a JUMP command. Once JUMP is enabled, TRAP will recognize only the EJUMP command unless END is encountered, in which case the program will terminate. Refer to the JUMP command for more detail.

END

Terminates the current TRAP session.

EXTEND {N}

Calls an extended function (supplied by the user) for computations related to current dependent variable data. Here, N refers to a flag defined by user in his own function subroutine. This option requires at least a partial recompilation of TRAP. Refer to Sec. F for details.

NOTE: The EXTEND command must be enabled before the DEP command.

FILE {FNAME}

Defines a TRAP command input file "FNAME" to be used for the TRAP run. FILE is usually entered immediately on the execute line, but may also be entered interactively. In the latter case, the user should reset any existing options that are not desired with the CLEAR command before entering the command file mode. TRAP then becomes completely automatic and will not return control to the user.

NOTE: There is no abbreviation for this command.

FIXED {IDV N1} N2 N3

Defines the independent variable IDV as fixed from N1 to N2, with N3 increments (N3 is valid only when defining TIME or Z during the ROD mode). Either N3 alone or both N2 and N3 may be null. The independent variable IDV must be RA or RADIUS, TH or THETA, TI or TIME, X, or Z. If IDV is TIME, then N1, N2, and N3 must be floating-point integers. If IDV is not TIME, then all parameters must be integer valued, except when defining rod heights. For example:

```
ROD 3
FI Z 3.25 4.57 5
```

defines rod number 3 at heights from 3.25 meters to 4.57 meters, with 5 equally spaced increments. In this mode, N1 and N2 are floating-point integers, and N3 is integer valued. This example would result in five plots if the NOADV option is not enabled. Also,

```
ROD 3
FI Z 5 FACTOR SHIFT
2.2 3.1 5.7 5.8 6.5
```

would produce five plots as before, except that the increments are not equal. In this format, the line following the FIXED command must contain monotonically

increasing (space-delimited) values of Z. The number of Z-values to be input are defined by the integer (in this example, "5") following FI Z.

FACTOR and SHIFT (optional) operate on the Z-values and can be used to convert Z from meters to other units, while simultaneously performing a linear shift. For convenience, the translation is performed on the Z-values before further data retrieval. Therefore, the user must be certain that the values of FACTOR and SHIFT are correct, so that when TRAP performs the translation, the internal Z-values will be referenced from the bottom of the component, and not the bottom of the core. The values on the third line will be plotted; thus, incorrectly translated Z-values will not necessarily be obvious.

For example:

```
ROD 1
FI Z 3 0.0254 1.5
    0.0 6.5 14.8
```

references the core at 0.0, 6.5, and 14.8 inches, assuming the distance from the bottom of the component to the bottom of the core is 1.5 meters. Internally, TRAP would reference the rod data at 1.5, 1.6651, and 1.87592 meters from the bottom of the component.

NOTE: FACTOR must be included with SHIFT, but if only FACTOR is provided, a default shift value of 0.0 is assumed.

FLEGEND X Y

Re-enables curve markers for the NBOX mode, and forces the legend to be plotted along with curve data. The legend will begin at the coordinate (X,Y) on the page. If X and Y are omitted, then TRAP will set them to X=6.5 and Y=6.25 (inches).

NOTE: X and Y are in page units (inches).

FNUM

Writes "FRAME NO. N" in the lower left-hand corner of the plot, where "N" is increased for each new plot. This option is intended for fiche output. (Also see the NUMBER option).

FORMAL

Formal mode. This option invokes the following features:

- Formal variable descriptions are extracted, if possible, from the file "LBLIST."
- DUPLX character style
- Header/Title(s)

NOTE: Any of the above features may be overridden by the user. Refer to the LBCALL command for further description of LBLIST.

FORMAL also invokes the DISSPLA character sets, fonts, etc. (Refer to the DISSPLA (Ref. II-7) manual for full details). TRAP uses the following conventions:

- "{" = switch to the lower-case alphabet
- "}" = return to the (standard) upper-case alphabet
- ":" = switch to the instruction alphabet

TRAP also internally defines and uses these instruction alphabet flags:

- Z = EH.5
- Z1 = EXHX
- Z2 = EH.5) 2:EXHX
- Z3 = EH.5) 3:EXHX
- Z4 = E.4) :EX

For example, when FORMAL is enabled, the line

$Y = A:Z2)X + :M6)D:MX)T:LH.5)X:LXHX) - - - (LOWER)-CASE - - - :M11) K:MX(F(X)DX)$

would be translated as:

$$Y = A^2X + \Delta T_x - - - \text{lower-CASE} - - - \int f(x)dx$$

FREE IDV N1 N2 N3 (See argument description below).

Defines independent variable IDV as free from N1 to N2, with increments of N3 units. The structure of these independent variables is the same as in the FIXED command. For defining rod heights, the usage is similar to the description under the FIXED command. Normally, N1 and N2 must be defined. However, an exception is made for the independent variable "TIME." That is, "FR TI" or "FREE TIME," equivalent to

$$\text{FREE TIME TB TE } \frac{TE - TB}{74.0}$$

causes TRAP to default to the beginning and end times of whatever component is defined. Also, "FR TI N" or "FREE TIME N" will define a time interval with N points. (Note that the default is 75 points.) The resulting time interval would be equivalent to

$$\text{FREE TIME TB TE } \frac{TE - TB}{N - 1}$$

This feature is particularly useful when running batch jobs, when the run duration is not known at the time the job is submitted.

GRID (See argument description below).

The GRID command has two formats, depending on its purpose:

GRID DX DY PAGE: draws a grid over the entire page.
(Used for label positioning.)

GRID NX NY PLOT: draws lines only across the plot area.
(Used for graph measuring.)

DX and DY are real valued, in page units (inches). NX and NY are integer valued, denoting the number of lines between tick marks. (NX or NY may be zero.) PAGE and PLOT are keywords spelled out as indicated.

HELP

Provides the user with a detailed description of commands which TRAP reads from the LBLIST file. (See LBCALL for further description of this file.) The full spelling of the command and the abbreviation (when applicable) are accepted. After enabling the HELP command, enter a command or its abbreviation for a detailed description of the command format and its usage. To exit the HELP session (and re-enter the interactive command mode), enter a single period.

HISTORY

Plots the update history of the version of TRAC used to create the user's component files. File HIST must be local; if not, no harm is done, and TRAP continues with the next command.

NOTE: This option is primarily intended to be used in the TEK (Tektronix session) mode, because TRAP automatically outputs the contents of COMP00 in any other mode.

IMAGE (See argument description below).

IMAGE is used in the TRAP GIVE mode for predetermining the plot or film image size. [For more detail, consult the writeups on PFILM (Ref. II-5) and PESP (Ref. II-6).] The command has two formats:

IMAGE X1 Y1 becomes IMAGE=(EXACT X1 Y1)

IMAGE X1 X2 Y1 Y2 becomes IMAGE=(LIMITS X1 X2 Y1 Y2)

NOTE: There is no abbreviation for this command.

INCLUDE {FNAME} N KRVTYP KOLOR

Draws plot number N of the auxiliary file FNAME on the current plot area. If N is zero or null, and file FNAME is already open, the next sequential plot is drawn, but if file FNAME is not open, the first plot on the file is drawn. (Note that a plot may be several curves.) If N is a positive integer, the auxiliary file is rewound, and the first N-1 plots are skipped. If N has a value of zero (for additional line entries), it is treated as if it were blank or null. Curve types and colors may be selected within the auxiliary file itself by extending the number of points and bar-skip line in the following manner:

NOPTS SKIP KRVTYP KOLOR

where KRVTYP = 0, 1, 2, 3, 4, 5 results in SOLID, DASH, DOT, CHNDSH, CHN-DOT, and NOCURVE (that is, just the curve markers), respectively, and KOLOR = RED, YELLOW, GREEN, etc. [refer to the DISSPLA (Ref. II-7) manual for more detail]. The default curve type is DASH, and the default color is YELLOW. The INCLUDE command must be respecified before each DISPLAY command, unless

MOVIE is enabled. When MOVIE is enabled, one auxiliary file should be used (for efficiency) for coplotting with calculated data. The auxiliary file must consist of the same number of plots as the number of movie frames that automatically enable the INCLUDE command. Of course, more than one variable and corresponding auxiliary file may be plotted in the same movie. For further detail, see Sec. G.

The TRANS option creates a problem with handling auxiliary files. For instance, if an auxiliary file is written during the translate mode and is read later during a nontranslate mode, a scale conflict occurs. This problem is resolved in the following manner: TRAP will flag any auxiliary files written during the translate mode by writing "TRANS" as the second entry on the first line of the file. Then, whenever this file is read in while the translate mode is in effect, it will not be translated. The TRANS flag will not affect plots in the nontranslate mode. The manner will always be translated before plotting during the translate mode. Care should be taken when merging auxiliary files to ensure that plots consist entirely of files in the same unit because TRAP will not check for mixed-mode files.

In summary, when INCLUDE is used to plot auxiliary files in a nontranslate mode, the files are plotted as they are, regardless of any internal translate flags. However, when using INCLUDE to plot files in the translate mode, the dependent variable data from all auxiliary files will be translated unless they are flagged as being data from files that have already been translated.

JUMP

The user can use this command to initiate a branch over command file statements. TRAP will ignore all subsequent input commands (except END) until an EJUMP command is encountered.

NOTE: The user is cautioned that the LANGFL which TRAP creates in this mode will not necessarily echo the commands that are entered by the user interactively, or are contained in the TRAP input command file.

LABEL {NL H XC YC XL YL} ANG

Forces a line or block of text on the current plot. X and Y values are in page units (inches). The parameters are:

NL	=	number of text lines (which must follow the command)
H	=	character height (0.0 defaults to internal height)
XC	=	X-center of label box
YC	=	Y-center of label box
XL	=	X-length of label box
YL	=	Y-length of label box
ANG	=	angle of box (degrees) relative to the X-axis

NOTE: If ANG \neq 0, and COLOR is enabled, then the label lettering is shaded YELLOW. Thus, ANG=360.0 would be equivalent to ANG=0.0 with YELLOW

lettering (excellent for highlighting). Also, FORMAL is enabled by this command, if not already enabled, and only one label command call is allowed per plot.

For purposes of identifying curves, the LABEL command may also be used to plot a short curve of a specified type (solid, dot, dash, etc.) immediately before the text description. To do this, the first four characters of the label line must be TYPE, followed by an integer identifier as described in the INCLUDE writeup in this section. The integer identifier must be separated by a blank, and the entire line must be in the following format:

TYPE NNNNN	10 character/word machines
TYPE NNN	8 character/word machines

LBCALL {VARNAME}

Communicates with the LBLIST (dependent variable label look-up) file. The name VARNAME is saved to be used by the look-up routine, and the corresponding variable description (found in LBLIST) will be used for the Y-axis label. Thus, the user can make his own variable descriptions to be used by LBLIST. The primary purpose of this option is to save the user the tedium of making multiple redundant Y-axis definitions. Using LBCALL, only one definition is necessary for any number of plots. Auxiliary file plots can also be labelled with this command. That is, the auxiliary file dependent variable may be defined in the LBLIST label look-up file. This command does not affect the auxiliary file name; it affects only the labelling of the plot. To allow automatic label description look-up, and to accommodate the secondary axis descriptions and units required by the DUALAXIS and TRANS options, the label description look-up file LBLIST includes a primary dependent variable axis unit, a secondary dependent variable axis unit, and the necessary conversion factors to translate from the primary unit to the secondary unit.

NOTE: FORMAL is enabled by this command, if not already enabled. LBLIST also contains information used by the HELP option.

LEGEND {NAME1} NAME2

Replaces the TRAP-supplied legend box data with NAME1 and NAME2. NAME1 and NAME2 are two 10-character (or fewer) space-delimited words. These labels may be expanded using the OVERRIDE option into two 80-character lines, including blanks. This option is intended to be used with only one TRAC curve. However, several auxiliary curves may be included in the plot.

MACRO {N}

Draws component number N with the X and Y axes referenced in meters. (This option is not valid for 3D components, but the SLICE options will display mesh cells which are proportionately accurate for measurement purposes.)

MAIN

MAIN is used to clear the SIDE command for a steam generator or tee component. This command is equivalent to "CLEAR SIDE" or "CL SI," but it was created to make switching between the main and sides of a tee component (or subcomponent) easier.

NOTE: The abbreviation for MAIN is MN, and not MA.

MOVIE

Puts TRAP into a movie mode. For the run duration, one leader, one caption, and one trailer are scheduled, regardless of the number of individual "DISPLAY" commands which comprise the run. This allows the user to plot several variables and components in various modes during the duration of the movie. The user may choose to slow down the movie (by decreasing ΔT) for periods of greater interest, or to speed up the movie (by increasing ΔT) for periods of lesser interest. Highlighting is accomplished by separating the movie into multiple contiguous time domains, and defining a (usually single-line) label for each domain. TRAP assumes that TIME is the independent variable that indexes the movie frames, so the user must have TIME defined accordingly.

MRGAUX {SINKFILE AUXFILE₁ AUXFILE₂ . . . AUXFILE_N}

Merges auxiliary files into a designated sink file. This option permits the user to plot separate auxiliary files at the same time. The sink file must not be any of the auxiliary files, although it may be an existing file. In the latter case, it is completely rewritten. N must not exceed 6. If more than six files are to be merged, repeat the MRGAUX command.

NOTE: Auxiliary file merging is performed immediately upon receipt of the MRGAUX command. DISPLAY does not affect this command.

NAMAUX

Assigns an 80-character title, which must be the next line of text, to the next auxiliary file to be created. The primary purpose of this option is to provide a means of labelling auxiliary files created with the OPAUX and WRAUX commands. The NAMAUX command should immediately precede the WRAUX or OPAUX command. NAMAUX is cleared after OPAUX or WRAUX terminates.

NOTE: This file name will overwrite the first (title) line of the TRAP input command file, producing a plot title conflict if TITLE is entered on the execute line.

NCOMP

Prevents the automatic TRAP plotting of the HIST history file if the file is local. This option must be enabled before a DISPLAY command is encountered.

!FILL

Prevents color-filling the full page background for an NBOX color plot. That is, the plot area is left untouched by the color shading and only its border is shaded. This feature is used for plotting curves on color film for any color other than white.

NBOX

Disables the legend box mode, extends the plot area over the previous legend box area, and disables the curve marker mode. Multiple curves are characterized in the following manner:

SOLID DASH, CHNDSH, and CHNDOT.

(The patterns will repeat in the same order if more than four).

Refer to the INCLUDE option for selecting curve colors in auxiliary files. Refer to the FLEGEND command if a legend and curve markers are desired in this mode.

NLOGO

Suppresses the "LOS ALAMOS" logo on plots when FORMAL is enabled.

NOADV

Suppresses the automatic frame advance for the current plot. Multiple curves per plot may be drawn by fixing an independent variable with multiple values.

NOTE: The rod radial values are the only exception. Because of the complex and unique structure of the rod component, only one rod radial value may be plotted per request. However, the WRAUX feature may be used to create plots for different rod radial values which may then be merged into single plots using the MRGAUX feature.

NPLOT

Suppresses all but INCLUDE-defined curve(s) for the current plot. This option is used for plotting auxiliary file data independently.

NOTE: Because any existing computed data will be overwritten by the auxiliary file cleared, computed data must be redefined. The maximum number of points allowed in this mode is approximately 1165.

NQUANT {N}

Defines N quantized shades and/or colors for the SLICE and USRDRW options. For color, N must be positive, and the next line must have N GPALET colors (ranging from 1-60). If B/W is desired, N may be negative, in which case no further input is expected. Also see the QREV and TABLE commands.

NOTE: Consult the writeup on GPALET (Ref. II-4) for the available colors.

NUMBER

Writes a small sequential plot number in the lower left-hand corner of the plot. (Also see the FNUM option).

OPAUX SINK [OPERATION-OPERAND] KRV TYP KOLOR

Where:

OPAUX or OP

The aux command recognized by TRAPI.

SINK

The output aux file name. Can be an existing local file name or one will be created with the default number of time points (5000).

[OPERATION-OPERAND]

The mathematical operation that is to be performed on the aux file "SINK." It can have either of the following two forms.

1. It can be any of the following stand-alone operators, which will operate on everything to the left of it:

SMOOTH - The arithmetic average of the data points. The next line must be the begin time, end time, and the desired time for smoothing. The order is mandatory.

RECP - reciprocal of "SINK" (1/SINK)

INT - The point-wise cumulative integral

DER - The derivative of "SINK"

Function - where "function" can be any of the following intrinsic functions: ABS, AINT, ALOG, ALOG10, SIN, COS, EXP, TANH, SINH, COSH, ATANH, SQRT, ATAN, ACOS, ASIN, TAN.

2. It can also have the "operation-operand" form where "operation" can be any one of the following operations: +, -, /, *, **, SET. The operand can be either an integer or real constant or the name of a local AUX FILE.

KRV TYP

The number of the curve type desired: 0 = Solid, 1 = Dash, etc.

KOLOR

The name of the color desired: WHITE, YELLOW, etc.

Example:

$\{[(1.0 - \text{ALPHA}) * \text{ROHL} * 9.801]/2\} + \text{PRESS1} - \text{PRESS2}$

can be expressed as

OP SINK SET $1.0 - \text{ALPHA} * \text{ROHL} * 9.801/2 + \text{PRESS1} - \text{PRESS2}$ 0 WHITE .

SINK, ALPHA, ROHL, PRESS1, PRESS2 = Existing local aux files.

1.0, 9.801, 2 = Real and integer constants. Values like 1.0e06 are also permitted

SET, -, +, *, / = Operators.

The only requirement is that the length of the constant not exceed the number of characters per machine word (8 for a Cray). The "SET" operator resets everything to the left of it to the value on its right. In this case, it resets the data contents of the existing aux file SINK to the value of 1.0. If the file SINK did not already exist, it would have been created with 1165 points. This could result in the SINK and ALPHA files being incompatible as the result of a difference in the number of points.

All of the AUX files used in a given OPAUX command line must be generated with an equal number of points or values.

ORIENT ANG

ORIENT is used in the TRAP GIVE mode. ANG is the orientation of all plots (in degrees) relative to the medium line of direction. Default is 0 degrees.

NOTE: There is no abbreviated form for this command.

OVERRIDE {OP} HGT

Overrides the title, X- or Y-axis, legend, character style, etc. "OP" may be: T1, T11, T12, LX, LY, LZ, LD, LE, DE, SI, SC, DU, UNIT, HGT, or LENGTH. T1, T11, T12, LX, LY, LZ, LD, and LE refer to: Title (3 types), X-label, Y-label, Z-label, dual-axis label, and legend respectively. (A text line must follow each of these commands.) For a two-line title, use OV T1 for each line. Also, T11 or T12 will replace any existing first or second title line, respectively. DE, SI, SC, and DU refer to character styles: DEFAULT, SIMPLX, SCMPLEX, and DUPLX respectively. Special characters such as mathematics symbols are not allowed with the DUPLX character style. If special characters are desired while retaining DUPLX as the background character style, first invoke the SIMPLX (or SCMPLEX) character style: OV SI (or SC), then follow with OV DU. [Refer to the DISSPLA (Ref. II-7) manual for examples of the various character styles and symbols.] HGT is the character height of the override. (Blank or 0.0 defaults to the TRAP-defined height, which varies for the different labels.) Typical values are 0.15-0.22 inches.

UNIT and LENGTH are only used for the USRDRW command, and set the unit reference vector value and the maximum length of the individual cell vectors. Vectors which would exceed the specified length stay at that length, but become dashed to indicate that their true lengths are more. In these two cases, the HGT value becomes the unit vector and physical length respectively.

NOTE: TRAP will interpret a dollar sign in column one as a blank line. Also, FORMAL is enabled by this command, if not already enabled.

PAGE {format}

PAGE is used in the TRAP GIVE mode, for PESP (Ref. II-6) only, for defining the page size. Format options are:

STANDARD . . . 2-in. space between images

ABUT no space between images

NOTE: There is no abbreviated form for this command.

PAUSE (a one-word message may be associated with any pause)

Interrupts TRAP processing, prints PAUSE (message), and waits for the user to press RETURN. When RETURN is pressed, processing resumes with the next command. PAUSE may be used to temporarily interrupt TRAP, especially when auxiliary files are being created, to hand-modify or check these files. The one-word message may not contain embedded blanks.

PORT {X₁ X₂ Y₁ Y₂}

Allows plotting in constrained areas of the current frame. The numbers X₁, X₂, Y₁, and Y₂ are floating-point integers, where X₁ = lower X-value of window, X₂ = upper X-value, Y₁ = lower Y-value, and Y₂ = upper Y-value. All values are in the closed interval [0,1]. For example:

PORT 0.0 0.5 0.5 1.0 or PORT 0.5 1.0 0.5 1.0

cause the plot to be displayed in the upper left quadrant or the upper right quadrant, respectively, of the plane.

QREV

Reverses order of the quantized level shading/colors for the SLICE and USRDRW options. For example, for B/W plotting, the mass flow-rate dependent variable would look more realistic if the low-to-high rates were shaded dark-to-light. Also, reversing the shading for the vapor-fraction-dependent variable gives a more realistic plot. That is, water would have dark shading, and steam would have light shading. Also see the NQUANT and TABLE options.

REVIEW

Prints out all commands and abbreviations recognized by TRAP.

ROD {N}

Sets up internal logic for vessel/core rod structure for retrieval of rod number N. Command must be enabled prior to any RA, TH, or Z definitions. Also see the description of the formats for FIXED Z and FREE Z, when the ROD command is enabled. Rod referenced in meters from the bottom of the vessel or core.

NOTE: Only one radial node may be referenced in any one plot command. See the NOADV command for information on multiple curves per plot.

SCALE DV AMIN AMAX (See argument description below).

Forces the scale for the dependent variable DV with AMIN and AMAX as the absolute minimum and maximum scale values for all axes. The user is cautioned that the data are not clipped using this option. If clipping is desired, refer to the CLIP command. If DV = X, then the horizontal axis is scaled. If DV = Y, then the vertical axis is scaled. **WARNING:** both axes must be scaled if either is to be scaled. The user can usually get by with scaling only one axis if a previous plot has been made, in which case the unscaled axis uses the previous scale. If DV = ABS, and AMIN and AMAX are null, then the scaling assumes the minimum and maximum of the previous plot. (There is no default; a previous plot must have been made).

NOTE: The DEP command must be enabled prior to the SCALE command if a dependent variable name is to be used in the command. The SCALE command is automatically cleared when the DEP command is encountered.

SIDE

Sets up internal logic for handling side data for the steam generator and tee components.

SLAB

Sets up logic for slab geometry for the vessel vertical SLICE option. SLAB is initiated as follows:

Enter the SLAB command.
Set the plot up for a vertical SLICE,
but enter only one THETA value.
(TRAP permits only one THETA value for the SLAB option, but
requires two THETA values for the SLICE option.)

NOTE: The abbreviated command for SLAB is SB, not SL.

SLICE

Displays a horizontal or vertical section of the three-dimensional vessel, shaded or colored by a quantized table of dependent variable values (default is five levels). Both horizontal and vertical slice displays are enabled by this command. Also see the NQUANT, QREV, SLAB, and TABLE options, which pertain to this command. Horizontal and vertical are distinguished by selecting the following variables:

Horizontal:
FIXED TIME T
FIXED Z N
FREE RA M1 M2
FREE TH N1 N2
SLICE

Vertical:
FIXED TIME T

FIXED TH N1 N2
FREE RA N1 N2
FREE Z N1 N2
SLICE

SMOOTH

Applies a spline fit to the current curve. (SMOOTH also applies to auxiliary file curves.) The curve is, specifically, a parametric cubic spline developed originally by Ferguson (Ref. II-8) and implemented by ISSCO for DISSPLA (Ref. II-7).

SUMMARY

Prints out all commands enabled, parameter settings, limits, etc.

NOTE: X and Y are in page units (inches).

TABLE

Defines the shading of the N quantized levels (not necessarily linear) for the SLICE and USRDRW options. The following line must be N+1 boundaries in percentages (that is, 0.0-100.0). References must be with respect to the scaled minimum (0.0%) to the scaled maximum (100.0%) of the dependent variable. Also see the NQUANT and QREV options.

THICKLIN {N}

Causes TRAP to draw curves N thicknesses wide (instead of 1). This feature is useful for making color plots on a colored background. For most purposes, a thickness of two or three is adequate. If N is four or more, sharp peaks become severely exaggerated.

NOTE: The abbreviation for THICKLIN is TK, not TH.

TRANS YMULT YSHIFT

Sets up a secondary axis scale (like the DUALAXIS option), but this scale is the only one plotted. That is, the SI scale is ignored, and the secondary scale is plotted in its place. YMULT and YSHIFT have the same meaning and application as in the DUALAXIS option. TRANS and DUALAXIS are incompatible options and TRAP will not allow both to be simultaneously enabled.

TSCALE {YLOW YHIGH}

Sets an absolute scale for the translated dependent variable. The low and high values must be in the same units as the translated variable. Thus, the user can scale his translated plots in the desired units without having to precalculate the SI scale (to use with the SCALE command) beforehand. Use this option only with TRANS, since the DUALAXIS scale is predetermined by the SI scale on the left side of the graph.

USRDRW (See argument description below).

Displays, graphically, a simultaneous color-quantized diagram of a system of connected components (such as a loop, tee and/or steam generator with both primary and secondary sides, etc.). This option is particularly useful when producing 16mm color movies.

Actually, the components need not be connected. The only way TRAP can determine the variable assignments for each cell is completely determined by the user; hence the name USRDRW. It is essential that the user define his cell sequence in exactly the same order that TRAP will extract the respective cells. This applies also to the location of blocks of cells pertaining to components.

To set up the plot, a set of cells from a component must be defined as free. For example:

FREE X 1 10

If the component is a vessel, one or two of the three variables (RA,TH,Z) may be defined as free. To understand the order in which TRAP will extract the variables, consider the variables as one single multi-indexed variable:

DEPV = V(RA,TH,Z)

(To remember the order of RA, TH, and Z, simply alphabetize them). Then, as in FORTRAN, the inside or leftmost variables vary faster than the variables to the right. For example, if either of the following are defined,

FR RA 1 5

FR Z 1 11

or

FR Z 1 11

FR RA 1 5

TRAP would extract the variables in the following order for V(RA,Z):

V(1.1), V(2.1), ..., V(5.1), V(1.2), V(2.2), ..., V(5.2), ..., V(5.11)

After setting up the component variable definitions for each component, but before defining the last component, the command

USRDRW (or US)

must be given to signal TRAP to create a corresponding component scratch file containing the dependent variable values which will be plotted. The above procedure must be used for all but the last component of the system. When the last component (or the first, if only one) is reached, the USRDRW command is given along with the name of the auxiliary file which contains the cell boundaries (that is,

vertexes) of the system in the proper order. For movie purposes, some additional options are available and are included.

```
USRDRW {file} VECTORS MOVAVG N MDEP M SHADES
```

Before explaining the additional options, note that MOVAVG N and MDEP M occur in pairs. In the above example, TRAP would perform a moving average over $2N$ movie frames and plot only the averaged "middle" frame.

For example, if N were 10, and the movie frames were normally produced at 20 frames per second, then each frame under the MOVAVG 10 mode would be plotted one per second (representing an average of 20×0.05 s each).

MDEP M is used only with the Extended Function output, and informs TRAP of the total number of output dependent variables to be used in the movie.

SHADES instructs TRAP to color-quantize the individual cells according to the same type of scheme as used in the SLICE options. SHADES and VECTORS may be simultaneously enabled.

The cell boundary file must have the following format:

```
NCELLS
NCELL1
X1 Y1 X2 Y2 X3 Y3 ... XNCELL1 YNCELL1
NCELL2
X1 Y1 X2 Y2 X3 Y3 ... XNCELL2 YNCELL2
.
.
.
NCELLNCELLS
X1 Y1 X2 Y2 X3 Y3 ... XNCELLNCELLS YNCELLNCELLS
```

where NCELLS is the total number of cells to be plotted (maximum 100); NCELL₁ is the number of vertexes (maximum 10) of the first cell; etc. The order of the cell boundaries should be clockwise, and the values should be floating-point values. The scale is arbitrary, since TRAP will normalize before plotting. Note that the cells need not be rectangular; a convex polygon will suffice. Using all of the 10 vertexes, cell boundaries which are rounded can be drawn.

USRDRW (See argument description below).

Displays, graphically, a simultaneous color-quantized diagram of a system of connected components (such as a loop, tee and/or steam generator with both primary and secondary sides, etc.). This option is particularly useful when producing 16mm color movies.

Actually, the components need not be connected. The only way TRAP can determine the variable assignments for each cell is completely determined by the user; hence the name USRDRW. It is essential that the user define his cell sequence in exactly the same order that TRAP will extract the respective cells. This applies also to the location of blocks of cells pertaining to components.

To set up the plot, a set of cells from a component must be defined as free. For example:

```
FREE X 1 10
```

If the component is a vessel, one or two of the three variables (RA,TH,Z) may be defined as free. To understand the order in which TRAP will extract the variables, consider the variables as one single multi-indexed variable:

$$DEPV = V(RA,TH,Z)$$

(To remember the order of RA, TH, and Z, simply alphabetize them). Then, as in FORTRAN, the inside or leftmost variables vary faster than the variables to the right. For example, if either of the following are defined,

```
FR RA 1 5
FR Z 1 11
or
FR Z 1 11
FR RA 1 5
```

TRAP would extract the variables in the following order for V(RA,Z):

$$V(1,1), V(2,1), \dots, V(5,1), V(1,2), V(2,2), \dots, V(5,2), \dots, V(5,11)$$

After setting up the component variable definitions for each component, but before defining the last component, the command

```
USRDRW (or US)
```

must be given to signal TRAP to create a corresponding component scratch file containing the dependent variable values which will be plotted. The above procedure must be used for all but the last component of the system. When the last component (or the first, if only one) is reached, the USRDRW command is given along with the name of the auxiliary file which contains the cell boundaries (that is,

VECTOR {X_f Y_f X_t Y_t TYPE} MODE

Causes a vector to be drawn from (X_f, Y_f) to (X_t, Y_t). X and Y are in page units (inches). MODE may be SOLID (default), DASH, DOT, CHNDSH, or CHNDOT.

TYPE=wxyz determines the type of vector:

w = ratio of width to length $0 \leq w \leq 5$

x = line thickness $0 \leq x \leq 6$

y = form:

0 → solid

1 → white

2 → open

3 → closed

z = location of arrowhead:

0 no arrowhead

1 at (X_t, Y_t)

2 at both ends

3 both ends pointing to (X_t, Y_t)

Up to 10 vectors can be defined by repeated VECTOR commands.

VIEW {X Y Z}

Defines the viewpoint for a perspective view of a surface. (X,Y,Z) is the viewpoint, where the workbox is considered a rectangular solid of dimension {(xmax - xmin), (ymax - ymin), (zmax - zmin)}.

WALL

Sets up internal logic for handling wall temperatures. In this option, "FI RA N" refers to the Nth wall node. The steam generator effective wall temperatures have only one node.

WRAUX {FNAME} FORMAT KVRTYP KOLOR NOMARK

Creates an on-the-spot auxiliary file "FNAME" from the data currently residing in the graphics plot buffers. If format="E15," the auxiliary file will be written in 2(1PE15.8) (extended) format. If format is omitted, or anything else, the auxiliary file will be written in 2(1PE10.3) format. KVRTYP = 0, 1, 2, 3, 4, or 5. (Refer to the INCLUDE command for more detail). KOLOR = WHITE, GREEN, etc. If the sixth entry is "NOMARK," the point-skip value of the auxiliary file is set to zero (causing no curve markers to be plotted, unless overridden by the INCLUDE command itself). If any optional parameter is desired, all preceding parameters must be included, even if meaningless (for example, "DUMMY") mnemonics must be used.

NOTE: The auxiliary file is created immediately upon receipt of the WRAUX command. DISPLAY does not affect this option.

XTRANS {XMULT XSHIFT}

Performs an analogous operation to TRANS on the X-axis. The user, for example, can shift the origin of a vessel core to 0.0 m (instead of the height above the component bottom), by entering a multiplication factor of 1.0 and a linear shift factor of the (negative) height of the lower plenum.

3TITLE

Plots a third title line (text line must follow). This command sets up a small title line immediately above the plot area.

E. SLICE Options

The user can plot a graphic display of the three-dimensional vessel component using the horizontal and vertical SLICE options. Refer to the USRDRW option for a similar type of graphics plot. For example,

```
COMPID 42
FORMAL
BORDER
FIXED TIME 0.75
FIXED Z 9
FREE RA 1 5
FREE TH 1 8
SLICE
DEP TL
DISPLAY .
```

and

```
COMPID 42
FORMAL
BORDER
FIXED TH 1 4
FREE RA 1 5
FREE Z 1 11
SLICE
DEP TL
DISPLAY .
```

Set up SLICE plots (horizontal and vertical, respectively). The second plot displays two "wedges," namely those for TH = 1, and TH = 4. A slab model can be plotted using the SLICE option. For SLICE plots, TH must have two values defined, and for SLAB plots, TH must have only one value defined.

F. Extended Function Capability

A user may be interested in plotting dependent variables that are combinations of the variables on a COMPIj file. He wishes to transform a given set of dependent variable functions (by a subroutine EXTFNC, supplied by the user) to another set

EXTFNC: D_1, \dots, D_k E_1, \dots, E_m .

where the domains for the functions D_i , E_j are given in the plot request.

For example, consider a vessel and the following set of TRAP commands (EXTEND is a flag for the extended function option)

```
COM 42
FI TH 1
FI RA 1
FI Z 1
FR TI 0. 1. .1
EXTEND
DEP ALPHA P TL
DI
EN .
```

In this example, the actual dependent variable appearing in the plot is

$$\text{NEWVAR} = \text{ALPHA}^2 \cdot \text{P} / \text{TL} \cdot \text{R}_1$$

and the plot would involve NEWVAR vs TIME at the (1,1,1)th TH, R, Z cell. The function domains are all identical as specified in the plot request. Additionally, the user needs geometric information because the first radial width, R_1 , occurs in the equation above.

The command language statements that enable the extended function option are:

```
EXTEND N
DEP  $D_1$   $D_2$  ...  $D_k$  .
```

where each D_i is a dependent variable name and is defined over the same spatial and time domains given in the plot request. "N" is a flag available to EXTFNC as an internal switch for equation selection. Subroutine EXTFNC is a subroutine that transforms a given set of dependent variables to another by special mathematical and logical relations of interest to the user. It is written by him and recompiled into TRAP. EXTFNC is enabled by TRAP repeatedly each time required.

The calling sequence for EXTFNC is:

```
CALL EXTFNC (LENX, DX, LENZ, DZ, LENR, DR,
LENTH, DTH, VOL, FA, MDEP,
DEPIN1, DEPIN2, DEPIN3, DEPIN4, DEPIN5,
DPOUT1, DPOUT2, LABL1, LABL2, NAM1, NAM2, ICODE) .
```

where

LENX = Length of array given in plot request
DX = Array of values, length LENX
LENZ = length of array given in plot request
DZ = array of values, length LENZ
LENR = length of array given in plot request
DR = array of values, length LENR
LENTH = length of array given in plot request

DTH = array of values, length LENTH
 VOL = array of volumes, 1D or 3D component
 FA = array of flow areas, 1D (LENX+1) or 3D component
 MDEP = number of incoming dependent variables on input;
 number of outgoing dependent variables on output
 DEPIN1 = first input dependent variable array
 DPOUT1 = first dependent variable output array
 LABL1 = axis label for DPOUT1, 2 words: output
 NAM1 = DPOUT1 mnemonic: output
 ICODE = input flag set by user in EXTEND command, which is
 available for use by subroutine EXTFNC.

Five incoming dependent variables can be used along with any or all of the available DEPIN arrays, depending on how the DEP command is entered in the plot request. Also, two dependent variable output arrays exist.

The structure of the dependent variable data arrays depends on component type and on the nature of the plot request, specifically the independent variables definition. If the component is one-dimensional, then the values in the dependent variable arrays are ordered for increasing cell number. If the component is three-dimensional, then the dependent variable values are ordered first for increasing θ -cell value, then for increasing R-cell value, and finally for increasing Z-cell value. The dependent variable values in the DEPIN arrays exist only over the spatial independent variable ranges, as given in the plot request.

For example, if the component is a pipe, the plot request is

```

    TITLE
    COM 7
    FI X 4 6
    FR TI 0. 1. .1
    EX
    DE TL
    DI
    EN
  
```

The dependent variables are ordered as follows

```

    DEPIN1(1) - TL at 4th cell
    DEPIN1(2) - TL at 5th cell
    DEPIN1(3) - TL at 6th cell
  
```

If the component is a vessel, the plot request consists of the following command statements:

```

    TITLE
    COM 42
    FI RA 3 4
    FI TH 1 7
    FI Z 4 5
    FR TI 0. 1. .1
    EX
  
```


DE TL
DI
EN .

The dependent variables are ordered as follows:

DEPIN1(1) - TL at TH, R, Z cell (1, 3, 4)

DEPIN1(7) - TL at TH, R, Z cell (7, 3, 4)

DEPINL(8) - TL at TH, R, Z cell (1, 4, 4)

DEPIN1(14) - TL at TH, R, Z cell (7, 4, 4)

DEPIN1(15) - TL at TH, R, Z cell (1, 3, 5)

DEPIN1(21) - TL at TH, R, Z cell (7, 3, 5)

DEPIN1(22) - TL at TH, R, Z cell (1, 4, 5)

DEPIN1(28) - TL at TH, R, Z cell (7, 4, 5)

TRAP performs all SCM/LCM data transfers and manages the time edit blocks as they are given to EXTFNC. TRAP enacts the EXTFNC subroutine after it interpolates dependent variable data at time values supplied by the user. TRAP also checks for full SCM/LCM buffers as well as for the movie option. This minimizes the programming involvement for the user. An example of the subroutine EXTFNC follows. Let

$$D = (D_1/D_2) \cdot V,$$

where

D_1 = first dependent variable.

D_2 = second dependent variable.

V = cell volume, and

D = transformed dependent variable.

Specifically, we are considering a vessel and are interested in the case in which

D_1 = pressure and

D_2 = liquid temperature.

Our command set to TRAP is

```
COM 42
FI RA 1
FI TH 1
FI Z 1
FR TI 0. 1.
EX
DE P TL
DI
EN
```

This produces a plot of (P/TL)·VOL vs TIME at the (1.1.1)th vessel cell. The EXTFNC listing for this simple example is given in Table II-1. TRAP must be at least partially compiled for this option, and if any common blocks in CLIFILE)* are modified, then a full compilation is mandatory.

To generate a partial compilation using a new EXTFNC (extended function) subroutine, first compile the new EXTFNC subroutine separately to produce a relocatable binary file. Then, in the LDR (loader) utility, use the relocatable binary file from the new subroutine and the relocatable binary file from a prior compilation of TRAP as input to the loader by the BIN directive.

```
LDR BIN = (NEWSUB, TRAPBIN) . . . , etc.
```

The order of the binary files is important if a subroutine exists in more than one binary file. If that is the case, then the first one encountered is the one that gets loaded.

G. Superposition of Auxiliary Data

1. The INCLUDE operation co-plots data in a auxiliary file with data supplied by the TRAC computer program, or the auxiliary file can be plotted by itself using the NPLOT option. Scaling is automatic, unless specified by the SCALE command. The user is responsible for consistency

The format for the auxiliary file is (mandatory parameters in brackets)

```
{I} J T
{K L} M N
{WORDS}
{X1 Y1} W1
```

```
{Xm Ym} Wm
```

where the entries per line are defined as follows.

* CLIFILE is the common blocks definition file for TRAC.

TABLE II-1
LISTING OF EXTFNC

```

1      SUBROUTINE EXTFNC(LENX,DX,LENZ,DZ,LENR,DR,
2      $ LENTH,DTH,VOL,FA,MDEP,
3      $ DEPIN1,DEPIN2,DEPIN3,
4      $ DEPIN4,DEPIN5,DPOUT1,DPOUT2,LABL1,LABL2,
5      $ NAM1,NAM2,ICODE)
6 C
7 C   DUMMY PROGRAM-USER SUPPLIES HIS OWN IN CASE
8 C   THE EXTENDED FUNCTION OPTION IS USED
9 C
10 C  * NOTE: THIS SUBROUTINE IS CALLED FROM OVERLAY 2.0 (2C0). *
11 *CALL.COMMN0
12 *CALL.COMMN1
13 *CALL.COMMN7
14 *CALL.COMMN13
15 C      DEPNEW = DEPIN1/DEPIN2 * VOL
16 C
17 C      USED IN CONTEXT OF EXTENDED FUNCTION CAPABILITY
18 C
19 C      THIS IS TEST VERSION ONLY
20 C
21 C
22      REAL DX(1), DZ(1), DR(1), DTH(1), FA(1),
23      $ DEPIN1 (LENTH,LENR,LENZ), DEPIN2 (LENTH,LENR,LENZ),
24      $ DEPIN3 (LENTH, LENR, LENZ), DEPIN4, (LENTH,LENR,LENZ),
25      $ DEPIN5 (LENTH,LENR,LENZ), DPOUT1 (LENTH,LENR,LENZ),
26      $ DPOUT2 (LENTH,LENR,LENZ), VOL (LENTH,LENR,LENZ)
27      INTEGER LABL1(2), LABL2(2)
28 C
29 C***TRANSFORM VARIABLES
30      DO 10 K=1,LENZ
31      DO 10 J=1, LENR
32      DO 10 I=1, LENTH
33      DPOUT1(I,J,K) = DEPIN1(I,J,K)/DEPIN2(I,J,K)*VOL(I,J,K)
34      10 CONTINUE
35 C***DEFINE LABELS & MNEMONICS
36      LABL1(1) = 9H(P/T)*VOL
37      LABL1(2) = 1H
38      NAM1 = 7HP/T*VOL
39      MDEP = 1
40      RETURN
41      END

```

- I = Number of curves in the file.
- J = Format: "E15" implies 2(E15.8) format. Blank (or anything else) implies 2(E10.3) format.
- T = Translate Flag: if T = "TRANS", then the current auxiliary file will not be translated if TRAP is plotting in the TRANS mode.
- K = Number of points in the curve.
- L = Point skip indicator: the program plots error bars at every L-th point. If L = 0, no curve markers or error bars are plotted.
- M = Curve type: 0 = solid, 1 = dash, 2 = dot, 3 = chaindash, 4 = chaindot, and 5 = no curve drawn between points.
- N = Color (white, green, yellow, etc.).

(NOTE: I, K, M are entered as space-delimited integers, J and N are ASCII or blank).

WORDS = Curve descriptor (2A10). (May be a blank line). JPLAX will give up to 15 characters to an auxfile.

(X,Y) = Rectangular coordinates of error bar centers.

W = Half width of error bar. (May be blank or zero).

X, Y, W are given in format 3(E10.3) [or 3(E15.8) if J=E15].

2. Auxiliary files can be created on the spot operated on, and displayed in a single session.

For example,

```

COM 42
FIXED RA 4
FIXED THETA 2
FIXED Z 9
FREE TIME 75
DEP P
WRAUX LEV9 E15

```

would create an auxiliary file "LEV9" containing 75 time-points (in the AUXFILE format), with extended-accuracy 2(E15.8) format.

The user might continue with

```

.
.
FIXED Z 8
WRAUX LEV8 E15

```

to create a second auxiliary file "LEV8." These two files may then be subtracted (or some other linear operation) to form a third:

```

.
.
OPAUX LEV9M8 + LEV9 - LEV8 0 WHITE .

```

Here, "LEV9M8" represents the pressure differential between level 9 and level 8, for the above defined values of RA, TH, and Z. Finally, the resulting pressure differential could be plotted by itself.

```
INCLUDE LEV9M8
DISPLAY
```

Note that NPLOT was enabled to exclude any currently defined data in the plot buffers.

H. TRAC Dependent Variable Names and Descriptions

Following a list of all dependent variable names and descriptions as found in the LBLIST file. Each description refers to the (primary) axis unit (SI). To illustrate the techniques and symbols required to invoke the DISSPLA upper case and lower case, special symbols, and different alphabets, the descriptions are printed just as they are stored in file LBLIST. The user can construct his own descriptions using this list as a guide.

ALPHA	V(OID) F(RACTION)
ALPHAP	P(UMP) V(OID) F(RACTION)
ALREAC	V(OID)-F(RACTION)R(EACTIVITY)
ALV	H(EAT) T(RANSFER)C(OEFFICIENT)ALV (W/K)
AM	A(IR) M(ASS){(KG)}
AREA	V(ALVE) A(REA){(M^2)}
AVRODTMP	R(OD) T(EMPERATURE)
BSA	I(NTEGRATED) A(IR) M(ASS) F(LOW){(KG)}
BXA	A(IR) M(ASS) F(LOW) {(KG/S)}
CB	C(ONTROL) B(LOCK)
CEMFR	C(ORE)-O(UTLET) M(ASS) F(LOW){(KG/S)}
CHTI	H(EAT) T(RANSFER) C(OEFFICIENT) CHTI (W/K)
CIMFRL	C(ORE)-I(NLET) L(IQUID) M(ASS) F(LOW) {(KG/S)}
CIMFRV	C(ORE)-I(NLET) V(APOR) M(ASS) F(LOW) {(KG/S)}
CMASS	C(ORE) L(IQUID) M(ASS) {(KG/S)}
CMFLOW	C(ORE)-I(NLET) M(ASS) F(LOW) {(KG/S)}
COMFRL	C(ORE)-O(UTLET) L(IQUID) M(ASS) F(LOW) {(KG/S)}
COMFRV	C(ORE)-O(UTLET) V(APOR) M(ASS) F(LOW) {(KG/S)}
CONC	S(OLUTE) C(ONCENTRATION){(KG/KG LIQUID)}
CPOW	H(EATER) P(OWER) (W)
CPUTOT	T(OTAL) CPU T(IME) {(S)}
CRLIQFR	C(ORE) L(IQUID) V(OLUME) F(RACTION)
CRPRESS	C(ORE) A(VERAGE) P(RESSURE) (P/A)
DBREAC	D(ISSOLVED)-B(ORON) R(EACTIVITY)
DCFLOW	D(OWNCOMER) M(ASS) F(LOW){(KG/S)}
DCLQVOL	D(OWNCOMER) L(IQUID) V(OLUME) F(RACTION)
DELT	T(IME)-S(TEP) S(IZE) {(S)}

DELTAP P(UMP) :M6)D:MX)P (P(A))
 DMASS D(OWNCOMER) L(IQUID) M(ASS) ((KG))
 DPRMAX M(AXIMUM) F(RACTIONAL) P(RESSURE) C(HANGE)
 DR :M6)D:MX)R ((M))
 DTHETA :M6)D:MX)-THETA ((RAD))
 DTLMAX M(AXIMUM) L(IQUID) -T(EMPERATURE) C(HANGE) (K)
 DTRMAX M(AXIMUM) R(OD)-T(EMPERATURE) C(HANGE) (K)
 DTSMAX M(AXIMUM) S(LAB)-T(EMPERATURE) C(HANGE) (K)
 DTVMAX M(AXIMUM) V(APOR)-T(EMPERATURE) C(HANGE) (K)
 DX :M6)D:MX)X ((M))
 DZ :M6)D:MX)Z ((M))
 EFFSTG S(TAGE) E(FFICIENCY)
 FA F(LOW) A(REA) ((M):Z2))
 FLOWAREA F(LOW) A(REA) ((M):Z2))
 FRICIZ A(XIAL) I(NTERFACIAL) S(HEAR) C(OEFFICIENT)
 HEAD F(UMP) H(EAD) ((M):Z2)/(S):Z2))
 HEIGHT W(ATER) L(LEVEL) ((M))
 HILG I(NSIDE) L(IQUID) H(EAT)-T(RANSFER) C(OEFFICIENT) (W/(M):Z24)K)
 HIVG I(NSIDE) V(APOR) H(EAT)-T(RANSFER) C(OEFFICIENT) (W/(M):Z24)K)
 HL L(IQUID) H(EAT)-T(RANSFER) C(OEFFICIENT) (W/(M):Z24)K)
 HLV I(NTERFACIAL) H(EAT)-T(RANSFER) C(OEFFICIENT) (W/K)
 HOLG O(UTSIDE) L(IQUID) H(EAT)-T(RANSFER) C(OEFFICIENT) (W/(M):Z24)K)
 HOVG O(UTSIDE) V(APOR) H(EAT)-T(RANSFER) C(OEFFICIENT) (W/(M):Z24)K)
 HTLSCI I(NSIDE) H(EAT) L(OSS) (W)
 HTLSCO O(UTSIDE) H(EAT) L(OSS) (W)
 HTLSGI S(TEAM)-G(ENERATOR) I(NSIDE) H(EAT) L(OSS) (W)
 HTLSGO S(TEAM)-G(ENERATOR) O(UTSIDE) H(EAT) L(OSS) (W)
 HV V(APOR) H(EAT)-T(RANSFER) C(OEFFICIENT) (W/(M):7274)K)
 ICJ S(OURCE) A(DJACENT) C(OMPONENT)
 IDGHI I(NSIDE) H(EAT)-T(RANSFER) R(EGIME)
 IDGHO O(UTSIDE) H(EAT)-T(RANSFER) R(EGIME)
 IDR H(EAT)-T(RANSFER) R(EGIME)
 IDRGR R(OD) H(EAT)-T(RANSFER) R(EGIME)
 IDRGS S(LAB) H(EAT)-T(RANSFER) R(EGIME)
 IHTF R(OD) H(EAT)-T(RANSFER) R(EGIME)
 IMFLOW I(NTEGRATED) M(ASS) F(LOW) ((KG))
 ISRC S(OURCE) C(ELL) N(UMBER)
 ISRF S(OURCE) F(ACE) N(UMBER)
 ISRL S(OURCE) L(LEVEL) N(UMBER)
 MFLOW M(ASS) F(LOW) ((KG/S))
 MFRL L(IQUID) A(XIAL) M(ASS) F(LOW) ((KG/S))
 MFRLR L(IQUID) R(ADIAL) M(ASS) F(LOW) ((KG/S))
 MFRLT L(IQUID) A(ZIMUTHAL) M(ASS) F(LOW) ((KG/S))
 MFRLZ L(IQUID) A(XIAL) M(ASS) F(LOW) ((KG/S))

MFRV	V(APOR) A(XIAL) M(ASS) F(LOW) ((KG/S))
MFRVR	V(APOR) R(ADIAL) M(ASS) F(LOW) ((KG/S))
MFRVT	V(APOR) A(ZIMUTHAL) M(ASS) F(LOW) ((KG/S))
MFRVZ	V(APOR) A(XIAL) M(ASS) F(LOW) ((KG/S))
NSRL	S(OURCES) (ON) L(LEVEL)
OMEGA	S(PPEED) ((RAD/S))
P	P(RESSURE) (P(A))
PA	A(IR) P(ARTIAL) P(RESSURE) (P(A))
PDC	D(OWNCOMER) A(VERAGE) P(RESSURE) (P(A))
PFLOW	P(UMP) M(ASS) F(LOW) ((KG/S))
PGRÉAC	P(ROGRAMMED) R(EACTIVITY)
PLP	L(OWER)-P(LENUM) A(VERAGE) P(RESSURE) (P(A))
PMASS	L(IQUID) M(ASS) B(ELOW) D(OWNCOMER) ((KG))
POWER	P(OWER) (W)
POWR1	H(EATER) P(OWER) (W)
POWR2	H(EATER) P(OWER) (W)
POWSTG	S(TAGE) P(OWER) O(UTPUT) (W)
PUP	U(PPER)-P(LENUM) A(VERAGE) P(RESSURE) (P(A))
QRD	R(OD) H(EAT) T(RANSFER) (W)
QRDTOT	T(OTAL) R(OD) H(EAT) T(RANSFER) (W)
QSL	S(LAB) H(EAT) T(RANSFER) (W)
QSLTOT	T(OTAL) S(LAB) H(EAT) T(RANSFER) (W)
RDAN	M(ACROSCOPIC) A(IR) D(ENSITY) ((KG/M):Z3)
RHOL	L(IQUID) D(ENSITY) ((KG/M):Z3)
RHOM	M(IXTURE) D(ENSITY) ((KG/M):Z3)
RHOP	P(UMP) M(IXTURE) D(ENSITY) ((KG/M):Z3)
RHOV	V(APOR) D(ENSITY) ((KG/M):Z3)
RMCKN	R(EACTOR) M(ULTIPLICATION) C(ONSTANT) (K)
ROAN	M(ACROSCOPIC) A(IR) D(ENSITY) ((KG/M):Z3)
RODTMP	R(OD) T(EMPERATURE) (K)
RPOWER	C(ORE) P(OWER) (W)
SMOM	P(UMP) M(OMENTUM) S(OURCE) ((M/S):Z2)
SOLID	P(LATED) S(SOLID)/C(ELL) V(OLUME) ((KG/M):Z3)
SV	S(IGNAL) V(ARIABLES)
TCILMF	I(NTEGRATED) C(ORE)-I(NLET) L(IQUID) M(ASS) F(LOW) ((KG))
TCIVMF	I(NTEGRATED) C(ORE)-I(NLET) V(APOR) M(ASS) F(LOW) ((KG))
TCOLMF	I(NTEGRATED) C(ORE)-O(UTLET) L(IQUID) M(ASS) F(LOW) ((KG))
TCORE	C(ORE) A(VERAGE) L(IQUID) T(EMPERATURE) (K)
TCOVMF	I(NTEGRATED) C(ORE)-O(UTLET) V(APOR) M(ASS) F(LOW) ((KG))
TCREAC	C(OOLANT)-T(EMPERATURE) R(EACTIVITY)
TDC	D(OWNCOMER) A(VERAGE) L(IQUID) T(EMPERATURE) (K)
TFREAC	F(UEL)-T(EMPERATURE) R(EACTIVITY)
TIMET	T(RANSIENT) T(IME) ((S))
TL	L(IQUID) T(EMPERATURE) (K)

TLP	L(OWER)-P(LENUM) A(VERAGE) L(IQUID) T(EMPERATURE) (K)
TNSTEP	T(OTAL) T(IME) S(TEPS)
TORQUE	P(UMP) T(ORQUE) ((KG):Z4(M):Z2)/(S):Z2)
TRAMAX	M(AXIMUM) A(VERAGE)-R(OD) T(EMPERATURE) (K;
TRBSIG	T(URBINE) G(OVERNING) S(IGNAL)
TRHMAX	M(AXIMUM) H(OT)-R(OD) T(EMPERATURE) (K)
TSAT	S(ATURATION) T(EMPERATURE) (K)
TSCORE	C(ORE) A(VERAGE) S(ATURATION) T(EMPERATURE) (K)
TSDC	D(OWNCOMER) A(VERAGE) S(ATURATION) T(EMPERATURE) (K)
TSLP	L(OWER)-P(LENUM) A(VERAGE) S(ATURATION) T(EMPERATURE) (K)
TSUP	U(PPER)-P(LENUM) A(VERAGE) S(ATURATION) T(EMPERATURE) (K)
TUP	U(PPER)-P(LENUM) A(VERAGE) L(IQUID) T(EMPERATURE) (K)
TV	V(APOR) T(EMPERATURE) (K)
TW	W(ALL) T(EMPERATURE) (K)
TWGI	E(FFFECTIVE) W(ALL) T(EMPERATURE) (K)
VFLOW	V(OLUMETRIC) F(LOW) ((M):Z3)/(S)
VFLOWP	P(UMP) V(OLUMETRIC) F(LOW) ((M):Z3)/(S)
VL	L(IQUID) V(ELOCITY) ((M/S))
VLN-R	L(IQUID) R(ADIAL) V(ELOCITY) ((M/S))
VLN-T	L(IQUID) A(ZIMUTHAL) V(ELOCITY) ((M/S))
VLN-Z	L(IQUID) A(XIAL) V(ELOCITY) ((M/S))
VLOSS	L(IQUID) V(OLUME) D(ISCHARGED) ((M):Z3)
VLPLIQ	L(OWER)-P(LFNUM) L(IQUID) F(RACTION)
VLPLM	L(OWER)-P(LENUM) L(IQUID) M(ASS)((KG))
VMASS	V(ESSEL) L(IQUID) M(ASS) ((KG))
VMFR	V(APOR) M(ASS) F(LOW) ((KG/S))
VOL	V(OLUME) ((M):Z3)
VSFLOW	V(ESSEL) M(ASS) F(LOW)((KG/S))
VUPLIQ	U(PPER)-P(LENUM) L(IQUID) V(OLUME) F(RACTION)
VUPLM	U(PPER)-P(LENUM) L(IQUID) M(ASS) ((KG))
VV	V(APOR) V(ELOCITY) ((M/S))
VVN-R	V(APOR) R(ADIAL) V(ELOCITY) ((M/S))
VVN-T	V(APOR) A(ZIMUTHAL) V(ELOCITY) ((M/S))
VVN-Z	V(APOR) A(XIAL) V(ELOCITY) ((M/S))
ZHT	A(XIAL) P(OSITION) ((M))

I. Programming Guidelines

The TRAP code was developed and is currently maintained at Los Alamos with a maintenance utility called HISTORIAN, which is proprietary software of OPCODE, Inc. The selection of HISTORIAN as our maintenance code provides us with the capability to be upwardly compatible with the CDC UPDATE utility and with the flexibility to transmit the program library to other installations which have HISTORIAN and still preserve the history of the corrections.

The TRAP code has been structured with some degree of portability by the use of conditional directives (*IF DEFs) placed in the code's program library. Here at Los Alamos

the TRAP code has been run on a CDC 7600 using the LTSS operating system and on a CRAY machine which uses the CTSS operating system. The respective compiler-ready FORTRAN source for the systems mentioned above can be extracted from the program library using the following UPDATE/HISTORIAN *DEFIN : options.

CRAY	CDC-7600
CRAY	L7600
EIGHTB	VDM
NOLCM	TENB
VDM	OVLAY

Other *DEFINE options which exist in the program library are IBM, S7600, HEX, and VAX. An external user would have to supplement the list or create a combination of *DEFINE options that are appropriate for his system. For a description of the *DEFINE options, please see your TRAC manual.

Any external organization which has considered installing TRAP on their system must make sure that the DSSPLA graphics software, which is proprietary software of Integrated Software Systems Corporation, exists on their system. TRAP is very dependent on this graphics package and currently uses version 9.2 with updates for version 10 available for users who have the latest version.

The calling sequence for the TRAP code and its associated subroutines has been included in Fig. II-2.

J. Examples

In this section we present examples of code execution, various code options, and input commands.

1. Examples of Code Execution. Plot liquid temperature vs cell position for component 16. Here, this component is a pressurizer and is a one-dimensional model. First generate the component data file, COMP16, by running EXCON on TRCGRF (upper-case entries are supplied by the user):

```
EXCON GO / 1 1.2
Enter Space Delimited Component Numbers
16
Input accepted. Now Running Excon . . .
```

Component Comp16 Finished: .100000E-04 To .101080E+01 With 38 Time-Points.

All Done .

1	trap	calls fillup file setdev setlcm chkln defdat	qassign exit lrgbuf second close wfile	noprmt golay gwidth tming pltdat dfclos	qgetsym diolay gstart probdf assign tming	ljb libdisp getufi givep bfaloc
2	defdat	calls pblock called by trap	blockq	tsrch	movlev	
3	pltdat	calls movlev gplbeg mac1d case04 case10 case18 close called by trap	adv gtitle movcap case02 case11 case19	gdone gplend lims case09 case12 fnlplt	gtebeg graf2d case03 case08 case14 endpl	gteend case20 case01 case21 case16 chkln
4	probdf	calls rset dep includ fexist help1 called by trap	idl spaux	idky summry fillup z2b mrgaux	compid fof dict skmpr review	clr ljb pltinf override ljust
5	adv	calls resct called by pltdat	no: rdr	title	endpl	
6	auxil	calls decod thkcrv chndsh dofr called by graf2d	ljb curve chndot messag	height marker psplit skmpr	klip dash reset lims	solor dot grace
7	backgd	calls solor called by graf2d	ffill			
8	box	called by hslice	usrdrw	vslice		
9	case02	calls fillup called by pltdat	detval	movlev	setlbl	grf3d
10	case03	calls fillup graf2d called by pltdat	movlev	detval	srch	bfin
11	case04	calls fillup called by pltdat	movlev	setlbl	graf2d	srch
12	case04	calls fillup called by pltdat	assign	usrdrw	srch	close

Fig. II-2.
Program subroutine function entry.

13	case08	calls detval grf3d called by pltdat	bfin	movlev	fillup	setlbl
14	case09	calls detval called by pltdat	bfin	movlev	hslice	
15	case10	calls fillup grf3d called by pltdat	detval	setlbl	bfin	movlev
16	case11	calls fillup close called by pltdat	setlbl	assign	movlev	usrdrw
17	case12	calls fillup setlbl called by pltdat	detval graf2d	srch	bfin	movlev
18	case14	calls fillup grf3d called by pltdat	detval	setlbl	bfin	movlev
19	case16	calls fillup setlbl called by pltdat	detval graf2d	srch	bfin	movlev
20	case18	calls fillup setlbl called by pltdat	detval graf2d	srch	bfin	movlev
21	case19	calls fillup called by pltdat	movlev	setlbl	graf2d	
22	case20	calls fexist height close called by pltdat	assign messag	setup rjust	lib z2b	nchar fillup
23	case21	calls detval called by pltdat	movlev	bfin	vslice	error
24	dua'	calls reset dofr called by fnlplt	height angle graf2d	yintax messag	ygraxs	ynonum
25	ffill	calls fill called by backgd	setup	usrdrw	vslice	

Fig. II-2 (cont).

26	fnlplt	calls nchar tlogo called by pltdat	height	messag	dual	plotno
27	fonts	calls reset mx2alf z3use called by graf2d	scmplx mx3alf z4use	simplx zuse	duplx z1use	mx1alf z2use
28	graf2d	setup calls fillup waux nibrdr messag ynonum backgd intno psplin curve endpl called by pltdat case18	grf3d usrdrw dolr reset int:xs angle xgraxs scolor quadfr dash realno	hslice vslice nchar physor title graf ygraxs gridd frame chndsh dual	mac1d lookup yaxang headr grfacs grace labl klip chndot plotno	movcap auxil fonts height xnonum nocek vector marker thkcrv tlogo
29	grf3d	calls dolr headr graf3d scolor tlogo called by case01	case02 case19 lookup physor nocek bckgnd endpl	case03 noibrdr gridd height	case12 axes3d surtrn messag	case16 title view surmat plotno
30	grfacs	calls ynonum rintax rreal called by graf2d	case08 ynonum height rmess	case10 graf xrl	case14 reset mshift	yintax rlint
31	gridd	calls grid height called by graf2d	dash messag	strtp vector	connpt	reset
32	headr	calls height called by graf2d	grf3d nchar	hslice headin	usrdrw vslice	
33	hslice	calls intens fonts scolor circle gpale box plotno called by case09	grf3d movlev nocek gridd curve share dolr tlogo	mac1d reset title labl bounds intno nchar endpl	physor grace height sector messag fillup	nibrdr graph thkcrv blnk1 xrl lookup
34	lr	calls nchar curve called by graf2d	dolr dot reset hslice	height chndsh messag usrdrw	angle chndot vslice	scolor thkcrv

Fig. II-2 (cont).

35	mac1d	calls movlev dolr graph thkcrv called by pltdat	reset nchar ystaxs curve	nobrd headr scol plotno	fonts graf bckgnd tlogo	title ygraxs messag endpl
36	movcap	calls reset scolor called by pltdat	fonts height	physor messag	title setlbl	graf endpl
37	plotno	calls height called by fnlplt usrdrw	intno	messag		
38	quadfr	calls strtpt called by graf2d	graf2d vslice connpt	grf3d	hslice	mac1d
39	setup	calls reset fonts tlogo called by case20	physor scolor endpl	nochek ffill	title height	grace messag
40	tlogo	calls height called by fnlplt setup	messag	grf3d vslice	hslice	mac1d
41	usrdrw	calls intens reset grace height bounds realno box called by case04	assign physor title lbl gpalet setlbl ffill	close nobrd graf dash shade vector plotno	dolr fonts scolor rlvec numode intno tlogo	nchar nochek gridd curve messag xrl endpl
42	vslice	calls intens nochek graf thkcrv ffill vector lookup plotno called by case21	case11 reset dolr scolor box1 shade sector nur.ode tlogo	physor nchar gridd curve intno strtpt realno endpl	nobrd title height bounds messag connpt xrl	fonts grace lbl gpalet dash fillup box
43	xrl	called by grfax	hslice	usrdrw	vslice	
44	bfaloc	called by trap	compid			
45	bfclos	called by trap	compid			

Fig. II-2 (cont).

46	bfin	calls error called by case02 case14 dict	rdlcm	case08 case16 pblock	case09 case18 varhd	case10 case21	case12 datamv
47	hfout	calls error called by movlcm calls	wrlcm				
48	blockq	varhd called by defdat	geom	movlev			
49	bounds	called by hslice	usrdrw	vslice			
50	box1	called by vslice					
51	check	calls skpr called by probd					
52	circle	called by hslice					
53	clr	calls ljb called by probd	skpr	fillup	rset		
54	compid	calls chklun fexist bfaloc called by probd	bfclos skpr	close qassign	z2b getuff	ljust setlcm	
55	datamv	calls bfin called by tsrch					
56	decod	z2b called by auxil	idkey	mrgaux	opaux		
57	dep	calls ljb called by probd	skpr				
58	detval	calls movlev called by case01 case12 wfile	case02 case14	case08 case16	case09 case18	case10 case21	
59	dict	calls bfin called by probd	varhd	skpr			
60	dolr	calls nchar called by auxil labl	duel mac1d	graf2d usrdrw	grf3d vslice	hslice	

Fig. II-2 (cont).

61	efun	calls movlev called by movlcm	extfnc			
62	efunct	called by opaux				
63	error	calls exita called by case21	bfin	bfout	givep	
62	extfnc	called by efun				
65	file	calls ljb called by trap	fexist	qassign		
66	fillup	called by trap case04 case14 graf2d override	probd case08 case16 hslice rset	case01 case10 case18 vslice	case02 case11 case19 clr	case03 case12 case20 givep
67	fixglch					
68	fof	calls ljb called by probd	skipr	ivalue		
69	geom	calls unpak called by blockq	movlev	values		
70	givep	calls fillup called by trap	ljb	nchar	qrunit	error
71	help1	calls skipr close called by probd	fexist	ljb	lookup	z2b
72	idkey	calls decod called by probd				
73	idky	calls ljb called by probd	skipr			
74	includ	calls ljb qassign called by probd	fexist z2b	skipr skpau	chklun	close
75	inte:p	called by tsrch				
76	jvalue	called by fof				
77	klip	called by auxil	graf2d			

Fig. II-2 (cont).

78	lims	called by pltdat	auxii				
79	ljb	calls ljust called by trap dep idky scal	z2b				
			probd file includ	auxil fof mrgaux	case20 givep opaux	clr help1 override	
80	lookup	calls qassign called by graf2d	close	skpr			
81	ipack		grf3d	hslice	vslice	help1	
82	movicm	called by varhd calls movlev called by tsrch	efun	bfout	srch		
83	mrgaux	calls ljb decod called by probd	fexist	chklun	close	qassign	
84	nchar	called by case20 labl givep	fnlpt mac1d	graf2d usrdrw	headr vslice	hslice dolr	
85	opaux	calls fexist ljb called by probd	chklun skpr	close efunct	sign	decod	
86	override	calls ljb called by probd	skpr	fillup			
87	pblock	calls savstr called by defdat	bfin				
88	pltinf	called by probd					
89	rd'cm	called by bfin calls skpr					
90	review	called by probd calls fillup called by probd					
91	rset	calls fillup called by probd				clr	
92	savstr	called by pblock calls ljb called by probd					

Fig. II-2 (cont).

94	search	called by				
		tsrch				
95	sector	called by				
		hslice	vslice			
96	setlbl	called by				
		case01	case03	case08	case10	case11
		case12	case14	case16	case18	case19
		movcap	usrdrw			
97	setlcm	calls				
		memadj				
		called by				
		trap	compid			
98	skpaux	calls				
		skipr				
		called by				
		includ				
99	srch	called by				
		case02	case03	case04	case12	case16
		case18	movlcm			
100	suinmry	calls				
		skipr				
		called by				
		probd				
101	trng	calls				
		second	timing			
		called by				
		trap				
102	tsrch	calls	search	xtract	interp	movlcm
		datamv				
		called by				
		defdat				
103	unpak	calls				
		unpkit				
		called by				
		geom	xtract1	xtrct2	xtrct3	
104	unpkit	called by				
		unpak				
105	values	calls				
		movlev				
		called by				
		geom				
106	varhd	calls				
		bfin	lpack			
		called by				
		blockq	dict			
107	waux	called by				
		graf2d	wfile			
108	wfile	calls				
		movlev	detval	waux	wtrac2	wtrac
		called by				
		trap				
109	wrlcm	called by				
		bfout				
110	wtrac	called by				
		wfile				

Fig. II-2 (cont).

111	wtrac2	called by					
		wfile					
112	xtract	calls					
		xtrct1	xtrct2	xtrct3			
		called by					
113	xtrct1	tsrch					
		calls					
		unpak					
		called by					
114	xtrct2	xtract					
		calls					
		unpak					
		called by					
115	xtrct3	xtract					
		calls					
		unpak					
		called by					
116	z2b	xtract					
		called by					
		probd	case20	compid	decod	help1	
		includ	ljb				
117	y3dmat						
118	x3dmat						
		library	subroutines				
1	qassign						
		called by					
		trap	case20	compid	file	includ	
		lookup	mrgaux	opaux			
2	noprmt						
		called by					
		trap					
3	qgetsum						
		called by					
		trap					
4	exit						
		called by					
		trap					
5	golay						
		called by					
		trap					
6	diolay						
		called by					
		trap					
7	libdisp						
		called by					
		trap					
8	setdev						
		called by					
		trap					
9	lrgbuf						
		called by					
		trap					
10	gwidth						
		called by					
		trap					
11	gstart						
		called by					
		trap					
12	getufl						
		called by					
		trap					
13	second						
		called by	compid				
		trap					
		called by	tming				
		trap					

Fig. II-2 (cont).

14	timing	called by trap	tming			
15	chklun	called by trap opaux	pltdat	compid	includ	mrgaux
16	close	called by trap usrdrw mrgaux	pltdat compid opaux	case04 help1	case11 includ	case20 lookup
17	assign	called by trap	case04	case11	usrdrw	
18	movlev	called by defdat case08 case14 hslice geom	pltdat case09 case16 mac1d movic:n	case01 case10 case18 blockq values	case02 case11 case19 detval wfile	case03 case12 case21 efun
19	gdone	called by pltdat				
20	gtebeg	called by pltdat				
21	gteend	called by pltdat				
22	gplbeg	called by pltdat				
23	gtitle	called by pltdat				
24	gplend	called by pltdat				
25	endpl	called by pltdat mac1d	adv movcap	graf2d setup	grf3d usrdrw	hslice vslice
26	fexist	called by probdf includ	case20 mrgaux	compid opaux	file	help1
27	rjust	called by probdf	case20			
28	skipr	called by probdf dep includ skpaux	auxil dict lookup summry	check fof opaux	clr help1 override	compid idky review
29	ljust	called by probdf	compid	ljb		

Fig. II-2 (cont).

30	reset	called by adv grf3d mac1d	auxil grfaxes movcap	dual gridd setup	fonts hslice usrdrw	graf2d labl vslice
31	nobrd	called by adv usrdrw	graf2d vslice	grf3d	hslice	mac1d
32	title	called by adv movcap	graf2d setup	grf3d usrdrw	hslice vslice	mac1d
33	height	called by auxil grf3d labl usrdrw	case20 grfaxes movcap vslice	dual gridd plotno	fnlplt headr setup	graf2d hslice tlogo
34	scolor	called by auxil labl vslice	backgd mac1d	graf2d movcap	grf3d setup	hslice usrdrw
35	thkcrv	called by auxil vslice	graf2d	hslice	labl	mac1d
36	curve	called by auxil usrdrw	graf2d vslice	hslice	labl	mac1d
37	marker	called by auxil	graf2d			
38	dash	called by auxil vslice	graf2d	gridd	labl	usrdrw
39	dot	called by auxil	labl			
40	chndsh	called by auxil	graf2d	labl		
41	chndot	called by auxil	graf2d	labl		
42	psplin	called by auxil	graf2d			
43	grace	called by auxil vslice	graf2d	hslice	setup	usrdrw
44	messag	called by auxil grf3d movcap vslice	case20 gridd plotno	dual hslice setup	fnlplt labl tlogo	graf2d mac1d usrdrw
45	yintax	called by dual	grfaxes			

Fig. II-2 (cont).

46	ygraxs	called by dual	graf2d	mac1d		
47	ynonum	called by dual	graf2d	grfaxes		
48	angle	called by dual	graf2d	labl		
49	fill	called by fhill				
50	scmplx	called by fonts				
51	simplx	called by fonts				
52	duplx	called by fonts				
53	mx1alf	called by fonts				
54	mx2alf	called by fonts				
55	mx3alf	called by fonts				
56	zuse	called by fonts				
57	z1use	called by fonts				
58	z2use	called by fonts				
59	z3use	called by fonts				
60	z4use	called by fonts				
61	physor	called by graf2d usrdrw	grf3d vslice	hslice	movcap	setup
62	yaxang	called by graf2d				
63	intaxs	called by graf2d				
64	graf	called by graf2d vslice	grfaxes	mac1d	movcap	usrdrw
65	xnonum	called by graf2d	grfaxes			
66	xgraxs	called by graf2d				

Fig. II-2 (cont).

67	nochek	called by graf2d vslice	grf3d	hslice	setup	usrdrw
68	vector	called by graf2d	gridd	usrdrw	vslice	
69	intno	called by graf2d	hslice	plotno	usrdrw	vslice
70	frame	called by graf2d				
71	realno	called by graf2d	usrdrw	vslice		
72	axes3d	called by grf3d				
73	view	called by grf3d				
74	graf3d	called by grf3d				
75	surtrn	called by grf3d				
76	surmat	called by grf3d				
77	bckgnd	called by grf3d	mac1d			
78	xintax	called by grfax				
79	mshift	called by grfax				
80	rlint	called by grfax				
81	rlreal	called by grfax				
82	rlmess	called by grfax				
83	grid	called by gridd				
84	strtp	called by gridd				
85	connpt	called by gridd	quadfr	vslice		
86	headin	called by gridd	quadfr	vslice		
87	intens	called by headr				
		called by hslice	usrdrw	vslice		

Fig. II-2 (cont).

88	graph	called by		
		hslice	mac1d	
89	blink1	called by		
		hslice		
90	gpalet	called by		
		hslice	usrdrw	vslice
91	shade	called by		
		hslice	usrdrw	vslice
92	ystaxs	called by		
		mac1d		
93	rlvec	called by		
		usrdrw		
94	numode	called by		
		usrdrw	vslice	
95	exita	called by		
		error		
96	qunit	called by		
		givep		
97	memadj	called by		
		setlcm		

Fig. II-2 (cont).

Here, TRCGRF is the graphics file produced by program TRAC. Next run TRAP:

```

TRAP / 1 1.2
CTSS TRAP FOR INDEPENDENT VERSION.GT.12.0
enter output plot file label (max. 80 characters)
TEST CASE
Enter Problem Specification
COMPID 16
FIXED TIME .4
FREE X 1 5
DEP TL
DISPLAY
Plot No. 1 Completed.
END
All Done

```

We define the component ID number as 16; set TIME as the fixed independent variable at 0.4 s, set the cell position coordinate, X, as the free independent variable ranging from the first through the fifth cells; and define TL (liquid temperature) as the dependent variable.

Example.

Plot, for a three-dimensional component, a curve relating liquid temperature and axial position (the free independent variable). Let the component ID be 42 and assume the associated data on file COMP42 resides in the user's local file space. Let the values of the fixed independent variables be: 1st radial cell, 1st angular cell, $t = 0.5$ s. Also, for this example,

TABLE II-2
AUXFIL DATA

2		
811		
USPWR1		
1.	549.	1.
3.	551.	1.
4.	554.	2.
5.	558.	2.
6.	566.	2.
8.	580.	2.
10.	579.	2.
11.	571.	1.
7 1 4		
USPWR2		
1.?	550.	1.
3.5	552.	1.
4.5	556.	2.
5.5	562.	2.
6.5	570.	2.
8.5	580.	2.
10.5	577.	2.

superimpose data with error bars from two auxiliary data files. The information in this file is given in Table II-2. The interactive session is:

```

TRAP / 1 1.2
Enter General Title
TEST CASE
Enter Problem Specifications
COMPID 42
DEP TL
FIXED RADIUS 1
FIXED THETA 1
FIXED TIME .5
FREE Z 1 11
OV LE
TRAC DATA
INCLUDE AUXFIL
DISPLAY
Plot No. 1 Completed.
END
All Done .

```


The user sets the component identification (ID) at 42 (a vessel); fixes RADIUS and THETA as two of the independent variables (that is, he fixes both radius and theta at the first cell); fixes TIME at 0.5s, defines the free independent variable as Z (ranging from the first z-level through the eleventh z-level); defines the dependent variable mnemonic as TL (which is equivalent to liquid temperatures—these mnemonics are generated by TRAC and stored in the TRCGRF output file); and finally asks TRAP to co-plot all data found in the auxiliary file, AUXFIL. He finishes his plot problem definition with the DISPLAY command and terminates the program with the END statement.

For this example, if the user chooses not to be completely interactive, he prepares a file containing:

```
TEST CASE
COMPID 42
DEP TL
FIXED RADIUS 1
FIXED THETA 1
FIXED TIME .5
FREE Z 1 11
OV LE
TRAC DATA
INCLUDE AUXFIL
DISPLAY
END .
```

If we name this file TRAPIN, the terminal session is:

```
TRAP FILE TRAPIN / 1 1.2
```

```
Plot No. 1 Completed.
```

```
All Done .
```

Example.

This is the same as the first example, except that we also wish to plot a TL-vs-X curve for TIME = 0.7s. The session is:

```
TRAP / 1 1.2
CTSS TRAP FOR INDEPENDENT VERSION.GT.12.0
enter output plot file label (max. 80 characters)
TEST CASE
Enter Problem Specification
COMPID 16
FIXED TIME .4
FREE X 1 5
DEP TL
DISPLAY
Plot No. 1 Completed.
FIXED TIME .7
DISPLAY
Plot No. 2 Completed.
```

END
All Done

2. Examples of Option Use. Most of the following examples are based on a standard problem, USPWR1, which models a reactor system having 42 components.

Component number 42 is a VESSEL, where
NASX = number of axial levels = 11,
NRSX = number of radial rings = 5,
NTSX = number of sectors = 8.

Component number 6 is a TEE, where
NCELL1 = number of cells in primary tube = 2,
NCELL2 = number of cells in side tube = 6.

(a) Create COMPij files corresponding to components 6, 16, 42:

```
EXCON GO
Enter Space Delimited Component Numbers
6 16 42
```

Component Comp06 Finished: .100000E-04 To .101080E+01 With 38 Time-Points.
Component Comp16 Finished: .100000E-04 To .101080E+01 With 38 Time-Points.
Component Comp42 Finished: .100000E-04 To .101080E+01 With 38 Time-Points.

All Done

This reformats file TRCGRF (assuming it exists as a local file) to files COMPO6, COMP16, and COMP42.

(b) Create two plot frames during the same session and display them on the Tektronix screen.

```
TRAP FILE TRAPIN TEK / t v
Plot No. 1 Completed.
Plot No. 2 Completed.
All Done
```

where TRAPIN contains

```
TEST CASES
COM 42
FI RA 1
FI TH 3
FI Z 7
FR TI .005 1. .1
DE TL
SCALE TL 545. 500.
DI
CL RA TI
FI TI 0.2 1.0 0.2
FR RA 1 5
```

```
SM
NO
DI
EN .
```

The first plot graphs LIQUID TEMPERATURE as a function of TIME at the R=1, TH=3, Z=7 cell in the vessel. The second plot graphs LIQUID TEMPERATURE as a function of RADIUS at TIME = 0.2 s to 1.0 s at TIME = 0.2 s over vessel cells with TH = 3, Z = 7, (cell indexes). This second plot contains five curves smoothed by the SM option co-plotted on the same frame because the NOADV option is set.

(c) Create a movie of PRESSURE vs RADIUS:

```
TRAP FILE FILE1 16C GIVE / t v
Plot No. 1 Completed.
All Done
```

where FILE 1 contains

```
TEST CASE
COM 42
FI TH 7
FI Z 2
FI TI .001 1. .005
FR RA 1 5
DE P
MO
DI
END
```

This produces a short color movie of approximately 200 frames (not including header and trailer), lasting about 12 seconds when projected at 16 frames/s. The film files are automatically released to the FR80 plotter.

(d) Create two plots. The first is liquid heat-transfer coefficient vs time for the side tube of the tee, and the second is liquid temperature vs X for the pressurizer.

```
TRAP FILE TRAPIN 35C GIVE / t v
Plot No. 1 Completed.
Plot No. 2 Completed.
All Done
```

where TRAPIN contains

```
TEST CASES
COM 6
FI X 4
FR TI .005 1. .05
DE HL
SI
DI
```

```
CL ALL
COM 16
FI TI .4
FR X 1 5
DE TL
DI
END
```

The first plot request produces one 35-mm color frame. The second plot request also produces one 35-mm color frame. Both are automatically released to the system.

(e) In this example, we temporarily deal with another reactor system. Display component 9, which is a pipe component (nozzle) having 13 cells:

```
TRAP TEK / 1 1.2
  Enter General Title
    TEST CASE
  Enter Problem Specifications
    MA 9
    DI
  Plot No. 1 Completed.
    EN
  All Done
```

(f) Plot average rod temperature vs Z for rod number 13 at radial node number 8:

```
TRAP TEK / 1 1.2
  Enter General Title
    TEST CASE
  Enter Problem Specifications
    COM 42R
    ROD 13
    FI RA 8
    FR Z 0. 12. 12
    FI TI .2
    DE RODTMP
    DI
  Plot No. 1 Completed.
    EN
  All Done
```

(g) Example of use of DVNAMES and SUMMARY commands (see Tables II-3 and II-4 for output):

```
TRAP / 1 1.2
  Enter General Title
    TEST CASE
  Enter Problem Specifications
    COM 16
    DV
```

TABLE II-3
OUTPUT FROM USING DVNAMES COMMAND

DEPENDENT VARIABLE DICTIONARY
 COMP= 2 TYPE= STGEN NCELLS= 28 NODES = 0
 FILE TIME SPAN=0.1000E-03 TO 0.8003E+02
 NCELL1 = 10 NCELL2 = 17

ADJUNCT COMPONENT DATA

NO.	TYPE	NCELLS (MAIN)	NCELLS (SIDE)
1	PIPE	5	0
2	TEE	4	1
3	TEE	2	1

NCLP = 0 (NO MAIN TEE EXISTS)

MNEMONIC	CELL/GLOBAL	DESCRIPTION
MFLOW	C	MASS FLOW-RATE (KG/S)
RHOM	C	MIXTURE DENSITY (KG/M**3)
TSAT	C	SATURATION TEMP (DEG-K)
ALPHA	C	VOID FRACTION
VL	C	LIQUID VELOCITY (M/S)
VV	C	VAPOR VELOCITY (M/S)
TL	C	LIQUID TEMP (DEG-K)
TV	C	VAPOR TEMP (DEG-K)
P	C	PRESSURE (PA)
AM	C	AIR MASS (KG)
RDAN	C	AIR DENSITY (KG/M**3)
PA	C	PARTIAL PRESSURE (PA)
TWGI	C	TWGI (DEG-K)
HILG	C	HILG (W/M**2/DEG-K)
HIVG	C	HIVG (W/M**2/DEG-K)
HOLG	C	HOLG (W/M**2/DEG-K)
HOVG	C	HOVG (W/M**2/DEG-K)
IDGHI	C	INNER HT. TRANS. REGIME
IDGHO	C	OUTER HT. TRANS. REGIME
HLVPR	C	HLV PS (W/K)
HLVSEC	C	HLV SS (W/K)
HTLSCI	G	TUBE INNER HT. LOSS (W)
HTLSCO	G	TUBE OUTER HT. LOSS (W)
HTLSGI	G	GEN. INNER HT. LOSS (W)
HTLSGO	G	GEN. OUTER HT. LOSS (W)

FIX 3
 FR TI 0. 1. .1
 DE TL
 SU

TABLE II-4
OUTPUT FROM USING SUMMARY COMMAND

SUMMARY OF USER SPECIFICATIONS

COMPONENT ID = 16
PRIMARY KEYWORD OPTIONS ARE -
COMPID
DEP
FIXED
FREE
SUMMARY
DVNAMES
FREE INDEPENDENT VARIABLES ARE -
TIME
FIXED INDEPENDENT VARIABLES ARE -

SUMMARY OF USER SPECIFICATIONS (cont)

X
DEPENDENT VARIABLES ARE -
TL
X-CELL LIMITS 3 3
LIMITS FOR TIME VARIABLE ARE -
0.00000 1.00000 .10000

DI
Plot Request Completed
END
All Done

REFERENCES

- II-1. Safety Code Development Group, "TRAC-PF1/MOD1: An Advanced Best-Estimate Computer Program for Pressurized Water Reactor Thermal-Hydraulic Analysis," Los Alamos National Laboratory report NUREG/CR-3853, LA-10157-MS (July 1986).
- II-2. K. A. Taggart and D. R. Liles, "POST: A Postprocessor Computer Code for Producing Three-Dimensional Movies of Two-Phase Flow in a Reactor Vessel," Los Alamos National Laboratory report LA-NUREG-6954-MS (September 1977).
- II-3. D. Hyman, "PSCAN," CTSS Graphics Utility, Los Alamos National Laboratory, CCF Library, GR864 (November 1982).
- II-4. T. Reed, "GPALET," CTSS Graphics Utility, Los Alamos National Laboratory, CCF Library, GR750 (April 1981).
- II-5. M. Vigil, "PFILM," CTSS Utility Routine, Los Alamos National Laboratory, CCF Library, GR863 (June 1982).
- II-6. M. Vigil, "PESP," CTSS Utility Routine, Los Alamos National Laboratory, CCF Library, GR862 (November 1982).
- II-7. *DISSPLA Users Manual*, ISSCO, San Diego, California (1981).
- II-8. J. Ferguson, "Multivariable Curve Interpolation," *J. ACM* (April 1964).

III. EXTRACT

EXTRACT is a TRAC-PF1/MOD1 postprocessor that reads a TRCDMP file generated by TRAC-PF1/MOD1 (version 14.1 or greater). The code then extracts the NAMELIST and COMPONENT data from a selected dump and creates an output file, in standard TRAC load format. The output file is a partial TRACIN (TRAC input) file any parameters of which can be modified using a text editor. Once the remaining data are merged with the modified EXTRACT output file to create a complete TRACIN file, the calculation can be restarted from the selected dump. By using EXTRACT instead of rerunning the job from the beginning every time the input deck is changed, users can realize considerable savings in both computer and user resources for many types of simulations. EXTRACT can be used any time the component input must be changed and economics do not justify restarting the job from the beginning. It can also be used with relative ease for parallel sensitivity studies in parametric calculations.

A. Introduction

There has been an increasing demand for software like EXTRACT leading to the current independent code version. Many users have wanted versions of EXTRACT compatible with the TRAC versions they were using at the time.

The need for EXTRACT has been especially noted by users who wish to make changes to VESSEL data. For example, several users have wanted to modify the maximum rate of change of power value in their VESSEL input to limit the rate at which power can drop after scram for a LOCA. In this case the user's VESSEL needs to be changed on a restart, and few users welcome the exercise of doing this by hand.

In the past, our approach to resolving this user request was to inform the user which version of TRAC would work with our available EXTRACT version. If the user did not want to use that version of TRAC, many resources had to be expended to provide the user with a compatible EXTRACT code.

Many suggestions were made regarding how to provide a current EXTRACT version more easily as new versions of the TRAC code were created. These ranged from suggesting that the function of EXTRACT be made an option in the TRAC code to installing EXTRACT directly into TRAC using IF DEFs, so that a new version would not have to be generated every time a change was made to the TRAC code. Because we have had much success in using and managing independent versions of our other postprocessing codes, EXCON and TRAP, we decided to employ a standalone, independent version of EXTRACT as well.

The independent version of EXTRACT became possible only after modifying the TRAC generated TRCRST file (see Table III-1 for a current description of TRCDMP/TRCRST file). The changes, which consisted of freezing the order that the variables in the FLT, VLT, and PTB (fixed length, variable length and pointer tables) are written to the dump file, and adding the NAMELIST data to the dump file, provided needed data and stability required for a standalone code.

The current version of EXTRACT will be upgraded to reflect feedback from users. It is envisioned that EXTRACT will become another strong complement to our current TRAC postprocessor package.

TABLE III-1
DUMP-FILE OVERALL STRUCTURE

DATE	Date of file creation.
TIME	Time of file creation.
NCOMP	Number of components.
LENTBL	Actual length of fixed-length table.
MLNTBL	Frozen length of fixed-length table.
LENID	Length of problem ID.
ID	Problem ID.
LNLIST	Frozen length of NAMELIST variables.
NAMELIST	List of NAMELIST variables.
--*BFIO*--	Word indicating logical "end of file."
Time- Edit Data	Dump at first time.
--*BFIO*--	
Time- Edit Data	Dump at second time.
--*BFIO*--	
.	
.	etc.
.	
EOF	Physical "End of File" mark (fills remainder of I/O buffer).
EOF	
EOF	
LCOMP1	Length of first component dump.
	Fixed-length table for first component.
MLNVLT	Frozen length of variable-length table.
	Frozen variable-length table for first component.
MLNPTR	Frozen length of pointer-table.
	Frozen pointer table for first component.
	Array data for first component (only required arrays are dumped).
LCOMP2	Length of second component dump.
	Fixed-length table for second component.
MLNULT	Frozen length of variable-length table.
	Frozen variable-length table for second component.
MLNPTR	Frozen length of pointer table. Frozen pointer table for second component. Array data for second component (only required arrays are dumped).
LCOMP3	Length of third component dump. Fixed-length table for third component.
MLNULT	Frozen length of variable-length table.
	Frozen variable-length table for second component.
MLNPTR	Frozen length of pointer table. Frozen pointer table for second component. Array data for second component (only required arrays are dumped).

B. Basic Concepts and Capabilities

EXTRACT requires two input files: a TRCRST (a renamed TRCDMP) file and a user-created EXTIN file. EXTIN consists of user-selected options such as the dump number and the component numbers desired. Based on this input, EXTRACT will generate an output file containing all the NAMELIST information and the user-selected COMPONENT data. The output file, EXTOUT, is written in TRAC load format, which is the standard format used in TRACIN files (Ref. III-1). The user can then modify the EXTOUT file using any local editor.

The types of changes permitted can range from a simple task of changing the flow area in one cell to redesigning the mesh cell configuration of the entire component. The modified file can then be used to create a new restart TRACIN file.

EXTOUT is a partial TRACIN file lacking only part of the main data cards, the material property data, the control section (trips, signal, and control block data), and the TIME-EDIT information. After a user has created a new TRACIN file using the contents of EXTOUT file and supplied the remaining data, he can continue processing the TRAC job treating the TRACIN file as a restart deck.

When the TRACIN file created from the EXTOUT does not have control parameters that depend on their previous computed time-step values, and restarting the job at time=0 (zero) is acceptable, then the TRACIN file can be treated as a standard input deck. If the TRACIN file generated by EXTRACT contains core data in the VESSEL component, the user must set the NAMELIST option IEXVSS to 1. This is a new option that has been added to TRAC to inform the code that it is going to read an EXTRACT-generated VESSEL component. Because of the automatic fine-mesh noding (rezoning), there has always been a slight difference in the amount of data supplied by the dump file and the data required in the input file. Since an EXTRACT output file contains all the data found in a TRAC restart run, but has the appearance of a standard input deck, a means to distinguish the difference was needed.

The EXTOUT file created by EXTRACT will include any additional data which exists as a result of a special option being turned on through a NAMELIST or main data card selection. This capability does not require any additional user input but is handled automatically by EXTRACT.

EXTRACT also has the capability of reading a TRCRST file generated by a particular version of TRAC and producing an output file that will be compatible to a different version of TRAC. This option can be selected by assigning the appropriate values to CVERSION and DVERSION on the first input record of file EXTIN. Please note that if you attempt to create a newer version of an output file by setting DVERSION greater than CVERSION certain parameters required by the newer version will not be provided correctly. The same type of problem can exist in going in the opposite direction and the only solution is that the user input the correct value after editing the file.

C. Using EXTRACT

EXTRACT user must have the following files local: EXTRACT, TRCRST, and EXTIN. TRCRST is the renamed TRCDMP file from which the user wants to "extract" component information for subsequent use in a TRAC restart calculation. The EXTIN file is the EXTRACT input file. It contains at least four cards, depending on what the user wants to extract from the TRCRST file. In brief:

Card	Input Description
1	free-format input (two reals and an integer) indicating version being read (from TRCRST), version desired, and KFAC usage (0 for FRICs and 1 for KFACs)
2	-1 picks up last dump (or enter specific dump number)
3 on	999 to pick up all components (or list specific components one per line)
last line	0 (zero) is component list terminator and last record (required)

The output file created by EXTRACT has components in TRACIN-compatible format ready for editing into your TRACIN deck. A sample EXTIN input deck and run is shown in Figs. 3.1 and 3.2.

(NOTE: If the user extracts a VESSEL component that has a core, he/she must set the NAMELIST option IEXVSS to 1 in the new restart TRACIN deck whether it is to use the extracted VESSEL component data.) The following lines will compare the input differences based on the value of the NAMELIST variable IEXVSS.

IEXVSS=0 (default) vessel component was not EXTRACT generated.

Variable	Dimension	Description
BURN	ICRU-ICRL+1	Fuel burnup (MWD/MTU)
RFTN	NODES*(ICRU-ICRL+1)	Initial fuel rod temperature (K)

For IEXVSS=1 vessel component was generated by EXTRACT.

ZHT	NZMAX	Fine-mesh axial heights (m)
BURN	NZMAX	Fuel burnup (MWD/MTU)
RFTN	NODES*NZMAX	Restart fuel rod temperature (K)

The value of IEXVSS is meaningless if the VESSEL component does not contain a core. Also, in cases in which multivessel components are present in a TRACIN file, the value of IEXVSS will apply to all the VESSELS.

D. TRAC Programming Guidelines to Preserve Independence of EXTRACT

This version of EXTRACT is a standalone code and does not need to be regenerated every time a new TRAC program library is generated. The same version of EXTRACT will now be capable of reading dump files generated by the base TRAC-PF1/MOD1 version 14.1 code up to the latest version. Having this backwards and forwards capability on a standalone code required freezing the order of certain blocks of data written to the dump file and establishing some guidelines for adding and deleting data from the frozen blocks and from other write statements in the dump routines. This effort also changed the structure of the TRAC-PF1/MOD1 dump file (see Table III-1 for current description). The guidelines that users modifying the TRAC code need to know to preserve the backwards and forwards compatibility between the TRAC dump routines, the TRAC restart routines, and the EXTRACT code are:

1. All new NAMELIST variables must be equivalenced to the next available position in the dummy DVLT array in the DMPIT routine. For example, `INEW=IDVLT(N)`, where N is the next available position in array IDVLT, the integer equivalent of DVLT.
2. All new FLT (fixed-length table) variables must be equivalenced to the next available position in the dummy DVLT array in the DMPIT routine (between statement labels 90 and 95) (for examples of 1 and 2, see Table III-2).
3. All new VLT (variable-length table) and PTB (pointer table) variables must be equivalenced to the next available position in the corresponding DVLT array in their respective component-type dump routine (DPIPE, DTEE, etc.) (for examples see Table III-3).
4. The index position of the DVLT array for a NAMELIST, FLT, VLT, or PTB variable that has been deleted cannot be reassigned to a new variable added to any of the tables. Once an index on the DVLT has been assigned it can never be used again for a different variable even after the index becomes available when a variable has been deactivated.
5. For every new variable added to the NAMELIST, FLT, VLT, and PTB tables, a corresponding equivalence of the opposite type must be added to the restart routine that reads that particular table. For example, if I add the variable INEW to the NAMELIST table, I find out that the next available position on the DVLT array in the DMPIT routine is DVLT (56). I then equivalence the new variable to "`DVLT(56)=INEW`" and insert the line. But I also have to do a corresponding equivalence of the opposite type in routine RDREST. The new line would be "`INEW=DVLT(56)`." See Table III-4 and Table III-5 for additional examples.

All the guidelines up to this point pertain to preserving the frozen order of the data tables and adding new variables at the next available position. This is the only way to write a table variable to the DUMP file. Failure to do this correctly would result in the loss of the variable for restarts. The EXTRACT code will automatically read new variables added to the data tables even though additional information will sometimes be needed to instruct EXTRACT as to what to do with the new variable.

To delete an existing variable from the frozen tables one would merely delete the equivalencing line "`DVLT(N)=OLD`" on the dump routine and the corresponding "`OLD=DVLT(N)`" on the restart routine. Please note that the PT (pointer table) variables are not re-equivalenced from the DVLT array on the restart routines. The reason is that the TRAC code will recompute the pointer values based on the current version of the code. The pointer values from the dump file are only used by EXTRACT for error checking.

The last remaining guideline pertains to the addition, deletion, or modification of any calls to the subroutine BFOUT from any of the dump routines (DMPIT, DCOMP, DACCUM, DVSSL, etc.).

6. If a call to subroutine BFOUT from a dump routine is added, deleted, or modified, the following action must be taken.
 - a. Add, delete, or modify the CALL BFIN statement in the respective restart routine to correspond to the CALL BFOUT statement in the dump routine.

TABLE III-2

EXAMPLES FOR EQUIVALENCING THE NAMELIST AND FLT
VARIABLES TO THE DUMMY DVLT ARRAY IN ROUTINE DMPIT

```

*DC DMPIT
*IF -DEF, OVLAY, 1
    SUBROUTINE DMPIT
*IF DEF, OVLAY, 5
*IF -DEF, S7600, 1
    SUBROUTINE DMPIT
*IF DEF, S7600, 2
    OVERLAY (CBDUMP, 3, 0)
PROGRAM DMPIT

C
C     SUBROUTINE DMPIT GENERATES RESTART DUMPS
C
CALL CLEAR (0.0 DVLT(1), LNLIST)
DVLT(1) = ALP
DVLT(2) = CHM12
DVLT(3) = CHM13
DVLT(4) = CHM14
DVLT(5) = CHM15
DVLT(6) = CHM22
DVLT(7) = CHM23
DVLT(8) = CHM24
DVLT(9) = CHM25
DVLT(10) = DTSTRT
DVLT(11) = HSTN
IDVLT(12) = ADDED
IDVLT(13) = ICCFL
IDVLT(14) = ICDELT
IDVLT(15) = ICFLOW
IDVLT(16) = IDIAG
IDVLT(17) = IELV
IDVLT(18) = IGAS
IDVLT(19) = IGEOM3
IDVLT(20) = IH2SRC
IDVLT(21) = IHOR
IDVLT(22) = IHORG
IDVLT(23) = IKFAC
IDVLT(24) = IMFR
IDVLT(25) = INLAB
    
```

TABLE III-2 (cont)

IDVLT(26) = INVAN
 IDVLT(27) = IPOWR
 IDVLT(28) = ISGLCN
 IDVLT(29) = ISTO. F
 IDVLT(30) = LEVSTG
 IDVLT(31) = NCRG
 IDVLT(32) = NDIA1
 IDVLT(33) = NDIA3
 IDVLT(34) = NFRC1
 IDVLT(35) = NFRC2
 IDVLT(36) = NLT
 IDVLT(37) = NOAIR
 IDVLT(38) = NSEND
 IDVLT(39) = NSPL
 IDVLT(40) = NSPU
 IDVLT(41) = NVSLMX
 DVLT(42) = P
 DVLT(43) = PA
 DVLT(44) = QPPP
 DVLT(45) = TL
 DVLT(46) = YPOWR
 DVLT(47) = TV
 DVLT(48) = TW
 DVLT(49) = VL
 DVLT(50) = VV
 IDVLT(51) = IEXVSS

*IF DEF. INTERACT. :

IDVLT(52) = IPRCPV
 IDVLT(53) = NORMDP
 IDVLT(54) = NORMRS

C* * * * THE ORDER OF THE NAMELIST VARIABLES ARE FROZEN IN THE ABOVE ORDER.

C* * * * ALL NEW VARIABLES MUST BE ASSIGNED STARTING AT IDVLT(55) OR

C* * * * DVLT(55).

C IDVLT(56) = ISOLUT

C* * * * WRITE THE FROZEN NAMELIST VALUES.

C

C LOOP OVER CC = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100

C

C DO 500 ICOMP = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50

C

C READ FIXED LIST

C

TABLE III-2 (cont)

```

CALL RDLCLM(FLT(1), IA(LCOMPT+ICOMP-1), LENTBL)
CALL CLEAR (0.0, DVLT(1), MLNTBL)
DVLT(1) = HTLSCI
DVLT(2) = HTLSCO
IDVLT(3) = ICFLG
IDVLT(4) = ID
IDVLT(5) = IREST
IDVLT(6) = LENARR
IDVLT(7) = LENFV
IDVLT(8) = LENFV2
IDVLT(9) = LENPTR
IDVLT(10) = LENVLT
IDVLT(11) = LEXTRA
IDVLT(12) = LFFV
IDVLT(13) = LFFVN
IDVLT(14) = NCELLT
IDVLT(15) = NUMBM1
IDVLT(16) = NUMBM2
IDVLT(17) = NUMBM3
IDVLT(18) = NUMBN1
IDVLT(19) = NUMBN2
IDVLT(20) = NUMBN3
IDVLT(21) = NODES
IDVLT(22) = NUM
IDVLT(23) = NRVLT
IDVLT(24) = TYPE
*IF DEF, FOURB, 1
  NW=8
*IF DEF, EIGHTB, 1
  NW=4
*IF DEF, TENB, 1
  NW=3
  DO 95 I=1, NW
    IDVLT(I+24) = CTITLE(I)
  95 CONTINUE
  INDX=NW+2
C* * * * THE FIXED LENGTH TABLE HAS BEEN FROZEN IN THE ABOVE ORDER IF NEW
C* * * * VARIABLES ARE ADDED PLEASE BEGIN ASSIGNING THEM AS FOLLOWS:
C* * * * DVLT(INDX+1) OR IDVLT(INDX+1) = NEW-VARIABLE.

```

TABLE III-3

EXAMPLES FOR READING THE DUMMY DVLT ARRAY FROM A DUMP
FILE AND REASSIGNING THE VARIABLE TO THE NAMELIST
AND FLT TABLES CORRECTLY IN THE RDREST ROUTINE

```

*DC RL:LCM
*CA RDK
*DK RDREST
*IF -DEF, OVLAY, 1
    SUBROUTINE RDREST
*IF DEF, CVLAY, 5
*IF -DEF, $7600, 1
    SUBROUTINE RDREST
*IF DEF, $7600, 2
    OVERLAY (CBRDRES, 1, 3)
    PROGRAM RDREST
C
C     SUBROUTINE REDREST OBTAINS COMPONENT DATA FROM RESTART FILE
C
C* * * * READ THE NAMELIST VALUES FROM THE DUMP FILE.
C
    CALL CLEAR(0.0 DVLT(1), LNLISTO)
    CALL BFIN(IDVLT(1), LNLISTO, ICTRLP)
    ALP = DVLT(1)
    CHM12 = DVLT(2)
    CHM13 = DVLT(3)
    CHM14 = DVLT(4)
    CHM15 = DVLT(5)
    CHM22 = DVLT(6)
    CHM23 = DVLT(7)
    CHM24 = DVLT(8)
    CHM25 = DVLT(9)
    D*STRT = DVLT(10)
    H* N = DVLT(11)
    IADDED = IDVLT(12)
    ICCFL = IDVLT(13)
    ICDELTA = IDVLT(14)
    ICLFLOW = IDVLT(15)
    IDIAG = IDVLT(16)
    IELV = IDVLT(17)
    IGAS = IDVLT(18)
    IGEOM3 = IDVLT(19)

```

TABLE III-3 (cont)

IG2SRC = IDVLT(20)
IHOR = IDVLT(21)
IHORG = IDVLT(22)
IKFAC = IDVLT(23)
IMFR = IDVLT(24)
INLAB = IDVLT(25)
INVAN = IDVLT(26)
IPOWER = IDVLT(27)
ISOLCN = IDVLT(28)
ISTOPT = IDVLT(29)
LEVSTG = IDVLT(30)
NCRG = IDVLT(31)
NDIA1 = IDVLT(32)
NDIA3 = IDVLT(33)
NFRC1 = IDVLT(34)
NFRC3 = IDVLT(35)
NLT = IDVLT(36)
NOAIR = IDVLT(37)
NSENT = IDVLT(38)
NSPL = IDVLT(39)
NSPU = IDVLT(40)
NVSLMX = IDVLT(41)
P = DVLT(42)
PA = DVLT(43)
QPPP = DVLT(44)
TL = DVLT(45)
TPOWER = DVLT(46)
TV = DVLT(47)
TW = DVLT(48)
VL = DVLT(49)
VV = DVLT(50)
IEXVSS = IDVLT(51)

*IF DEF. INTERACT. 3

IPRCPV = IDVLT(52)
NORMDP = IDVLT(53)
NORMRS = IDVLT(54)

C* * * * PLEASE DEFINE NEW NEAMELIST VARIABLES STARTING AT POSITION DVLT(55)

C* * * * OR IDVLT(55)

C IF(IELV.EQ.IELVC) GO TO 1991

TABLE III-3 (cont)

200 CONTINUE

```

NRCOMP = NRCOMP+1
IF(NRCOMP.GT.DNCOMP) GO TO 221
CALL BFIN(LCOMP, 1, ICTRLR)
CALL CLEAR (O.O, DVLT(1), MLNTBO)
CALL BFIN (DVLT, MLNTBO, ICTRLR)
HTLSCI = DVLT(1)
HTLSCO = DVLT(2)
ICFLG = IDVLT(3)
ID = IDVLT(4)
IREST = IDVLT(5)
LENARR = IDVLT(6)
LENFV = IDVLT(7)
LENFV2 = IDVLT(8)
LENPTR = IDVLT(9)
LENVLT = IDVLT(10)
LEXTRA = IDVLT(11)
LFV = IDVLT(12)
LFVN = IDVLT(13)
NCELLT = IDVLT(14)
NUMBM1 = IDVLT(15)
NUMBM2 = IDVLT(16)
NUMBM3 = IDVLT(17)
NUMBN1 = IDVLT(18)
NUMBN2 = IDVLT(19)
NUMBN3 = IDVLT(20)
NODES = IDVLT(21)
NUM = IDVLT(22)
NRVLT = IDVLT(23)
TYPE = IDVLT(24)

```

*IF DEF, FOURB, 1

NW=8

*IF DEF, EIGHTB, 1

NW=4

*IF DEF, TENB, 1

NW=3

DO 205 I=1, NW

CTITLE(I) = IDVLT(I+24)

205 CONTINUE

INDX=NW+24

C* * * * PLEASE DEFINE NEW FLT VARIABLES STARTING AT POSITION DVLT(INDX+1)

C* * * * OR IDVLT (iINDX+1).

TABLE III-4

EXAMPLES FOR EQUIVALENCING THE VLT AND PTL VARIABLES
TO THE DUMMY DVLT ARRAY FOR A PIPE COMPONENT
IN THE DPIPE ROUTINE

```
*DK DPIPE
  SUBROUTINE DPIPE(COMPTR)
C
C
C*** WRITE THE FROZEN FLT TO THE DUMP FILE ***
C
  CALL BFOU1 (DVLT, MLNTBL, ICTRLD)
C
C
*** WRITE THE FROZEN LENGTH OF THE VLT ***
C
  CALL BFOU2 (MLNVLT, 1, ICTRLD)
C
  CALL CLEAR (0.0 DVLT(1), MLNVLT)
  DVLT(1) = AA1111
  DVLT(2) = BSMASS
  DVLT(3) = CPOW
  DVLT(4) = ENINP
  DVLT(5) = FL(1)
  DVLT(6) = FL(2)
  DVLT(7) = FL(1)
  DVLT(8) = FV(2)
  DVLT(9) = HOUTL
  DVLT(10) = HOUTV
  DVLT(11) = PLENT
  DVLT(12) = POWIN
  DVLT(13) = POWOFF
  DVLT(14) = QINT
  DVLT(15) = QOUT
  DVLT(16) = QP3IN
  DVLT(17) = QP3OFF
  DVLT(18) = RADIN
  DVLT(19) = RPOWMX
  DVLT(20) = RQP3MX
  DVLT(21) = TH
  DVLT(22) = TOUTL
  DVLT(23) = TOUTV
  DVLT(24) = VFLOW
  DVLT(25) = Z
```


TABLE III-4 (cont)

DVLT(26) = Z11111
IDVLT(27) = IA1111
IDVLT(28) = IACC
IDVLT(29) = ICHF
IDVLT(30) = ICJ1
IDVLT(31) = ICJ2
IDVLT(32) = ICONC
IDVLT(33) = IONOFF
IDVLT(34) = IPF
IDVLT(35) = IPOW
IDVLT(36) = IPOWSV
IDVLT(37) = IPOWTR
IDVLT(38) = IPP
IDVLT(39) = IQF
IDVLT(40) = IQP
IDVLT(41) = IQP3SV
IDVLT(42) = IQP3TR
IDVLT(43) = ISOLLB
IDVLT(44) = ISOLRB
IDVLT(45) = JS1
IDVLT(46) = JS2
IDVLT(47) = JUN1
IDVLT(48) = JUN2
IDVLT(49) = MAT
IDVLT(50) = NCELLS
IDVLT(51) = NONOFF
IDVLT(52) = NPOWRF
IDVLT(53) = NPOWSV
IDVLT(54) = NPOWTB
IDVLT(55) = NQP3RF
IDVLT(56) = NQP3SV
IDVLT(57) = NQP3TB
IDVLT(58) = TYPE1
IDVLT(59) = TYPE2
IDVLT(60) = Z11111

TABLE III-4 (cont)

```
C* * * * THE VARIABLE LENGTH TABLE IS FROZEN UP TO THIS POINT
C* * * * PLEASE ADD ALL NEW VARIABLES STARTIN AT IDVLT(61) OR DVLT(61)
C
C* * * * WRITE THE FROZEN VLT TO THE DUMP FILE * * * *
C
      CALL BFOUT (DVLT, MLNVLT, ICTRLD)
C
C
C* * * * WRITE THE FROZEN LENGTH OF THE POINTER TABLE FOR THE PIPE * * * *
C
      CALL BFOUT (MLNPTR, 1, ICTRLD)
C
      CALL RDLCM (PT, COMPTR+LENTBL+LENVLT, LENPTR)
C
      CALL CLEAR (0, IDVLT(1), MLNPTR)
C* * * * THE FOLLOWING ARE POINTERS FOUND IN *CD DUALPT
IDVLT(1) = LALP
IDVLT(2) = LALPD
IDVLT(3) = LALPDN
IDVLT(4) = LALPN
IDVLT(5) = LALV
IDVLT(6) = LALVN
IDVLT(7) = LARA
IDVLT(8) = LARAN
IDVLT(9) = LAREL
IDVLT(10) = LARELN
IDVLT(11) = LAREV
IDVLT(12) = LAREVN
IDVLT(13) = LARL
IDVLT(14) = LARLN
IDVLT(15) = LARV
IDVLT(16) = LARVN
IDVLT(17) = LBIT
IDVLT(18) = LBITN
IDVLT(19) = LCIF
IDVLT(20) = LCIFN
IDVLT(21) = LCONC
IDVLT(22) = LCONCN
IDVLT(23) = LD
IDVLT(24) = LDN
IDVLT(25) = LEA
IDVLT(26) = LEAN
```

TABLE III-4 (cont)

IDVLT(27) = LEL
IDVLT(28) = LELM
IDVLT(29) = LEV
IDVLT(30) = LEVN
IDVLT(31) = LGAM
IDVLT(32) = LGAMN
IDVLT(33) = LHIL
IDVLT(34) = LHILO
IDVLT(35) = LHIV
IDVLT(36) = LHIVO
IDVLT(37) = LHLV
IDVLT(38) = LHLVN
IDVLT(39) = LP
IDVLT(40) = LPA
IDVLT(41) = LPAN
IDVLT(42) = LPN
IDVLT(43) = LROA
IDVLT(44) = LROAN
IDVLT(45) = LROL
IDVLT(46) = LROLN
IDVLT(47) = LROV
IDVLT(48) = LROVN
IDVLT(49) = LS
IDVLT(50) = LSN
IDVLT(51) = LTD
IDVLT(52) = LTDN
IDVLT(53) = LTL
IDVLT(54) = LTLN
IDVLT(55) = LTV
IDVLT(56) = LTVN
IDVLT(57) = LTW
IDVLT(58) = LTWN
IDVLT(59) = LVL
IDVLT(60) = LVLN
IDVLT(61) = LVLT
IDVLT(62) = LVLTO
IDVLT(63) = LVM
IDVLT(64) = LVMN
IDVLT(65) = LVV
IDVLT(66) = LVVN

TABLE III-4 (cont)

IDVLT(67) = LVVT

IDVLT(68) = LVVTO

C* * * * THE FOLLOWING ARE POINTERS FOUND IN *CD HYDROPT

IDVLT(69) = LALPMN

IDVLT(70) = LALPMX

IDVLT(71) = LALPO

IDVLT(72) = LAM

IDVLT(73) = LARC

IDVLT(74) = LB

IDVLT(75) = LCFZ

IDVLT(76) = LCL

IDVLT(77) = LCPL

IDVLT(78) = LCPV

IDVLT(79) = LCV

IDVLT(80) = LDALVA

IDVLT(81) = LDFLDP

IDVLT(82) = LDFVDP

IDVLT(83) = LDRIV

IDVLT(84) = LDX

IDVLT(85) = LELEV

IDVLT(86) = LFA

IDVLT(87) = LFAVOL

IDVLT(88) = LFINAN

IDVLT(89) = LFRIC

IDVLT(90) = LFSMLT

IDVLT(91) = LGRAV

IDVLT(92) = LGRVOL

IDVLT(93) = LH(1)

IDVLT(94) = LH(2)

IDVLT(95) = LH(3)

IDVLT(96) = LHD

IDVLT(97) = LHFG

IDVLT(98) = LHGAM

IDVLT(99) = LHLA

IDVLT(100) = LHLATW

IDVLT(101) = LHVA

IDVLT(102) = LHVATW

IDVLT(103) = LKFAC

IDVLT(104) = LQP3F

IDVLT(105) = LQPPP

IDVLT(106) = LRF'S

IDVLT(107) = LRMEM

IDVLT(108) = LRMVM

TABLE III-4 (cont)

IDVLT(109) = LROM
IDVLT(110) = LRVMF
IDVLT(111) = LSIG
IDVLT(112) = LTRID
IDVLT(113) = LTSAT
IDVLT(114) = LTSSN
IDVLT(115) = LVISL
IDVLT(116) = LVISV
IDVLT(117) = LVLALP
IDVLT(118) = LVLVC
IDVLT(119) = LVLVOL
IDVLT(120) = LVLX
IDVLT(121) = LVOL
IDVLT(122) = LVR
IDVLT(123) = LVRV
IDVLT(124) = LVVVOL
IDVLT(125) = LVVX
IDVLT(126) = LWA
IDVLT(127) = LWAT
IDVLT(128) = LWFL
IDVLT(129) = LWFV

C* * * * THE FOLLOWING POINTERS ARE FROM *CD HEATPT

IDVLT(130) = LCPW
IDVLT(131) = LCW
IDVLT(132) = LDR
IDVLT(133) = LEMIS
IDVLT(134) = LHOL
IDVLT(135) = LHOV
IDVLT(136) = LQPPC
IDVLT(137) = LRN
IDVLT(138) = LRN2
IDVLT(139) = LROW
IDVLT(140) = LTOL
IDVLT(141) = LTOV

C* * * * THE FOLLOWING POINTERS ARE FROM *CD INTPT

IDVLT(142) = LIDR
IDVLT(143) = LMATID
IDVLT(144) = LNFF

TABLE III-4 (cont)

C* * * * THE FOLLOWING ARE ADDITIONAL POINTERS FOR A PIPE COMPONENT. * * * *

IDVLT(145) = LPOWRF

IDVLT(146) = LPOWTB

IDVLT(147) = LQP3RF

IDVLT(148) = LQP3TB

C* * * * THE ORDER OF THE VARIABLES IN THE POINTER TABLES ARE FROZEN UP TO

C* * * * THIS POINT. ALL NEW VARIABLES MUST BE ASSIGNED STARTING

C* * * * WITH IDVLT(149).

C

C* * * * WRITE THE FROZEN POINTER TABLE FOR A PIPE TO THE DUMP FILE * * * *

TABLE III-5

EXAMPLES FOR READING THE DUMMY DVLT ARRAY FROM THE
DUMP FILE AND REASSIGNING THE VARIABLES BACK TO
THE VLT ARRAY FOR A PIPE COMPONENT

```
*DK REPIPE
  SUBROUTINE REPIPE(ISTORE, MAXLEN, JFLAG, JUN, LCOMPTR)
C
C
C* * * * READ THE FROZEN LENGTH OF THE VLT (VARIABLE LENGTH TABLE * * * *
  CALL BFIN (MLNVLO, 1, ICTRLR)
C
* * * * READ THE FROZEN VLT FOR A PIPE COMPONENT FROM THE DUMP FILE.
C
  CALL CLEAR (0.0 DVLT(1), MLNVLO)
  CALL BFIN (DVLT(1), MLNVLO, ICTRLR)
  AA1111 = DVLT(1)
  BSMASS = DVLT(2)
  CPOW = DVLT(3)
  ENINP = DVLT(4)
  FL(1) = DVLT(5)
  FL(2) = DVLT(6)
  FV(1) = DVLT(7)
  FV(2) = DVLT(8)
  HOUTL = DVLT(9)
  HOUTV = DVLT(10)
  PLENT = DVLT(11)
  POWIN = DVLT(12)
  POWIFF = DVLT(13)
  QINT = DVLT(14)
  QOUT = DVLT(15)
  QP3IN = DVLT(16)
  QP3OFF = DVLT(17)
  RADIN = DVLT(18)
  PROWMX = DVLT(19)
  RQP3MX = DVLT(20)
  TH = DVLT(21)
  TOUTL = DVLT(22)
  TOUTV = DVLT(23)
  VFLOW = DVLT(24)
  Z = DVLT(25)
```


TABLE III-5 (cont)

Z11111 = DVLT(26)
IA1111 = IDVLT(27)
IACC = IDVLT(28)
ICHF = IDVLT(29)
ICJ1 = IDVLT(30)
ICJ2 = IDVLT(31)
ICONC = IDVLT(32)
IONOFF = IDVLT(33)
IPF = IDVLT(34)
IPOW = IDVLT(35)
IPOWSV = IDVLT(36)
IPOWTR = IDVLT(37)
IPP = IDVLT(38)
IQF = IDVLT(39)
IQP = IDVLT(40)
IQP3SV = IDVLT(41)
IQP3TR = IDVLT(42)
ISOLLB = IDVLT(43)
ISOLRB = IDVLT(44)
JS1 = IDVLT(45)
JS2 = IDVLT(46)
JUN1 = IDVLT(47)
JUN = IDVLT(48)
MAT = IDVLT(49)
NCELLS = IDVLT(50)
NONOFF = IDVLT(51)
NPOWRF = IDVLT(52)
NPOWSV = IDVLT(53)
NPOWTB = IDVLT(54)
NQP3RF = IDVLT(55)
NQP3SV = IDVLT(56)
NQP3TB = IDVLT(57)
TYPE1 = IDVLT(58)
TYPE2 = IDVLT(59)
ZI1111 = IDVLT(60)

C* * * * PLEASE DEFINE NEW VLT VARIABLES STARTING AT POSITION DVLT(61) OR
C* * * * IDVLT(61).

- b. Recompute the length of the variable "LEXTRA" in the respective input routine (ICORE, ISTGEN, etc.) which corresponds to the dump routine of the component type where the call to BFOUT was altered. This does not apply if the altered BFOUT statement was in dump routine DMPIT or DCOMP.
- c. Recompute the length of the variable "LCOMP" in all the component dump routines if the altered call to BFOUT occurred in dump routine DCOMP.
- d. Notify the EXTRACT maintenance person promptly of any changes regarding calls to subroutine BFOUT from any dump routine. Failure to do so results in the dump file being incompatible to EXTRACT.

REFERENCES

- III-1. Safety Code Development Group, "TRAC-PF1/MOD1: An Advanced Best-Estimate Computer Program for Pressurized Water Reactor Thermal-Hydraulic Analysis," Los Alamos National Laboratory report NUREG/CR-3858, LA-10157-MS (July 1986).

IV. CVRT

CVRT converts input for MOD1 or old versions of MOD2 into input for the current MOD2 version. Several MOD2 code-development activities have significantly changed the MOD2 input: the heat-structure input and storage were removed from the VESSEL component, and generalized heat-structure components were developed to model the core and vessel heat transfer. CVRT is an attempt to automate the process of converting old input to new MOD2 input. Because CVRT is under development, it still requires significant operator interaction after conversion runs.

The TRAC code (specifically its input routines) forms the basis of the CVRT code. TRAC routines not necessary for input processing were removed from the CVRT program library. The INLAB option in TRAC enables the writing of new input as the old input is read in from the CVRTIN file. Most of the changes associated with MOD2 input are in the vessel and heat-structure components. The capability to read a MOD2 deck and convert to another version of the MOD2 input was recently added to the CVRT code.

Because the CVRT code is being developed and tested, all output from the CVRT code should be used with caution. Any problems with the code should be reported to Victor Martinez (505-667-1122) at Los Alamos. The current version is version 1.0 and has been re-sequenced to aid in off-site maintenance. It is highly recommended that off-site users maintain the CVRT source with the HISTORIAN program, since error corrections and improvements to CVRT will be distributed as HISTORIAN commands. CVRT has the same IF DEFs that the original TRAC source contained; however, many of the IF DEFs in the main program have been removed. As off-site users provide us with information about problems with conversion on their operating systems, we will incorporate appropriate IF DEFs into the CVRT source so that off-site users do not have to resolve old conversion problems each time a new CVRT code is released.

A. CVRT Input and Execution

CVRT currently runs interactively and only requires two input values:

- VERI - The version number for the old input.
Acceptable input:
14.0 (the old input deck is for the MOD1 code)
3.7 (the old input deck is for the MOD2 code).
- VERO - The version number for the new input.
Acceptable input:
3.4, 3.5, or 3.8 if VERI is 14.0
3.8 if VERI is 3.7.
(3.8 input is equivalent to that for version 4.0 and is preferable to input for versions 3.4 and 3.5.)

If the MOD2 input changes again, then the CVRT program will be modified to perform the appropriate conversions. For example, if version 4.1 MOD2 input is different from version 4.0 input, CVRT will be modified to read version 4.0 input and write out version 4.1 input.

If the off-site operating system does not support interactive computing, then CVRT can easily be modified to run in batch mode reading VERI and VERO from an input file. CVRT requires that the old input file be on the CVRTIN file (device unit number 1) and will write the new input file on the INLAB file (device unit number 3). CVRT converts the free-format

input on CVRTIN to a fixed-format TRAC input on the CVTINP file (device unit number 5). Interactive input and output are to the TTY file (device unit number 59). Messages are written to the CVTMSG file (device unit number 7) and output to the CVTOUT file (device unit number 6).

B. Conversion Tasks Currently Performed by CVRT

When converting from a MOD1 version 14.0 input deck to a MOD2 version 3.8 input deck, CVRT performs the following conversion tasks.

1. Adds the wall surface roughness input to appropriate components. For the MOD2 code, wall surface roughness is input as a single number for selected components. This resulted in changes to almost all of the 1D and 3D component input.
2. Removes unused variables from the 1D component input for the MOD2 code CVRT takes this into account in preparing the new input file.
3. Removes heat-structure input from the VESSEL component for the MOD2 code input. Several new variables are available as input to the MOD2 VESSEL component (*i.e.*, IGEOM, etc.) and appropriate values are supplied by the CVRT code. Since the MOD1 vessel could only be a right cylinder, this is the option generated by CVRT.
4. Gathers heat-structure input for the VESSEL component and writes it out in the new format associated with the generalized heat-structure component. CVRT currently writes out a new generalized heat-structure component for each of the slab heat structures in the VESSEL component. If any of these heat structures have similar geometry, generalized heat-structure components are not necessary for each of the similar VESSEL heat slabs; geometry defined by one generalized heat structure can be duplicated in many different locations within a TRAC 3D VESSEL component. However, CVRT is unable to perform this automatically at present; if a user wishes to reduce the number of generalized heat-structure components, he or she will have to perform that manually. There will be one new generalized heat structure for each vessel heat slab, so this manual lumping of heat-slab input may be required for large vessels with several hundred heat slabs. In addition, CVRT figures out the connectivity from the heat-structure component to the vessel component, locating the new generalized heat structure in the appropriate vessel hydraulic cell.
5. Numbers the new generalized heat-structure components starting at 999, decreasing to 998, 997, etc. The MOD2 code currently requires that all heat-structure component numbers be larger than the largest hydraulic component's/component number. If the old TRAC input deck has hydraulic components with component numbers in this range, the component numbers will have to be changed.
6. Performs the appropriate conversion to make the MOD1 CFZs into MOD2 CFZs if IKFAC=0. In the MOD2 code, the 3D-VESSEL CFZ definition was changed to be consistent with the 1D FRIC definition. Therefore, both FRIC and CFZ in the MOD2 code are dimensionless and are both additive constant friction factors if IKFAC=0.

C. Conversion Tasks Not Performed by CVRT

The following is a list of tasks that must be performed manually on the INLAB file after executing CVRT and before attempting to run the MOD2 code. As resources permit, we will eliminate these manual tasks.

1. NAMELIST INPUT — The number of heat structures (NHTSTR) in the new TRAC input is a required NAMELIST input. CVRT will print a message with the correct number; however, because CVRT does not currently write the correct number into the NAMELIST input, the user must do this manually. In addition, any references to the following variables in the NAMELIST input should be removed.
 - a. NVSLMX — MOD2 always uses direct inversion and the capacitance method to solve the pressure matrix equations.
 - b. ICCFL — MOD2 currently does not include the CCFL model.
 - c. ISTOPT — The INLAB file will contain all of the default values for P, PA, etc. Also, if you have specified defaults in your NAMELIST (*i.e.*, P, PA, etc.), these also must be removed from the NAMELIST input on the INLAB file.
2. NCOMP — The number of components will increase because of all the new heat-structure components. Again, CVRT will print the correct number for NCOMP in a message in the INLAB file, but the user will have to modify the NCOMP input value manually.
3. IORDER ARRAY — The new component numbers for the new heat structures must appear in the IORDER array. CVRT will have written the new IORDER array input in the appropriate format to the INLAB file, but the user will have to move the new IORDER array to the appropriate location.
4. ROD AND SLAB INPUT — The generalized heat-structure input must appear in the input deck immediately before the time-step control cards. If the VESSEL component is the last component in the old TRAC input deck, then the ROD and SLAB input will appear in the correct locations in the INLAB file. If the VESSEL component in the old TRAC input deck is not last, or if there are two or more VESSEL components, then the new generalized heat-structure input must be moved to right in front of the time-step cards.
5. WARNING AND ??? CARDS — CVRT writes warning cards and "???" where the appropriate input is not known. A pattern search on WARNING and ??? will direct the user to comment cards that must be deleted as well as to other numbers that may have to be changed. There are typically two or three comment cards after the warning that should be deleted. A general gravity vector for the VESSEL component that CVRT writes out with ??? as input can typically be deleted because the default is a vertical VESSEL component.
6. SIGNAL VARIABLES — Signal variables with parameter numbers 18-19, 24-26, and 44-54 can no longer refer to a VESSEL component. The component numbers for these signal variables should be changed to the appropriate ROD component numbers.

DISTRIBUTION

Nuclear Regulatory Commission, R4, Laurel, Maryland	273
Technical Information Center, Oak Ridge, Tennessee	2
Los Alamos National Laboratory, Los Alamos, New Mexico (includes special distribution to TRAC User's Group)	<u>429</u>
Total	704

NRC FORM 335 (2-84) NRCM 1102 3201, 3202	U.S. NUCLEAR REGULATORY COMMISSION	1. REPORT NUMBER (Assigned by T/DC add Vol. No., if any)
BIBLIOGRAPHIC DATA SHEET		NUREG/CR-5071 LA-11214-MS
SEE INSTRUCTIONS ON THE REVERSE		
2. TITLE AND SUBTITLE	3. LEAVE BLANK	
TRAC Support Software	4. DATE REPORT COMPLETED	
5. AUTHOR(S)	MONTH	YEAR
R. P. Jenks V. Martinez	April	1988
7. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)	6. DATE REPORT ISSUED	
Los Alamos National Laboratory Los Alamos, NM 87545	MONTH	YEAR
10. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)	8. PROJECT/TASK/WORK UNIT NUMBER	
Division of Reactor Accident Analysis Office of Nuclear Regulatory Research U. S. Nuclear Regulatory Commission Washington, DC 20555	9. FIN OR GRANT NUMBER	
12. SUPPLEMENTARY NOTES	A7016	
13. ABSTRACT (200 words or less)	11. TYPE OF REPORT	
This manual provides users of the Transient Reactor Analysis Code (TRAC) with information about computer codes that can be used to support their analysis efforts. These codes are collectively referred to as TRAC support software.	Informal	
Specifically, documentation is provided to allow users to implement, apply, and interpret the output of the support software that is available. The TRAC code is an advanced best-estimate code for analyzing transients in thermal-hydraulic systems. Considerable input data are required for modeling a large thermal-hydraulic system and considerable output data are produced from the computer calculation for analysis; the support software assists users in their management of both the input and the output data.	10. PERIOD COVERED (Inclusive dates)	
Initially, this document contains sections that describe EXCON, TRAP, EXTRACT, and CVRT, major support software used routinely at Los Alamos National Laboratory.		
14. DOCUMENT ANALYSIS -- a. KEYWORDS DESCRIPTORS	15. AVAILABILITY STATEMENT	
b. IDENTIFIERS-OPEN ENDED TERMS	Unlimited	
	16. SECURITY CLASSIFICATION	
	(This page)	
	Unclassified	
	(This report)	
	Unclassified	
	17. NUMBER OF PAGES	
	108	
	18. PRICE	

Available from

Superintendent of Documents
U. S. Government Printing Office
Post Office Box 37082
Washington, D. C. 20513-7982

and

National Technical Information Service
Springfield, VA 22161