WRDC-TR-90-8007
Volume VIII
Part 20

AD-A248 928

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 20 - Forms Driven Forms Editor Product Specification

S. Barker

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

DTIC
S ELECTE D
APR 23 1992
D

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

92-10247

Approved for Public Release; Distribution is Unlimited

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

92 4 21 063

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) PS 620344402	5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR-90-8007 Vol. VIII, Part 20
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services 6b. OFFICE SYMBOL *(if applicable)*	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI
6c. ADDRESS *(City, State, and ZIP Code)* 2970 Presidential Drive Fairborn, OH, 45324-6209	7b. ADDRESS *(City, State, and ZIP Code)* WPAFB, OH 45433-6533
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF 8b. OFFICE SYMBOL *(if applicable)* WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464
8c. ADDRESS *(City, State, and ZIP Code)* Wright-Patterson AFB, Ohio 45433-6533	10. SOURCE OF FUNDING NOS.
11. TITLE *(Ir* Forms Driv See block 19	PROGRAM ELEMENT NO. 78011F PROJECT NO. 595600 TASK NO. F95600 WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Structural Dynamics Research Corporation: Barker, S., et al.	
13a. TYPE OF REPORT Final Report 13b. TIME COVERED 4/1/87-12/31/90	14. DATE OF REPORT *(Yr.,Mo.,Day)* 1990 September 30 15. PAGE COUNT 304
16. SUPPLEMENTARY NOTATION WRDC/MTI Project Priority 6203	

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify block no.)*
FIELD	GROUP	SUB GR.	
1308	0905		

19. ABSTRACT *(Continue on reverse if necessary and identify block number)*

This specification establishes the detailed design of the Forms Driven Form Editor computer program.

BLOCK 11:

INTEGRATED INFORMATION SUPPORT SYSTEM
Vol VIII -User Interface Subsystem

Part 20 - Forms Driven Forms Editor Product Specification

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. *(Include Area Code)* (513) 255-7371 22c. OFFICE SYMBOL WRDC/MTI

EDITION OF 1 JAN 73 IS OBSOLETE

DD FORM 1473, 83 APR

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface,and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

		Page
SECTION 1.0	SCOPE	1-1
1.1	Identification	1-1
1.2	Functional Summary	1-1
SECTION 2.0	DOCUMENTS	2-1
2.1	Reference Documents	2-1
2.2	Terms and Abbreviations	2-3
SECTION 3.0	REQUIREMENTS	3-1
3.1	Structural Description	3-1
3.1.1	Module Hierarchy	3-1
3.1.2	Module Descriptions	3-3
3.1.2.1	FDFE	3-3
3.1.2.2	PRSCMD	3-3
3.1.2.3	LISTIT	3-3
3.1.2.4	VIEW	3-4
3.1.2.5	FORMS LANGUAGE SOURCE ACCESS MODULES	3-4
3.1.2.6	EDTMOD	3-5
3.1.2.7	LISTFM	3-5
3.1.2.8	INSFRM	3-6
3.1.2.9	DRPFRM	3-6
3.1.2.10	EDTWHL	3-6
3.1.2.11	EDTFLD	3-6
3.1.2.12	LAYOUT	3-7
3.1.2.13	SCRMAN	3-7
3.1.2.14	CHGPOS	3-7
3.1.2.15	TRNSCR	3-7
3.1.2.16	TRNSTR	3-8
3.1.2.17	VALINP	3-8
3.1.2.18	GTNMFD	3-8
3.1.2.19	MODFLD	3-8
3.1.2.20	DELFLD	3-9
3.1.2.21	INSFLD	3-9
3.1.2.22	COPFRM	3-9
3.1.2.23	MODFRM	3-10
3.1.2.24	FLFMST	3-10
3.1.2.25	FLSTRC	3-10
3.1.2.26	FLWHST	3-10
3.1.2.27	GWHINP	3-11
3.1.2.28	GTCPFD	3-11
3.1.2.29	DRPWHL	3-11
3.1.2.30	MODWHL	3-11
3.1.2.31	INSWHL	3-12
3.2	Functional Flow	3-12
3.3	Interfaces	3-13
3.3.1	Application Interface	3-13
3.3.2	Forms Language Compiler	3-13
3.3.3	Operating System	3-14
3.4	Program Interrupts	3-14
3.5	Timing ans Sequencing Description	3-14
3.6	Special Control Features	3-14

3.7 Storage Allocation 3-14
3.7.1 Data Base Definition 3-14
3.7.1.1 File Descriptions 3-14
3.8 Object Code Creation 3-16
3.9 Adaptation Data 3-17
3.10 Detailed Design Description 3-17
3.10.1 Main Program List 3-17
3.10.2 Module List 3-19
3.10.3 External Routines List 3-25
3.10.4 Include File List 3-28
3.10.5 Where Include File Used List 3-30
3.10.6 Where External Routine Used List 3-45
3.10.7 Main Program Parts List 3-63
3.10.8 Module Documentation 3-74
3.10.9 Include File Description 3-237
3.10.10 Hierarchy Chart 3-251
3.11 Program Listings Comments 3-390

SECTION 4.0 QUALITY ASSURANCE PROVISIONS 4-1
4.1 Introduction and Definitions 4-1
4.2 Computer Programming and Test Evaluation 4-1

Accesion For
NTIS CRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

LIST OF ILLUSTRATIONS

Figure	Title	Page
3-1	FDFE Hierarchy Charts	3-2
3-2	FDFE Data Flow	3-12
3-3	FDFE Interface Diagram	3-13

SECTION 1

SCOPE

1.1 Identification

This specification establishes the detailed design of a computer program identified as the Forms Driven Form Editor, hereinafter referred to as FDFE. The FDFE is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Functional Summary

The FDFE is a software tool for creating and initializing form definitions. The FDFE displays a series of screens which request information from the user and visually show the form under construction. Once a form has been completed, the FDFE stores the form definition constructs needed to recreate the form. The stored form can be selected and modified.

The runtime UI or UIMS views the FDFE, which is part of the UIDS, as an application program which uses the Form Processor. Data to be selected or stored comes from or is passed to the Common Data Model (CDM) in the integrated implementation; otherwise, a file system is used. The FDFE also interacts with the Forms Language Compiler (FLAN) to translate between the forms language source and the compiled form definition.

The FDFE is a C program which makes extensive use of form language sources and compiled forms, performs interactive user input/output via the Form Processor (FP), and uses the FP to manage the compiled forms. The internal form data structure is the same as that used by the Form Processor.

SECTION 2

DOCUMENTS

2.1 Reference Documents

[1] Structural Dynamics Research Corporation, Application Interface Product Specification, PS 620144700 , 1 November 1985.

[2] Structural Dynamics Research Corporation, Forms Language Compiler Product Specification, PS 620144401 , 1 November 1985.

[3] Structural Dynamics Research Corporation, Form Processor Product Specification, PS 620144200 , 1 November 1985.

[4] Structural Dynamics Research Corporation, Rapid Application Generator Product Specification, PS 620144502 , 1 November 1985.

[5] Structural Dynamics Research Corporation, Report Writer Product Specification, PS 620144501 , 1 November 1985.

[6] Structural Dynamics Research Corporation, Text Editor Product Specification, PS 620144600 , 1 November 1985.

[7] Structural Dynamics Research Corporation, User Interface Services Product Specification, PS 620144100 , 1 November 1985.

[8] Structural Dynamics Research Corporation, Virtual Terminal Product Specification, PS 620144300 , 1 November 1985.

[9] Structural Dynamics Research Corporation, Forms Driven Form Editor Development Specification, DS 620144402B, 1 November 1985.

[10] Structural Dynamics Research Corporation, Forms Driven Form Editor Unit Test Plan, UTP620144402 , 1 November 1985.

[11] Structural Dynamics Research Corporation, Forms Driven Form Editor User Manual, UM 620144402 , 1 November 1985.

[12] "Designing a Portable Natural Language Database Query System", S. J. Kaplan, ACM Trans. on Database Sys. 9(1), 1984.

[13] "Document Formatting System: Survey, Concepts and Issues", R. Furuta, J. Scofield, A. Shaw, ACM Comp. Surveys 14(3), 1982.

[14] "Formal Grammar and Human Factors Design of an Interactive Graphics System", P. Reisner, IEEE Trans. on Software ENG. 7(2), 1981.

[15] HUMAN PERFORMANCE ENGINEERING: A GUIDE FOR SYSTEM DESIGNERS, R. Bailey; Prentice-Hall, Inc., (1982).

[16] ICAM DOCUMENTATION STANDARDS, ICAM DOCUMENT IDS 150120000C, 15 SEPTEMBER 1983.

[17] "Interactive Editing Systems: Parts I and II", N. Meyrowitz and Andries van Dam, ACM Comp. Surveys 14(3), 1982.

[18] "Programming Language Constructs for Screen Definition", L. A. Rowe and K. A. Shoens, IEEE Trans. on Software Eng. 9(1), 1983.

[19] THE PSYCHOLOGY OF HUMAN-COMPUTER INTERACTION, S. K. Card, T. P. Moran and A. Newell, Lawrence Erlbaum Associates, Inc. (1983).

[20] SOFTWARE PSYCHOLOGY: HUMAN FACTORS IN COMPUTER AND INFORMATION SYSTEMS, B. Shneiderman; Little, Brown and Co. (1982).

[21] General Electric Co., System Design Specification, 7 February 1983.

2.2 Terms and Abbreviations

American Standard Code for Information Interchange: (ASCII), the character set defined by ANSI X3.4 and used by most computer vendors.

Application Interface: (AI), subset of the IISS User Interface that consists of the callable routines that are linked with applications that use the Form Processor or Virtual Terminal. The AI enables applications to be hosted on computers other than the host of the User Interface.

Application Process: (AP), a cohesive unit of software that can be initiated as a unit to perform some function or functions.

Attribute: field characteristic such as blinking, highlighted, black, etc. and various other combinations. Background attributes are defined for forms or windows only. Foreground attributes are defined for items. Attributes may be permanent, i.e., they remain the same unless changed by the application program, or they may be temporary, i.e., they remain in effect until the window is redisplayed.

Device Drivers: (DD), software modules written to handle I/O for a specific kind of terminal. The modules map terminal specific commands and data to a neutral format. Device Drivers are part of the UI Virtual Terminal.

Display List: is similar to the open list, except that it contains only those forms that have been added to the screen and are currently displayed on the screen.

Extended Binary Coded Decimal Interchange Code: (EBCDIC), the character set used by a few computer vendors (notably IBM) instead of ASCII.

Field: two dimensional space on a terminal screen.

Form: structured view which may be imposed on windows or other forms. A form is composed of fields. These fields may be defined as forms, items, and windows.

Form Definition: (FD), forms definition language after compilation. It is read at runtime by the Form Processor.

Forms Definition Language: (FDL), the language in which electronic forms are defined.

Forms Driven Form Editor: (FDFE), subset of the FE which consists of a forms driven application used to create Form Definition files interactively.

Form Editor: (FE), subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor and the Forms Language Compiler.

Form Hierarchy: a graphic representation of the way in which forms, items and windows are related to their parent form.

Forms Language Compiler: (FLAN), subset of the FE that consists of a batch process that accepts a series of forms definition language statements and produces form definition files as output.

Form Processor: (FP), subset of the IISS User Interface that consists of a set of callable execution time routines available to an application program for form processing.

Form Processor Text Editor: (FPTE), subset of the Form Processor that consists of software modules that provide text editing capabilities to all users of applications that use the Form Processor.

IISS Function Screen: the first screen that is displayed after logon. It allows the user to specify the function he wants to access and the device type and device name on which he is working.

Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Item: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Message: descriptive text which may be returned in the standard message line on the terminal screen. They are used to warn of errors or provide other user information.

Message Line: a line on the terminal screen that is used to display messgaes.

Network Transaction Manager: (NTM), IISS subsystem that performs the coordination, communication and housekeeping functions required to integrate the Application Processes and System Services resident on the various hosts into a cohesive system.

Open List: a list of all the forms that have been and are currently open for an application process.

Operating System: (OS), software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Page: instance of forms in windows that are created whenever a form is added to a window.

Paging and Scrolling: a method which allows a form to contain more data than can be displayed with provisions for viewing any portion of the data buffer.

Physical Device: a hardware terminal.

Qualified Name: the name of a form, item or window preceded by the hierarchy path so that it is uniquely identified.

Subform: a form that is used within another form.

User Data: data which is either input by the user or output by the application programs to items.

User Interface: (UI), IISS subsystem that controls the user's terminal and interfaces with the rest of the system. The UI consists of two major subsystems: the User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Development System: (UIDS), collection of IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

User Interface Management System: (UIMS), the runtime UI. It consists of the Form Processor, Virtual Terminal, Application Interface, the User Interface Services and the Text Editor.

User Interface Monitor: (UIM), part of the Form Processor that handles messaging between the NTM and the UI. It also provides authorization checks and initiates applications.

User Interface Services: (UIS), subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

User Interface/Virtual Terminal Interface: (UI/VTI), another name for the User Interface.

Virtual Terminal: (VT), subset of the IISS User Interface that performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by the UI software which constitutes the virtual terminal definition. Specific terminals are then mapped against the virtual terminal software by specific software modules written for each type of real terminal supported.

Window: dynamic area of a terminal screen on which predefined forms may be placed at run time.

Window Manager: a facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.

SECTION 3

REQUIREMENTS

3.1 Structural Description

The general approach is to view the FDFE as a hierarchy of modules. The FDFE screens presented in the appendix are associated with only certain modules in the hierarchy based on the functionality being performed by the module.

3.1.1 Module Hierarchy

The following hierarchy chart shows the organization of the FDFE:

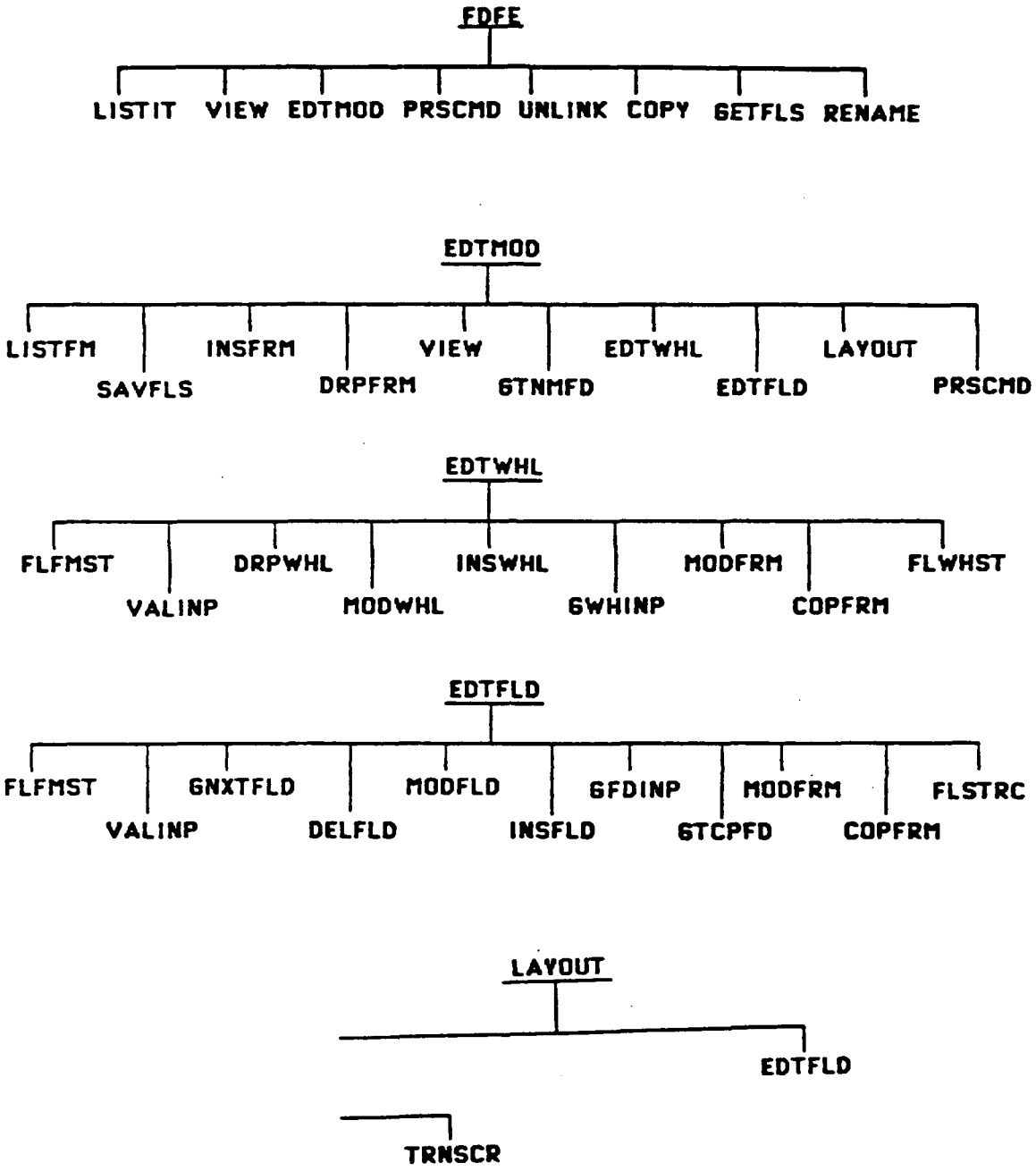


Figure 3-1 FDFE Hierarchy Charts

3.1.2 Module Descriptions

The following paragraphs describe the modules associated with each of the major sections of the FDFE.

3.1.2.1 FDFE:

This module is the main driver. It allows the user to choose among several file management options or to proceed to the edit task level. It controls the WRKTASK screen.

Input Parameters:
None
Output Parameters:
None

3.1.2.2 PRSCMD:

This module parses the command line for both the WRKTASK and EDTTASK screens to determine which other modules are to be called and what parameters are to be passed.

Input Parameters:
Pointer to command line
Array of parameter counts
Array of pointers to valid commands
Output Parameters:
Option chosen
Pointer to parameters to be passed to the module which will execute the option.
Number of parameters found in command line

3.1.2.3 LISTIT:

This module lists all of the Forms Definition Object or Forms Language Source in the user's specified forms language source or definition object libraries.

Input Parameters:
Pointer to "FDL" or "FD" string
Output Parameters:
Returns any error code or a NULL pointer if successful

3.1.2.4 VIEW:

This module displays a form just as it would appear on the screen when used by a program.

Input Parameters:
Pointer to name of form to view
Output Parameters:
Returns any error code or a NULL pointer if successful

3.1.2.5 FORMS LANGUAGE SOURCE ACCESS MODULES:

- UNLINK:

  This module drops a particular Forms Language Source.

  Input Parameters:
  Name of Forms Language Source
  Output Parameters:
  Returns any error code or a NULL pointer if successful

- COPY:

This module copies a particular Forms Language Source to another Forms Language Source of specified name.

Input Parameters:
Name of existing "from" Forms Language Source
Name of new or "to" Forms Language Source
Output Parameters:
Returns any error code or a NULL pointer if successful

- RENAME:

This module renames a particular Forms Language Source to a specified name.

Input Parameters:
Name of existing Forms Language Source
New name of Forms Language Source
Output Parameters:
Returns any error code or a NULL pointer if successful

- GETFLS:

This module retrieves a particular Forms Language Source and translates it into the internal data structure.

Input Parameters:
Name of existing Forms Language Source
Output Parameters:
Pointer to the "opened" form, the internal Forms Processor data structure

- SAVFLS:

This module saves a particular Forms Language Source (fdl file) and a Forms Definition Object (fd file) after it translates the internal data structure into forms language syntax.

Input Parameters:
Name of the form to be saved under flag indicating to write or not to write out the fd file
Pointer to list of forms to be written out
Output Parameters:
Returns any error code or a NULL pointer if successful

3.1.2.6 EDTMOD:

This module is the control module for all edit tasks. It controls the EDTTASK screen.

Input Parameters:
New/old form flag
Change/retrieve only flag
Output Parameters:
Returns any error code or a NULL pointer if successful

3.1.2.7 LISTFM:

This module lists all forms in the Forms Language Source on which work is being done.

Input Parameters:
None
Output Parameters:
Returns any error codes or NULL pointer if successful

3.1.2.8 INSFRM:

This module inserts a new form into the Forms Language Source on which work is being done.

Input Parameters:
  name of form
Output Parameters:
  Returns any error code or NULL pointer if successful

3.1.2.9 DRPFRM:

This module deletes a form from the Forms Language Source on which work is being done.

Input Parameters:
  Name of form
Output Parameters:
  Returns any error codes or NULL pointer if successful

3.1.2.10 EDTWHL:

This module allows the user to edit an entire form at once. It controls the presentation of the FORM EDIT screen.

Input Parameters:
  Read-only flag
  Name of form
Output Parameters:
  Returns any error code or NULL pointer if successful

3.1.2.11 EDTFLD:

This module allows the user to edit all fields of a form one at a time. It controls the presentation of the FIELD EDIT screen. It is also called from LAYOUT.

Input Parameters:
  Read-only flag
  col cursor position if coming from layout mode
  row cursor position if coming from layout mode
  pointer to internal form structure
  edit mode
Output Parameters:
  Returns any error code or NULL pointer if successful

3.1.2.12 LAYOUT:

This module allows the user to edit an entire form as it would appear when used (with regards to the location and size of fields) on one screen. The other needed information is filled in on the LAYOUT DESCRIPTION screen.

Input Parameters:
  Pointer to internal form structure
  Read-only fla
Output Parameters:
  Returns any error code or NULL pointer if successful

3.1.2.13 SCRMAN:

This module controls the first stage of layout edit mode - it manages the screen using the following three modules to translate internal structure to screen layout and vice versa.

Input Parameters:
  Read-only flag
  pointer to internal form structure
Output Parameters:
  Row position returned from GETCUR
  Col position returned from GETCUR

3.1.2.14 CHGPOS:

This module allows the user to change the location of a field in layout mode by indicating the "from" and "to" locations on the screen.

Input Parameters:
  pointer to internal form structure
Output Parameters:
  Returns any error code or NULL pointer if successful

3.1.2.15 TRNSCR:

This module translates the layout screen format to internal structure.

Input Parameters:
  Pointer to internal form structure
Output Parameters:
  Returns any error code or NULL pointer if successful

3.1.2.16 TRNSTR:

This module translates the internal structure to the layout screen format.

Input Parameters:
  Pointer to internal form structure
  Read only flag
Output Parameters:
  None

3.1.2.17 VALINP:

This module performs validation checks on fields. The objects to be validated are the values input on the FIELD EDIT and FORM EDIT screens.

Input Parameters:
Pointer to form to be validated
Pointer to field to be validated
Flag indicating type of validation
Output Parameters:
Returns TRUE if validation okay else returns FALSE

3.1.2.18 GTNMFD:

This module retrieves fields from the internal structure.

Input Parameters:
Pointer to 1st field in internal structure
Name of field to find
Output Parameters:
Pointer to field in the internal structure or NULL if could not find field

3.1.2.19 MODFLD:

This module modifies a field in the internal structure.

Input Parameters:
Pointer to parent of field
Pointer to pointer of field being modified
Pointer to screen changed information
Pointer to screen help info

Pointer to screen value info
Pointer to screen item info
Output Parameters:
Returns any error code or NULL pointer if successful

3.1.2.20 DELFLD:

This module deletes a field from the internal structure. This is the same function as that used by the Form Processor.

3.1.2.21 INSFLD:

This module inserts a field into the internal structure.

Input Parameters:
Address of pointer to field being inserted
Address where next field pointer will be inserted
Address where previous field pointer will be inserted
Pointer to parent of field
Pointer to screen field information
Pointer to screen help info
Pointer to screen value info
Pointer to screen item info
Recursion level
Output Parameters:
Returns any error code or NULL pointer if successful

3.1.2.22 COPFRM:

This module copies a Forms Language Source file into an alternate internal data structure and gets the pointer to the specified form.

Input Parameters:
Name of Forms Language Source file to copy from
Name of form to copy
Output Parameters:
Sets global variables:
Copyfls, name of Forms Language Source just copied
Copyfrm, name of form just copied
Altbuf, beginning of list containing all forms of Forms Language Source
Altfrm, pointer to form user wishes to copy

3.1.2.23 MODFRM:

This module updates information about the form.

Input Parameters:
Pointer to form internal structure
Pointer to screen form information
Output Parameters:
Returns any error code or NULL pointer if successful

3.1.2.24 FLFMST:

This module translates an FPD field structure for a form into the screen information structure.

Input Parameters:
Pointer to screen structure
Pointer to fpd form field
Output Parameters:
None

3.1.2.25 FLSTRC:

This module translates the FPD field structure to screen information structure for items, windows, and forms.

Input Parameters:
Pointer to fpd field
Pointer to screen field structure
Pointer to screen item help structure
Pointer to screen item value structure
Pointer to screen field domain check structure
Output Parameters:
Fills appropriate screen area with field information

3.1.2.26 FLWHST:

This module fills in the output screen FORM EDIT, and associates each field line on the output screen with the field's internal structure.

Input Parameters:
Pointer to form internal structure on which editing is to occur.

Output Parameters:
Fills output screen FORM EDIT area with form and field info and creates an external array of pointers

3.1.2.27 GWHINP:

This module gets all input for the FORM EDIT screen for the fields on the form being edited.

Input Parameters:
Pointer to form internal structure
Output Parameters:
PF key provided by OISCR
Returns any error code or NULL pointer if successful

3.1.2.28 GTCPFD:

This module gets the field at the located cursor position.

Input Parameters:
Pointer to form internal structure
Row cursor position
Col cursor position
Output Parameters:
Pointer to field at that location else
NULL if no field found

3.1.2.29 DRPWHL:

This module deletes all fields marked by the user on the FORM EDIT screen.

Input Parameters:
Pointer to internal form structure
Output Parameters:
PF key received by OISCR

3.1.2.30 MODWHL:

This module modifies existing fields as input by the user on the FORM EDIT screen.

Input Parameters:
Pointer to parent of field
Pointer to field being modified
Pointer to input screen structure
Pointer to help line on screen

Pointer to item value on screen
Pointer to item only info on screen
Output Parameters:
Returns any error code or NULL pointer if successful

3.1.2.31 INSWHL:

This module inserts all fields that have been entered on the FORM EDIT screen.

Input Parameters:
Pointer to form internal structure
Output Parameters:
Returns any error code or NULL pointer if successful

3.2 Functional Flow

Figure 3-2 is a data flow for the Forms Driven Form Editor.

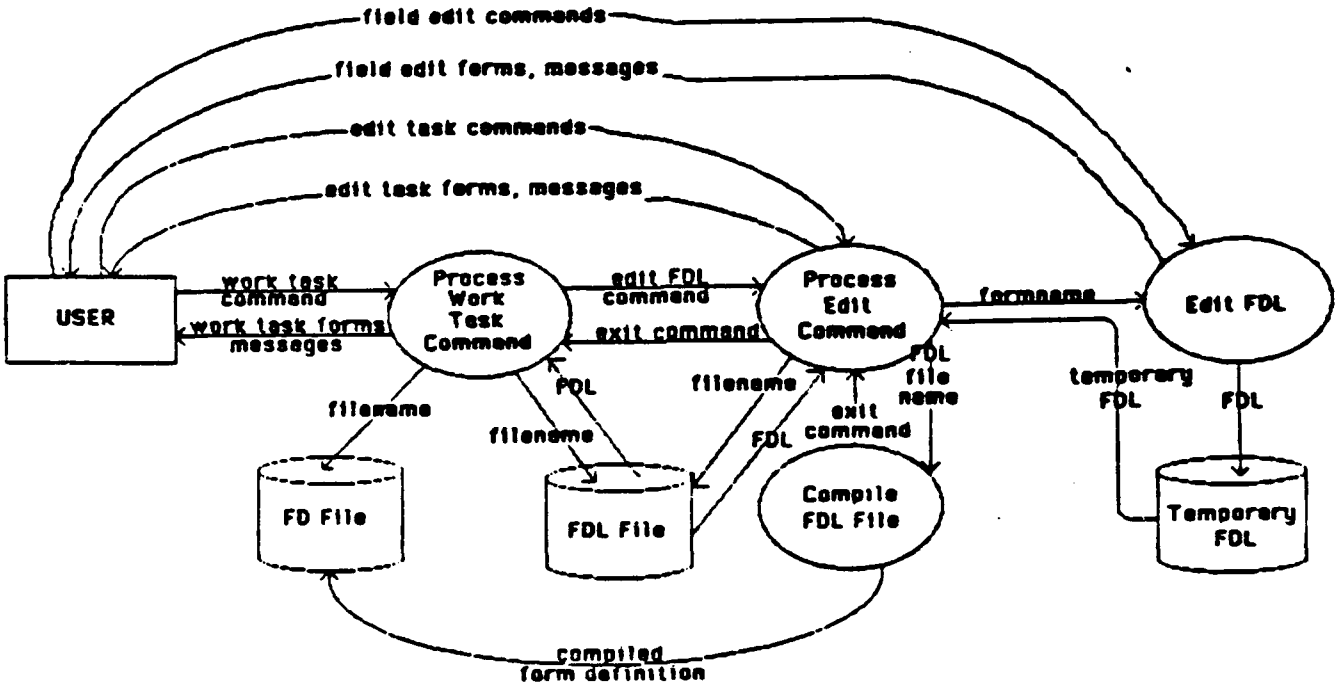


Figure 3-2 FDFE Data Flow

3.3 Interfaces

The FDFE interfaces directly with users as an IISS application. Physical terminals are assumed to have video display, a textual keyboard, four cursor positioning keys or key sequences, a help key or key sequence, an entry key, and four other keys to be used by the FDFE for special processing. The FDFE must interface with the AI, FLAN, and the operating system.

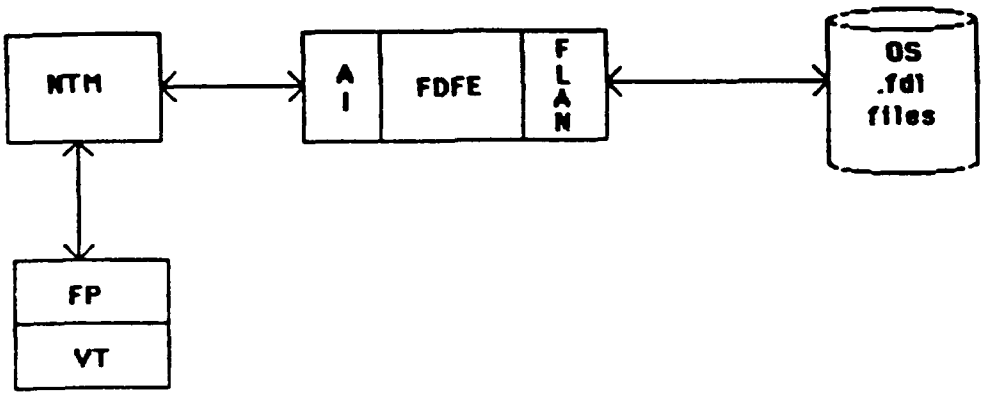


Figure 3-3 FDFE Interface Diagram

3.3.1 Application Interface

The FDFE interacts with users by calling appropriate routines of the Application Interface (AI). This interface creates messages which are sent to the Form Processor which moves information describing interactive terminal input and output and provides a link to users of the FDFE through the Virtual Terminal.

3.3.2 Forms Language Compiler

The FDFE uses the Forms Language Complier (FLAN) to convert forms language source into the Form Processor internal forms

structure. The FDFE also invokes the FLAN when the form under construction is to be stored.

3.3.3 Operating System

The FDFE stores form language source files (fdl files on the VAX) and compiled form definitions (fd files on the VAX). Form language source files may subsequently be compiled and displayed. The storage of the fdl files and fd files is system dependent. The VAX implementation uses the logicals IISSSLIB (for fdl files) to store/retrieve the files in/from the appropriate directory.

3.4 Program Interrupts

This section does not apply to the detailed design of the Forms Driven Form Editor.

3.5 Timing and Sequencing Description

The data flow diagram in section 3.2 and the detail design description in section 3.10 contain the procedural information for sequencing and control logic.

3.6 Special Control Features

The detailed design of the FDFE does not include any special control features as defined in the ICAM Documentation Standards manual.

3.7 Storage Allocation

The Forms Driven Form Editor executable is 351 blocks.

3.7.1 Data Base Definition

3.7.1.1 File Descriptions

1. FILE NAME: name.FDL - Form Definition Language file.

   PURPOSE: This file contains the language description of one or more forms. Compiling this file produces Form Definition files.

   DECLARATION:

   ```
   char line [132];
   ```

2. FILE NAME: name.TMP - Temporary FDL file.

   PURPOSE: This file contains the language description of one or more forms. It is created during an editing session from the FDFE's internal data structure. If this file compiles successfully, it is renamed to an FDL file, otherwise it remains so that the user can correct errors and recompile it.

   DECLARATION:

```
char line [132];
```

3. FILE NAME: formname.FD - Form Definition file. A complete description of the Form Definition file which is a binary file is contained in Appendix B of the Forms Language Compiler Development Specification (DS 620144401B). The name of this file is dependent upon the form it describes.

   PURPOSE: This file contains information about the structure and attributes of a form that is used at run time by the Form Processor.

   DECLARATION:

```
typedef struct      /* version number record */
   {
   char rectyp;     /* '1' */
   int  vernum;     /* current version number (2) */
   char linefeed;
   } VERREC;

typedef struct      /* form record */
   {
   char  form_name[10];   /* form name */
   char  background[10];  /* background name */
   short row;             /* starting row */
   short col;             /* starting col */
   short width;           /* width */
   short depth;           /* depth */
   short n_txtflds;       /* number of text fields */
   short n_datflds;       /* number of data fields */
   short s_txtbuf;        /* size of the text buffer */
   short s_defbuf;        /* size of the default buffer */
   char  linefeed;
   } FRMREC;
```

```
typedef struct      /* text record */
   {
   short row;       /* starting row */
   short col;       /* starting col */
   short len;       /* total length */
   char  linefeed;
   } TXTREC;

typedef struct      /* field record */
   {
   char  fld_name[10];   /* field name */
   char  fld_type;       /* field type (F, I, W, A) */
   short row;            /* starting row */
   short col;            /* starting col */
   short width;          /* field width */
   short depth;          /* field depth */
   int   min_value;      /* minimum value (if any) */
   int   max_value;      /* maximum value (if any) */
   char  helpline[80];   /* help text */
   char  disp_att[10];   /* display attribute */
   short n_formats;      /* number of formats */
   char  format[12][2];  /* format strings */
   short n_arydefs;      /* number of dimensions */
   struct      /* dimension specification */
      {
      char  dir;         /* repeat direction (H, V) */
      short cnt;         /* actual repeat count */
      short sp;          /* number of spaces between
                            repetitions */
      short dsp_size;    /* display repeat count */
      } array_def[3];
   char  linefeed;
   } FLDREC;

typedef struct {         /* run time relative positioning
                            info */
   POS posnod;
   NAME mynam, hnam, vnam;
   } RELREC;
```

3.8 Object Code Creation

The FDFE routines were compiled using a C compiler developed by Interactive Software under VAX/VMS.

3.9 Adaptation Data

The C source modules for the FDFE can be compiled using any UNIX version 7 compatible C compiler. The files FPDINI.H and GETFLS.C contain file names for the Form Definition and Form Definition Language files which may not port to systems other than VAX/VMS.

3.10 Detailed Design Description

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it is truly a main program entry point. If not, then it is merely called by other programs not being documented here.

FORMS DRIVEN FORM EDITOR Main Program List

Module Name	Purpose
CHKFLD	CHECK FIELD
CHKFRM	CHECK FORM
CSTASH	CHARACTER STASH
FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
MAKINT	MAKE EXPRESSION INTO AN INTEGER
MAKSTR	MAKE EXPRESSION INTO A STRING
MKPOS	MAKE POSITION NODE
VALINP/CCKVAL	CHECK VALUE

3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

FORMS DRIVEN FORM EDITOR Module List

Module Name	Purpose
ADDCHK	ADD POSITION TO CHECK LIST
ADDEXT	ADD EXTENSION TO FILE_NAME
CHKARY	CHECK ARRAY
CHKFLD	CHECK FIELD
CHKFRM	CHECK FORM
CHKPRM	CHECK PARAMETER
CPYFRM	COPY FORM
CSTASH	CHARACTER STASH
DRPFRM	DROP FORM
DRPWHL	DROP WHOLE
EDTFLD	EDIT FIELD
EDTMOD	EDIT MODE
EDTWHL	EDIT WHOLE
ERROR	ISSUE ERROR MESSAGE
EXPAND	EXPAND AN ARRAY
EXPAND/FIXFRM	FIX UP A FORM
FATAL	ISSUE FATAL ERROR MESSAGE
FDFE	FORMS DRIVEN FORM EDITOR
FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
FIFDST	FILL IN FIELD STRUCTURE

FORMS DRIVEN FORM EDITOR Module List

Module Name	Purpose
FLANCI	FLAN CALLABLE INTERFACE
FLDTYP	FIELD TYPE
FLFMST	FIELD TO FORM STRUCTURE TRANSLATION
FLSTRC	FIELD STRUCTURE TRANSLATION
FLWHST	FILL WHOLE STRUCTURE
FNDATT	FIND ATTRIBUTE
FREBUF	FREE BUFFER
GETFLS	GET FDL SOURCE FILE
GETFLS/TREEXP	TREE EXPRESSION
GETLEN	GET LENGTH
GFDINP	GET FIELD INPUT
GFLDPT	GET FIELD POINTER
GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
GNXTFD	GET NEXT FIELD
GNXTFD/NXTFLD	NEXT FIELD
GTCPFD	GET USING CURSOR POSITION FIELD
GTFDTX	GET FIELD TEXT
GTFDTX/GTXINF	GET TEXT INFORMATION
GTNMFD	GET NAMED FIELD
GWHINP	GET WHOLE INPUT
INSFLD	INSERT FIELD

FORMS DRIVEN FORM EDITOR Module List

Module Name	Purpose
INSFRM	INSERT FORM
INSWHL	INSERT WHOLE
LAYOUT	LAYOUT MODE
LISTFM	LIST FORMS
LISTIT	LIST IT
MAKINT	MAKE EXPRESSION INTO AN INTEGER
MAKSTR	MAKE EXPRESSION INTO A STRING
MKPOS	MAKE POSITION NODE
MODFLD	MODIFY FIELD
MODFRM	MODIFY FORM
MODFRM/FRETXT	FREE TEXT
MODWHL	MODIFY WHOLE
MYALLOC	MY MALLOC
PRCFIL	PROCESS TEMPORARY FILE
PREC	PRECEDENCE
PRSCMD	PARSE COMMAND
PUTERR	PUT ERROR
SAVFLS	SAVE FDL SOURCE
SCRMAN	SCREEN MANAGER
SCRMAN/CHGPOS	CHANG POSITION
SCRMAN/GETROW	GET ROW

FORMS DRIVEN FORM EDITOR Module List

Module Name	Purpose
TRNSCR	TRANSLATE SCREEN TO STRUCTURE
TRNSCR/FLCST	FILL LOCATION STRUCTURE
TRNSCR/FRLCST	FREE LOCATION STRUCTURES
TRNSCR/GTFMPMT	GET FORM PROMPT INFORMATION
TRNSCR/GTPINF	GET PROMPT INFORMATION
TRNSCR/LDPMINF	LOAD PROMPT INFORMATION
TRNSCR/MTCHPMT	MATCH PROMPT WITH FIELD
TRNSCR/PARSCRN	PARSE SCREEN DATA
TRNSCR/SPSYMB	SPECIAL SYMBOL CHECK
TRNSTR	TRANSLATE STRUCTURE TO SCREEN
TRNSTR/FLFLD	FILL FIELD
TRNSTR/FLPRMPT	FILL PROMPT
TRNSTR/GARINF	GET ARRAY INFORMATION
VALINP	VALIDATE INPUT
VALINP/CCKFLD	CHECK FIELD
VALINP/CCKFRM	CHECK FORM
VALINP/CCKHLP	CHECK HELP
VALINP/CCKITM	CHECK ITEM
VALINP/CCKNAM	CHECK NAME
VALINP/CCKPRM	CHECK PROMPT
VALINP/CCKRSV	CHECK FOR RESERVED WORD

FORMS DRIVEN FORM EDITOR Module List

Module Name	Purpose
VALINP/CCKVAL	CHECK VALUE
VIEW	VIEW A FORM
WARNING	ISSUE WARNING MESSAGE
WRTEXP	WRITE EXPRESSION
WRTFDL	WRITE FDL FILE
WRTFDL/ARYREF	ARRAY REFERENCE

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. The specification in which any module is documented may be found in the Module Documentation Index (Document Number CM 620100001). See section 3.10.6 for a list of the modules that call each of these external routines.

FORMS DRIVEN FORM EDITOR External Routines List

Module Name	First User
ABORT	VALINP
ABS	SCRMAN/CHGPOS
ACCESS	FDFE
ADDFRM	FDFE
ATOI	VALINP/CCKITM
BLEN	WRTFDL
CLSFRM	VIEW
COPFLD	EXPAND
COPY	FDFE
DELFLD	EXPAND/FIXFRM
ESCPY	VALINP/CCKITM
FCLOSE	GETFLS
FEOF	LISTIT
FERROR	PRCFIL
FGETS	PRCFIL
FOPEN	SAVFLS
FPRINTF	WRTFDL
FREE	TRNSCR/FRLCST
GATDEF	FIFDST
GDATA	GWHINP
GETCUR	SCRMAN
GWINDO	VIEW
INITAL	FDFE/MAIN
INITFP	FDFE/MAIN
ISALPHA	VALINP/CCKFLD
ISSPACE	TRNSCR/GTPINF
MAKFLD	INSFLD
MALLOC	MYALLOC
MATOI	VALINP/CCKFLD
MAX	CHKFLD
MEMCMP	VALINP
MEMCPY	GWHINP
MEMDGT	GITMD
MEMSET	PRSCMD
MIN	VALINP/CCKFLD
MITOA	TRNSCR
MKTEMP	LISTIT
OISCR	SCRMAN
PDATA	LISTIT
PMSGLC	SCRMAN
PMSGLS	LISTFM

FORMS DRIVEN FORM EDITOR External Routines List

Module Name	First User
PUTATT	EDTFLD
PUTCUR	PUTERR
RENAME	SAVFLS
REWIND	SAVFLS
RMVPAG	LISTIT
RSVATT	EXPAND/FIXFRM
SPRINTF	VALINP/CCKNAM
STRASN	CHKARY
STRCAT	SAVFLS
STRCHR	VALINP/CCKFRM
STRCMP	VALINP/CCKFLD
STRCPY	LISTIT
STRLEN	FDFE
STRNCMP	PRSCMD
STRNCPY	PRSCMD
STRRCHR	SCRMAN/GETROW
STRSPN	VALINP/CCKHLP
STRUPC	VALINP/CCKHLP
SYSMSG	VIEW
SYSTEM	LISTIT
TERMFP	FDFE/MAIN
TRMNAT	FDFE/MAIN
UNLINK	FDFE
WRTFRM	SAVFLS
YYPARSE	FLANCI

3.10.4 Include File List

The following is a list of all include files called in by modules being documented here. Each include file has a unique name regardless of the language being used. The purpose of each include file is listed as well. A more complete description of each include file is given in section 3.10.9. The purpose listed is the one that is in the source code of the include file.

A purpose of "**** PURPOSE NOT FOUND BY STRIPPER ****" indicates that a purpose statement was not written into the include file itself. The most common reason for this is that the include file comes from system libraries that were not developed by the project, such as 'C' libraries that are provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all the modules which call in each of these include files.

FORMS DRIVEN FORM EDITOR Include File List

File Name	Purpose
CTYPE	**** PURPOSE NOT FOUND BY STRIPPER ****
FDFE	FDFE DATA STRUCTURES
FDFEFM	FDFE FORM DEFINITIONS
FDFEINI	FDFE INITIALIZATIONS
FPCODE	FORM PROCESSOR RETURN CODES
FPD	FORM PROCESSOR DATA
FPDINI	FPD INITIALIZATION
FPPARM	FORM PROCESSOR PARAMETERS
NTM	NTM INTERFACE INCLUDE FILE
RW	REPORT WRITER DEFINITIONS
STDIO	**** PURPOSE NOT FOUND BY STRIPPER ****
STDTYP	STANDARD TYPE DEFINITIONS

3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
CTYPE		
	EDTFLD	EDIT FIELD
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	PUTERR	PUT ERROR
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/FL	FILL LOCATION STRUCTURE
	TRNSCR/FR	FREE LOCATION STRUCTURES
	TRNSCR/GT	GET FORM PROMPT INFORMATION
	TRNSCR/GT	GET PROMPT INFORMATION
	TRNSCR/LD	LOAD PROMPT INFORMATION
	TRNSCR/MT	MATCH PROMPT WITH FIELD
	TRNSCR/PA	PARSE SCREEN DATA
	TRNSCR/SP	SPECIAL SYMBOL CHECK
	VALINP	VALIDATE INPUT
	VALINP/CC	CHECK FIELD
	VALINP/CC	CHECK FORM
	VALINP/CC	CHECK HELP
	VALINP/CC	CHECK ITEM
	VALINP/CC	CHECK NAME
	VALINP/CC	CHECK PROMPT
	VALINP/CC	CHECK FOR RESERVED WORD
	VALINP/CC	CHECK VALUE
FDFE		
	ADDEXT	ADD EXTENSION TO FILE_NAME
	CHKPRM	CHECK PARAMETER
	CPYFRM	COPY FORM
	DRPFRM	DROP FORM
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
	FLFMST	FIELD TO FORM STRUCTURE TRANSLATION
	FLSTRC	FIELD STRUCTURE TRANSLATION

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	FLWHST	FILL WHOLE STRUCTURE
	FREBUF	FREE BUFFER
	GETFLS	GET FDL SOURCE FILE
	GETFLS/TR	TREE EXPRESSION
	GFDINP	GET FIELD INPUT
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GNXTFD	GET NEXT FIELD
	GNXTFD/NX	NEXT FIELD
	GTFDTX	GET FIELD TEXT
	GTFDTX/GT	GET TEXT INFORMATION
	GTNMFD	GET NAMED FIELD
	GWHINP	GET WHOLE INPUT
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM
	INSWHL	INSERT WHOLE
	LAYOUT	LAYOUT MODE
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	MODFLD	MODIFY FIELD
	MODFRM	MODIFY FORM
	MODFRM/FR	FREE TEXT
	MODWHL	MODIFY WHOLE
	PRCFIL	PROCESS TEMPORARY FILE
	PRSCMD	PARSE COMMAND
	PUTERR	PUT ERROR
	SCRMAN	SCREEN MANAGER
	SCRMAN/CH	CHANG POSITION
	SCRMAN/GE	GET ROW
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/FL	FILL LOCATION STRUCTURE
	TRNSCR/FR	FREE LOCATION STRUCTURES
	TRNSCR/GT	GET FORM PROMPT INFORMATION
	TRNSCR/GT	GET PROMPT INFORMATION
	TRNSCR/LD	LOAD PROMPT INFORMATION
	TRNSCR/MT	MATCH PROMPT WITH FIELD
	TRNSCR/PA	PARSE SCREEN DATA
	TRNSCR/SP	SPECIAL SYMBOL CHECK
	TRNSTR	TRANSLATE STRUCTURE TO SCREEN
	TRNSTR/FL	FILL FIELD
	TRNSTR/FL	FILL PROMPT

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	TRNSTR/GA	GET ARRAY INFORMATION
	VALINP	VALIDATE INPUT
	VALINP/CC	CHECK FIELD
	VALINP/CC	CHECK FORM
	VALINP/CC	CHECK HELP
	VALINP/CC	CHECK ITEM
	VALINP/CC	CHECK NAME
	VALINP/CC	CHECK PROMPT
	VALINP/CC	CHECK FOR RESERVED WORD
	VALINP/CC	CHECK VALUE
	VIEW	VIEW A FORM
FDFEFM		
	ADDEXT	ADD EXTENSION TO FILE_NAME
	CHKPRM	CHECK PARAMETER
	CPYFRM	COPY FORM
	DRPFRM	DROP FORM
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
	FLFMST	FIELD TO FORM STRUCTURE TRANSLATION
	FLSTRC	FIELD STRUCTURE TRANSLATION
	FLWHST	FILL WHOLE STRUCTURE
	FREBUF	FREE BUFFER
	GETFLS	GET FDL SOURCE FILE
	GETFLS/TR	TREE EXPRESSION
	GFDINP	GET FIELD INPUT
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GNXTFD	GET NEXT FIELD
	GNXTFD/NX	NEXT FIELD
	GTFDTX	GET FIELD TEXT
	GTFDTX/GT	GET TEXT INFORMATION
	GTNMFD	GET NAMED FIELD

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	GWHINP	GET WHOLE INPUT
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM
	INSWHL	INSERT WHOLE
	LAYOUT	LAYOUT MODE
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	MODFLD	MODIFY FIELD
	MODFRM	MODIFY FORM
	MODFRM/FR	FREE TEXT
	MODWHL	MODIFY WHOLE
	PRCFIL	PROCESS TEMPORARY FILE
	PRSCMD	PARSE COMMAND
	PUTERR	PUT ERROR
	SCRMAN	SCREEN MANAGER
	SCRMAN/CH	CHANG POSITION
	SCRMAN/GE	GET ROW
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/FL	FILL LOCATION STRUCTURE
	TRNSCR/FR	FREE LOCATION STRUCTURES
	TRNSCR/GT	GET FORM PROMPT INFORMATION
	TRNSCR/GT	GET PROMPT INFORMATION
	TRNSCR/LD	LOAD PROMPT INFORMATION
	TRNSCR/MT	MATCH PROMPT WITH FIELD
	TRNSCR/PA	PARSE SCREEN DATA
	TRNSCR/SP	SPECIAL SYMBOL CHECK
	TRNSTR	TRANSLATE STRUCTURE TO SCREEN
	TRNSTR/FL	FILL FIELD
	TRNSTR/FL	FILL PROMPT
	TRNSTR/GA	GET ARRAY INFORMATION
	VALINP	VALIDATE INPUT
	VALINP/CC	CHECK FIELD
	VALINP/CC	CHECK FORM
	VALINP/CC	CHECK HELP
	VALINP/CC	CHECK ITEM
	VALINP/CC	CHECK NAME
	VALINP/CC	CHECK PROMPT
	VALINP/CC	CHECK FOR RESERVED WORD
	VALINP/CC	CHECK VALUE
	VIEW	VIEW A FORM

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
FDFEINI		
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
FPCODE		
	ADDCHK	ADD POSITION TO CHECK LIST
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CPYFRM	COPY FORM
	CSTASH	CHARACTER STASH
	DRPFRM	DROP FORM
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	EXPAND	EXPAND AN ARRAY
	EXPAND/FI	FIX UP A FORM
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
	FLANCI	FLAN CALLABLE INTERFACE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE
	GETFLS	GET FDL SOURCE FILE
	GETFLS/TR	TREE EXPRESSION
	GFDINP	GET FIELD INPUT
	GFLDPT	GET FIELD POINTER
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GNXTFD	GET NEXT FIELD
	GNXTFD/NX	NEXT FIELD
	GTFDTX	GET FIELD TEXT

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	GTFDTX/GT	GET TEXT INFORMATION
	GWHINP	GET WHOLE INPUT
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM
	INSWHL	INSERT WHOLE
	LAYOUT	LAYOUT MODE
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MKPOS	MAKE POSITION NODE
	MODFLD	MODIFY FIELD
	MODFRM	MODIFY FORM
	MODFRM/FR	FREE TEXT
	MODWHL	MODIFY WHOLE
	MYALLOC	MY MALLOC
	PRCFIL	PROCESS TEMPORARY FILE
	PUTERR	PUT ERROR
	SAVFLS	SAVE FDL SOURCE
	SCRMAN	SCREEN MANAGER
	SCRMAN/CH	CHANG POSITION
	SCRMAN/GE	GET ROW
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/FL	FILL LOCATION STRUCTURE
	TRNSCR/FR	FREE LOCATION STRUCTURES
	TRNSCR/GT	GET FORM PROMPT INFORMATION
	TRNSCR/GT	GET PROMPT INFORMATION
	TRNSCR/LD	LOAD PROMPT INFORMATION
	TRNSCR/MT	MATCH PROMPT WITH FIELD
	TRNSCR/PA	PARSE SCREEN DATA
	TRNSCR/SP	SPECIAL SYMBOL CHECK
	TRNSTR	TRANSLATE STRUCTURE TO SCREEN
	TRNSTR/FL	FILL FIELD
	TRNSTR/FL	FILL PROMPT
	TRNSTR/GA	GET ARRAY INFORMATION
	VALINP	VALIDATE INPUT
	VALINP/CC	CHECK FIELD
	VALINP/CC	CHECK FORM
	VALINP/CC	CHECK HELP
	VALINP/CC	CHECK ITEM

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	VALINP/CC	CHECK NAME
	VALINP/CC	CHECK PROMPT
	VALINP/CC	CHECK FOR RESERVED WORD
	VALINP/CC	CHECK VALUE
	VIEW	VIEW A FORM
	WRTEXP	WRITE EXPRESSION
FPD		
	ADDCHK	ADD POSITION TO CHECK LIST
	ADDEXT	ADD EXTENSION TO FILE_NAME
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CHKPRM	CHECK PARAMETER
	CPYFRM	COPY FORM
	CSTASH	CHARACTER STASH
	DRPFRM	DROP FORM
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	EXPAND	EXPAND AN ARRAY
	EXPAND/FI	FIX UP A FORM
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
	FIFDST	FILL IN FIELD STRUCTURE
	FLANCI	FLAN CALLABLE INTERFACE
	FLDTYP	FIELD TYPE
	FLFMST	FIELD TO FORM STRUCTURE TRANSLATION
	FLSTRC	FIELD STRUCTURE TRANSLATION
	FLWHST	FILL WHOLE STRUCTURE
	FNDATT	FIND ATTRIBUTE
	FREBUF	FREE BUFFER
	GETFLS	GET FDL SOURCE FILE
	GETFLS/TR	TREE EXPRESSION
	GFDINP	GET FIELD INPUT

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	GFLDPT	GET FIELD POINTER
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GNXTFD	GET NEXT FIELD
	GNXTFD/NX	NEXT FIELD
	GTCPFD	GET USING CURSOR POSITION FIELD
	GTFDTX	GET FIELD TEXT
	GTFDTX/GT	GET TEXT INFORMATION
	GTNMFD	GET NAMED FIELD
	GWHINP	GET WHOLE INPUT
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM
	INSWHL	INSERT WHOLE
	LAYOUT	LAYOUT MODE
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MKPOS	MAKE POSITION NODE
	MODFLD	MODIFY FIELD
	MODFRM	MODIFY FORM
	MODFRM/FR	FREE TEXT
	MODWHL	MODIFY WHOLE
	MYALLOC	MY MALLOC
	PRCFIL	PROCESS TEMPORARY FILE
	PREC	PRECEDENCE
	PRSCMD	PARSE COMMAND
	PUTERR	PUT ERROR
	SAVFLS	SAVE FDL SOURCE
	SCRMAN	SCREEN MANAGER
	SCRMAN/CH	CHANG POSITION
	SCRMAN/GE	GET ROW
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/FL	FILL LOCATION STRUCTURE
	TRNSCR/FR	FREE LOCATION STRUCTURES
	TRNSCR/GT	GET FORM PROMPT INFORMATION
	TRNSCR/GT	GET PROMPT INFORMATION
	TRNSCR/LD	LOAD PROMPT INFORMATION
	TRNSCR/MT	MATCH PROMPT WITH FIELD
	TRNSCR/PA	PARSE SCREEN DATA
	TRNSCR/SP	SPECIAL SYMBOL CHECK

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	TRNSTR	TRANSLATE STRUCTURE TO SCREEN
	TRNSTR/FL	FILL FIELD
	TRNSTR/FL	FILL PROMPT
	TRNSTR/GA	GET ARRAY INFORMATION
	VALINP	VALIDATE INPUT
	VALINP/CC	CHECK FIELD
	VALINP/CC	CHECK FORM
	VALINP/CC	CHECK HELP
	VALINP/CC	CHECK ITEM
	VALINP/CC	CHECK NAME
	VALINP/CC	CHECK PROMPT
	VALINP/CC	CHECK FOR RESERVED WORD
	VALINP/CC	CHECK VALUE
	VIEW	VIEW A FORM
	WRTEXP	WRITE EXPRESSION
	WRTFDL	WRITE FDL FILE
	WRTFDL/AR	ARRAY REFERENCE
FPDINI		
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
FPPARM		
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
	GFDINP	GET FIELD INPUT
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GWHINP	GET WHOLE INPUT

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM
	LAYOUT	LAYOUT MODE
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	PUTERR	PUT ERROR
	SCRMAN	SCREEN MANAGER
	SCRMAN/CH	CHANG POSITION
	SCRMAN/GE	GET ROW
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/FL	FILL LOCATION STRUCTURE
	TRNSCR/FR	FREE LOCATION STRUCTURES
	TRNSCR/GT	GET FORM PROMPT INFORMATION
	TRNSCR/GT	GET PROMPT INFORMATION
	TRNSCR/LD	LOAD PROMPT INFORMATION
	TRNSCR/MT	MATCH PROMPT WITH FIELD
	TRNSCR/PA	PARSE SCREEN DATA
	TRNSCR/SP	SPECIAL SYMBOL CHECK
	VALINP	VALIDATE INPUT
	VALINP/CC	CHECK FIELD
	VALINP/CC	CHECK FORM
	VALINP/CC	CHECK HELP
	VALINP/CC	CHECK ITEM
	VALINP/CC	CHECK NAME
	VALINP/CC	CHECK PROMPT
	VALINP/CC	CHECK FOR RESERVED WORD
	VALINP/CC	CHECK VALUE
	VIEW	VIEW A FORM
NTM		
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
RW		
	ADDCHK	ADD POSITION TO CHECK LIST
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CSTASH	CHARACTER STASH
	FLANCI	FLAN CALLABLE INTERFACE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE
	GFLDPT	GET FIELD POINTER
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MKPOS	MAKE POSITION NODE
	MYALLOC	MY MALLOC
	WRTEXP	WRITE EXPRESSION
STDIO		
	ADDCHK	ADD POSITION TO CHECK LIST
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CSTASH	CHARACTER STASH
	EDTFLD	EDIT FIELD
	FLANCI	FLAN CALLABLE INTERFACE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE
	GETFLS	GET FDL SOURCE FILE
	GETFLS/TR	TREE EXPRESSION
	GFLDPT	GET FIELD POINTER
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	LISTIT	LIST IT
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MKPOS	MAKE POSITION NODE
	MYALLOC	MY MALLOC
	PRCFIL	PROCESS TEMPORARY FILE
	PREC	PRECEDENCE

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	SAVFLS	SAVE FDL SOURCE
	WRTEXP	WRITE EXPRESSION
	WRTFDL	WRITE FDL FILE
	WRTFDL/AR	ARRAY REFERENCE
STDTYP		
	ADDCHK	ADD POSITION TO CHECK LIST
	ADDEXT	ADD EXTENSION TO FILE_NAME
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CHKPRM	CHECK PARAMETER
	CPYFRM	COPY FORM
	CSTASH	CHARACTER STASH
	DRPFRM	DROP FORM
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	ERROR	ISSUE ERROR MESSAGE
	EXPAND	EXPAND AN ARRAY
	EXPAND/FI	FIX UP A FORM
	FATAL	ISSUE FATAL ERROR MESSAGE
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
	FIFDST	FILL IN FIELD STRUCTURE
	FLANCI	FLAN CALLABLE INTERFACE
	FLDTYP	FIELD TYPE
	FLFMST	FIELD TO FORM STRUCTURE TRANSLATION
	FLSTRC	FIELD STRUCTURE TRANSLATION
	FLWHST	FILL WHOLE STRUCTURE
	FNDATT	FIND ATTRIBUTE
	FREBUF	FREE BUFFER
	GETFLS	GET FDL SOURCE FILE
	GETFLS/TR	TREE EXPRESSION
	GETLEN	GET LENGTH

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	GFDINP	GET FIELD INPUT
	GFLDPT	GET FIELD POINTER
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GNXTFD	GET NEXT FIELD
	GNXTFD/NX	NEXT FIELD
	GTCPFD	GET USING CURSOR POSITION FIELD
	GTFDTX	GET FIELD TEXT
	GTFDTX/GT	GET TEXT INFORMATION
	GTNMFD	GET NAMED FIELD
	GWHINP	GET WHOLE INPUT
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM
	INSWHL	INSERT WHOLE
	LAYOUT	LAYOUT MODE
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MKPOS	MAKE POSITION NODE
	MODFLD	MODIFY FIELD
	MODFRM	MODIFY FORM
	MODFRM/FR	FREE TEXT
	MODWHL	MODIFY WHOLE
	MYALLOC	MY MALLOC
	PRCFIL	PROCESS TEMPORARY FILE
	PREC	PRECEDENCE
	PRSCMD	PARSE COMMAND
	PUTERR	PUT ERROR
	SAVFLS	SAVE FDL SOURCE
	SCRMAN	SCREEN MANAGER
	SCRMAN/CH	CHANG POSITION
	SCRMAN/GE	GET ROW
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/FL	FILL LOCATION STRUCTURE
	TRNSCR/FR	FREE LOCATION STRUCTURES
	TRNSCR/GT	GET FORM PROMPT INFORMATION
	TRNSCR/GT	GET PROMPT INFORMATION
	TRNSCR/LD	LOAD PROMPT INFORMATION
	TRNSCR/MT	MATCH PROMPT WITH FIELD
	TRNSCR/PA	PARSE SCREEN DATA

FORMS DRIVEN FORM EDITOR Where-include-file-used List

Include File	Module Name	Module Purpose
	TRNSCR/SP	SPECIAL SYMBOL CHECK
	TRNSTR	TRANSLATE STRUCTURE TO SCREEN
	TRNSTR/FL	FILL FIELD
	TRNSTR/FL	FILL PROMPT
	TRNSTR/GA	GET ARRAY INFORMATION
	VALINP	VALIDATE INPUT
	VALINP/CC	CHECK FIELD
	VALINP/CC	CHECK FORM
	VALINP/CC	CHECK HELP
	VALINP/CC	CHECK ITEM
	VALINP/CC	CHECK NAME
	VALINP/CC	CHECK PROMPT
	VALINP/CC	CHECK FOR RESERVED WORD
	VALINP/CC	CHECK VALUE
	VIEW	VIEW A FORM
	WARNING	ISSUE WARNING MESSAGE
	WRTEXP	WRITE EXPRESSION
	WRTFDL	WRITE FDL FILE
	WRTFDL/AR	ARRAY REFERENCE

3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
ABORT		
	VALINP	VALIDATE INPUT
ABS		
	CHKARY	CHECK ARRAY
	CHKFRM	CHECK FORM
	CHKFRM	CHECK FORM
	EXPAND	EXPAND AN ARRAY
	SCRMAN/CHGCHANG	POSITION
	TRNSTR/GARGET	ARRAY INFORMATION
	WRTFDL/ARYARRAY	REFERENCE
ACCESS		
	FDFE	FORMS DRIVEN FORM EDITOR
ADDFRM		
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	GWHINP	GET WHOLE INPUT
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	SCRMAN	SCREEN MANAGER
	VIEW	VIEW A FORM
ATOI		
	VALINP/CCKCHECK	FIELD
	VALINP/CCKCHECK	ITEM

FORMS DRIVEN FORM EDITOR Where-external-routine-used L

System Module	Module Name	Module Purpose
BLEN		
	CHKFLD	CHECK FIELD
	FLSTRC	FIELD STRUCTURE TRANSLATION
	GETFLS	GET FDL SOURCE FlLE
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	WRTFDL	WRITE FDL FILE
CLSFRM		
	VIEW	VIEW A FORM
COPFLD		
	EXPAND	EXPAND AN ARRAY
	EXPAND/FIXFIX	UP A FORM
COPY		
	FDFE	FORMS DRIVEN FORM EDITOR
DELFLD		
	DRPFRM	DROP FORM
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EXPAND/FIXFIX	UP A FORM
	FLANCI	FLAN CALLABLE INTERFACE
	FREBUF	FREE BUFFER
	MODFLD	MODIFY FIELD
	SAVFLS	SAVE FDL SOURCE
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
ESCPY		
	ADDEXT	ADD EXTENSION TO FILE_NAME
	CHKPRM	CHECK PARAMETER
	CPYFRM	COPY FORM
	FIFDST	FILL IN FIELD STRUCTURE
	GETFLS	GET FDL SOURCE FILE
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GTFDTX	GET FIELD TEXT
	GTNMFD	GET NAMED FIELD
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM
	MODFRM	MODIFY FORM
	SAVFLS	SAVE FDL SOURCE
	SCRMAN/GETGET	ROW
	VALINP/CCKCHECK	FIELD
	VALINP/CCKCHECK	FORM
	VALINP/CCKCHECK	HELP
	VALINP/CCKCHECK	ITEM
	VALINP/CCKCHECK	FOR RESERVED WORD
	WRTFDL	WRITE FDL FILE
FCLOSE		
	FDFE	FORMS DRIVEN FORM EDITOR
	GETFLS	GET FDL SOURCE FILE
	LISTIT	LIST IT
	SAVFLS	SAVE FDL SOURCE
FEOF		
	LISTIT	LIST IT
FERROR		

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	PRCFIL	PROCESS TEMPORARY FILE
FGETS		
	PRCFIL	PROCESS TEMPORARY FILE
FOPEN		
	FDFE	FORMS DRIVEN FORM EDITOR
	GETFLS	GET FDL SOURCE FILE
	LISTIT	LIST IT
	SAVFLS	SAVE FDL SOURCE
FPRINTF		
	WRTFDL	WRITE FDL FILE
FREE		
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GTFDTX	GET FIELD TEXT
	MODFLD	MODIFY FIELD
	MODFRM/FRE	FREE TEXT
	TRNSCR/FRL	FREE LOCATION STRUCTURES
	WRTEXP	WRITE EXPRESSION
GATDEF		
	FIFDST	FILL IN FIELD STRUCTURE
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
GDATA		
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	GFDINP	GET FIELD INPUT
	GWHINP	GET WHOLE INPUT
	SCRMAN	SCREEN MANAGER
GETCUR		
	SCRMAN	SCREEN MANAGER
	SCRMAN/CHG	CHANG POSITION
GWINDO		
	VIEW	VIEW A FORM
INITAL		
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
INITFP		
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
ISALPHA		
	VALINP/CCK	CHECK FIELD

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	VALINP/CCKCHECK	HELP
ISSPACE		
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/GTPGET	PROMPT INFORMATION
	TRNSCR/PARPARSE	SCREEN DATA
MAKFLD		
	INSFLD	INSERT FIELD
	INSFRM	INSERT FORM
MALLOC		
	GETFLS	GET FDL SOURCE FILE
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GTFDTX	GET FIELD TEXT
	MODFRM	MODIFY FORM
	MYALLOC	MY MALLOC
	TRNSCR/FLCFILL	LOCATION STRUCTURE
	TRNSCR/GTPGET	PROMPT INFORMATION
MATOI		
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GTFDTX/GTXGET	TEXT INFORMATION
	INSFLD	INSERT FIELD
	MODFLD	MODIFY FIELD
	MODFRM	MODIFY FORM
	SCRMAN/GETGET	ROW
	VALINP/CCKCHECK	FIELD
MAX		

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	GETFLS	GET FDL SOURCE FILE
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	INSFLD	INSERT FIELD
	SCRMAN/CHG	CHANG POSITION
	TRNSCR/GTP	GET PROMPT INFORMATION
	TRNSCR/MTC	MATCH PROMPT WITH FIELD
	TRNSTR/FLF	FILL FIELD
MEMCMP		
	CHKPRM	CHECK PARAMETER
	CPYFRM	COPY FORM
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
	GFDINP	GET FIELD INPUT
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GNXTFD	GET NEXT FIELD
	GWHINP	GET WHOLE INPUT
	INSWHL	INSERT WHOLE
	LAYOUT	LAYOUT MODE
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	MODWHL	MODIFY WHOLE
	PRSCMD	PARSE COMMAND
	PUTERR	PUT ERROR
	SCRMAN	SCREEN MANAGER
	SCRMAN/CHG	CHANG POSITION
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	VALINP	VALIDATE INPUT
	VALINP/CCK	CHECK FIELD
	VALINP/CCK	CHECK FORM
	VALINP/CCK	CHECK HELP

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	VALINP/CCKCHECK	ITEM
	VALINP/CCKCHECK	NAME
	VALINP/CCKCHECK	PROMPT
	VIEW	VIEW A FORM
MEMCPY		
	CHKFLD	CHECK FIELD
	CPYFRM	COPY FORM
	EDTFLD	EDIT FIELD
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	FLSTRC	FIELD STRUCTURE TRANSLATION
	GETFLS	GET FDL SOURCE FILE
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	GWHINP	GET WHOLE INPUT
	LISTFM	LIST FORMS
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/LDP	LOAD PROMPT INFORMATION
	WRTEXP	WRITE EXPRESSION
MEMDGT		
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
MEMSET		
	CHKFLD	CHECK FIELD
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	FLFMST	FIELD TO FORM STRUCTURE TRANSLATION
	FLSTRC	FIELD STRUCTURE TRANSLATION
	FLWHST	FILL WHOLE STRUCTURE
	GETFLS	GET FDL SOURCE FILE

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	LISTFM	LIST FORMS
	MODFLD	MODIFY FIELD
	PRCFIL	PROCESS TEMPORARY FILE
	PRSCMD	PARSE COMMAND
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSTR	TRANSLATE STRUCTURE TO SCREEN
MIN		
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE
	SCRMAN/CHG	CHANG POSITION
	TRNSCR/GTP	GET PROMPT INFORMATION
	TRNSCR/PAR	PARSE SCREEN DATA
	TRNSTR/FLF	FILL FIELD
	TRNSTR/FLP	FILL PROMPT
	VALINP/CCK	CHECK FIELD
	VALINP/CCK	CHECK FORM
MITOA		
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSCR/GTF	GET FORM PROMPT INFORMATION
	TRNSCR/LDP	LOAD PROMPT INFORMATION
MKTEMP		
	LISTIT	LIST IT
OISCR		
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	GFDINP	GET FIELD INPUT
	GWHINP	GET WHOLE INPUT
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	SCRMAN	SCREEN MANAGER
	VIEW	VIEW A FORM
PDATA		
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	GFDINP	GET FIELD INPUT
	GWHINP	GET WHOLE INPUT
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	SCRMAN	SCREEN MANAGER
PMSGLC		
	EDTFLD	EDIT FIELD
	EDTWHL	EDIT WHOLE
	PUTERR	PUT ERROR
	SCRMAN	SCREEN MANAGER
PMSGLS		
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	ERROR	ISSUE ERROR MESSAGE
	FATAL	ISSUE FATAL ERROR MESSAGE
	FDFE	FORMS DRIVEN FORM EDITOR
	INSFLD	INSERT FIELD
	INSWHL	INSERT WHOLE

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	MODFLD	MODIFY FIELD
	MODWHL	MODIFY WHOLE
	PUTERR	PUT ERROR
	SCRMAN	SCREEN MANAGER
	SCRMAN/CHG	CHANG POSITION
	TRNSTR/FLF	FILL FIELD
	VIEW	VIEW A FORM
	WARNING	ISSUE WARNING MESSAGE
PUTATT		
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	GWHINP	GET WHOLE INPUT
	PUTERR	PUT ERROR
PUTCUR		
	EDTFLD	EDIT FIELD
	GWHINP	GET WHOLE INPUT
	PUTERR	PUT ERROR
RENAME		
	FDFE	FORMS DRIVEN FORM EDITOR
	SAVFLS	SAVE FDL SOURCE
REWIND		
	SAVFLS	SAVE FDL SOURCE

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
RMVPAG		
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTMOD	EDIT MODE
	EDTWHL	EDIT WHOLE
	FDFE	FORMS DRIVEN FORM EDITOR
	GWHINP	GET WHOLE INPUT
	LISTFM	LIST FORMS
	LISTIT	LIST IT
	SCRMAN	SCREEN MANAGER
	VIEW	VIEW A FORM
RSVATT		
	EXPAND	EXPAND AN ARRAY
	EXPAND/FIX	FIX UP A FORM
SPRINTF		
	ADDEXT	ADD EXTENSION TO FILE_NAME
	DRPWHL	DROP WHOLE
	EDTFLD	EDIT FIELD
	EDTWHL	EDIT WHOLE
	ERROR	ISSUE ERROR MESSAGE
	FATAL	ISSUE FATAL ERROR MESSAGE
	FDFE	FORMS DRIVEN FORM EDITOR
	FLFMST	FIELD TO FORM STRUCTURE TRANSLATION
	FLSTRC	FIELD STRUCTURE TRANSLATION
	GETFLS	GET FDL SOURCE FILE
	GETFLS/TRE	TREE EXPRESSION
	GFDINP	GET FIELD INPUT
	GWHINP	GET WHOLE INPUT
	INSWHL	INSERT WHOLE
	LISTIT	LIST IT
	MODWHL	MODIFY WHOLE

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	PRCFIL	PROCESS TEMPORARY FILE
	SAVFLS	SAVE FDL SOURCE
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	TRNSTR/FLF	FILL FIELD
	VALINP	VALIDATE INPUT
	VALINP/CCK	CHECK FIELD
	VALINP/CCK	CHECK FORM
	VALINP/CCK	CHECK NAME
	WARNING	ISSUE WARNING MESSAGE
	WRTEXP	WRITE EXPRESSION
	WRTFDL/ARY	ARRAY REFERENCE
STRASN		
	CHKARY	CHECK ARRAY
	CHKFRM	CHECK FORM
STRCAT		
	ADDEXT	ADD EXTENSION TO FILE_NAME
	GETFLS	GET FDL SOURCE FILE
	GETFLS/TRE	TREE EXPRESSION
	LISTIT	LIST IT
	PUTERR	PUT ERROR
	SAVFLS	SAVE FDL SOURCE
	WRTFDL/ARY	ARRAY REFERENCE
STRCHR		
	PRCFIL	PROCESS TEMPORARY FILE
	VALINP/CCK	CHECK FIELD
	VALINP/CCK	CHECK FORM
	VALINP/CCK	CHECK HELP
	VALINP/CCK	CHECK ITEM

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
STRCMP		
	EXPAND/FIX	FIX UP A FORM
	FNDATT	FIND ATTRIBUTE
	GFLDPT	GET FIELD POINTER
	GTNMFD	GET NAMED FIELD
	VALINP/CCK	CHECK FIELD
	VALINP/CCK	CHECK HELP
	VALINP/CCK	CHECK FOR RESERVED WORD
	WRTFDL	WRITE FDL FILE
STRCPY		
	CSTASH	CHARACTER STASH
	FLSTRC	FIELD STRUCTURE TRANSLATION
	LISTIT	LIST IT
	PUTERR	PUT ERROR
	SAVFLS	SAVE FDL SOURCE
STRLEN		
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CSTASH	CHARACTER STASH
	EDTFLD	EDIT FIELD
	EDTWHL	EDIT WHOLE
	ERROR	ISSUE ERROR MESSAGE
	FATAL	ISSUE FATAL ERROR MESSAGE
	FDFE	FORMS DRIVEN FORM EDITOR
	GETFLS	GET FDL SOURCE FILE
	GTNMFD	GET NAMED FIELD
	LISTFM	LIST FORMS
	PRSCMD	PARSE COMMAND
	SCRMAN/GET	GET ROW
	TRNSCR	TRANSLATE SCREEN TO STRUCTURE
	VALINP/CCK	CHECK FIELD
	VALINP/CCK	CHECK HELP
	WARNING	ISSUE WARNING MESSAGE
	WRTEXP	WRITE EXPRESSION

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
STRNCMP		
	CPYFRM	COPY FORM
	PRSCMD	PARSE COMMAND
	VALINP/CCKCHECK	HELP
	WRTFDL	WRITE FDL FILE
STRNCPY		
	PRSCMD	PARSE COMMAND
STRRCHR		
	ADDEXT	ADD EXTENSION TO FILE_NAME
	GETFLS	GET FDL SOURCE FILE
	PRCFIL	PROCESS TEMPORARY FILE
	SAVFLS	SAVE FDL SOURCE
	SCRMAN/GETGET	ROW
STRSPN		
	VALINP/CCKCHECK	FIELD
	VALINP/CCKCHECK	HELP
STRUPC		
	VALINP/CCKCHECK	HELP
SYSMSG		
	CHKFLD	CHECK FIELD
	GETFLS	GET FDL SOURCE FILE
	GITMD	GET ITEM DATA AND INSERT IN STRUCTURE

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
	GTFDTX	GET FIELD TEXT
	LISTIT	LIST IT
	MODFRM	MODIFY FORM
	SAVFLS	SAVE FDL SOURCE
	VIEW	VIEW A FORM
SYSTEM		
	LISTIT	LIST IT
TERMFP		
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
TRMNAT		
	FDFE/MAIN	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
UNLINK		
	FDFE	FORMS DRIVEN FORM EDITOR
	LISTIT	LIST IT
WRTFRM		
	SAVFLS	SAVE FDL SOURCE
YYPARSE		
	FLANCI	FLAN CALLABLE INTERFACE

FORMS DRIVEN FORM EDITOR Where-external-routine-used List

System Module	Module Name	Module Purpose
------	------	-------

3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more that once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external "routine". The Purpose of the Main Program module is listed as well.

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
CHKFLD	Purpose-->	CHECK FIELD
	BLEN	External routine
	ERROR	Well-defined module
	FATAL	Well-defined module
	FNDATT	Well-defined module
	FREE	External routine
	MALLOC	External routine
	MAX	External routine
	MEMCPY	External routine
	MEMSET	External routine
	MYALLOC	Well-defined module
	PMSGLS	External routine
	SPRINTF	External routine
	STRCMP	External routine
	STRLEN	External routine
	SYSMSG	External routine
	WRTEXP	Well-defined module

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
CHKFRM	Purpose-->	CHECK FORM
	ABS	External routine
	ADDCHK	Well-defined module
	CHKARY	Well-defined module
	ERROR	Well-defined module
	FATAL	Well-defined module
	FLDTYP	Well-defined module
	FNDATT	Well-defined module
	FREE	External routine
	GFLDPT	Well-defined module
	MALLOC	External routine
	MAX	External routine
	MYALLOC	Well-defined module
	PMSGLS	External routine
	SPRINTF	External routine
	STRASN	External routine
	STRCMP	External routine
	STRLEN	External routine
	WARNING	Well-defined module

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
CSTASH	Purpose-->	CHARACTER STASH
	FATAL	Well-defined module
	MALLOC	External routine
	MYALLOC	Well-defined module
	PMSGLS	External routine
	SPRINTF	External routine
	STRCPY	External routine
	STRLEN	External routine

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
FDFE/MAIN	Purpose-->	MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
	ABORT	External routine
	ABS	External routine
	ACCESS	External routine
	ADDEXT	Well-defined module
	ADDFRM	External routine
	ATOI	External routine
	BLEN	External routine
	CHKPRM	Well-defined module
	CLSFRM	External routine
	COPY	External routine
	CPYFRM	Well-defined module
	DELFLD	External routine
	DRPFRM	Well-defined module
	DRPWHL	Well-defined module
	EDTFLD	Well-defined module
	EDTMOD	Well-defined module
	EDTWHL	Well-defined module
	ESCPY	External routine
	FCLOSE	External routine
	FDFE	Well-defined module
	FEOF	External routine
	FERROR	External routine
	FGETS	External routine
	FIFDST	Well-defined module
	FLANCI	Well-defined module
	FLFMST	Well-defined module
	FLSTRC	Well-defined module
	FLWHST	Well-defined module
	FNDATT	Well-defined module
	FOPEN	External routine
	FPRINTF	External routine
	FREBUF	Well-defined module
	FREE	External routine
	GATDEF	External routine
	GDATA	External routine
	GETCUR	External routine
	GETFLS	Well-defined module
	GETFLS/TREEXP	Well-defined module

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
	GETLEN	Well-defined module
	GFDINP	Well-defined module
	GITMD	Well-defined module
	GNXTFD	Well-defined module
	GNXTFD/NXTFLD	Well-defined module
	GTCPFD	Well-defined module
	GTFDTX	Well-defined module
	GTFDTX/GTXINF	Well-defined module
	GTNMFD	Well-defined module
	GWHINP	Well-defined module
	GWINDO	External routine
	INITAL	External routine
	INITFP	External routine
	INSFLD	Well-defined module
	INSFRM	Well-defined module
	INSWHL	Well-defined module
	ISALPHA	External routine
	ISSPACE	External routine
	LAYOUT	Well-defined module
	LISTFM	Well-defined module
	LISTIT	Well-defined module
	MAKFLD	External routine
	MALLOC	External routine
	MATOI	External routine
	MAX	External routine
	MEMCMP	External routine
	MEMCPY	External routine
	MEMDGT	External routine
	MEMSET	External routine
	MIN	External routine
	MITOA	External routine
	MKTEMP	External routine
	MODFLD	Well-defined module
	MODFRM	Well-defined module
	MODFRM/FRETXT	Well-defined module
	MODWHL	Well-defined module
	OISCR	External routine
	PDATA	External routine
	PMSGLC	External routine
	PMSGLS	External routine

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
	PRCFIL	Well-defined module
	PREC	Well-defined module
	PRSCMD	Well-defined module
	PUTATT	External routine
	PUTCUR	External routine
	PUTERR	Well-defined module
	RENAME	External routine
	REWIND	External routine
	RMVPAG	External routine
	SAVFLS	Well-defined module
	SCRMAN	Well-defined module
	SCRMAN/CHGPOS	Well-defined module
	SCRMAN/GETROW	Well-defined module
	SPRINTF	External routine
	STRCAT	External routine
	STRCHR	External routine
	STRCMP	External routine
	STRCPY	External routine
	STRLEN	External routine
	STRNCMP	External routine
	STRNCPY	External routine
	STRRCHR	External routine
	STRSPN	External routine
	STRUPC	External routine
	SYSMSG	External routine
	SYSTEM	External routine
	TERMFP	External routine
	TRMNAT	External routine
	TRNSCR	Well-defined module
	TRNSCR/FLCST	Well-defined module
	TRNSCR/FRLCST	Well-defined module
	TRNSCR/GTFMPMT	Well-defined module
	TRNSCR/GTPINF	Well-defined module
	TRNSCR/LDPMINF	Well-defined module
	TRNSCR/MTCHPMT	Well-defined module
	TRNSCR/PARSCRN	Well-defined module
	TRNSCR/SPSYMB	Well-defined module
	TRNSTR	Well-defined module
	TRNSTR/FLFLD	Well-defined module
	TRNSTR/FLPRMPT	Well-defined module

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
	TRNSTR/GARINF	Well-defined module
	UNLINK	External routine
	VALINP	Well-defined module
	VALINP/CCKFLD	Well-defined module
	VALINP/CCKFRM	Well-defined module
	VALINP/CCKHLP	Well-defined module
	VALINP/CCKITM	Well-defined module
	VALINP/CCKNAM	Well-defined module
	VALINP/CCKPRM	Well-defined module
	VALINP/CCKRSV	Well-defined module
	VIEW	Well-defined module
	WRTFDL	Well-defined module
	WRTFDL/ARYREF	Well-defined module
	WRTFRM	External routine
	YYPARSE	External routine

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
MAKINT	Purpose-->	MAKE EXPRESSION INTO AN INTEGER
	FATAL	Well-defined module
	MALLOC	External routine
	MYALLOC	Well-defined module
	PMSGLS	External routine
	SPRINTF	External routine
	STRLEN	External routine

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
MAKSTR	Purpose-->	MAKE EXPRESSION INTO A STRING
	FATAL	Well-defined module
	MALLOC	External routine
	MYALLOC	Well-defined module
	PMSGLS	External routine
	SPRINTF	External routine
	STRLEN	External routine

FORMS DRIVEN FORM EDITOR Main Program Parts List

Main Pgm Name	Module Name	Module Type
MKPOS	Purpose-->	MAKE POSITION NODE
	FATAL	Well-defined module
	MALLOC	External routine
	MYALLOC	Well-defined module
	PMSGLS	External routine
	SPRINTF	External routine
	STRLEN	External routine

3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME:	Name of program Module.
PURPOSE:	Purpose of Module as detailed in the source code.
LANGUAGE:	Programming language source code is written in. The choices are: VAX-11 FORTRAN C (I/S-1 Workbench 'C') VAX-11 COBOL
MODULE TYPE:	Whether a Program, Subroutine, or Function.
SOURCE FILE:	Name of Source File from file specification.
SOURCE FILE TYPE:	Source File Extension from file specification.
HOST:	Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
SUBSYSTEM:	IISS sub-system this file resides in.
SUBDIRECTORY:	Sub-directory of that subsystem in which this file resides.
DOCUMENTATION GROUP:	Name of documentation group of which this source file is a member.
DESCRIPTION:	A description of the module as obtained from the source code.
ARGUMENTS:	The arguments with which this routine is called if it is a Subroutine or a Function.
INCLUDE FILES:	A list of all the files that are included into this module as well as their purposes.

ROUTINES CALLED:	Subroutines or Functions, either documented or external, called by this module, if any.
CALLED DIRECTLY BY:	The documented routines which call this module, if any.
USED IN MAIN PROGRAM(S):	The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   ADDCHK
PURPOSE:                ADD POSITION TO CHECK LIST
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     VOID ADDCHK(POSPTR)
        POS *POSPTR;

  DESCRIPTION
     ADDS THE SPECIFIED POSITION TO THE OVERLAP CHECK LIST

ARGUMENTS:
----------
  POSPTR =        POS *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  FLDTYP      - FIELD TYPE
  ERROR       - ISSUE ERROR MESSAGE


CALLED DIRECTLY BY:
-------------------
  CHKFRM      - CHECK FORM


USED IN MAIN PROGRAM(S):
------------------------
  CHKFRM      - CHECK FORM
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   ADDEXT
PURPOSE:                ADD EXTENSION TO FILE_NAME
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            ADDEXT
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     VOID ADDEXT(NAME, FLG, FILE_NAME)
       ENAME    NAME;
       INT      FLG;
       CHAR     FILE_NAME[];


     INPUTS/OUTPUTS:


     INPUTS:
       ENAME    NAME; NAME OF FILE WITHOUT EXTENSION.
       INT      FLG;  TYPE OF EXTENSION AND DIRECTORY TO ADD.


     OUTPUTS:
       CHAR     FILE_NAME[] NEW NAME WITH EXTENSION ADDED.


  DESCRIPTION
    CONCATENATES THE SPECIFIED EXTENSION ONTO THE NAME AND
                 RETURNS
    IT IN FILE_NAME.

ARGUMENTS:
----------
  NAME =         CHAR []
  FLG =         INT
  FILE_NAME =          CHAR []


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

ESCPY
STRRCHR
STRCAT
SPRINTF

CALLED DIRECTLY BY:

FDFE - FORMS DRIVEN FORM EDITOR

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   CHKARY
PURPOSE:                CHECK ARRAY
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     VOID CHKARY(ARYPTR)
        FIELD *ARYPTR;

  DESCRIPTION
     GENERATES POSITIONS FOR EACH ELEMENT OF AN ARRAY FOR
                        OVERLAP CHECKING

ARGUMENTS:
----------
  ARYPTR =        FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  MYALLOC     - MY MALLOC
  ABS
  STRASN


CALLED DIRECTLY BY:
-------------------
  CHKFRM      - CHECK FORM


USED IN MAIN PROGRAM(S):
------------------------
  CHKFRM      - CHECK FORM
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   CHKFLD
PURPOSE:                CHECK FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     VOID CHKFLD()

  DESCRIPTION
     CHECKS THE CURRENT FIELD FOR COMPLETENESS AND CONSISTENCY


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  FNDATT      - FIND ATTRIBUTE
  ERROR       - ISSUE ERROR MESSAGE
  MEMSET
  MAX
  FREE
  WRTEXP      - WRITE EXPRESSION
  BLEN
  MEMCPY
  SYSMSG
  MYALLOC     - MY MALLOC
  STRLEN
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   CHKFRM
PURPOSE:                CHECK FORM
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     VOID CHKFRM()

  DESCRIPTION
     CHECKS THE CURRENT FORM FOR COMPLETENESS AND CONSISTENCY


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  WARNING     - ISSUE WARNING MESSAGE
  ADDCHK      - ADD POSITION TO CHECK LIST
  CHKARY      - CHECK ARRAY
  ABS
  STRLEN
  FREE
  FLDTYP      - FIELD TYPE
  ERROR       - ISSUE ERROR MESSAGE
  GFLDPT      - GET FIELD POINTER
  ABS
  MAX
  STRASN
  FNDATT      - FIND ATTRIBUTE
```

FORMS DRIVEN FORM EDITOR Module Documentation

NAME: CHKPRM
PURPOSE: CHECK PARAMETER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: BOOL ()
SOURCE FILE: CHKPRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FDFE
DOCUMENTATION GROUP: FDFE

DESCRIPTION:

SYNOPSIS
BOOL CHKPRM(PARAMTR,PARSIZ,RTPARM)
CHAR PARAMTR[];
INT PARSIZ;
CHAR RTPARM[];

INPUTS/OUTPUTS:

INPUTS:
PARAMTR - CHAR STING (NON NULL TERMINATED) CONTAINING
THE PARAMETER
BEING CHECKED FOR NOT BEING ENTERED.
PARSIZE - LENGTH OF CHARACTER STRING CONTAINING
PARAMETER

OUTPUTS:
RTPARM - NULL TERMINATED STRING CONTAINING PARAMETER
RETURNS A BOOL FALSE IF PARAMETER IS A BLANK

DESCRIPTION
CHECKS TO MAKF SURE THAT PARAMETER HAS BEEN ENTERED BY
USER AND COPYS
IT INTO A NULL TERMINATED STRING PROVIDED BY CALLER

ARGUMENTS:

PARAMTR = CHAR []
PARSIZ = INT
RTPARM = CHAR []

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS

```
  FPD          - FORM PROCESSOR DATA
  FDFEFM       - FDFE FORM DEFINITIONS
  FDFE         - FDFE DATA STRUCTURES

ROUTINES CALLED:
----------------
  MEMCMP
  ESCPY

CALLED DIRECTLY BY:
-------------------
  EDTMOD       - EDIT MODE
  FDFE         - FORMS DRIVEN FORM EDITOR

USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN  - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   CPYFRM
PURPOSE:                COPY FORM
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            CPYFRM
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    CHAR *CPYFRM(COPFLS,COPFRM)
         ENAME COPFLS;
         ENAME COPFRM;


     INPUTS/OUTPUTS:


     INPUTS:
       COPFLS - NON NULL TERMINATED STRING CONTAINING THE NAME
                    OF FLS FILE
                FROM WHICH USER WISHES TO COPY FORM.
       COPFLS - NON NULL TERMINATED STRING CONTAINING THE NAME
                    OF THE FORM.

     OUTPUTS:
       THIS ROUTINE FILLS (AND IS THE ONLY ROUTINE WHICH
                    WRITES TO THESE
       VARIABLES) SEVERAL GLOBAL VARIABLES:
          ALTBUF - BEINING OF LINK LIST CONTAINING ALL FORMS
                    ETC. OF FLS.
          ALTFRM - POINTER TO FORM WHICH THE USER WHISH TO
                    COPY INTO THE
                   SCREEN DATA AREA.
          COPYFLS - NAME OF FLS JUST COPIED - USED TO SAVE ON
                    CALLS TO
                   GETFLS.
          COPYFRM - NAME OF FORM JUST COPIED - USED TO SAVE ON
                    CALLS TO
                   GTNMFD.

  DESCRIPTION
          THIS ROUTINE IS USED TO COPY AN FLS FILE INTO AN
                    ALTERNATE INTERNAL
          DATA STRUCTURE AND GETS THE POINTER TO THE FORM
                    INTERESTED IN.  IS
          USED BY WHLEDT AND FLDEDT TO RETRIEVE DATA
```

CONCERNING A FIELD(S)
ON A FORM NOT NOW BEING WORKED ON IN AN FLS THAT IS
NOT BEING WORKED
ON, FOR POSSIBLE INSERTION BY THE USER.

ARGUMENTS:

COPFLS = ENAME
COPFRM = ENAME

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA
FDFEFM - FDFE FORM DEFINITIONS
FDFE - FDFE DATA STRUCTURES

ROUTINES CALLED:

MEMCMP
STRNCMP
ESCPY
GETFLS - GET FDL SOURCE FILE
GTNMFD - GET NAMED FIELD
MEMCPY

CALLED DIRECTLY BY:

EDTFLD - EDIT FIELD
EDTWHL - EDIT WHOLE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   CSTASH
PURPOSE:                CHARACTER STASH
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     CHAR *CSTASH(S)
        CHAR *S;

  DESCRIPTION
     SAVES THE SPECIFIED CHARACTER STRING AND RETURNS A
                     POINTER TO IT

ARGUMENTS:
----------
  S =        CHAR *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  STRCPY
  STRLEN
  MYALLOC     - MY MALLOC
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   DRPFRM
PURPOSE:                DROP FORM
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            DRPFRM
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    CHAR *DRPFRM(FRMNAM)
         ENAME FRMNAM;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMNAM - NON NULL TERMINATED STRING CONTAINING THE NAME
                      OF FORM TO
                BE DELETED.

     OUTPUTS:
       RETURNS A NULL IF FOUND THE FORM A ND DELETED IT AND
                      RETURNS ERROR
       CODE IF COULD NOT FIND FORM TO DELETE.

  DESCRIPTION
       DELETES A FORM NAMED BY USER FROM INTERNAL DATA
                      STRUCTURE - CALLS
       DELFLD TO DO ACTUAL WORK OF DELETING.

ARGUMENTS:
----------
  FRMNAM =        ENAME


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

GTNMFD - GET NAMED FIELD
DELFLD

CALLED DIRECTLY BY:

EDTMOD - EDIT MODE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   DRPWHL
PURPOSE:                DROP WHOLE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            DRPWHL
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *DRPWHL(FRMPNT,PFKEY)
          FIELD          *FRMPNT;
          INT            *PFKEY;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM FROM WHICH FIELDS MARKED WILL
                BE DROPPED

     OUTPUTS:
       PFKEY - RETURNS TO CALL THE PFKEY RECEIVED FROM OISCR.

  DESCRIPTION
       DROPS ALL FEILDS MARKED BY USER

ARGUMENTS:
----------
  FRMPNT =        FIELD *
  PFKEY =        INT *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  ADDFRM
```

MEMCMP
DELFLD
GDATA
PMSGLS
RMVPAG
OISCR
MEMSET
FLSTRC - FIELD STRUCTURE TRANSLATION
SPRINTF
PDATA
PUTATT

CALLED DIRECTLY BY:

EDTWHL - EDIT WHOLE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   EDTFLD
PURPOSE:                EDIT FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDTFLD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
      CHAR *EDTFLD(FRMPNT,RDONLY,MODE,ROW,COL)
          FIELD          *FRMPNT;
          BOOL           RDONLY;
          BOOL           MODE;
          INT            ROW;
          INT            COL;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM WORKING ON
       RDONLY - FLAG INDICATING WHETHER READ ONLY IS ALLOWED
                OR NOT
       MODE   - FLAG INDICATING WHICH MODE NOW IN(LAYOUT OR
                EDIT FIELD)
       ROW    - CURSOR POSITION COMING FROM LAYOUT (WILL BOTH
                BE 0 IF
       COL      IN FIELD EDIT MODE.

     OUTPUTS:
       IN EDIT FIELD MODE:
          RETURNS A NULL IF QUITING
          RETURNS AN ERROR CODE IF ABNORMALLY TERMINATED
       IN LAYOUT MODE:
          RETURNS A NULL IF GOING TO LAYOUT AGAIN
          RETURNS A EXITFDFE IF QUITING
          RETURNS AN ERROR CODE IF ABNORMALLY TERMINATED
  DESCRIPTION
       THIS IS THE DRIVER MODULE FOR FIELD EDIT MODE AND THE
                DETAILED EDIT
       MODE FOR LAYOUT. IT ALLOWS USER TO INSERT,MODIFY AND
                DELETE FIELDS
       ONE AT A TIME AS WELL AS MODIFYING FORM INFORMATION
                (PROMPT AND
```

BACKGROUND INFORMATION) AND TO COPY FIELD DATA FROM
ANOTHER FORM IN
ANTHER FLS FILE INTO SCREEN DATA AREA.

ARGUMENTS:

```
FRMPNT =      FIELD *
RDONLY =      BOOL
MODE =       BOOL
ROW =       INT
COL =       INT
```

INCLUDE FILES:

```
STDIO      - **** PURPOSE NOT FOUND BY STRIPPER ****
STDTYP     - STANDARD TYPE DEFINITIONS
CTYPE      - **** PURPOSE NOT FOUND BY STRIPPER ****
FPPARM     - FORM PROCESSOR PARAMETERS
FPCODE     - FORM PROCESSOR RETURN CODES
FPD        - FORM PROCESSOR DATA
FDFEFM     - FDFE FORM DEFINITIONS
FDFE       - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

```
ADDFRM
MEMCMP
RMVPAG
GDATA
PMSGLS
VALINP     - VALIDATE INPUT
PMSGLC
CPYFRM     - COPY FORM
GNXTFD     - GET NEXT FIELD
GFDINP     - GET FIELD INPUT
MODFLD     - MODIFY FIELD
INSFLD     - INSERT FIELD
MODFRM     - MODIFY FORM
PUTCUR
OISCR
PUTATT
SPRINTF
MEMSET
MEMCPY
STRLEN
FLSTRC     - FIELD STRUCTURE TRANSLATION
DELFLD
PDATA
GTCPFD     - GET USING CURSOR POSITION FIELD
FLFMST     - FIELD TO FORM STRUCTURE TRANSLATION
```

CALLED DIRECTLY BY:

EDTMOD - EDIT MODE
LAYOUT - LAYOUT MODE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   EDTMOD
PURPOSE:                EDIT MODE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDTMOD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *EDTMOD(NEWOLD,EDTFLAG)
          BOOL   NEWOLD;
          INT    EDTFLAG;


     INPUTS/OUTPUTS:


     INPUTS:
       NEWOLD - FLAG INDICATING WHETER FORM IS NEW OR AN OLD
                FORM
       EDTFLAG - FLAG IDICATING TYPE OF EDITING CHANGE/SELECT

     OUTPUTS:
       RETURNS NULL IF QUIT KEY PRESSED
       RETURNS EXITFDFE IF EXIT OPTION CHOSE.
       RETURNS ERROR CODE IF ABNOMALLY TERMINATED

  DESCRIPTION
        MAIN DRIVER MODULE FOR EDIT MODE. IN ADDITION TO
                ALLOWING
        USER TO CHOOSE EDIT MODE(WHOLE,FIELD,LAYOUT), USER CAN
                VIEW
        FORM, GET LIST OF FORMS IN FLS AND SAVE (AND COMPILE)
                SOURCE.

ARGUMENTS:
----------
  NEWOLD =        BOOL
  EDTFLAG =        INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
```

```
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  ADDFRM
  MEMCMP
  FREBUF      - FREE BUFFER
  PRSCMD      - PARSE COMMAND
  CHKPRM      - CHECK PARAMETER
  MEMSET
  GDATA
  PMSGLS
  RMVPAG
  SAVFLS      - SAVE FDL SOURCE
  DRPFRM      - DROP FORM
  INSFRM      - INSERT FORM
  LAYOUT      - LAYOUT MODE
  EDTFLD      - EDIT FIELD
  EDTWHL      - EDIT WHOLE
  GTNMFD      - GET NAMED FIELD
  LISTFM      - LIST FORMS
  OISCR
  PUTATT
  PDATA


CALLED DIRECTLY BY:
-------------------
  FDFE        - FORMS DRIVEN FORM EDITOR


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   EDTWHL
PURPOSE:                EDIT WHOLE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EDTWHL
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *EDTWHL(FRMPNT,RDONLY)
          FIELD          *FRMPNT;
          BOOL           RDONLY;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTS TO FORM WORKING ON
       RDONLY - FLAG INDICATING WHETHER OR NOT IN READ ONLY
                MODE

     OUTPUTS:
       RETURNS A ERROR CODE IF EXIT ABNORMALLY
       RETURNS A NULL IF EXIT NORMALLY

  DESCRIPTION
       THIS MODULE IS THE MAIN DRIVER MODULE FOR WHOLE EDT MODE

ARGUMENTS:
----------
  FRMPNT =        FIELD *
  RDONLY =        BOOL


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

MEMSET	
CPYFRM	- COPY FORM
FLWHST	- FILL WHOLE STRUCTURE
DRPWHL	- DROP WHOLE
ADDFRM	
MEMCMP	
GDATA	
PMSGLS	
RMVPAG	
GWHINP	- GET WHOLE INPUT
PMSGLC	
MODFRM	- MODIFY FORM
MODWHL	- MODIFY WHOLE
INSWHL	- INSERT WHOLE
VALINP	- VALIDATE INPUT
OISCR	
PUTATT	
SPRINTF	
MEMCPY	
STRLEN	
FLFMST	- FIELD TO FORM STRUCTURE TRANSLATION
PDATA	

CALLED DIRECTLY BY:

EDTMOD	- EDIT MODE

USED IN MAIN PROGRAM(S):

FDFE/MAIN	- MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

NAME: ERROR
PURPOSE: ISSUE ERROR MESSAGE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: FLANERR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FDFE
DOCUMENTATION GROUP: FDFE

DESCRIPTION:

SYNOPSIS

```
VOID ERROR(S, A, B, C, D, E, F)
   CHAR *S, *A, *B, *C, *D, *E, *F;
```

DESCRIPTION

PRINTS AN ERROR MESSAGE ON STDERR AND INCREMENTS THE NUMBER OF ERRORS

ARGUMENTS:

S =	CHAR *
A =	CHAR *
B =	CHAR *
C =	CHAR *
D =	CHAR *
E =	CHAR *
F =	CHAR *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

PMSGLS
STRLEN
SPRINTF

CALLED DIRECTLY BY:

CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM
ADDCHK - ADD POSITION TO CHECK LIST

USED IN MAIN PROGRAM(S):

CHKFLD	- CHECK FIELD
CHKFRM	- CHECK FORM

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   EXPAND
PURPOSE:                EXPAND AN ARRAY
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            EXPAND
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
      CHAR *EXPAND(FDP, USELST)
          FIELD *FDP;
          FIELD **USELST;


     INPUTS:
          FIELD *FDP;  ** THE FORM YOU WISH EXPANDED **
          FIELD **USELST; ** WHERE TO LOOK FOR EXPANDING
                          SUBFORMS


  DESCRIPTION
      THIS GUY IS RESPOSIBLE FOR EXPANDING AN ARRAY WHICH WAS
                          PARTIALLY
      CONSTRUCTED BY FLAN. IT TAKES A POINTER TO THE FORM TO
                          BE EXPANDED
      AND A POINTER TO THE POINTER TO THE LIST FROM WHICH
                          SUBFORMS MAY BE
      TAKEN. IF A SUBFORM IS NOT FOUND THE FIELD'S DISPLAY
                          ATTRIBUTE IS
      SET TO INPUT. THE CASE WHERE BOTH A FIELD AND THE
                          SUBFORM HAVE
      PROMPTS IS RESOLVED BE CREATING A SPECIAL FIELD TO HOLD
                          THE FIELD'S
      PROMPTS. USELST MUST BE A POINTER TO A POINTER BECAUSE
                          DELFLD IS USED
      AND THAT'S WHAT IT NEEDS.

ARGUMENTS:
----------
  FDP =         FIELD *
  USELST =         FIELD **
```

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

RSVATT	
FNDATT	- FIND ATTRIBUTE
ABS	
COPFLD	
EXPAND/FIXFRM	- FIX UP A FORM

CALLED DIRECTLY BY:

EXPAND/FIXFRM	- FIX UP A FORM

FORMS DRIVEN FORM EDITOR Module Documentation

NAME: EXPAND/FIXFRM
PURPOSE: FIX UP A FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: EXPAND
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FDFE
DOCUMENTATION GROUP: FDFE

DESCRIPTION:

FIXES A SUBFORM BY LOCATING IT AND ATTACHING IT IN PLACE AND EXPANDING IT IF REQUIRED.
NOTE: FIELDS WITH PROMPTS AND SUBFORMS WITH PROMPTS CAUSE A SPECIAL FIELD TO BE CREATED.

ARGUMENTS:

NDP = FIELD **
USELST = FIELD **

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

EXPAND - EXPAND AN ARRAY
DELFLD
RSVATT
FNDATT - FIND ATTRIBUTE
COPFLD
STRCMP

CALLED DIRECTLY BY:

EXPAND - EXPAND AN ARRAY

FORMS DRIVEN FORM EDITOR Module Documentation

NAME: FATAL
PURPOSE: ISSUE FATAL ERROR MESSAGE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: FLANERR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FDFE
DOCUMENTATION GROUP: FDFE

DESCRIPTION:

SYNOPSIS

```
VOID FATAL(S, A, B, C, D, E, F)
   CHAR *S, *A, *B, *C, *D, *E, *F;
```

DESCRIPTION

PRINTS A FATAL MESSAGE ON STDERR AND EXITS

ARGUMENTS:

S =	CHAR *
A =	CHAR *
B =	CHAR *
C =	CHAR *
D =	CHAR *
E =	CHAR *
F =	CHAR *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

SPRINTF
STRLEN
PMSGLS

CALLED DIRECTLY BY:

MYALLOC - MY MALLOC

USED IN MAIN PROGRAM(S):

CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM

CSTASH	- CHARACTER STASH
MAKINT	- MAKE EXPRESSION INTO AN INTEGER
MAKSTR	- MAKE EXPRESSION INTO A STRING
MKPOS	- MAKE POSITION NODE

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FDFE
PURPOSE:                FORMS DRIVEN FORM EDITOR
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            FDFE
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  LANGUAGE:  C

  DESCRIPTION:
  ------------

    DESCRIPTION
     MAIN INCLUDE FILE FOR FDFE


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
  FDFEINI     - FDFE INITIALIZATIONS
  FPDINI      - FPD INITIALIZATION
  NTM         - NTM INTERFACE INCLUDE FILE


ROUTINES CALLED:
----------------
  RMVPAG
  VIEW        - VIEW A FORM
  RENAME
  COPY
  GETFLS      - GET FDL SOURCE FILE
  EDTMOD      - EDIT MODE
  UNLINK
  FCLOSE
  FOPEN
  STRLEN
  MEMCPY
  ACCESS
  ADDEXT      - ADD EXTENSION TO FILE_NAME
  LISTIT      - LIST IT
  SPRINTF
```

CHKPRM - CHECK PARAMETER
PRSCMD - PARSE COMMAND
GDATA
OISCR
PMSGLS
PUTATT
MEMCMP
ADDFRM
MEMSET

CALLED DIRECTLY BY:

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FDFE/MAIN
PURPOSE:                MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR
                        (FDFE)
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            FDFE
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     FDFE()


     INPUTS/OUTPUTS:
          NONE

     INPUTS:


     OUTPUTS:


  DESCRIPTION
        MAIN DRIVER MODULE FOR THE FORMS DRIVEN FORMS EDITOR.
                        IT ALLOWS
        USER TO CHOOSE AMONG SEVERAL FILE MANAGEMENT OPTIONS
                        AS WELL AS
        EDIT MODES.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
  FDFEINI     - FDFE INITIALIZATIONS
  FPDINI      - FPD INITIALIZATION
  NTM         - NTM INTERFACE INCLUDE FILE


ROUTINES CALLED:
----------------
  INITAL
```

MEMCMP
INITFP
FDFE - FORMS DRIVEN FORM EDITOR
TERMFP
TRMNAT

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FIFDST
PURPOSE:                FILL IN FIELD STRUCTURE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FIFDST
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    CHAR *FIFDST(FLDPNT,FLDNAM,ROW,COL,WDTH,DPTH,FLDTYP,ATTNAM)
            FIELD   *FLDPNT;
            ENAME   FLDNAM;
            INT     ROW;
            INT     COL;
            INT     WDTH;
            INT     DPTH;
            CHAR    FLDTYP[1];
            ENAME   ATTNAM;


     INPUTS/OUTPUTS:


     INPUTS:
       FLDNAM - NAME OF FIELD CONCERNED
       ROW    - ROW OF FIELD CONCERNED
       COL    - COL OF FIELD CONCERNED
       WDTH   - WIDTH OF FIELD CONCERNED
       DPTH   - DEPTH OF FIELD CONCERNED
       FLDTYP - TYPE OF FIELD CONCERNED
       ATTNAM - ATTRIBUTE NAME OF FIELD CONCERNED

     OUTPUTS:
       NONE - BUT MODIFIES INTERNAL DATA STRUCTURE.

  DESCRIPTION
       THIS MODULE FILL IN THE FILD INFORMATION PASSED IT BY
                   CALLER.

ARGUMENTS:
----------
  FLDPNT =        FIELD *
  FLDNAM =        ENAME
  ROW =         INT
  COL =         INT
  WDTH =         INT
```

```
  DPTH =        INT
  FLDTYP =        CHAR [1]
  ATTNAM =        ENAME
```

INCLUDE FILES:

```
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPD         - FORM PROCESSOR DATA
```

ROUTINES CALLED:

```
  ESCPY
  GATDEF
```

CALLED DIRECTLY BY:

```
  MODFLD      - MODIFY FIELD
  MODFRM      - MODIFY FORM
```

USED IN MAIN PROGRAM(S):

```
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FLANCI
PURPOSE:                FLAN CALLABLE INTERFACE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     CHAR *FLANCI(FPTR)
        FILE *FPTR;

     INPUTS:
        FPTR - FILE TO BE COMPILED

  DESCRIPTION
     COMPILES THE SPECIFIED FILE INTO THE LOCAL OPEN LIST.

ARGUMENTS:
----------
  FPTR =        FILE *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  YYPARSE
  DELFLD


CALLED DIRECTLY BY:
-------------------
  GETFLS      - GET FDL SOURCE FILE
  SAVFLS      - SAVE FDL SOURCE


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FLDTYP
PURPOSE:                FIELD TYPE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
    CHAR *FLDTYP(C)
       CHAR C;

  DESCRIPTION
    RETURNS A STRING OF THE SPECIFIED FIELD TYPE

ARGUMENTS:
----------
  C =        CHAR


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


CALLED DIRECTLY BY:
-------------------
  CHKFRM      - CHECK FORM
  ADDCHK      - ADD POSITION TO CHECK LIST


USED IN MAIN PROGRAM(S):
------------------------
  CHKFRM      - CHECK FORM
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FLFMST
PURPOSE:                FIELD TO FORM STRUCTURE TRANSLATION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            FLFMST
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
       VOID FLFMST(DP, FP)
          FIELD *DP;
          STRUCT FRMINFO *FP;

     INPUTS:
       FIELD *DP; ** INPUT FPD FORM FIELD **
       STRUCT FRMINFO *FP; ** OUTPUT DISPLAY FORM **


  DESCRIPTION
      TRANSLATE FPD FIELD (DP) TO A FRMINFO (FP).

ARGUMENTS:
----------
  DP =        FIELD *
  FP =        STRUCT FRMINFO *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  SPRINTF
  MEMSET


CALLED DIRECTLY BY:
-------------------
  EDTFLD      - EDIT FIELD
  EDTWHL      - EDIT WHOLE
  TRNSCR      - TRANSLATE SCREEN TO STRUCTURE
```

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FLSTRC
PURPOSE:                FIELD STRUCTURE TRANSLATION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            FLSTRC
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    VOID FLSTRC(DP, COM, HELP, VAL, CHKS)
       FIELD *DP;
       STRUCT COMINFO *COM;
       STRUCT ITMHELP *HELP;
       STRUCT ITMVAL  *VAL;
       STRUCT ITMCHKS *CHKS;

     INPUTS:
       FIELD *DP; ** INPUT FIELD TO BE TRANSLATED **
       STRUCT COMINFO *COM; ** OUTPUT COMINFO/FLDINFO **
       STRUCT ITMHELP *HELP; ** OUTPUT ITMHELP **
       STRUCT ITMVAL  *VAL; ** OUTPUT ITMVAL **
       STRUCT ITMCHKS *CHKS; ** OUTPUT ITMCHKS **

  DESCRIPTION
     TRANSLATES AN FPD FIELD TO COMINFO/FLDINFO(COM)
                      STRUCTURE, ITMHELP(HELP)
     STRUCTURE, ITMVAL(VAL) STRUCTURE, AND ITMCHKS(CHKS)
                      STRUCTURE.

ARGUMENTS:
----------
  DP =         FIELD *
  COM =         STRUCT COMINFO *
  HELP =         STRUCT ITMHELP *
  VAL =         STRUCT ITMVAL *
  CHKS =         STRUCT ITMCHKS *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

SPRINTF
MEMSET
MEMCPY
BLEN
STRCPY

CALLED DIRECTLY BY:

DRPWHL	- DROP WHOLE
EDTFLD	- EDIT FIELD
FLWHST	- FILL WHOLE STRUCTURE
TRNSCR	- TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FLWHST
PURPOSE:                FILL WHOLE STRUCTURE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            FLWHST
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     VOID FLWHST(FRMPNT)
          FIELD         *FRMPNT;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM FROM WHICH INFORMATION WILL
                     RETIEVED AND
       AND PUT IN SCREEN DATA AREA.

     OUTPUTS:
       NONE - BUT DOES FILL EXTERNAL SCREEN DATA AREA AND
                     EXTERNAL ARAY OF
       POINTERS.

  DESCRIPTION
       THIS MODULE CALLS FLSTRC FOR EACH FIELD IN FORM AND
                     STORES POINTER TO
       FIELD IN PNTARY WHICH IS A GLOBAL ARRAY USEDED TO
                     ASSOCIATE EACH FIELD
       ON SCREEN WITH ITS INTERNAL STRUCTURE.

ARGUMENTS:
----------
  FRMPNT =        FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

MEMSET
FLSTRC - FIELD STRUCTURE TRANSLATION

CALLED DIRECTLY BY:

EDTWHL - EDIT WHOLE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FNDATT
PURPOSE:                FIND ATTRIBUTE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          ATTMAP * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     ATTMAP     *FNDATT(S)
        CHAR *S;

  DESCRIPTION
     RETURNS A POINTER TO THE SPECIFIED ATTRIBUTE IN THE
                        ATTRIBUTE MAP

ARGUMENTS:
----------
  S =         CHAR *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  STRCMP


CALLED DIRECTLY BY:
-------------------
  EXPAND/FIXFRM - FIX UP A FORM
  EXPAND      - EXPAND AN ARRAY
  VALINP/CCKFRM - CHECK FORM
  VALINP/CCKFLD - CHECK FIELD
  CHKFLD      - CHECK FIELD
  CHKFRM      - CHECK FORM
```

USED IN MAIN PROGRAM(S):

CHKFLD	- CHECK FIELD
CHKFRM	- CHECK FORM
FDFE/MAIN	- MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   FREBUF
PURPOSE:                FREE BUFFER
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            FREBUF
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     VOID FREBUF(BUFPNT)
          FIELD **BUFPNT;


     INPUTS/OUTPUTS:


     INPUTS:
       BUFPNT - ADDRESS OF BUFFER TO BE DELETED

     OUTPUTS:
       NONE

  DESCRIPTION
       THIS MODULE CALLS DELFLD FOR ALL FIELDS IN A
                        BUFFER(ALTBUF
       OR WRKBUF).

ARGUMENTS:
----------
  BUFPNT =        FIELD **


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  DELFLD
```

CALLED DIRECTLY BY:

EDTMOD	- EDIT MODE
GETFLS	- GET FDL SOURCE FILE

USED IN MAIN PROGRAM(S):

FDFE/MAIN	- MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GETFLS
PURPOSE:                GET FDL SOURCE FILE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            GETFLS
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
      CHAR *GETFLS(FRMNAM, PDP)
         CHAR FRMNAM[];
         FIELD **PDP;

     INPUTS:
       CHAR FRMNAM[]; ** THE NAME OF THE FILE CONTAINING THE
                      FDL SOURCE
                           THERE IS NO FILE EXTENSION **

     OUTPUTS:
       FIELD **PDP; ** A POINTER WHICH IS TO BE SET TO THE
                      'OPENED' FORMS

  DESCRIPTION
    GIVEN A FORM NAME OPENS THE FDL SOURCE FILE AND CALLS
                      FLANCI TO CREATE
    AN FPD STRUCTURE.

ARGUMENTS:
----------
  FRMNAM =        CHAR []
  PDP =         FIELD **


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  FREBUF      - FREE BUFFER
```

ESCPY
SPRINTF
STRRCHR
STRCAT
FOPEN
SYSMSG
BLEN
MEMSET
MEMCPY
FLANCI - FLAN CALLABLE INTERFACE
FCLOSE
GETFLS/TREEXP - TREE EXPRESSION
MAX
STRLEN
MALLOC

CALLED DIRECTLY BY:

CPYFRM - COPY FORM
FDFE - FORMS DRIVEN FORM EDITOR

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GETFLS/TREEXP
PURPOSE:                TREE EXPRESSION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            GETFLS
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  STATIC VOID TREEXP(STR, EP)
     CHAR STR[];
     ENODE *EP;

  CREATES AN ALGEBRAIC EXPRESSION GIVEN ITS SYNTATIC TREE
                        REPRESENTATION.
  THIS PROCEDURE IS ALSO USED BY GETFLS WHEN FETCHING SOURCE
                        FORMS.

ARGUMENTS:
----------
  STR =         CHAR []
  EP =         ENODE *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  PREC        - PRECEDENCE
  SPRINTF
  GETFLS/TREEXP - TREE EXPRESSION
  STRCAT


CALLED DIRECTLY BY:
-------------------
  GETFLS/TREEXP - TREE EXPRESSION
  GETFLS      - GET FDL SOURCE FILE
```

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GETLEN
PURPOSE:                GET LENGTH
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            GETLEN
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     INT GETLEN(BUF,BUFLEN)
          CHAR          BUF[];
          REGISTER INT BUFLEN;


     INPUTS/OUTPUTS:


     INPUTS:
       BUF    - NON NULL TERMINATED STRING THE 'NON BLANK'
                      LENGTH OF WHICH
                IS RETURNED
       BUFLEN - ACTUAL LENGTH OF BUFFER
     OUTPUTS:
       LENGTH (NON BLANK) OF STRING IS RETURNED

  DESCRIPTION
        THIS MODULE RETURNS THE "NON BLANK" LENGTH OF A NON
                      NULL TERMINATED
        STRING.

ARGUMENTS:
----------
  BUF =         CHAR []
  BUFLEN =         INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS


CALLED DIRECTLY BY:
-------------------
  GITMD       - GET ITEM DATA AND INSERT IN STRUCTURE
  GTFDTX/GTXINF - GET TEXT INFORMATION
  MODFRM      - MODIFY FORM
```

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GFDINP
PURPOSE:                GET FIELD INPUT
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            GFDINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *GFDINP(PFKEY,FRMPNT,FLDPNT)
         FIELD          *FRMPNT;
         FIELD          *FLDPNT;
         INT            *PFKEY;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - FORM WORKING ON
       FLDPNT - FIELD WORKING ON

     OUTPUTS:
       PFKEY - RETURN TO CALLER PFKE RECEIVED FROM OISCR
       RETRUNS NULL IF NOMALLY TERMINATED AND AN ERRROR CODE IF
       ABNORMALLY TERMINATED

  DESCRIPTION
       THIS MODULE GETS INPUT UPPERCASES FIELDS WHICH MUST BE
                      UPPER CASE
       AND CALLS VALINP IF PFKEY IS ENTER AND IF TASK = INSERT
                      OR MODIFY.

ARGUMENTS:
----------
  PFKEY =         INT *
  FRMPNT =         FIELD *
  FLDPNT =         FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
```

```
  FPD          - FORM PROCESSOR DATA
  FDFEFM       - FDFE FORM DEFINITIONS
  FDFE         - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

```
  PDATA
  MEMCMP
  VALINP       - VALIDATE INPUT
  GDATA
  OISCR
  SPRINTF
```

CALLED DIRECTLY BY:

```
  EDTFLD       - EDIT FIELD
```

USED IN MAIN PROGRAM(S):

```
  FDFE/MAIN    - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GFLDPT
PURPOSE:                GET FIELD POINTER
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          FIELD * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     FIELD *GFLDPT(FLDPTR, S)
        FIELD *FLDPTR;
        CHAR  *S;

  DESCRIPTION
     RETURN A POINTER TO THE NAMED FIELD ON THE SPECIFIED FORM.

ARGUMENTS:
----------
  FLDPTR =        FIELD *
  S =         CHAR *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  STRCMP


CALLED DIRECTLY BY:
-------------------
  CHKFRM      - CHECK FORM


USED IN MAIN PROGRAM(S):
------------------------
  CHKFRM      - CHECK FORM
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GITMD
PURPOSE:                GET ITEM DATA AND INSERT IN STRUCTURE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            GITMD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *GITMD (FLDPNT,ITMHLP,ITMVAL,ITMCHKS)
         FIELD           *FLDPNT;
         CHAR            *ITMHLP;
         CHAR            *ITMVAL;
         STRUCT ITMCHKS *ITMCHKS;


     INPUTS/OUTPUTS:


     INPUTS:
       FLDPNT - FIELD IN WHICH PUTTING INFORMATION RECIEVED
                      FROM USER
       ITMHLP - HELP LINE INPUT BY USER
       ITMVAL - ITEM DEFAULT VALUE INPUT BY USER
       ITMCHKS - CHECKING DESIRED BY USER

     OUTPUTS:
       RETURNS ERROR CODE IF ABNORMAL TERMINATION
       RETURNS NULL IF NORMAL TERMINATION

  DESCRIPTION
       THIS MODULE INSERTS DESIRED CHANGES INTO ITEM DATA
                      STRUCTURE OF
       INTERNAL DATA STRUCTURE

ARGUMENTS:
----------
  FLDPNT =        FIELD *
  ITMHLP =        CHAR *
  ITMVAL =        CHAR *
  ITMCHKS =        STRUCT ITMCHKS *


INCLUDE FILES:
--------------
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
```

```
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

```
  FREE
  MAX
  GETLEN      - GET LENGTH
  MALLOC
  SYSMSG
  ESCPY
  MEMDGT
  MATOI
  BLEN
  MEMSET
  MEMCPY
  MIN
  MEMCMP
```

CALLED DIRECTLY BY:

```
  INSFLD      - INSERT FIELD
  MODFLD      - MODIFY FIELD
```

USED IN MAIN PROGRAM(S):

```
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GNXTFD
PURPOSE:                GET NEXT FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          FIELD * ()
SOURCE FILE:            GNXTFD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    FIELD *GNXTFD(FRMPNT,FLDPNT,PFKEY)
          FIELD  *FRMPNT;
          FIELD  *FLDPNT;
          INT    PFKEY;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM WORKING ON
       FLDPNT - CURRENT FIELD POINTER
       PFKEY  - PFKEY PASSED BY CALLER FROM OISCR

     OUTPUTS:
       RETURNS - POINTER TO NEXT FIELD IF FOUND
       RETURNS - NULL IF NO FIELD FOUND

  DESCRIPTION
       THIS MODULE GETS THE NEXT FIELD TO BE VIEWED OR WORKED
                       ON
       USING USER INPUT

ARGUMENTS:
----------
  FRMPNT =        FIELD *
  FLDPNT =        FIELD *
  PFKEY =        INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

MEMCMP
GTNMFD - GET NAMED FIELD
GNXTFD/NXTFLD - NEXT FIELD

CALLED DIRECTLY BY:

EDTFLD - EDIT FIELD

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GNXTFD/NXTFLD
PURPOSE:                NEXT FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          FIELD * ()
SOURCE FILE:            GNXTFD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
    STATIC FIELD *NXTFLD(FLDPNT, TYP, DRCTN)
        FIELD *FLDPNT;
        CHAR  TYP;
        CHAR  DRCTN;


     INPUTS/OUTPUTS:


     INPUTS:
       FLDPNT - CURRENT FIELD POINTER
       TYP    - TYPE OF FIELD INTERESTED IN
       DRCTN  - DIRECTION WANT TO SEARCH

     OUTPUTS:
       RETURNS - POINTER TO NEXT FIELD IF FOUND
       RETURNS - NULL IF NO FIELD FOUND

  DESCRIPTION
       THIS MODULE GETS THE NEXT FIELD OF A SPECIFIED TYPE
                      SEARCHING IN A
       SPECIFIED DIRECTION TO BE VIEWED OR WORKED ON.

ARGUMENTS:
----------
  FLDPNT =        FIELD *
  TYP =         CHAR
  DRCTN =         CHAR


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

GNXTFD/NXTFLD - NEXT FIELD

CALLED DIRECTLY BY:

GNXTFD/NXTFLD - NEXT FIELD
GNXTFD - GET NEXT FIELD

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GTCPFD
PURPOSE:                GET USING CURSOR POSITION FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          FIELD * ()
SOURCE FILE:            GTCPFD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     FIELD *GTCPFD(FRMPNT,ROW,COL)
         FIELD *FRMPNT;
         INT   ROW;
         INT   COL;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM WORKING ON
       ROW    - CURSOR POSITION OF DESIRED FIELD
       COL
     OUTPUTS:
       RETURNS A POINTER TO FIELD AT INDICATED LOCATION IF ONE
       RETURNS A NULL IF NO FIELD FOUND

  DESCRIPTION
    THIS MODULE RETURNS A POINTER TO FIELD AT ROW AND COL
                      GIVEN BY CALLER

ARGUMENTS:
----------
  FRMPNT =        FIELD *
  ROW =         INT
  COL =         INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPD         - FORM PROCESSOR DATA
```

CALLED DIRECTLY BY:

EDTFLD - EDIT FIELD
SCRMAN/CHGPOS - CHANG POSITION
TRNSCR - TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GTFDTX
PURPOSE:                GET FIELD TEXT
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            GTFDTX
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *GTFDTX(FLDPNT,COMINFO)
          FIELD *FLDPNT;
          STRUCT COMINFO *COMINFO;

     INPUTS/OUTPUTS:


     INPUTS:
       FLDPNT   - POINTER TO FIELD WORKING ON
       COMINFO - POINTER TO STRUCTURE CONTAINING USER INPUT
     OUTPUTS:
       RETURNS NULL IF TERNIMATED NORMALLY
       RETURNS ERROR CODE IF TERMINATED ABNORMALLY

  DESCRIPTION
       THIS MODULE GETS TEXT FIELDS AND INSERTS THEM INTO
                      INTERNAL STRUCTRUE
       OF FIELD CONCERNED.

ARGUMENTS:
----------
  FLDPNT =          FIELD *
  COMINFO =          STRUCT COMINFO *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  FREE
```

GTFDTX/GTXINF - GET TEXT INFORMATION
MALLOC
SYSMSG
ESCPY

CALLED DIRECTLY BY:

INSFLD - INSERT FIELD
MODFLD - MODIFY FIELD

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GTFDTX/GTXINF
PURPOSE:                GET TEXT INFORMATION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            GTFDTX
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
    STATIC VOID GTXINF(COMINFO,PROW,PCOL,PLNGTH)
        STRUCT COMINFO *COMINFO;
        INT            *PROW, *PCOL, *PLNGTH;

     INPUTS/OUTPUTS:


     INPUTS:
       COMINFO - POINTER TO STRUCTURE CONTAINING USER INPUT
     OUTPUTS:
       PROW    - PROMPT (OR TEXT) ROW
       PCOL    - PROMPT (OR TEXT) COL
       PLNGTH  - PROMPT (OR TEXT) LENGTH


  DESCRIPTION
       THIS MODULE GETS TEXT FIELD INFORMATION AND PASSES IT
                    BACK TO CALLER

ARGUMENTS:
----------
  COMINFO =       STRUCT COMINFO *
  PROW =        INT *
  PCOL =        INT *
  PLNGTH =        INT *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

GETLEN - GET LENGTH
MATOI

CALLED DIRECTLY BY:

GTFDTX - GET FIELD TEXT

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GTNMFD
PURPOSE:                GET NAMED FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          FIELD * ()
SOURCE FILE:            GTNMFD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     FIELD *GTNMFD(FLDPNT,FLDNAM)
         FIELD *FLDPNT;
         ENAME FLDNAM;


     INPUTS/OUTPUTS:


     INPUTS:
       FLDPNT - POINTER TO FIRST FIELD IN LIST
       FLDNAM - NON NULL TERMINATED STRING WITH FIELD NAME
                    WANTED

     OUTPUTS:
       RETURNS A POINTER TO FIELD WITH NAME GIVEN IF FOUND
       RETURNS NULL IF NO FIELD FOUND WITH NAME GIVEN

  DESCRIPTION
       THIS MODULE RETURNS A POINTER TO FIELD STRUCTURE WITH
                     GIVEN NAME
       OR A NULL IF COULD NOT FIND SUCH A FIELD.

ARGUMENTS:
----------
  FLDPNT =        FIELD *
  FLDNAM =        ENAME


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

ESCPY
STRLEN
STRCMP

CALLED DIRECTLY BY:

CPYFRM	- COPY FORM
DRPFRM	- DROP FORM
EDTMOD	- EDIT MODE
GNXTFD	- GET NEXT FIELD
INSFRM	- INSERT FORM

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   GWHINP
PURPOSE:                GET WHOLE INPUT
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            GWHINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *GWHINP(FRMPNT,PFKEY)
         FIELD          *FRMPNT;
         INT            *PFKEY;


     INPUTS/OUTPUTS:

     INPUTS:
       FRMPNT - POINTER TO FORM BEING WORKED ON

     OUTPUTS:
       PFKEY  - PFKEY PASSED BACK TO CALLER FROM OISCR
       RETURNS NULL IF NORMAL TERMINATION
       RETURNS ERROR CODE IF ABNORMAL TERMINATION


  DESCRIPTION
     THIS MODULE GETS ALL INPUT FOR ALL SCREENS CALLING VALINP
     AND FLWHST FOR ALL SCREENS.

ARGUMENTS:
----------
  PFKEY =         INT *
  FRMPNT =         FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

ADDFRM
MEMCMP
MEMCPY
RMVPAG
VALINP - VALIDATE INPUT
PUTCUR
OISCR
SPRINTF
GDATA
PUTATT
PDATA

CALLED DIRECTLY BY:

EDTWHL - EDIT WHOLE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   INSFLD
PURPOSE:                INSERT FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            INSFLD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR

*INSFLD(FLDPNT,NXTFLD,PRVFLD,PARPNT,COMINFO,ITMHLP,ITMVAL,ITMCHK
S,LEVE
          FIELD             **FLDPNT;
          FIELD             **NXTFLD;
          FIELD             **PRVFLD;
          FIELD             *PARPNT;
          STRUCT COMINFO    *COMINFO;
          CHAR              *ITMHLP;
          CHAR              *ITMVAL;
          STRUCT ITMCHKS    *ITMCHKS;
          INT               LEVEL;


     INPUTS/OUTPUTS:

     INPUTS:
       FLDPNT   - ADDRESS OF POINTER TO FIELD BEING INSERTED
       PRVFLD   - ADDRESS WHERE PREVIOUS FIELD POINTER WILL BE
                      INSERTED
       NXTFLD   - ADDRESS WHERE NEXT FIELD POINTER WILL BE
                      INSERTED
       PARPNT   - POINTER TO PARENT OF FIELD
       COMINFO  - POINTER TO STRUCTURE CONTAINING INPUT FOR
                      GENERAL FIELD
                  INFORMATION OBTAINED FROM USER
       ITMHLP   - POINTER TO CHAR STRING CONTAINING HELP LINE
       ITMVAL   - POINTER TO CHAR STRING CONTAINING DEFAULT
                      VALUE FOR ITEM
       ITMCHKS  - POINTER TO STRUCTURE CONTAINING USER INPUT
                      FOR ITM CHECKS
       LEVEL    - LEVEL OF OF RECURSION

     OUTPUTS:
       RETURNS NULL IF NORMAL TERMINATION
       RETURNS ERROR CODE IF ABNORMAL TERMINATION
```

```
  DESCRIPTION
       THIS MODULE INSERTS FIELD INTO DATA STRUCTURE - CALLS
                      MAKFLD TO
       MAKE THE ACTUAL INSERTION THEN INSERTING PARTICULAR
                      INFORMATION
       FOR FIELD(WINDOW,ITEM,FORM,OR ARRAY)

ARGUMENTS:
----------
  FLDPNT =         FIELD **
  NXTFLD =         FIELD **
  PRVFLD =         FIELD **
  PARPNT =         FIELD *
  COMINFO =         STRUCT COMINFO *
  ITMHLP =         CHAR *
  ITMVAL =         CHAR *
  ITMCHKS =         STRUCT ITMCHKS *
  LEVEL =         INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MATOI
  INSFLD      - INSERT FIELD
  PMSGLS
  ESCPY
  GATDEF
  MAKFLD
  MAX
  GTFDTX      - GET FIELD TEXT
  GITMD       - GET ITEM DATA AND INSERT IN STRUCTURE


CALLED DIRECTLY BY:
-------------------
  EDTFLD      - EDIT FIELD
  INSFLD      - INSERT FIELD
  INSWHL      - INSERT WHOLE
  MODFLD      - MODIFY FIELD
  TRNSCR      - TRANSLATE SCREEN TO STRUCTURE


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   INSFRM
PURPOSE:                INSERT FORM
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            INSFRM
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *INSFRM(FRMNAM)
         ENAME FRMNAM;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMNAM - NON NULL TERMINATED STRING CONTAINING NAME OF
                     FRM TO BE
       INSERTED.

     OUTPUTS:
       RETURNS A NULL IF SUCCESSFUL
       RETURNS AN ERROR CODE IF FORM ALREADY EXIST OR
                     ABNORMALLY TERMINATED.

  DESCRIPTION
       THIS MODULE INSERTS TOP LEVEL FORM FILLING IN DEFAULT
                     VALUES

ARGUMENTS:
----------
  FRMNAM =        ENAME


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

GTNMFD - GET NAMED FIELD
ESCPY
GATDEF
MAKFLD

CALLED DIRECTLY BY:

EDTMOD - EDIT MODE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   INSWHL
PURPOSE:                INSERT WHOLE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            INSWHL
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *INSWHL(FRMPNT)
         FIELD          *FRMPNT;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM WORKING ON

     OUTPUTS:
       RETURNS NULL IF NORMAL TERMINATION
       RETURNS ERROR CODE IF ABNORMALLY TERMINATED

  DESCRIPTION
        THIS MODUL INSERTS ALL FIELDS USER ENTERED ON WHOLE
                     EDIT.
        IT USES THE GLOBAL DATA AREA.

ARGUMENTS:
----------
  FRMPNT =        FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MEMCMP
```

SPRINTF
PMSGLS
INSFLD - INSERT FIELD

CALLED DIRECTLY BY:

EDTWHL - EDIT WHOLE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   LAYOUT
PURPOSE:                LAYOUT MODE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            LAYOUT
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *LAYOUT(FRMPNT,RDONLY)
         FIELD *FRMPNT;
         BOOL   RDONLY;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM BEING WORKED ON
       RDONLY - FALG INDICATING WHETER IN READ ONLY MODE OR NOT

     OUTPUTS:
       RETURNS NULL IF NORMAL TERMINATION
       RETURNS ERROR CODE IF ABNORMAL TERMINATION

  DESCRIPTION
       THIS MODULE IS THE MAIN SWITCHER FOR LAYOUT MODE
                      BETWEEN ACTUAL
       LAYOUT

ARGUMENTS:
----------
  FRMPNT =          FIELD *
  RDONLY =          BOOL


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

SCRMAN - SCREEN MANAGER
MEMCMP
EDTFLD - EDIT FIELD

CALLED DIRECTLY BY:

EDTMOD - EDIT MODE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   LISTFM
PURPOSE:                LIST FORMS
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            LISTFM
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *LISTFM()


     INPUTS/OUTPUTS:


     INPUTS:
          NONE

     OUTPUTS:
       RETURNS NULL IF NORMAL TERMINATION
       RETURNS ERROR CODE IF ABNORMAL TERMINATION

  DESCRIPTION
       THIS MODULE LIST ALL FORMS IN CURRENT FLS FILE ON SCREEN


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MEMSET
  MEMCPY
  ADDFRM
  MEMCMP
  RMVPAG
  PMSGLS
```

OISCR
PDATA
STRLEN

CALLED DIRECTLY BY:

EDTMOD - EDIT MODE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   LISTIT
PURPOSE:                LIST IT
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            LISTIT
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    CHAR *LISTIT(TYPE)
       BOOL TYPE;


     INPUTS/OUTPUTS:


     INPUTS:
          CHAR *TYPE    POINTER TO THE CHARACTER STRING "FDL"
                        OR "FD"


     OUTPUTS:
          RETURNS SYSERR IF THE SYSTEM FUNCTION FAILS
                        OTHERWISE RETURNS A
          NULL POINTER.

  DESCRIPTION
       USES THE SYSTEM FUNCTION TO DO A DIRECTORY COMMAND FOR
                        *.FDL OR *.FD
       FILES AND REDIRECTS THE OUTPUT INTO A TEMPORARY FILE.
                        THEN CALLS PRCFIL
       TO READ THE FILE AND FILL IN THE FILENAMES ON THE
                        SCREEN.

ARGUMENTS:
----------
  TYPE =        BOOL


INCLUDE FILES:
--------------
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
```

```
  FPD          - FORM PROCESSOR DATA
  FDFEFM       - FDFE FORM DEFINITIONS
  FDFE         - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MKTEMP
  STRCPY
  STRCAT
  SPRINTF
  SYSMSG
  PRCFIL       - PROCESS TEMPORARY FILE
  PDATA
  ADDFRM
  MEMCMP
  FEOF
  FCLOSE
  UNLINK
  RMVPAG
  PMSGLS
  OISCR
  FOPEN
  SYSTEM


CALLED DIRECTLY BY:
-------------------
  FDFE         - FORMS DRIVEN FORM EDITOR


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN  - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   MAKINT
PURPOSE:                MAKE EXPRESSION INTO AN INTEGER
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          ENODE * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     ENODE *MAKINT(EP)
        ENODE *EP;

  DESCRIPTION
     CONVERT THE SPECIFIED EXPRESSION TO INTEGER AND RETURN
                        POINTER TO NEW
     EXPRESSION.

ARGUMENTS:
----------
  EP =        ENODE *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  MYALLOC     - MY MALLOC
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   MAKSTR
PURPOSE:                MAKE EXPRESSION INTO A STRING
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          ENODE * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     ENODE *MAKSTR(EP)
        ENODE *EP;

  DESCRIPTION
     CONVERT THE SPECIFIED EXPRESSION TO STRING AND RETURN
                        POINTER TO NEW
     EXPRESSION.

ARGUMENTS:
----------
  EP =          ENODE *


INCLUDE FILES:
--------------
  STDTYP        - STANDARD TYPE DEFINITIONS
  STDIO         - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD           - FORM PROCESSOR DATA
  RW            - REPORT WRITER DEFINITIONS
  FPCODE        - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  MYALLOC       - MY MALLOC
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   MKPOS
PURPOSE:                MAKE POSITION NODE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          POS * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     POS *MKPOS(HPOS, HMIN, HLOC, HREF, VPOS, VMIN, VLOC, VREF)
        INT HPOS, HMIN, HLOC;
        CHAR *HREF;
        INT VPOS, VMIN, VLOC;
        CHAR *VREF;

  DESCRIPTION
     CREATES THE SPECIFIED POSITION NODE AND ADDS IT TO THE
                        LIST.  HPOS AND
     VPOS ARE THE REFERENCE POINTS ON THE CURRENT FIELD, HMIN
                        AND VMIN ARE THE
     LOCATION RELATIVE TO THE REFERENCE FIELD, HLOC AND VLOC
                        ARE THE REFERENCE
     POINTS ON THE REFERENCE FIELD, AND HREF AND VREF ARE THE
                        REFERENCE
     FIELDS.

ARGUMENTS:
----------
  HPOS =          INT
  HMIN =          INT
  HLOC =          INT
  HREF =          CHAR *
  VPOS =          INT
  VMIN =          INT
  VLOC =          INT
  VREF =          CHAR *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
```

ROUTINES CALLED:

MYALLOC - MY MALLOC

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   MODFLD
PURPOSE:                MODIFY FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            MODFLD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *MODFLD(PARPNT,FLDPNT,COMINFO,ITMHLP,ITMVAL,ITMCHKS)
         FIELD           *PARPNT;
         FIELD           **FLDPNT;
         STRUCT COMINFO  *COMINFO;
         CHAR            *ITMHLP;
         CHAR            *ITMVAL;
         STRUCT ITMCHKS  *ITMCHKS;


     INPUTS/OUTPUTS:


     INPUTS:
       PARPNT   - POINTER TO PARENT FIELD
       FLDPNT   - ADDRESS OF POINNTER TO FIELD TO BE MODIFIED
       CCMINFO  - POINTER TO STRUCTURE CONTAINING USER INPUT
                      FOR GENERAL
                  FIELD INFORMATION
       ITMHLP   - POINTER TO CHAR STRING CONTAINING HELP LINE
                      INPUT BY USER
       ITMVAL   - POINTER TO DEFAULT VALUE FOR ITEMS INPUT BY
                      USER
       ITMCHKS  - POINTER TO STRUCTURE CONTAINING ITEM
                      VALIDATION CHECKS
                  INPUT BY USER

     OUTPUTS:
       RETURNS NULL IF NORMAL TERMINATION
       RETURNS ERROR CODE IF ABNORMAL TERMINATION

  DESCRIPTION
       THIS MODULE MODIFIES EXISTING FIELD IN ACCORDENCE WITH
                      USER INPUT

ARGUMENTS:
----------
  PARPNT =        FIELD *
```

```
  FLDPNT =        FIELD **
  COMINFO =        STRUCT COMINFO *
  ITMHLP =        CHAR *
  ITMVAL =        CHAR *
  ITMCHKS =        STRUCT ITMCHKS *
```

INCLUDE FILES:

```
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

```
  PMSGLS
  DELFLD
  INSFLD      - INSERT FIELD
  FREE
  MATOI
  FIFDST      - FILL IN FIELD STRUCTURE
  GTFDTX      - GET FIELD TEXT
  GITMD       - GET ITEM DATA AND INSERT IN STRUCTURE
  MEMSET
```

CALLED DIRECTLY BY:

```
  EDTFLD      - EDIT FIELD
  MODWHL      - MODIFY WHOLE
  TRNSCR      - TRANSLATE SCREEN TO STRUCTURE
```

USED IN MAIN PROGRAM(S):

```
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   MODFRM
PURPOSE:                MODIFY FORM
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            MODFRM
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *MODFRM(FRMPNT,FRMINFO)
        FIELD           *FRMPNT;
        STRUCT FRMINFO *FRMINFO;

     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT  - POINTER TO FORM'S INTERNAL STRUCTURE
       FRMINFO - ADDRESS OF USER INPUT
     OUTPUTS:
       RETURNS A NULL IF SUCCESS AND POINTER TO ERROR CODE
                    STRING IF ERROR

  DESCRIPTION
        THIS MODULE MODIFIES INTERNAL STRUCTURE FOR FORM
                     POINTED TO BY FRMPNT
        USING USER INPUT POINTED TO BY FRMINFO.

ARGUMENTS:
----------
  FRMPNT =        FIELD *
  FRMINFO =        STRUCT FRMINFO *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MATOI
```

```
  FIFDST      - FILL IN FIELD STRUCTURE
  MODFRM/FRETXT - FREE TEXT
  SYSMSG
  ESCPY
  GETLEN      - GET LENGTH
  MALLOC
```

CALLED DIRECTLY BY:

```
  EDTFLD      - EDIT FIELD
  EDTWHL      - EDIT WHOLE
  TRNSCR      - TRANSLATE SCREEN TO STRUCTURE
```

USED IN MAIN PROGRAM(S):

```
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   MODFRM/FRETXT
PURPOSE:                FREE TEXT
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            MODFRM
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     STATIC VOID FRETXT(FLDPNT)
        FIELD *FLDPNT;

     INPUTS/OUTPUTS:


     INPUTS:
       FLDPNT  - POINTER TO FIELD'S INTERNAL STRUCTURE
     OUTPUTS:
       NONE

  DESCRIPTION
       THIS MODULE FREES UP TEXT BUFFFER'S FOR THE FIELD
                      POINTED TO BY FLDPNT.

ARGUMENTS:
----------
  FLDPNT =        FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  FREE


CALLED DIRECTLY BY:
-------------------
  MODFRM      - MODIFY FORM
```

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   MODWHL
PURPOSE:                MODIFY WHOLE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            MODWHL
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *MODWHL(FRMPNT)
         FIELD    *FRMPNT;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM BEING WORKED ON

     OUTPUTS:
       RETURNS A NULL IF NORMAL TERMINATION
       RETURNS A ERROR CODE IF ABNORMAL TERMINATION

  DESCRIPTION
     THIS MODULE MODIFIES ALL FIELDS ON A FORM BY CALLING
                     MODFLD FOR EACH
     FIELD ON FORM

ARGUMENTS:
----------
  FRMPNT =        FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MODFLD      - MODIFY FIELD
```

MEMCMP
SPRINTF
PMSGLS

CALLED DIRECTLY BY:

EDTWHL - EDIT WHOLE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   MYALLOC
PURPOSE:                MY MALLOC
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     CHAR *MYALLOC(SIZE)
        UNSIGNED SIZE;

  DESCRIPTION
     ALLOCATE THE SPECIFIED MEMORY IF POSSIBLE, ELSE ISSUE
                        FATAL ERROR

ARGUMENTS:
----------
  SIZE =        UNSIGNED


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  FATAL       - ISSUE FATAL ERROR MESSAGE
  MALLOC


CALLED DIRECTLY BY:
-------------------
  CHKFLD      - CHECK FIELD
  CHKARY      - CHECK ARRAY
  CSTASH      - CHARACTER STASH
  WRTEXP      - WRITE EXPRESSION
  MKPOS       - MAKE POSITION NODE
  MAKINT      - MAKE EXPRESSION INTO AN INTEGER
  MAKSTR      - MAKE EXPRESSION INTO A STRING
```

USED IN MAIN PROGRAM(S):

CHKFLD	- CHECK FIELD
CHKFRM	- CHECK FORM
CSTASH	- CHARACTER STASH
MAKINT	- MAKE EXPRESSION INTO AN INTEGER
MAKSTR	- MAKE EXPRESSION INTO A STRING
MKPOS	- MAKE POSITION NODE

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   PRCFIL
PURPOSE:                PROCESS TEMPORARY FILE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            PRCFIL
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    CHAR *PRCFIL(FILE_PTR)
       FILE *FILE_PTR;


     INPUTS/OUTPUTS:


     INPUTS:
          FILE *FILE_PTR FILE POINTER TO THE TEMPORARY FILE
                         CONTAINING THE
          DIRECTORY LIST.


     OUTPUTS:


  DESCRIPTION
    PROCESSES THE TEMPORARY FILE FINDING ALL THE FILE NAMES
                       AND PLACES EACH
    ONE INTO THE SCREEN.

ARGUMENTS:
----------
  FILE_PTR =        FILE *


INCLUDE FILES:
--------------
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

STRCHR
FGETS
STRRCHR
SPRINTF
MEMSET
FERROR

CALLED DIRECTLY BY:

LISTIT - LIST IT

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   PREC
PURPOSE:                PRECEDENCE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            WRTFDL
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  INT PREC(EP)
     ENODE *EP;

  RETURNS THE PRECEDENCE OF AN EXPRESSION. THIS ROUTINE IS
                      USED IN
  TREEXP TO DETERMINE IF AN EXPRESSION REQUIRES PARENTHESIS.

ARGUMENTS:
----------
  EP =          ENODE *


INCLUDE FILES:
--------------
  STDTYP        - STANDARD TYPE DEFINITIONS
  STDIO         - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD           - FORM PROCESSOR DATA


ROUTINES CALLED:
----------------
  PREC          - PRECEDENCE


CALLED DIRECTLY BY:
-------------------
  GETFLS/TREEXP - TREE EXPRESSION
  PREC          - PRECEDENCE


USED IN MAIN PROGRAM(S):
------------------- ----
  FDFE/MAIN  - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   PRSCMD
PURPOSE:                PARSE COMMAND
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            PRSCMD
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     INT PRSCMD(CMD,PARSIZ,CMDNAM,PARAMTR,NUMPARM)
         CHAR          CMD[];
         INT           PARSIZ[MAXPARM];
         CHAR          *CMDNAM[];
         CHAR          *PARAMTR[MAXPARM];
         INT           *NUMPARM;


     INPUTS/OUTPUTS:


     INPUTS:
       CMD - COMMAND LINE
       PARSIZ   - ARRAY OF PARAMETER SIZES
       CMDNAM   - ARRAY OF CHAR POINTERS TO LEGITIMATE COMMANDS
     OUTPUTS:
       PARAMTR - PARAMETERS ARE RETURN TO THIS ARRAY
       NUMPARM - NUMBER OF PARAMETERS FOUND
       RETURNS INT VALUE CORRESPONDING TO OPTION CHOSEN

  DESCRIPTION
     THIS MODULE PARSES THE COMMAND LINE AND RETURNS INT NUM
                    CORRESPONDING TO
     OPTION AND PUTS IN 'NUMPARM' THE NUMBER OF PARAMETERS
                    FOUND AND IN
     PARAMTR THE ACTUAL PARAMETERS FOUND.

ARGUMENTS:
----------
  CMD =         CHAR []
  PARSIZ =         INT [MAXPARM ]
  CMDNAM =         CHAR * []
  PARAMTR =         CHAR * [MAXPARM ]
  NUMPARM =         INT *
```

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FDFEFM	- FDFE FORM DEFINITIONS
FDFE	- FDFE DATA STRUCTURES

ROUTINES CALLED:

STRNCPY
MEMCMP
MEMSET
STRNCMP
STRLEN

CALLED DIRECTLY BY:

EDTMOD	- EDIT MODE
FDFE	- FORMS DRIVEN FORM EDITOR

USED IN MAIN PROGRAM(S):

FDFE/MAIN	- MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

NAME: PUTERR
PURPOSE: PUT ERROR
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: BOOL ()
SOURCE FILE: VALINP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FDFE
DOCUMENTATION GROUP: FDFE

DESCRIPTION:

```
BOOL PUTERR(PATH, EXT, STR)
   CHAR PATH[], EXT[], STR[];
```

SET THE ERROR ATTRIBUTE FOR THE INVALID FIELD AND PUT THE CURSOR THERE.
PUTERR(PATH, EXTENSTION, EXPLANATION)

ARGUMENTS:

```
PATH =       CHAR []
EXT =       CHAR []
STR =       CHAR []
```

INCLUDE FILES:

```
STDTYP     - STANDARD TYPE DEFINITIONS
CTYPE      - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD        - FORM PROCESSOR DATA
FPCODE     - FORM PROCESSOR RETURN CODES
FPPARM     - FORM PROCESSOR PARAMETERS
FDFEFM     - FDFE FORM DEFINITIONS
FDFE       - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

```
STRCPY
STRCAT
PUTATT
MEMCMP
PMSGLC
PMSGLS
PUTCUR
```

CALLED DIRECTLY BY:

```
VALINP/CCKFRM - CHECK FORM
```

VALINP/CCKFLD - CHECK FIELD
VALINP/CCKHLP - CHECK HELP
VALINP/CCKITM - CHECK ITEM
VALINP/CCKNAM - CHECK NAME

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   SAVFLS
PURPOSE:                SAVE FDL SOURCE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            SAVFLS
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    CHAR *SAVFLS(FRMNAM, SAVFD, DP)
       CHAR FRMNAM[];
       BOOL SAVFD;
       FIELD *DP;


     INPUTS:
       CHAR FRMNAM[]; ** THE NAME THE FORM IS THE BE SAVED
                         UNDER
       BOOL SAVFD; ** IF TRUE THE .FD FILES ARE ALSO WRITTEN **
       FIELD *DP; ** POINTER TO LIST OF FORMS TO BE WRITTEN
                         OUT


  DESCRIPTION
    A LIST OF FORMS POINTED TO BY DP IS TO BE WRITTEN OUT AS
                         .FDL SOURCE.
    THE FORMS ARE FIRST WRITTEN TO A TEMPORARY FILE WHICH IS
                         FLANED TO
    CHECK FOR ERRORS IN VALUE EXPRESSIONS AND OVERLAPPING
                         FIELDS. IF
    THERE ARE NO ERRORS THE FILE IS RENAMED. IF THE SAVFD FLAG
                         IS TRUE
    THE .FD FILES ARE ALSO WRITTEN.

ARGUMENTS:
----------
  FRMNAM =        CHAR []
  SAVFD =        BOOL
  DP =        FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
```

```
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:
----------------
  ESCPY
  SPRINTF
  STRCPY
  STRRCHR
  STRCAT
  FOPEN
  SYSMSG
  FLANCI      - FLAN CALLABLE INTERFACE
  WRTFDL      - WRITE FDL FILE
  REWIND
  FCLOSE
  RENAME
  WRTFRM
  DELFLD

CALLED DIRECTLY BY:
-------------------
  EDTMOD      - EDIT MODE

USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   SCRMAN
PURPOSE:                SCREEN MANAGER
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            SCRMAN
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE
```

DESCRIPTION:

SYNOPSIS

```
CHAR *SCRMAN(FRMPNT,RDONLY,ROW,COL)
    FIELD *FRMPNT;
    BOOL   RDONLY;
    INT    *ROW;
    INT    *COL;
```

INPUTS/OUTPUTS:

INPUTS:
- FRMPNT - POINTER TO FORM WORKING ON
- RDOLNY - FLAG INDICATING WHETHER IN READ ONLY MODE OR NOT

OUTPUTS:
- ROW - ROW RECEIVED FROM GETCUR ROUTINE
- COL - COL RECEIVED FROM GETCUR ROUTINE

DESCRIPTION

THIS MODUE MANAGES THE SCREEN LAYOUT MODE: CALLING ROUTINES TO
TRANSLATE INTERNAL STRUCTURE TO SCREEN LAYOUT FORMAT AND VICE VERSA
HANLING ALL INPUT THROUGH AN OISCR CHECKING THE VALUE OF PFKEY RETURNED
BY OISCR AND ERORR CODES RETURNED BY TRANLATE ROUTINES TO DETERMINE
WHAT ACTION SHOULD BE TAKEN.

ARGUMENTS:

```
FRMPNT =        FIELD *
RDONLY =        BOOL
ROW =       INT *
COL =       INT *
```

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
FPPARM	- FORM PROCESSOR PARAMETERS
FPCODE	- FORM PROCESSOR RETURN CODES
FPD	- FORM PROCESSOR DATA
FDFEFM	- FDFE FORM DEFINITIONS
FDFE	- FDFE DATA STRUCTURES

ROUTINES CALLED:

SCRMAN/CHGPOS	- CHANG POSITION
SCRMAN/GETROW	- GET ROW
TRNSCR	- TRANSLATE SCREEN TO STRUCTURE
ADDFRM	
MEMCMP	
RMVPAG	
GETCUR	
GDATA	
OISCR	
TRNSTR	- TRANSLATE STRUCTURE TO SCREEN
PMSGLS	
PMSGLC	
PDATA	

CALLED DIRECTLY BY:

LAYOUT	- LAYOUT MODE

USED IN MAIN PROGRAM(S):

FDFE/MAIN	- MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   SCRMAN/CHGPOS
PURPOSE:                CHANG POSITION
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            SCRMAN
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
    STATIC CHAR *CHGPOS(FRMPNT)
       FIELD          *FRMPNT;

     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM WORKING ON

     OUTPUTS:
       STANDARD FORM PROCESSOR RETURN CODE

  DESCRIPTION

       THIS MODUE ALTERS THE INTERNAL DATA STRUCTURE OF FIELD
                      MARKED SO THAT
       ITS NEW ROW =  ROW OBTAINED FROM GETCUR AND THE COL =
                      COL OBTAINED
       FROM GETCUR +  1. IT SENDS THE USER APPROPRIATE ERROR
                      MESSAGES IF HE/SHE
       MAKES AN ERROR (NOT MARKING A FIELD TO B MOVED, FOR
                      EXAMPLE).

ARGUMENTS:
----------
  FRMPNT =        FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

SCRMAN/GETROW - GET ROW
GTCPFD - GET USING CURSOR POSITION FIELD
MAX
MIN
ABS
TRNSTR - TRANSLATE STRUCTURE TO SCREEN
PMSGLS
GETCUR
MEMCMP

CALLED DIRECTLY BY:

SCRMAN - SCREEN MANAGER

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   SCRMAN/GETROW
PURPOSE:                GET ROW
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          INT ()
SOURCE FILE:            SCRMAN
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
    STATIC INT GETROW(FQ_NAM)
       PATH     FQ_NAM;

     INPUTS/OUTPUTS:


     INPUTS:
       FQ-NAM - FULLY QUALIFIED NAME RETURNED FROM GETCUR

     OUTPUTS:
       RETURNS ROW OR 0 IF NOT ARRAY

  DESCRIPTION

       THIS MODULE RETURNS ROW BASED ON ARRAY INDEX

ARGUMENTS:
----------
  FQ_NAM =        PATH


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  ESCPY
  STRRCHR
  MATOI
  STRLEN
```

CALLED DIRECTLY BY:

SCRMAN - SCREEN MANAGER
SCRMAN/CHGPOS - CHANG POSITION

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSCR
PURPOSE:                TRANSLATE SCREEN TO STRUCTURE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            TRNSCR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     CHAR *TRNSCR(FRMPNT)
         FIELD          *FRMPNT;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM BEING WORKED ON

     OUTPUTS:
       RETURN NULL IF NO ERRORS
       RETURN ERROR CODE IF NO EITHER USER(OVERLAP) ERRORS OR
                      SYSTEM ERRORS

  DESCRIPTION
     THIS MODULE VALIDATES AND TRANSLATES USER INPUT AT LAYOUT
                      MODE TO
     INTERNAL DATA STRUCTURE.

ARGUMENTS:
----------
  FRMPNT =        FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

MEMSET
TRNSCR/LDPMINF - LOAD PROMPT INFORMATION
INSFLD - INSERT FIELD
MODFLD - MODIFY FIELD
TRNSCR/FLCST - FILL LOCATION STRUCTURE
TRNSCR/PARSCRN - PARSE SCREEN DATA
MEMCMP
ISSPACE
TRNSCR/SPSYMB - SPECIAL SYMBOL CHECK
TRNSCR/GTFMPMT - GET FORM PROMPT INFORMATION
TRNSCR/GTPINF - GET PROMPT INFORMATION
TRNSCR/MTCHPMT - MATCH PROMPT WITH FIELD
TRNSCR/FRLCST - FREE LOCATION STRUCTURES
MODFRM - MODIFY FORM
DELFLD
GTCPFD - GET USING CURSOR POSITION FIELD
FLSTRC - FIELD STRUCTURE TRANSLATION
MITOA
SPRINTF
MEMCPY
STRLEN
FLFMST - FIELD TO FORM STRUCTURE TRANSLATION

CALLED DIRECTLY BY:

SCRMAN - *SCREEN MANAGER*

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSCR/FLCST
PURPOSE:                FILL LOCATION STRUCTURE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            TRNSCR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC CHAR *FLCST(ROW,COL,DPTH,WDTH)
         REGISTER INT ROW,COL,DPTH,WDTH;


     INPUTS/OUTPUTS:


     INPUTS:
       ROW  - ROW WHERE FIELD IS LOCATED
       COL  - COL WHERE FIELD IS LOCATED
       DPTH - DEPTH OF FIELD LOCATED
       WDTH - DEPTH OF FIELD LOCATED

     OUTPUTS:
       RETURN NULL IF NO ALLOCATION ERRORS
       RETURN ALCERR IF ALOCATION ERRROR.

  DESCRIPTION
     THIS MODULE CREATES AND FILLS FIELD LOCATION STRUCTURE.

ARGUMENTS:
----------
  ROW =         INT
  COL =         INT
  DPTH =         INT
  WDTH =         INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

MALLOC

CALLED DIRECTLY BY:

TRNSCR/PARSCRN - PARSE SCREEN DATA
TRNSCR - TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

NAME: TRNSCR/FRLCST
PURPOSE: FREE LOCATION STRUCTURES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: TRNSCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FDFE
DOCUMENTATION GROUP: FDFE

DESCRIPTION:

SYNOPSIS
STATIC VOID FRLCST()

INPUTS/OUTPUTS:

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION
THIS MODULE FREES UP ALL DATA LOCATION STRUCTURES ALLOCATED BY TRNSCR
AND ITS DEPENDENT STATIC SUB MODULES (ALL FLDLOC AND PMTLOC STRUCTURES).

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS
FPD	- FORM PROCESSOR DATA
FDFEFM	- FDFE FORM DEFINITIONS
FDFE	- FDFE DATA STRUCTURES

ROUTINES CALLED:

FREE

CALLED DIRECTLY BY:

TRNSCR	- TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSCR/GTFMPMT
PURPOSE:                GET FORM PROMPT INFORMATION
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            TRNSCR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC CHAR *GTFMPMT(ROW,COL,N)
        REGISTER INT ROW,COL,N;


     INPUTS/OUTPUTS:


     INPUTS:
       ROW - ROW OF PROMPT ASSOCIATED WITH FORM (ON SCREEN
                ARRAY)
       COL - COL OF PROMPT ASSOCIATED WITH FORM (ON SCREEN
                ARRAY)
       N   - NUMBER OF THE PROMPT ASSOCIATED WITH FORM

     OUTPUTS:
       NONE

  DESCRIPTION
     THIS MODUE GETS FORM PROMPT INFO FOR PROMPT AND PUTS IT
                INTO COMMON
     DATA STRUCTURE WHICH INSFRM AND MODFRM USE

ARGUMENTS:
----------
  ROW =        INT
  COL =        INT
  N =        INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

MITOA

CALLED DIRECTLY BY:

TRNSCR - TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSCR/GTPINF
PURPOSE:                GET PROMPT INFORMATION
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            TRNSCR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC CHAR *GTPINF(ROW,COL,DIRFLG,RCOL,FSTFLG,ERRFLG)
        REGISTER INT  ROW,COL,DIRFLG;
        INT           *RCOL;
        BOOL          *FSTFLG,*ERRFLG;


     INPUTS/OUTPUTS:


     INPUTS:
       ROW    - ROW WHERE PROMPT SYMBOL WAS FOUND
       COL    - COL WHERE PROMPT SYMBOL WAS FOUND
       DIRFLG - TYPE OF PROMPT (DIRECTION OF ASSOCIATION)
       FSTFLG - FIRST PROMPT IN COL FLAG



     OUTPUTS:
       RCOL   - LOCATION TO RESUME SEARCH THOUGH SCREEN ARRAY.
       FSTFLG - FIRST PROMPT IN COL FLAG
       ERRFLG - SET TO TRUE IF PROMPT IS OPEN ENDED

  DESCRIPTION
     THIS MODULE DETERMINES LOCATION OF PROMPT IF IT CAN AND
                    CREATE A
     PROMPT LOCATION STRUCTURE WITH LOCATION AND CONTENT OF
                    PROMPT.

ARGUMENTS:
----------
  ROW =         INT
  COL =         INT
  DIRFLG =         INT
  RCOL =         INT *
  FSTFLG =         BOOL *
  ERRFLG =         BOOL *
```

INCLUDE FILES:

STDTYP	- STANDARD TYPE DEFINITIONS
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS
FPD	- FORM PROCESSOR DATA
FDFEFM	- FDFE FORM DEFINITIONS
FDFE	- FDFE DATA STRUCTURES

ROUTINES CALLED:

MAX
MIN
MALLOC
ISSPACE
TRNSCR/SPSYMB - SPECIAL SYMBOL CHECK

CALLED DIRECTLY BY:

TRNSCR - TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSCR/LDPMINF
PURPOSE:                LOAD PROMPT INFORMATION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            TRNSCR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC VOID LDPMINF(FLCPNT)
        STRUCT FLDLOC *FLCPNT;

     INPUTS/OUTPUTS:


     INPUTS:
       FLCPNT - POINTER TO FIELD LOCATION STRUCTURE WHOSE
                      PROMPT IS
                BEING WORKED ON

     OUTPUTS:

  DESCRIPTION
     THIS MODULE LOADS PROMPT INFO OF LAYOUT FIELD INTO COMMON
                      DATA
     STRUCTURE USED BY AND PASSED TO INSFLD(MODFLD) FIELD ETC.

ARGUMENTS:
----------
  FLCPNT =        STRUCT FLDLOC *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MEMCPY
  MITOA
```

CALLED DIRECTLY BY:

TRNSCR - TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSCR/MTCHPMT
PURPOSE:                MATCH PROMPT WITH FIELD
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            TRNSCR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC BOOL MTCHPMT()


     INPUTS/OUTPUTS:


     INPUTS:
       NONE

     OUTPUTS:
       RETURNS MATCHED/NOT_MATCHED FLAG

  DESCRIPTION
     THIS MODULE MATCHES PROMPTS WITH FIELDS.  IF PROMPT IS
                     UNMATCHED OR
     AMBIGUOUSLY MATCHED RETURN FAILURE ELSE PUT POINTER TO
                     PROMPT LOCATION
     FIELD INTO FIELD LOCATION FIELD PROMT POINTER.


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MAX
```

CALLED DIRECTLY BY:

TRNSCR - TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSCR/PARSCRN
PURPOSE:                PARSE SCREEN DATA
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            TRNSCR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC CHAR *PARSCRN(ROW,COL,MAXCOL)
        REGISTER INT ROW,COL,MAXCOL;


     INPUTS/OUTPUTS:


     INPUTS:
       ROW    - ROW IN WHICH BEGINNING FIELD SYMBOL OCCURED
       COL    - COL IN WHICH BEGINNING FIELD SYMBOL OCCURED
       MAXCOL - MAX COL IN WHICH ENDING FIELD SYMBOL CA BE
                      FOUND WITHOUT
                AN OVERLAP ERROR OCCURING.

     OUTPUTS:
       RETURN NULL IF NO ERRORS
       RETURN ERROR CODE IF  EITHER OVERLAP ERRORS OR SYSTEM
                      ERRORS

  DESCRIPTION
     THIS MODULE PARSES SCREEN TO DETERMINE THE LOCATION OF
                      FIELDS
     WHILE CHECKING FOR OVERLAPPING OF OTHER FIELDS OR PROMPTS.

ARGUMENTS:
----------
  ROW =         INT
  COL =         INT
  MAXCOL =         INT


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
```

FPD - FORM PROCESSOR DATA
FDFEFM - FDFE FORM DEFINITIONS
FDFE - FDFE DATA STRUCTURES

ROUTINES CALLED:

TRNSCR/SPSYMB - SPECIAL SYMBOL CHECK
ISSPACE
TRNSCR/PARSCRN - PARSE SCREEN DATA
TRNSCR/FLCST - FILL LOCATION STRUCTURE
MIN

CALLED DIRECTLY BY:

TRNSCR/PARSCRN - PARSE SCREEN DATA
TRNSCR - TRANSLATE SCREEN TO STRUCTURE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSCR/SPSYMB
PURPOSE:                SPECIAL SYMBOL CHECK
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            TRNSCR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC BOOL SPSYMB(SCRCHR)
         CHAR SCRCHR;


     INPUTS/OUTPUTS:


     INPUTS:
       SCRCHR - CHARACTER TO BE CHECKED

     OUTPUTS:
       RETURNS TRUE/FALSE FLAG

  DESCRIPTION
     THIS MODULE RETURNS TURE IF CHAR A SPECIAL SYMBOL ELSE
                      RETURNS FALSE

ARGUMENTS:
----------
  SCRCHR =        CHAR


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


CALLED DIRECTLY BY:
-------------------
  TRNSCR/GTPINF - GET PROMPT INFORMATION
  TRNSCR/PARSCRN - PARSE SCREEN DATA
  TRNSCR      - TRANSLATE SCREEN TO STRUCTURE
```

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSTR
PURPOSE:                TRANSLATE STRUCTURE TO SCREEN
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            TRNSTR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
     VOID TRNSTR(FRMPNT,FLDPNT,RDONLY)
         FIELD *FRMPNT;
         FIELD *FLDPNT;
         BOOL  RDONLY;


     INPUTS/OUTPUTS:


     INPUTS:
       FRMPNT - POINTER TO FORM BEING WORKED ON
       FLDPNT - POINTER TO FIELD BEING WORKED ON IF NULL DO
                    ENTIRE FORM
       RDONLY - FLAG INDICATING WHETHER IN READ ONLY MODE OR
                    NOT

     OUTPUTS:
       NONE

  DESCRIPTION
       THIS MODULE TRANSLATES INTERNAL DATA STRUCTURE INTO
                    SCREEN LAYOUT
       FORMAT.

ARGUMENTS:
----------
  FRMPNT =        FIELD *
  FLDPNT =        FIELD *
  RDONLY =        BOOL


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
```

```
  FPD          - FORM PROCESSOR DATA
  FDFEFM       - FDFE FORM DEFINITIONS
  FDFE         - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MEMSET
  TRNSTR/FLPRMPT - FILL PROMPT
  TRNSTR/FLFLD - FILL FIELD


CALLED DIRECTLY BY:
-------------------
  SCRMAN       - SCREEN MANAGER
  SCRMAN/CHGPOS - CHANG POSITION


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN  - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSTR/FLFLD
PURPOSE:                FILL FIELD
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            TRNSTR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC VOID FLFLD(FLDPNT,RDONLY)
         FIELD *FLDPNT;
         BOOL  RDONLY;


     INPUTS/OUTPUTS:


     INPUTS:
       FLDPNT - POINTER TO FIELD BEING WORKED ON
       RDONLY - FLAG INDICATING WHETHER IN READ ONLY MODE OR
                NOT

     OUTPUTS:
       NONE

  DESCRIPTION
       THIS MODULE TRANSLATES INTERNAL DATA STRUCTURE OF ONE
                  FIELD INTO SCREEN
       LAYOUT FORMAT.

ARGUMENTS:
----------
  FLDPNT =        FIELD *
  RDONLY =        BOOL


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

TRNSTR/FLPRMPT - FILL PROMPT
PMSGLS
SPRINTF
MIN
MAX
TRNSTR/GARINF - GET ARRAY INFORMATION

CALLED DIRECTLY BY:

TRNSTR - TRANSLATE STRUCTURE TO SCREEN

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSTR/FLPRMPT
PURPOSE:                FILL PROMPT
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            TRNSTR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC VOID FLPRMPT(FLFPNT,TXTPNT,PRMSYM)
        FIELD           *FLDPNT;
        TEXT            *TXTPNT;
        CHAR            PRMSYM;

     INPUTS/OUTPUTS:


     INPUTS:
       FLDPNT - POINTER TO FIELD ON WHICH IS FOUND TEXT
                   (PROMPT)
       TXTPNT - POINTER TO TEXT (PROMPT) BEING WORKED ON
       PRMSYM - PROMT SYMBOL (LAYOUT PROMPT SYMBOLS)
     OUTPUTS:
       NONE

  DESCRIPTION
       THIS MODULE FILLS SCREEN STRUCTURE WIHT PROMPT STING
                   FROM TEXT
       STRUCTURE AT THE APPROPRIATE ROW AND COL WITH THE
                   APPROPRIATE SYMBOL
       PASSED DOWN IN PRMSYM.

ARGUMENTS:
----------
  FLDPNT =        FIELD *
  TXTPNT =        TEXT *
  PRMSYM =        CHAR


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

MIN

CALLED DIRECTLY BY:

TRNSTR/FLFLD - FILL FIELD
TRNSTR - TRANSLATE STRUCTURE TO SCREEN

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   TRNSTR/GARINF
PURPOSE:                GET ARRAY INFORMATION
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            TRNSTR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     STATIC VOID GARINF(FLDPNT, WDTH, DPTH, ARWDTH, ARDPTH)
        FIELD           *FLDPNT;
        INT             *WDTH, *DPTH, *ARWDTH, *ARDPTH;


     INPUTS/OUTPUTS:


     INPUTS:
       FLDPNT - POINTER TO ARRAY BEING WORKED ON
       RDONLY - FLAG INDICATING WHETHER IN READ ONLY MODE OR
                NOT

     OUTPUTS:
       WDTH   - WIDTH OF FIRST MEMBER OF ARRAY
       DPTH   - DEPTH OF FIRST MEMBER OF ARRAY
       ARWDTH - WIDTH OF WHOLE OF ARRAY
       ARDPTH - DEPTH OF WHOLE OF ARRAY

  DESCRIPTION
       THIS MODULE OBTAINS ROW, COL, WIDTH, DEPTH, OF FIRST
                MEMBER OF ARRAY
       AND THE WIDTH AND DEPTH OF ENTIRE ARRAY AND RETURNS
                THEM TO CALLER

ARGUMENTS:
----------
  FLDPNT =        FIELD *
  WDTH =        INT *
  DPTH =        INT *
  ARWDTH =        INT *
  ARDPTH =        INT *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES
```

```
  FPD          - FORM PROCESSOR DATA
  FDFEFM       - FDFE FORM DEFINITIONS
  FDFE         - FDFE DATA STRUCTURES

ROUTINES CALLED:
----------------
  ABS

CALLED DIRECTLY BY:
-------------------
  TRNSTR/FLFLD - FILL FIELD

USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN  - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   VALINP
PURPOSE:                VALIDATE INPUT
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            VALINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    BOOL VALINP(FDP, DP, EDTTYP, BLNKSTOP)
       FIELD *FDP;
       FIELD *DP;
       INT   EDTTYP;
       BOOL  BLNKSTOP;

     INPUTS:
       FIELD *FDP; ** POINTER TO FORM CONTAINING FIELD TO BE
                     VALIDATED
       FIELD *DP;  ** POINTER TO FIELD TO BE VALIDATED **
       INT   EDTTYP; ** INDICATES THE COMBINATION OF THINGS TO
                     VALIDATE
       BOOL  BLNKSTOP; ** TRUE IF CHECKING IS TO STOP ON THE
                     FIRST BLANK FIELD


  DESCRIPTION
    PERFORMS VALIDATION CHECKS ON FIELDS. FDP AND DP INDICATE
                     THE FORM
    AND FIELD TO BE VALIDATED. EDTTYP INDICATES THE
                     COMBINATION OF THINGS
    TO BE VALIDATED. THE OBJECTS TO BE VALIDATED ARE CONTAINED
                     IN GLOBAL
    DATA. IF THE OBJECTS PASS THE VALIDATION CHECKS VALINP
                     RETURNS A TRUE.

ARGUMENTS:
----------
  FDP =         FIELD *
  DP =        FIELD *
  EDTTYP =        INT
  BLNKSTOP =        BOOL


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
```

```
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MEMCMP
  SPRINTF
  VALINP/CCKITM - CHECK ITEM
  ABORT
  VALINP/CCKFRM - CHECK FORM
  VALINP/CCKHLP - CHECK HELP
  VALINP/CCKFLD - CHECK FIELD
  VALINP/CCKNAM - CHECK NAME


CALLED DIRECTLY BY:
-------------------
  EDTFLD      - EDIT FIELD
  EDTWHL      - EDIT WHOLE
  GFDINP      - GET FIELD INPUT
  GWHINP      - GET WHOLE INPUT


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

NAME: VALINP/CCKFLD
PURPOSE: CHECK FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: BOOL ()
SOURCE FILE: VALINP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FDFE
DOCUMENTATION GROUP: FDFE

DESCRIPTION:

```
STATIC BOOL CCKFLD(PATH, ARPATH, FLDPTR, DP, FDP)
   PATH PATH[], ARPATH[];
   STRUCT COMINFO *FLDPTR;
   FIELD *DP, *FDP;
```

VALIDATE A FIELD.
CCKFLD (FIELD PATH, ARRAY PATH, POINTER TO DISPLAY, POINTER TO FPD FIELD)

ARGUMENTS:

```
PATH =        PATH []
ARPATH =        PATH []
FLDPTR =        STRUCT COMINFO *
DP =        FIELD *
FDP =        FIELD *
```

INCLUDE FILES:

```
STDTYP      - STANDARD TYPE DEFINITIONS
CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD         - FORM PROCESSOR DATA
FPCODE      - FORM PROCESSOR RETURN CODES
FPPARM      - FORM PROCESSOR PARAMETERS
FDFEFM      - FDFE FORM DEFINITIONS
FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

```
MATOI
VALINP/CCKPRM - CHECK PROMPT
MIN
ATOI
SPRINTF
MEMCMP
FNDATT      - FIND ATTRIBUTE
STRCHR
```

STRCMP
VALINP/CCKRSV - CHECK FOR RESERVED WORD
STRLEN
STRSPN
PUTERR - PUT ERROR
ISALPHA
ESCPY

CALLED DIRECTLY BY:

VALINP - VALIDATE INPUT

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   VALINP/CCKFRM
PURPOSE:                CHECK FORM
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            VALINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE
```

DESCRIPTION:

```
  STATIC BOOL CCKFRM(PATH, FRMPTR)
     CHAR PATH[];
     STRUCT FRMINFO *FRMPTR;

  VALIDATE A FORM.
  CCKFRM (PATH TO FORM, POINTER TO DISPLAY FORM)
```

ARGUMENTS:

```
  PATH =        CHAR []
  FRMPTR =        STRUCT FRMINFO *
```

INCLUDE FILES:

```
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

```
  SPRINTF
  STRCHR
  MIN
  MEMCMP
  PUTERR      - PUT ERROR
  FNDATT      - FIND ATTRIBUTE
  ESCPY
```

CALLED DIRECTLY BY:

```
  VALINP      - VALIDATE INPUT
```

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   VALINP/CCKHLP
PURPOSE:                CHECK HELP
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            VALINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  STATIC BOOL CCKHLP(PATH, HLPPTR)
     PATH PATH;
     STRUCT ITMHELP *HLPPTR;

  VALIDATE HELP.
  CCKHLP(PATH TO FIELD, POINTER TO DISPLAY FOR HELP)

ARGUMENTS:
----------
  PATH =        PATH
  HLPPTR =        STRUCT ITMHELP *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  STRCMP
  STRSPN
  ISALPHA
  STRNCMP
  PUTERR      - PUT ERROR
  STRLEN
  STRUPC
  ESCPY
  MEMCMP
  STRCHR
```

CALLED DIRECTLY BY:

VALINP - VALIDATE INPUT

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   VALINP/CCKITM
PURPOSE:                CHECK ITEM
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            VALINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  STATIC BOOL CCKITM(PATH, ITMPTR)
     PATH PATH;
     STRUCT ITMCHCK *ITMPTR;

  VALIDATE ITEM DOMAIN TYPE STUFF.
  CCKITM(FIELD PATH, POINTER TO DISPLAY)

ARGUMENTS:
----------
  PATH =        PATH
  ITMPTR =        STRUCT ITMCHCK *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  ATOI
  ESCPY
  MEMCMP
  PUTERR      - PUT ERROR
  STRCHR


CALLED DIRECTLY BY:
-------------------
  VALINP      - VALIDATE INPUT
```

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   VALINP/CCKNAM
PURPOSE:                CHECK NAME
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            VALINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  STATIC BOOL CCKNAM(FDP, BLNKSTOP)
     FIELD *FDP;
     BOOL BLNKSTOP;

  CHECK FOR DUPLICATE NAMES ON WHLEDT

ARGUMENTS:
----------
  FDP =         FIELD *
  BLNKSTOP =          BOOL


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MEMCMP
  PUTERR      - PUT ERROR
  SPRINTF


CALLED DIRECTLY BY:
-------------------
  VALINP      - VALIDATE INPUT


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   VALINP/CCKPRM
PURPOSE:                CHECK PROMPT
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            VALINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  STATIC BOOL CCKPRM(POS)
     CHAR POS[];

  CHECK THE FIELD PROMPT LOCATION FOR A LEGAL POSITION.
  CCKPRM(STRING)

ARGUMENTS:
----------
  POS =         CHAR []


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  MEMCMP


CALLED DIRECTLY BY:
-------------------
  VALINP/CCKFLD - CHECK FIELD


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN  - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

NAME: VALINP/CCKRSV
PURPOSE: CHECK FOR RESERVED WORD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: BOOL ()
SOURCE FILE: VALINP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FDFE
DOCUMENTATION GROUP: FDFE

DESCRIPTION:

```
STATIC BOOL CCKRSV(FLDNAM)
   CHAR FLDNAM[];

CHECK THE FIELD NAME AGAINST THE RESERVED WORD LIST.
CCKRSV(STRING)
```

ARGUMENTS:

```
FLDNAM =        CHAR []
```

INCLUDE FILES:

```
STDTYP      - STANDARD TYPE DEFINITIONS
CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD         - FORM PROCESSOR DATA
FPCODE      - FORM PROCESSOR RETURN CODES
FPPARM      - FORM PROCESSOR PARAMETERS
FDFEFM      - FDFE FORM DEFINITIONS
FDFE        - FDFE DATA STRUCTURES
```

ROUTINES CALLED:

```
ESCPY
STRCMP
```

CALLED DIRECTLY BY:

```
VALINP/CCKFLD - CHECK FIELD
```

USED IN MAIN PROGRAM(S):

```
FDFE/MAIN  - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   VALINP/CCKVAL
PURPOSE:                CHECK VALUE
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          BOOL ()
SOURCE FILE:            VALINP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  STATIC BOOL CCKVAL(PATH, VALPTR)
     PATH PATH;
     STRUCT ITMVAL *VALPTR;

  VALIDATE AN EXPRESSION (ACTUALLY DONE BY FLAN).
  CCKVAL(FIELD TO PATH, POINTER TO DISPLAY FOR VALUE)

ARGUMENTS:
----------
  PATH =        PATH
  VALPTR =        STRUCT ITMVAL *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPPARM      - FORM PROCESSOR PARAMETERS
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   VIEW
PURPOSE:                VIEW A FORM
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            VIEW
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    CHAR *VIEW(FORMNAME)
       NAME FORMNAME;


     INPUTS/OUTPUTS:


     INPUTS:
          NAME FORMNAME NAME OF THE FORM TO BE VIEWED.


     OUTPUTS:


  DESCRIPTION
       RETURNS STANDARD FORM PROCESSOR ERROR STRINGS OR IF
                       SUCCESSFUL, A NULL
       POINTER.

ARGUMENTS:
----------
  FORMNAME =        NAME


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  FPPARM      - FORM PROCESSOR PARAMETERS
  FPCODE      - FORM PROCESSOR RETURN CODES
  FPD         - FORM PROCESSOR DATA
  FDFEFM      - FDFE FORM DEFINITIONS
  FDFE        - FDFE DATA STRUCTURES


ROUTINES CALLED:
----------------
  ADDFRM
```

MEMCMP
SYSMSG
CLSFRM
RMVPAG
GWINDO
PMSGLS
OISCR

CALLED DIRECTLY BY:

FDFE - FORMS DRIVEN FORM EDITOR

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   WARNING
PURPOSE:                ISSUE WARNING MESSAGE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            FLANERR
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  SYNOPSIS
     VOID WARNING(S, A, B, C, D, E, F)
        CHAR *S, *A, *B, *C, *D, *E, *F;

  DESCRIPTION
     PRINTS A WARNING MESSAGE ON STDERR

ARGUMENTS:
----------
  S =         CHAR *
  A =         CHAR *
  B =         CHAR *
  C =         CHAR *
  D =         CHAR *
  E =         CHAR *
  F =         CHAR *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS


ROUTINES CALLED:
----------------
  PMSGLS
  STRLEN
  SPRINTF


CALLED DIRECTLY BY:
-------------------
  CHKFRM      - CHECK FORM


USED IN MAIN PROGRAM(S):
------------------------
  CHKFRM      - CHECK FORM
```

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   WRTEXP
PURPOSE:                WRITE EXPRESSION
LANGUAGE:               C
MODULE TYPE:            FUNCTION
FUNCTION TYPE:          CHAR * ()
SOURCE FILE:            FLANSP
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FE
DOCUMENTATION GROUP:    FDFE/FLAN

DESCRIPTION:
------------
  SYNOPSIS
     CHAR *WRTEXP(EP)
        ENODE *EP;

     INPUTS:
        EP - EXPRESSION TO WRITE

     OUTPUTS:
        RETURNS A POINTER TO THE WRITTEN EXPRESSION OR NULL
                     FOR ERRORS

  DESCRIPTION
     RETURNS A POINTER TO THE CHARACTER STRING REPRESENTING
                     THE GIVEN
     EXPRESSION, OR NULL IF AN ERROR IS DETECTED.

ARGUMENTS:
----------
  EP =        ENODE *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA
  RW          - REPORT WRITER DEFINITIONS
  FPCODE      - FORM PROCESSOR RETURN CODES


ROUTINES CALLED:
----------------
  FREE
  WRTEXP      - WRITE EXPRESSION
  MEMCPY
  MYALLOC     - MY MALLOC
  STRLEN
  SPRINTF
```

CALLED DIRECTLY BY:

CHKFLD	- CHECK FIELD
WRTEXP	- WRITE EXPRESSION

USED IN MAIN PROGRAM(S):

CHKFLD	- CHECK FIELD

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   WRTFDL
PURPOSE:                WRITE FDL FILE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            WRTFDL
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------

  SYNOPSIS
    VOID WRTFDL(FRMPTR, FILPTR)
       FIELD *FRMPTR;
       FILE  *FILPTR;

     INPUTS:
       FIELD *FRMPTR; ** THE LIST OF FORMS FOR WHICH .FDL IS
                         TO BE WRITTEN
       FILE  *FILPTR; ** THE FILE TO WHICH THE FORMS ARE TO BE
                         WRITTEN

  DESCRIPTION
    GIVEN A POINTER TO A LIST OF FORMS (FRMPTR) AND A FILE
                         POINTER (FILPTR),
    THE SOURCE LANGUAGE REPRESENTATION OF THE FPD LIST IS
                         CREATED.

ARGUMENTS:
----------
  FRMPTR =        FIELD *
  FILPTR =        FILE *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA


ROUTINES CALLED:
----------------
  FPRINTF
  BLEN
  ESCPY
  STRCMP
  STRNCMP
  WRTFDL/ARYREF - ARRAY REFERENCE
```

CALLED DIRECTLY BY:

SAVFLS - SAVE FDL SOURCE

USED IN MAIN PROGRAM(S):

FDFE/MAIN - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)

FORMS DRIVEN FORM EDITOR Module Documentation

```
NAME:                   WRTFDL/ARYREF
PURPOSE:                ARRAY REFERENCE
LANGUAGE:               C
MODULE TYPE:            SUBROUTINE
FUNCTION TYPE:          VOID ()
SOURCE FILE:            WRTFDL
SOURCE FILE TYPE:       .C
HOST:
SUBSYSTEM:              UI
SUBDIRECTORY:           FDFE
DOCUMENTATION GROUP:    FDFE

DESCRIPTION:
------------
  STATIC VOID ARYREF(ARYDEF, DP)
     CHAR ARYDEF[];
     FIELD *DP;

  CREATES THE FORMS LANGUAGE REPRESENTATION OF AN ARRAY
                      SPECIFICATION
  FROM THE UID.CURFPD->

ARGUMENTS:
----------
  ARYDEF =        CHAR []
  DP =         FIELD *


INCLUDE FILES:
--------------
  STDTYP      - STANDARD TYPE DEFINITIONS
  STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
  FPD         - FORM PROCESSOR DATA


ROUTINES CALLED:
----------------
  ABS
  SPRINTF
  STRCAT


CALLED DIRECTLY BY:
-------------------
  WRTFDL      - WRITE FDL FILE


USED IN MAIN PROGRAM(S):
------------------------
  FDFE/MAIN   - MAIN MODULE FOR FORMS DRIVEN FORMS EDITOR (FDFE)
```

3.10.9 Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code. The language it is written in is also given.

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FDFE
PURPOSE: FDFE DATA STRUCTURES
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
MAIN INCLUDE FILE FOR FDFE

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FDFEFM
PURPOSE: FDFE FORM DEFINITIONS
LANGUAGE: C

DESCRIPTION:

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FDFEINI
PURPOSE: FDFE INITIALIZATIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INITIALIZING INCLUDE FILE FOR FDFE

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FFFV2
PURPOSE: FORM FILE FORMAT - VERSION 2
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
RECORD LAYOUTS FOR THE BINARY FORM DEFINITION FILE

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FLAN
PURPOSE: FLAN INTERNAL STRUCTURES
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
AUXILIARY DATA STRUCTURES USED BY FLAN.

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FPCODE
PURPOSE: FORM PROCESSOR RETURN CODES
LANGUAGE: C

DESCRIPTION:

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FPD
PURPOSE: FORM PROCESSOR DATA
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
DATA DEFINITIONS FOR ALL FORM PROCESSOR (INCLUDING MONITER) DATA.

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FPDINI
PURPOSE: FPD INITIALIZATION
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INITIALIZED VERSION OF UID FOR INCLUSION IN MAIN PROGRAM.

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: FPPARM
PURPOSE: FORM PROCESSOR PARAMETERS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION: THESE DATA DEFINITIONS ARE USED
IN THE FORM PROCESSOR ROUTINES.

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: NTM
PURPOSE: NTM INTERFACE INCLUDE FILE
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INCLUDE FILE FOR NTM INTERFACE

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: RW
PURPOSE: REPORT WRITER DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION

FORMS DRIVEN FORM EDITOR Include File Description

FILE NAME: STDTYP
PURPOSE: STANDARD TYPE DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
THIS FILE ENSURES THAT THE FOLLOWING STANDARD TYPES ARE AVAILABLE:

FLOAT	- SINGLE PRECISION FLOAT
DOUBLE	- DOUBLE PRECISION FLOAT
LONG	- 32 BIT (OR LARGER) SIGNED INTEGER
LBITS	- 32 BITS (OR MORE) FOR BIT MANIPULATION
INT	- NATURAL SIZE SIGNED INTEGER
UNSIGNED	- NATURAL SIZE UNSIGNED INTEGER
BOOL	- NATURAL SIZE LOGICAL (ZERO / NON-ZERO ONLY)
SHORT	- 16 BIT (OR LARGER) SIGNED INTEGER
USHORT	- 16 BIT (OR LARGER) UNSIGNED INTEGER
BITS	- 16 BITS (OR MORE) FOR BIT MANIPULATION
CHAR	- SINGLE MACHINE CHARACTER (REAL CHARACTERS ALWAYS POSITIVE)
TINY	- 8 BIT (OR LARGER) SIGNED INTEGER
UTINY	- 8 BIT (OR LARGER) UNSIGNED INTEGER
TBITS	- 8 BITS (OR MORE) FOR BIT MANIPULATION
TBOOL	- 8 BIT (OR LARGER) LOGICAL (ZERO / NON-ZERO ONLY)
METACHAR	- 16 BIT (OR LARGER) AUGMENTED CHARACTER (SIGNED)
VOID	- FUNCTION THAT RETURNS NO VALUE
FORTRAN	- STORAGE CLASS FOR FOREIGN (NON-C) ROUTINES OR C ROUTINES WHICH ARE CALLABLE FROM FOREIGN ROUTINES

SINCE NOT ALL COMPILERS SUPPORT USHORT, TINY, AND UTINY, THE FUNCTIONS
USHORT(), TINY(), AND UTINY() SHOULD BE USED WHENEVER REFERENCING THEM.

IN ADDITION, THE FOLLOWING UTILITY MACROS ARE DEFINED:

LURSHIFT(N, B)	- UNSIGNED LONG RIGHT SHIFT
MAX(A, B)	- MAXIMUM OF A AND B
MIN(A, B)	- MINIMUM OF A AND B

FORMS DRIVEN FORM EDITOR Include File Description

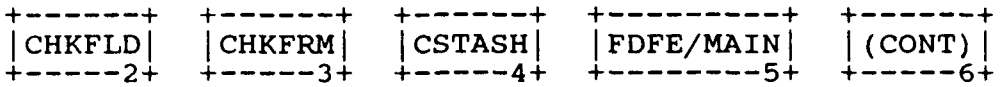
```
ABS(A)          - ABSOLUTE VALUE OF A
STRASN(A, B)    - TRANSPORTABLE A = B FOR STRUCTURES
NULL            - NULL POINTER VALUE (0)
TRUE            - 1
FALSE           - 0
SUCCESS         - EXIT(SUCCESS) INDICATES SUCCESSFUL
                  COMPLETION
FAILURE         - EXIT(FAILURE) INDICATES ERRORS

THE FOLLOWING SYMBOLS SHOULD BE DEFINED BASED ON THE
                  COMPILER BEING USED:
USHORT  - COMPILER SUPPORTS UNSIGNED SHORT
TINY    - COMPILER TREATS CHAR AS SIGNED
UTINY   - CHAR IS SIGNED AND COMPILER SUPPORTS
             UNSIGNED CHAR
VOID    - COMPILER SUPPORTS VOID
FORTRAN - COMPILER SUPPORTS FORTRAN
STRASN  - DEFINE APPROPRIATE MACRO
SUCCESS - DEFINE APPROPRIATE VALUE IF NOT 0
FAILURE - DEFINE APPROPRIATE VALUE IF NOT 1
```

3.10.10 Hierarchy Chart

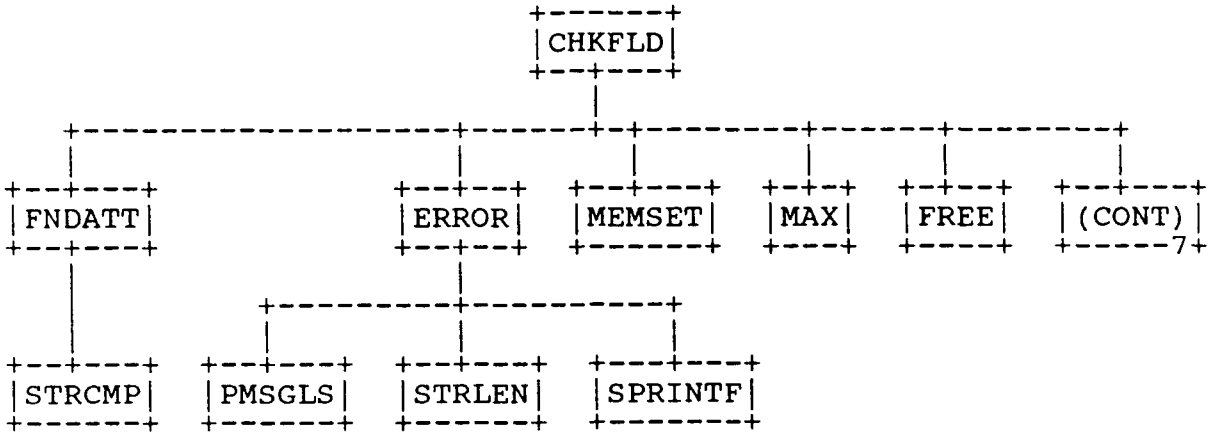
The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the first page are considered Main Programs as described in section 3.10.1 above.

There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.

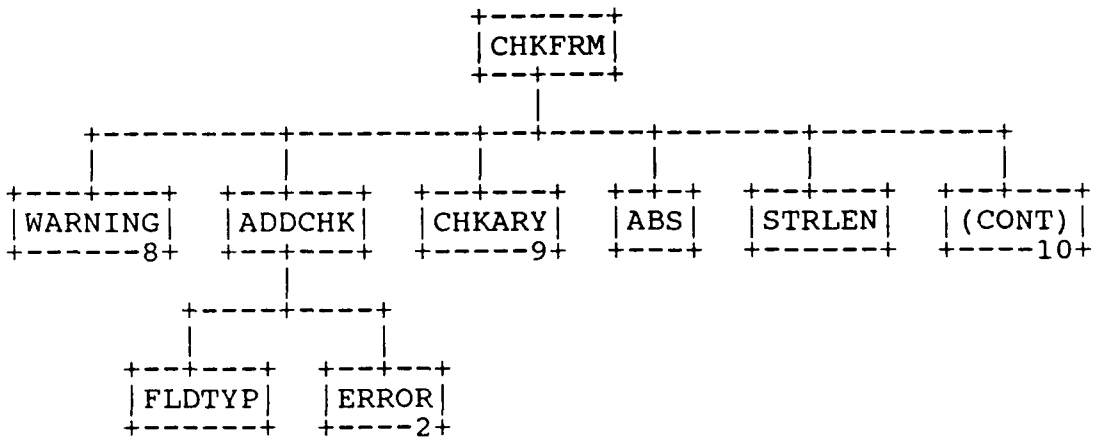
1


CHKFLD
CHKFRM
CSTASH
FDFE/MAIN
(CONT)
2
3
6

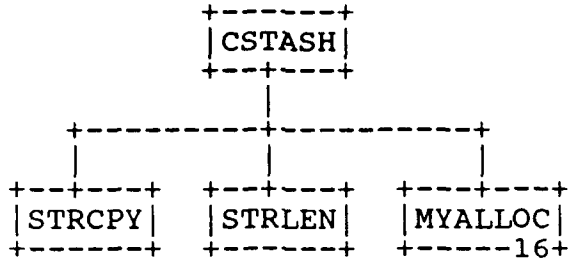
2


CHKFLD
FNDATT
ERROR
MEMSET
MAX
FREE
(CONT)
7
STRCMP
PMSGLS
STRLEN
SPRINTF

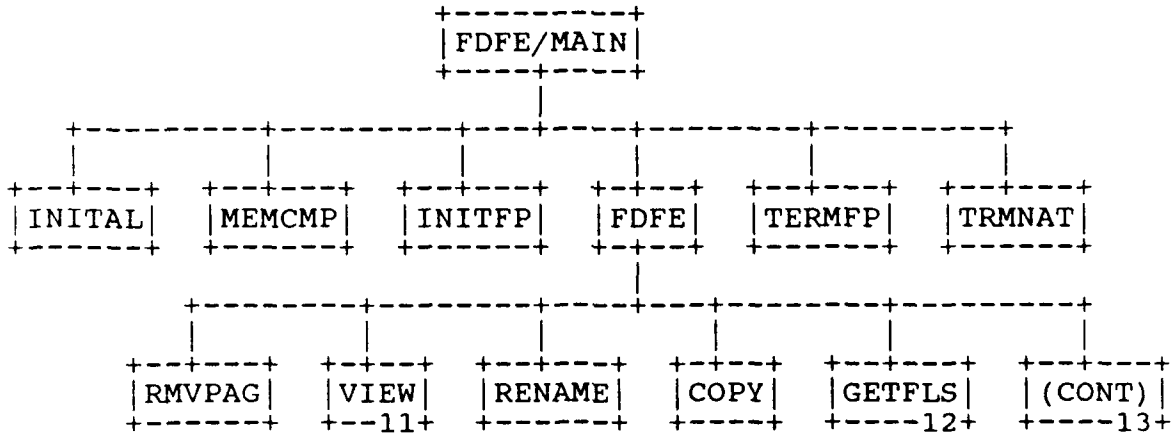
3


CHKFRM
WARNING
8
ADDCHK
CHKARY
9
ABS
STRLEN
(CONT)
10
FLDTYP
ERROR
2

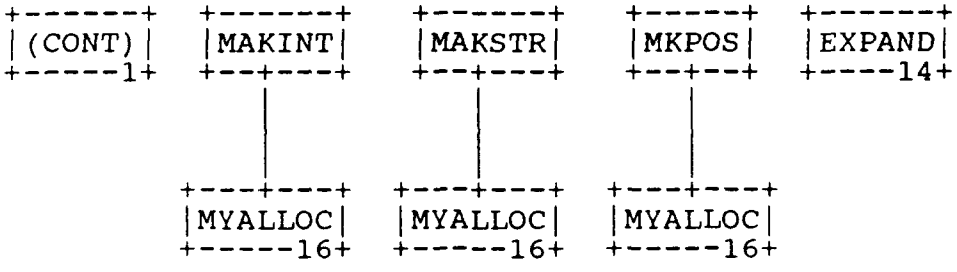
4


CSTASH
STRCPY
STRLEN
MYALLOC
16

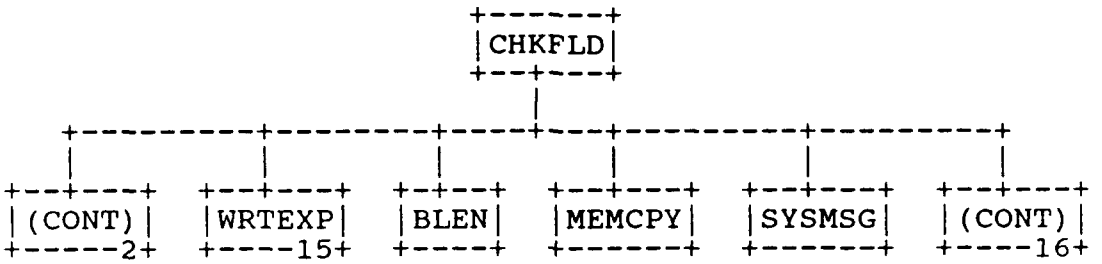
5


FDFE/MAIN
INITAL
MEMCMP
INITFP
FDFE
TERMFP
TRMNAT
RMVPAG
VIEW
11
RENAME
COPY
GETFLS
12
(CONT)
13

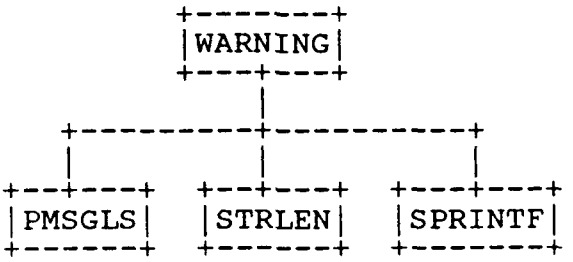
6


(CONT)
1
MAKINT
MAKSTR
MKPOS
EXPAND
14
MYALLOC
16
MYALLOC
16
MYALLOC
16

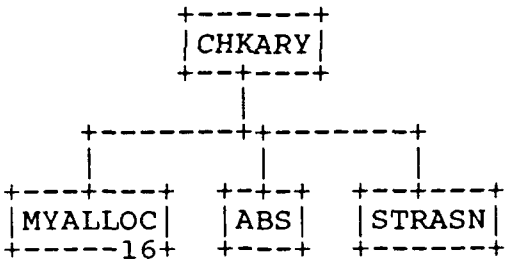
7


CHKFLD
(CONT)
2
WRTEXP
15
BLEN
MEMCPY
SYSMSG
(CONT)
16

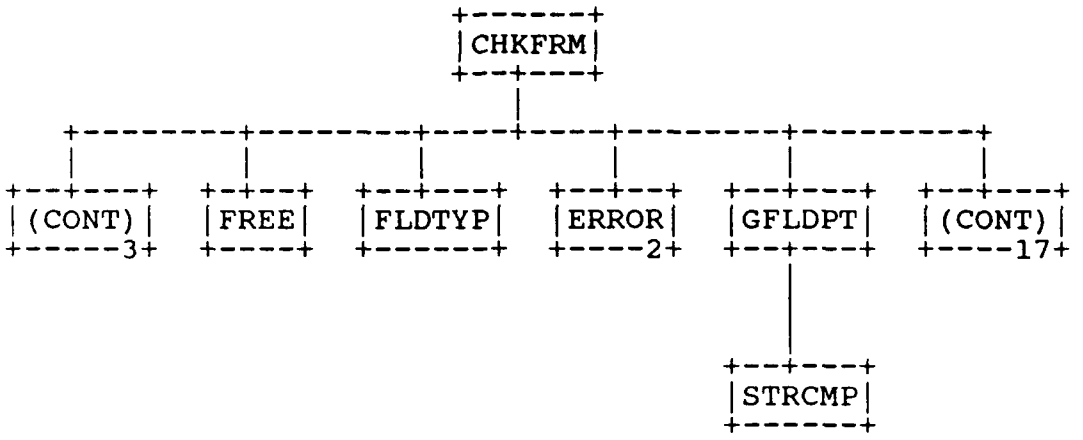
8


WARNING
PMSGLS
STRLEN
SPRINTF

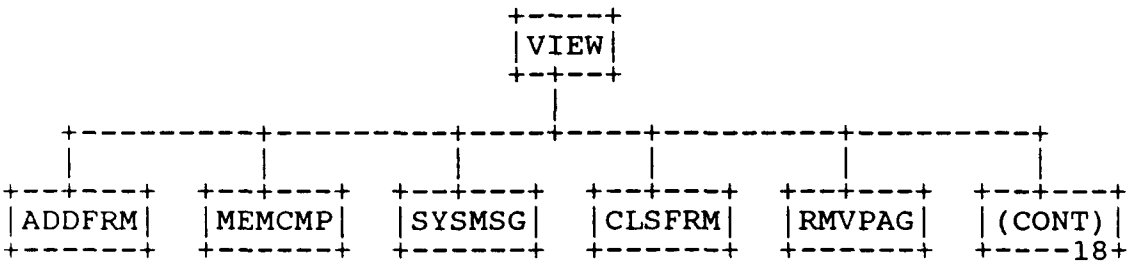
9


CHKARY
MYALLOC
16
ABS
STRASN

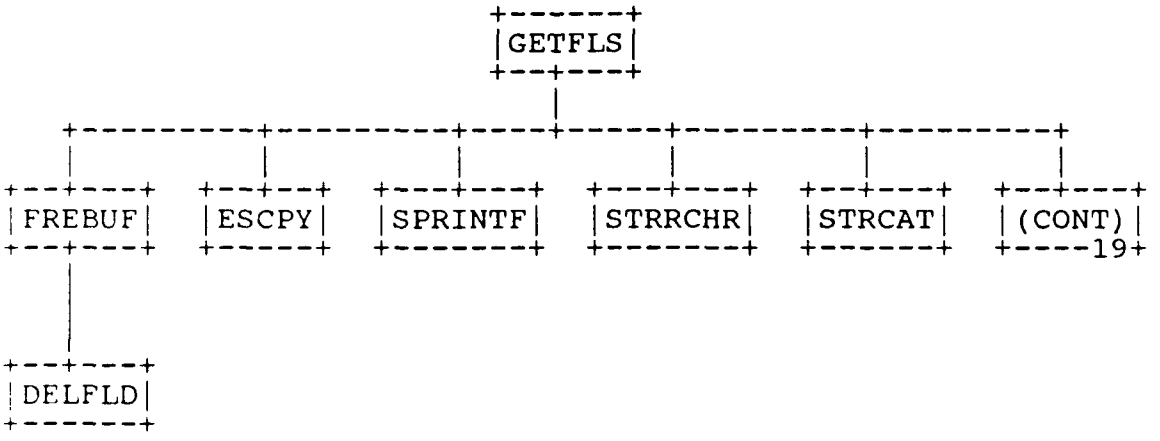
10


CHKFRM
(CONT)
3
FREE
FLDTYP
ERROR
2
GFLDPT
(CONT)
17
STRCMP

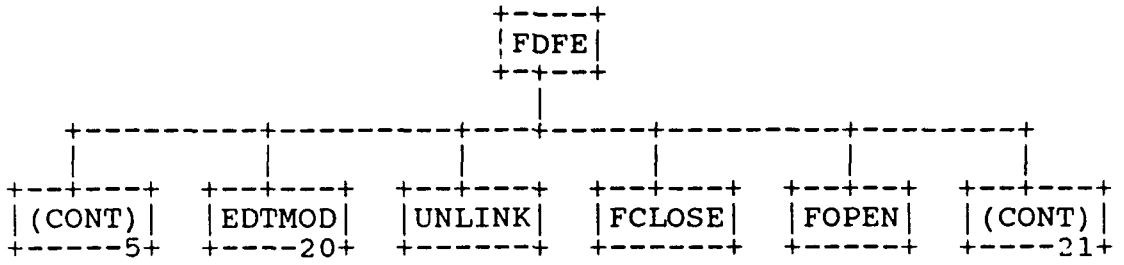
11


VIEW
ADDFRM
MEMCMP
SYSMSG
CLSFRM
RMVPAG
(CONT)
18

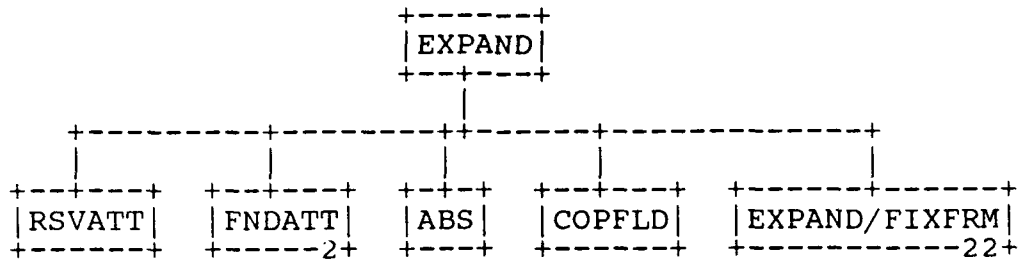
12


GETFLS
FREBUF
ESCPY
SPRINTF
STRRCHR
STRCAT
(CONT)
19
DELFLD

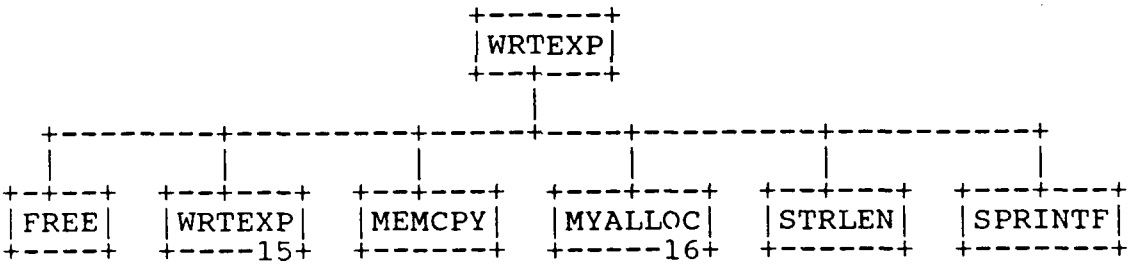
13


FDFE
(CONT)
5
EDTMOD
20
UNLINK
FCLOSE
FOPEN
(CONT)
21

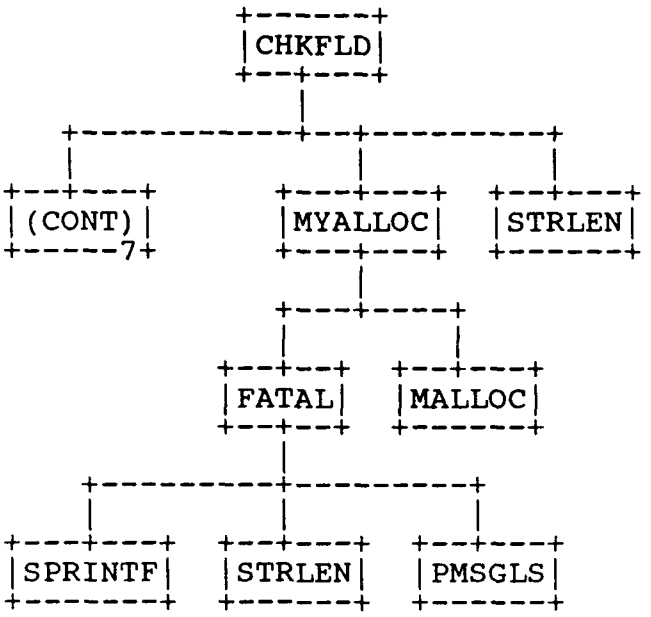
14


EXPAND
RSVATT
FNDATT
2
ABS
COPFLD
EXPAND/FIXFRM
22

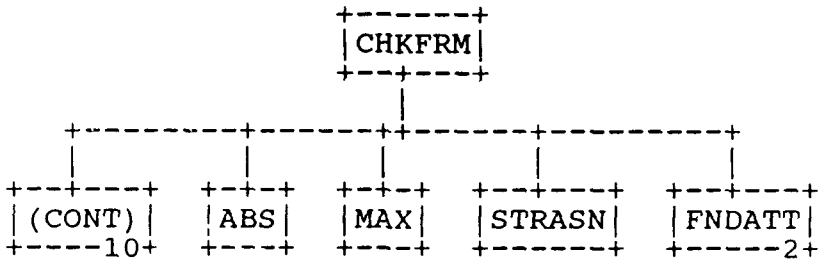
15


WRTEXP
FREE
WRTEXP
15
MEMCPY
MYALLOC
16
STRLEN
SPRINTF

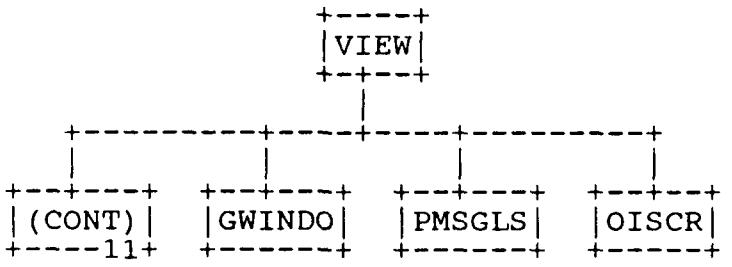
16


CHKFLD
(CONT)
7
MYALLOC
STRLEN
FATAL
MALLOC
SPRINTF
STRLEN
PMSGLS

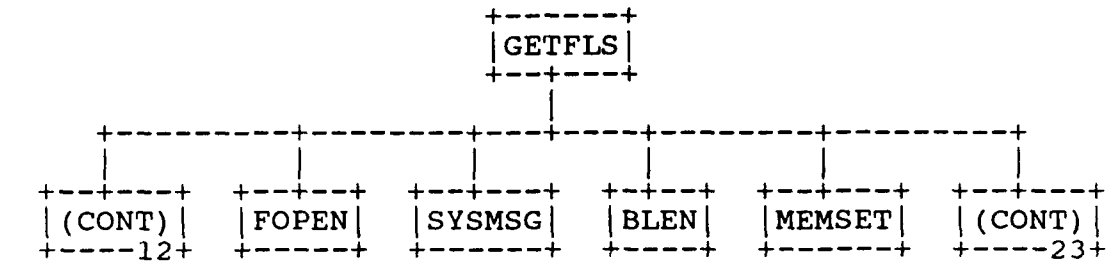
17


CHKFRM
(CONT)
10
ABS
MAX
STRASN
FNDATT
2

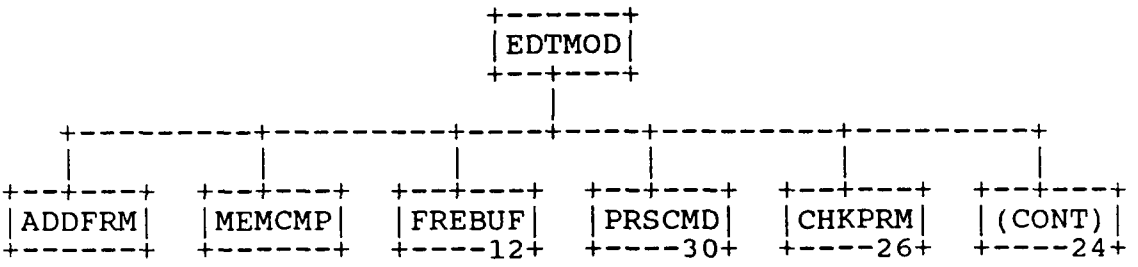
18


VIEW
(CONT)
11
GWINDO
PMSGLS
OISCR

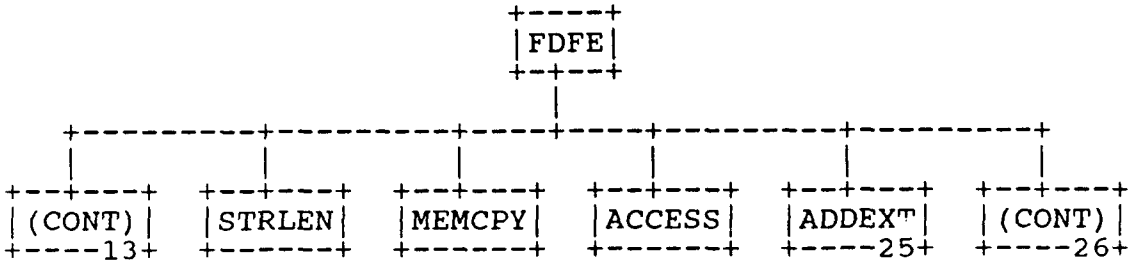
19


GETFLS
(CONT)
12
FOPEN
SYSMSG
BLEN
MEMSET
(CONT)
23

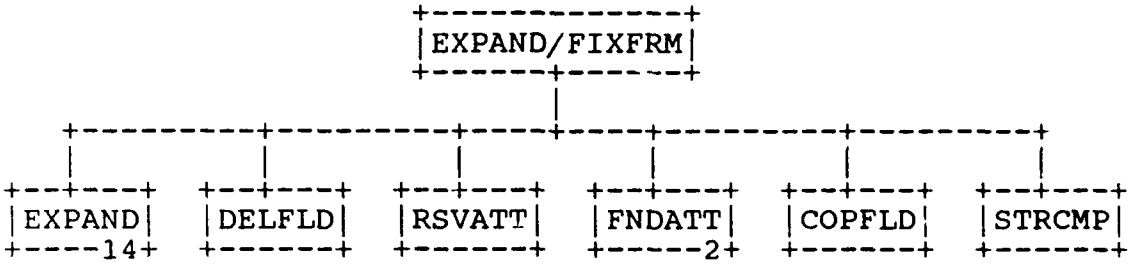
20


EDTMOD
ADDFRM
MEMCMP
FREBUF
12
PRSCMD
30
CHKPRM
26
(CONT)
24

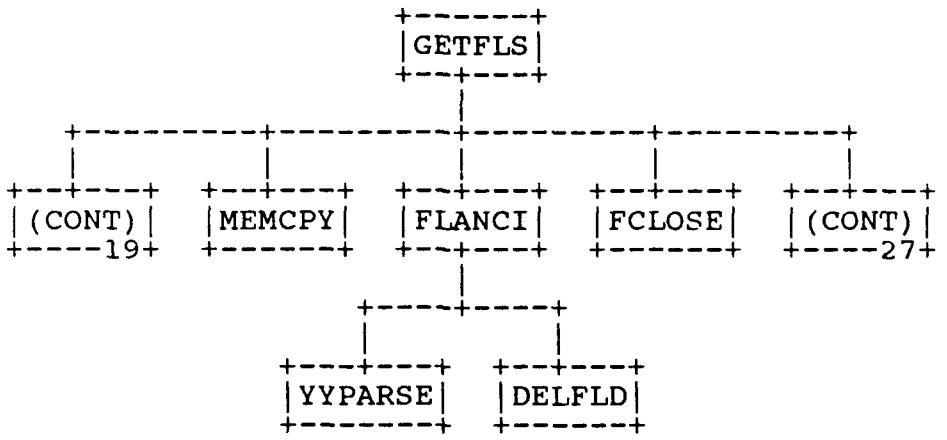
21


FDFE
(CONT)
13
STRLEN
MEMCPY
ACCESS
ADDEXT
25
(CONT)
26

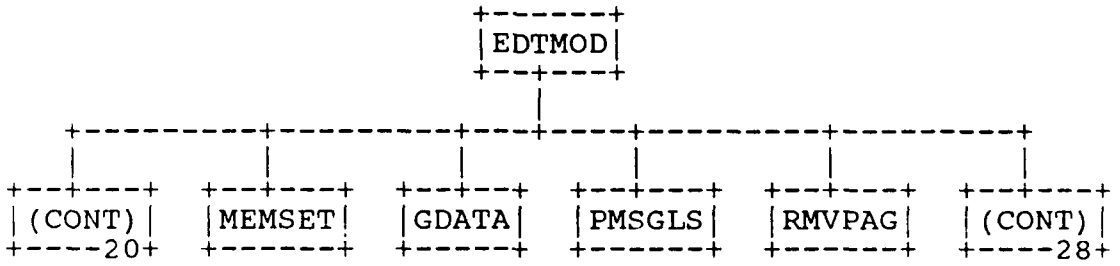
22


EXPAND/FIXFRM
EXPAND
14
DELFLD
RSVATT
FNDATT
2
COPFLD
STRCMP

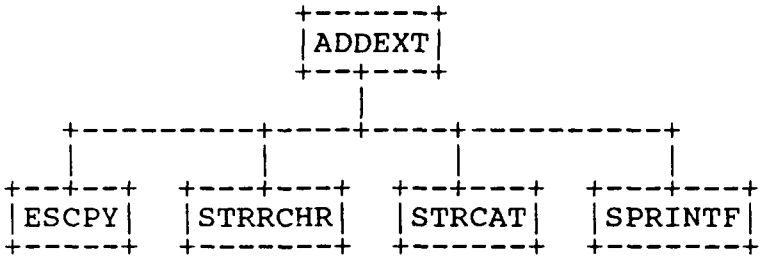
23


GETFLS
(CONT)
19
MEMCPY
FLANCI
FCLOSE
(CONT)
27
YYPARSE
DELFLD

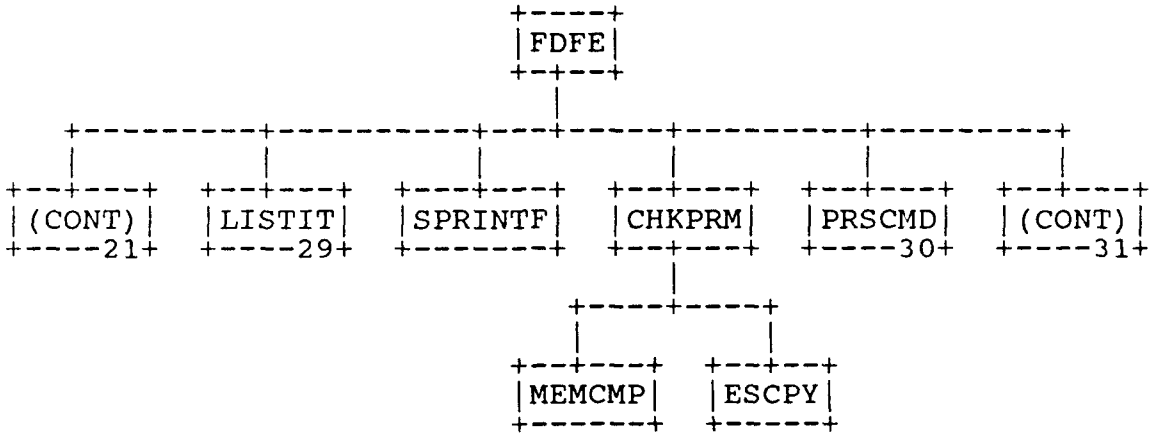
24


EDTMOD
(CONT)
20
MEMSET
GDATA
PMSGLS
RMVPAG
(CONT)
28

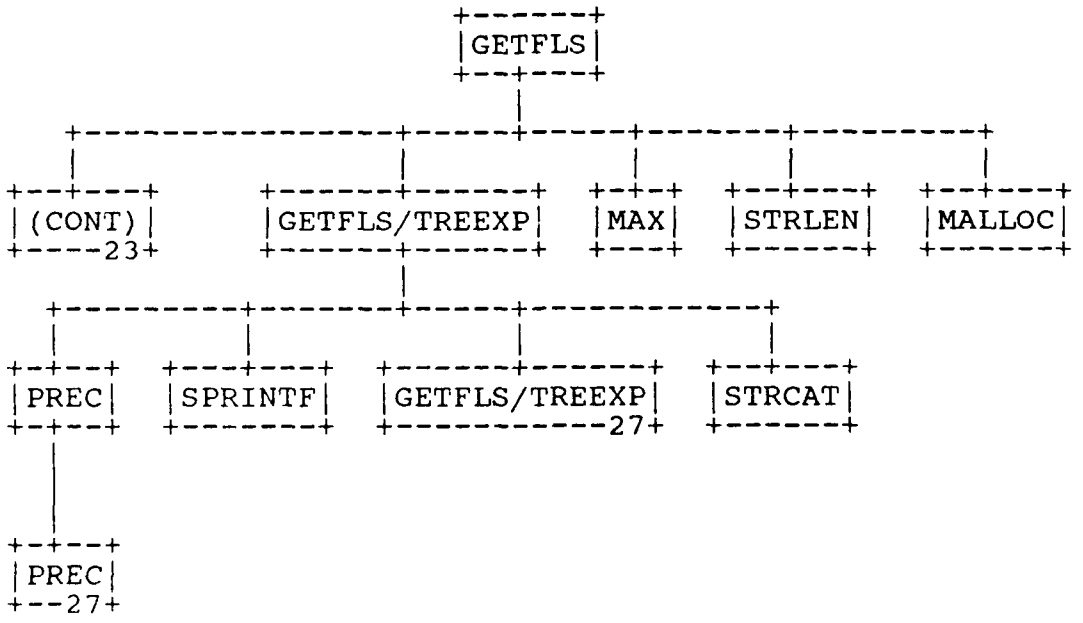
25


ADDEXT
ESCPY
STRRCHR
STRCAT
SPRINTF

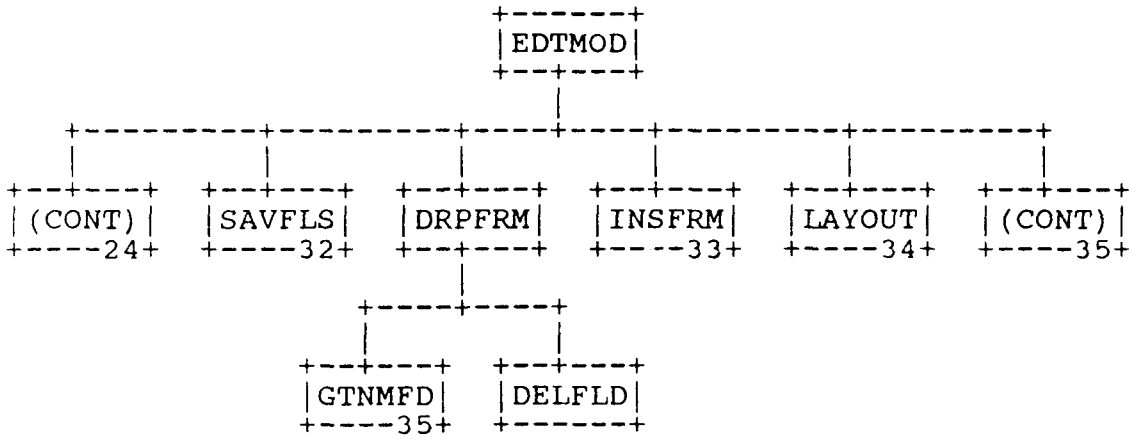
26


FDFE
(CONT)
21
LISTIT
29
SPRINTF
CHKPRM
PRSCMD
30
(CONT)
31
MEMCMP
ESCPY

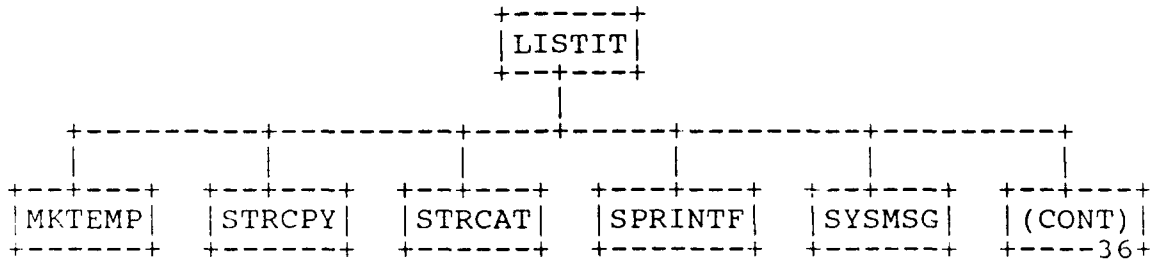
27


GETFLS
(CONT)
23
GETFLS/TREEXP
MAX
STRLEN
MALLOC
PREC
SPRINTF
GETFLS/TREEXP
27
STRCAT
PREC
27

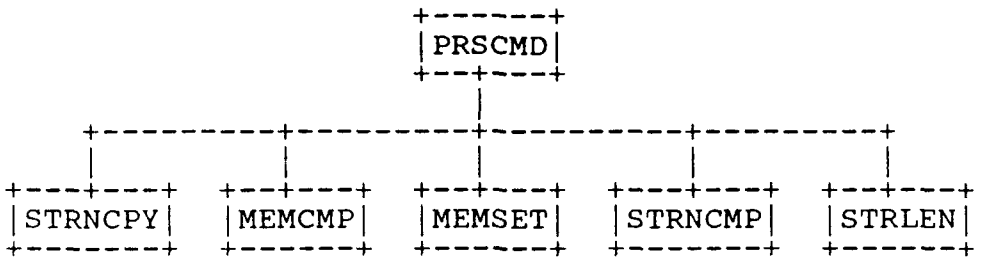
28


EDTMOD
(CONT)
24
SAVFLS
32
DRPFRM
INSFRM
33
LAYOUT
34
(CONT)
35
GTNMFD
35
DELFLD

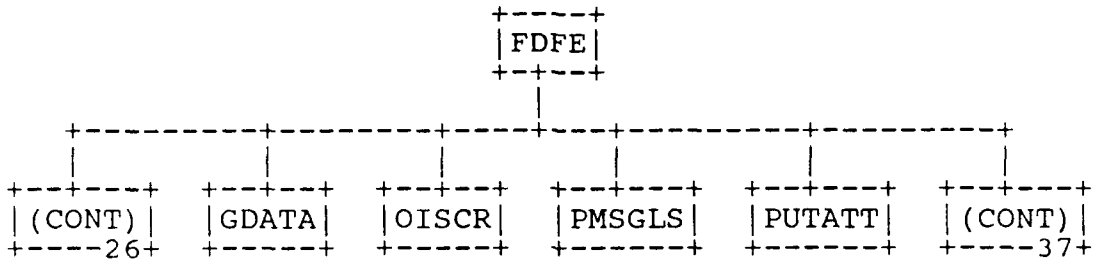
29


LISTIT
MKTEMP
STRCPY
STRCAT
SPRINTF
SYSMSG
(CONT)
36

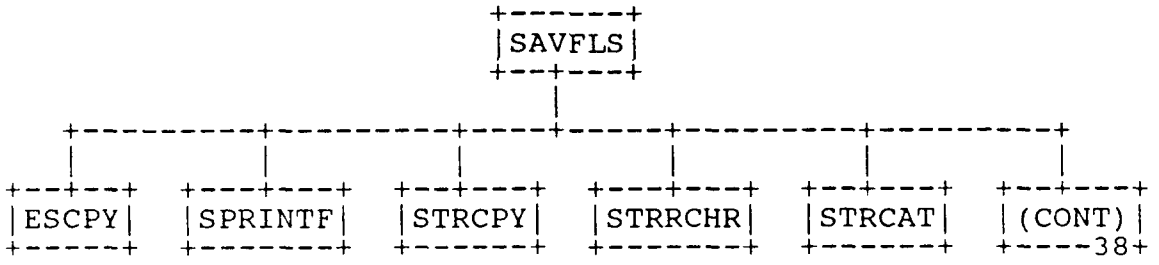
30


PRSCMD
STRNCPY
MEMCMP
MEMSET
STRNCMP
STRLEN

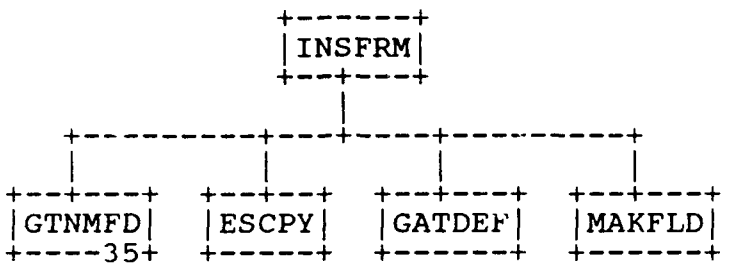
31


FDFE
(CONT)
26
GDATA
OISCR
PMSGLS
PUTATT
(CONT)
37

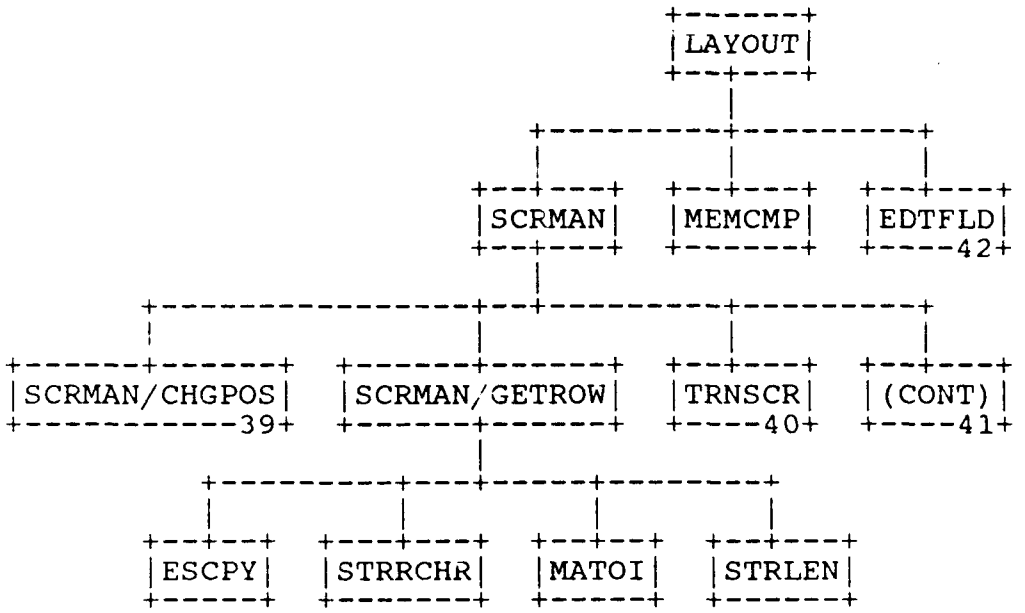
32


SAVFLS
ESCPY
SPRINTF
STRCPY
STRRCHR
STRCAT
(CONT)
38

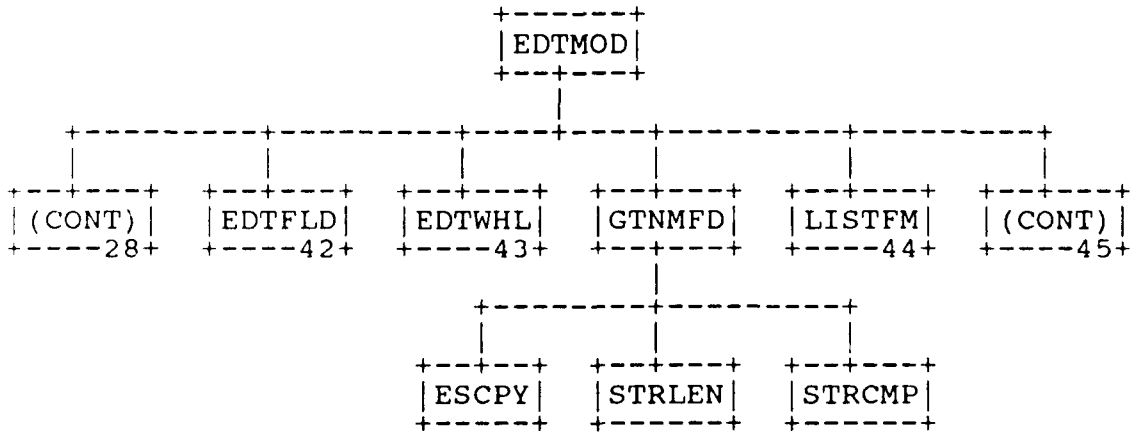
33


INSFRM
GTNMFD
35
ESCPY
GATDEF
MAKFLD

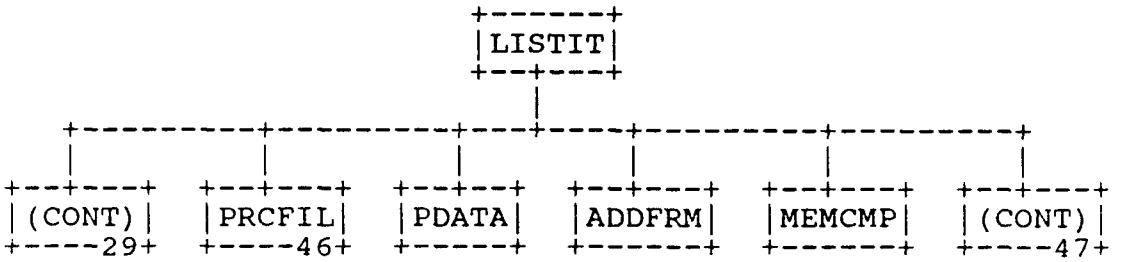
34


LAYOUT
SCRMAN
MEMCMP
EDTFLD
42
SCRMAN/CHGPOS
39
SCRMAN/GETROW
TRNSCR
40
(CONT)
41
ESCPY
STRRCHR
MATOI
STRLEN

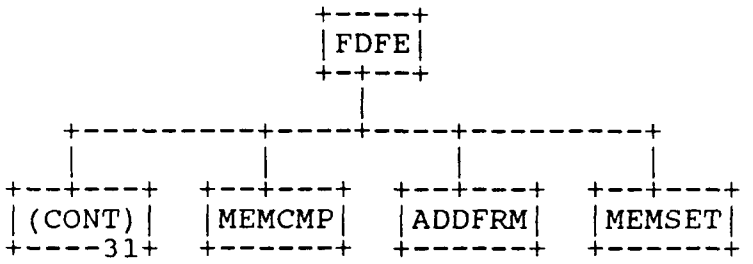
35


EDTMOD
(CONT)
28
EDTFLD
42
EDTWHL
43
GTNMFD
LISTFM
44
(CONT)
45
ESCPY
STRLEN
STRCMP

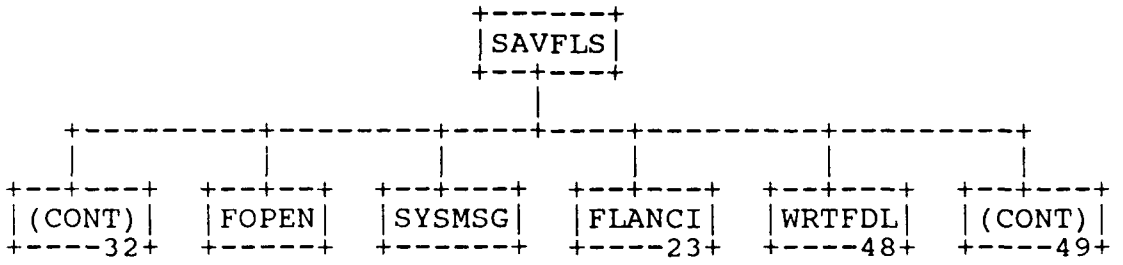
36


LISTIT
(CONT)
29
PRCFIL
46
PDATA
ADDFRM
MEMCMP
(CONT)
47

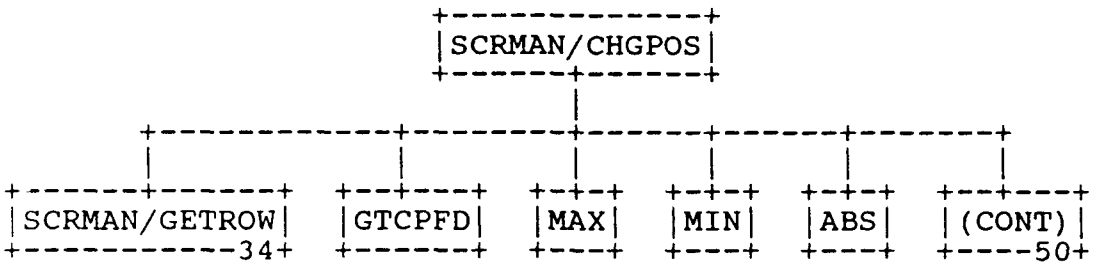
37


FDFE
(CONT)
31
MEMCMP
ADDFRM
MEMSET

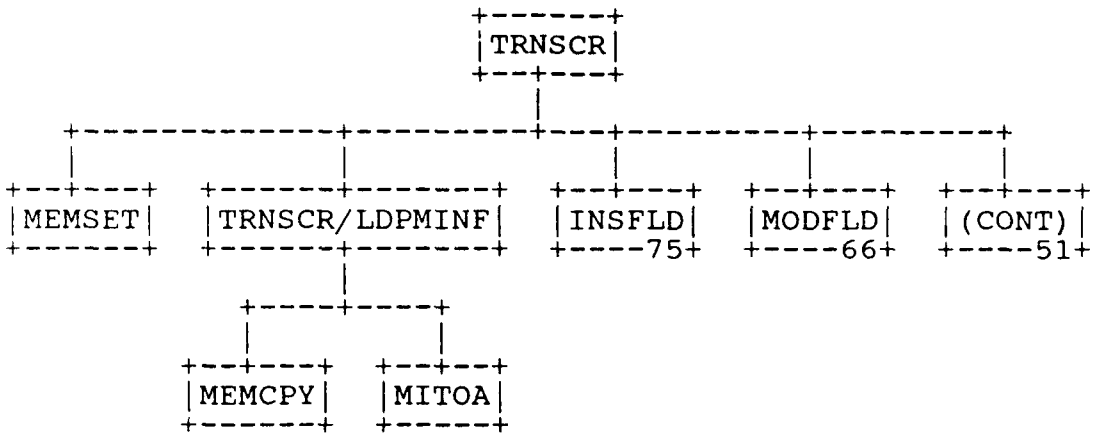
38


SAVFLS
(CONT)
32
FOPEN
SYSMSG
FLANCI
23
WRTFDL
48
(CONT)
49

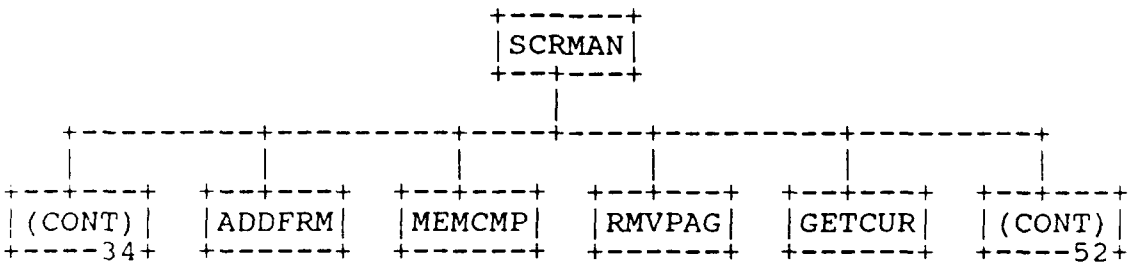
39


SCRMAN/CHGPOS
SCRMAN/GETROW
34
GTCPFD
MAX
MIN
ABS
(CONT)
50

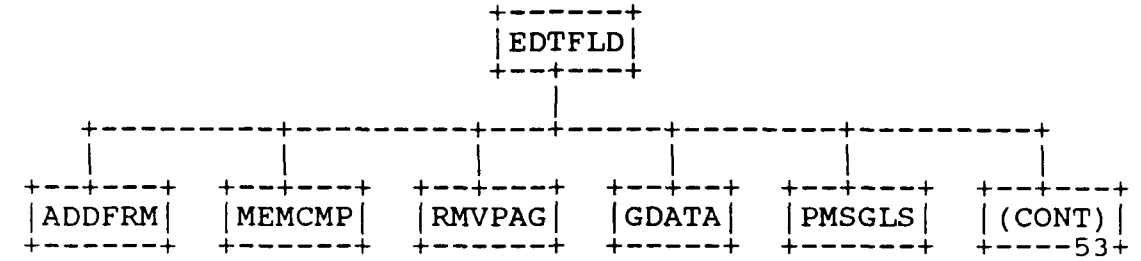
40


TRNSCR
MEMSET
TRNSCR/LDPMINF
INSFLD
75
MODFLD
66
(CONT)
51
MEMCPY
MITOA

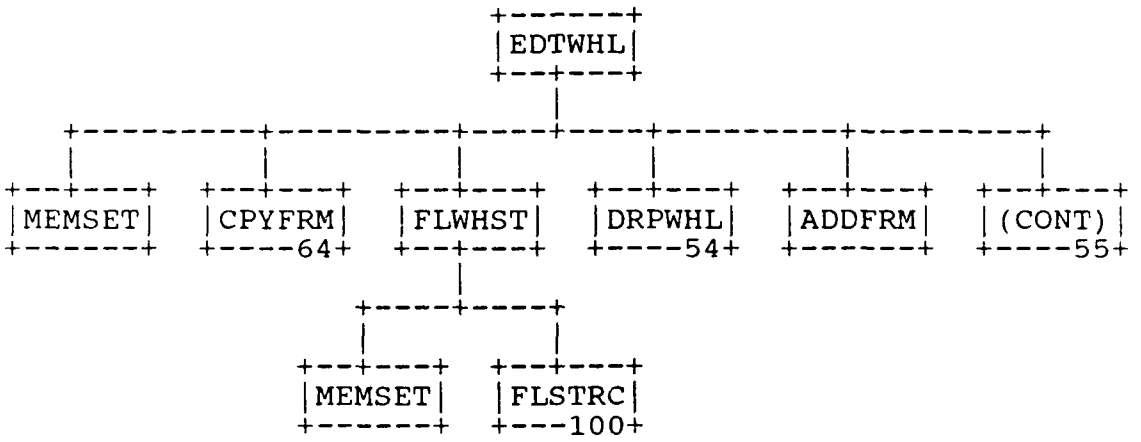
41


SCRMAN
(CONT)
34
ADDFRM
MEMCMP
RMVPAG
GETCUR
(CONT)
52

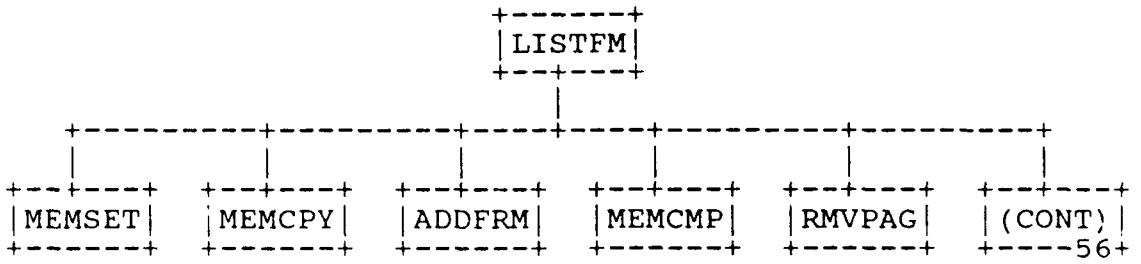
42


EDTFLD
ADDFRM
MEMCMP
RMVPAG
GDATA
PMSGLS
(CONT)
53

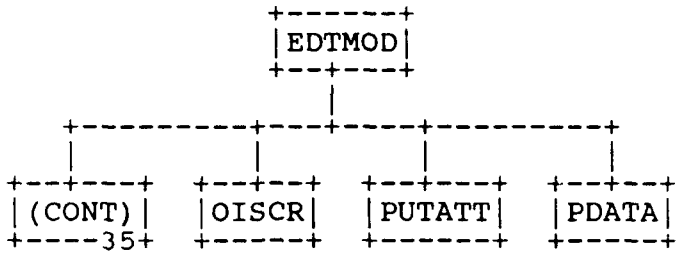
43


EDTWHL
MEMSET
CPYFRM
64
FLWHST
DRPWHL
54
ADDFRM
(CONT)
55
MEMSET
FLSTRC
100

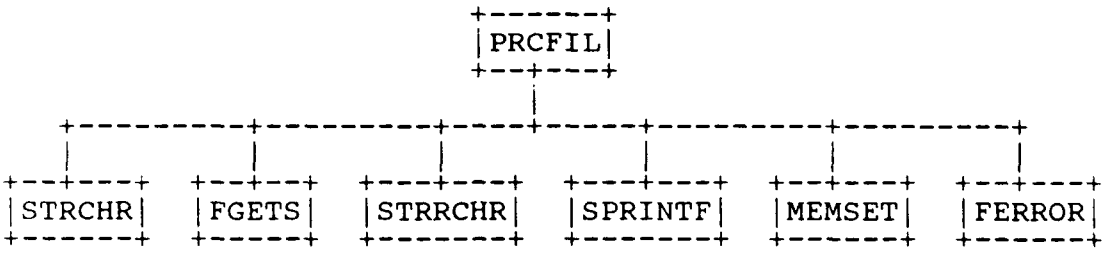
44


LISTFM
MEMSET
MEMCPY
ADDFRM
MEMCMP
RMVPAG
(CONT)
56

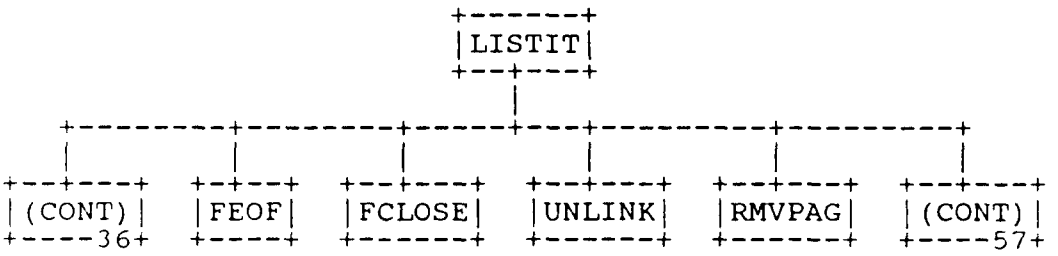
45


EDTMOD
(CONT)
35
OISCR
PUTATT
PDATA

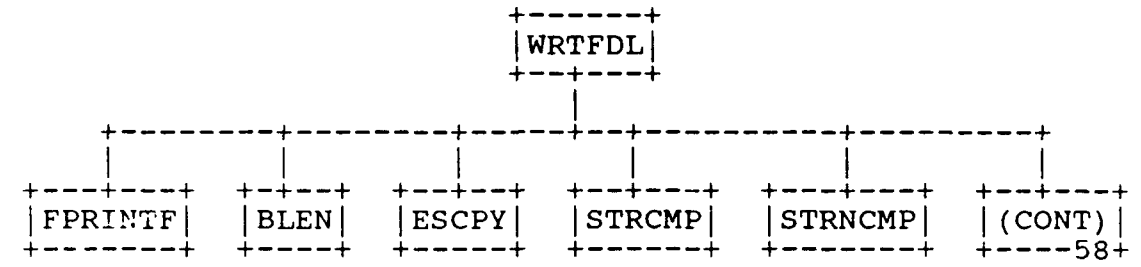
46


PRCFIL
STRCHR
FGETS
STRRCHR
SPRINTF
MEMSET
FERROR

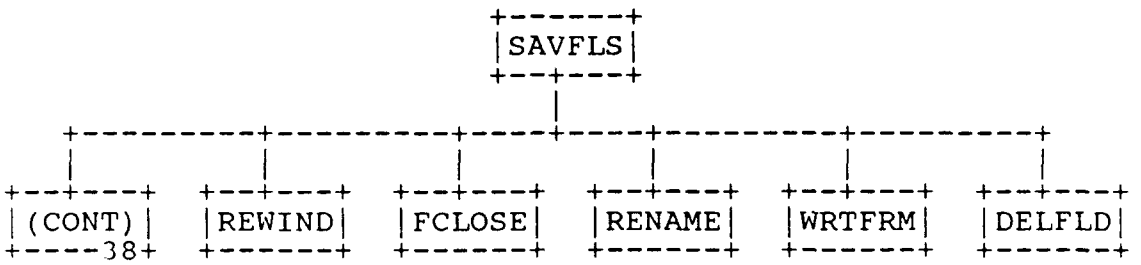
47


LISTIT
(CONT)
36
FEOF
FCLOSE
UNLINK
RMVPAG
(CONT)
57

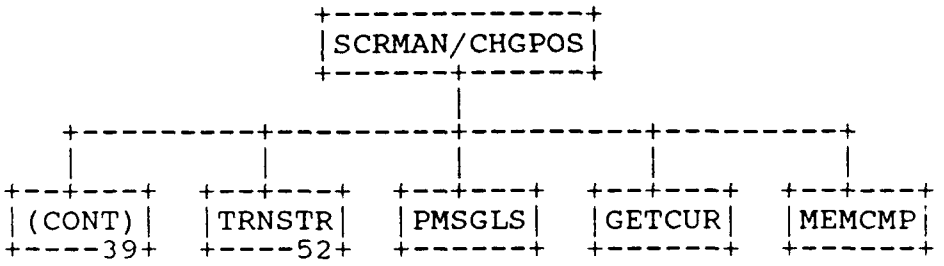
48


WRTFDL
FPRINTF
BLEN
ESCPY
STRCMP
STRNCMP
(CONT)
58

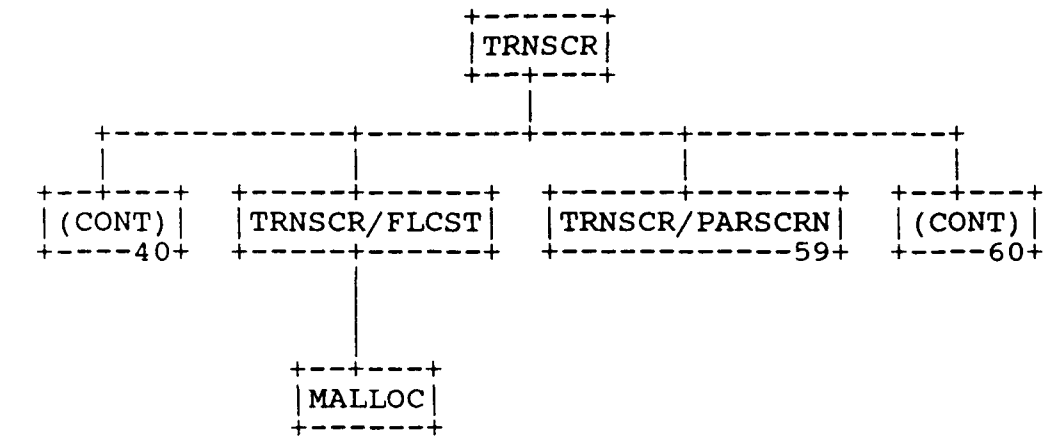
49


SAVFLS
(CONT)
38
REWIND
FCLOSE
RENAME
WRTFRM
DELFLD

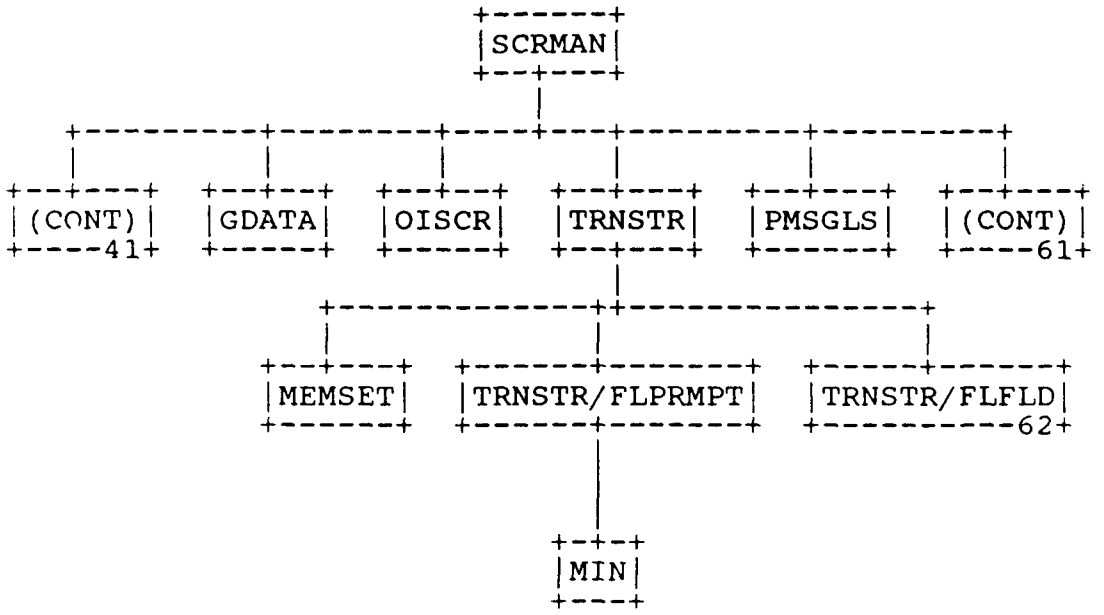
50


SCRMAN/CHGPOS
(CONT)
39
TRNSTR
52
PMSGLS
GETCUR
MEMCMP

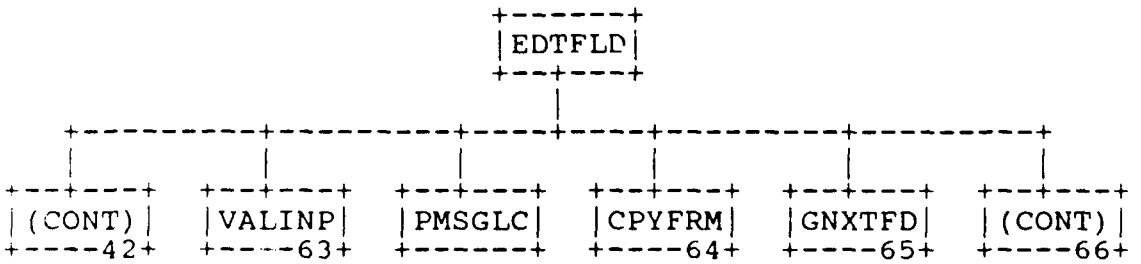
51


TRNSCR
(CONT)
40
TRNSCR/FLCST
MALLOC
TRNSCR/PARSCRN
59
(CONT)
60

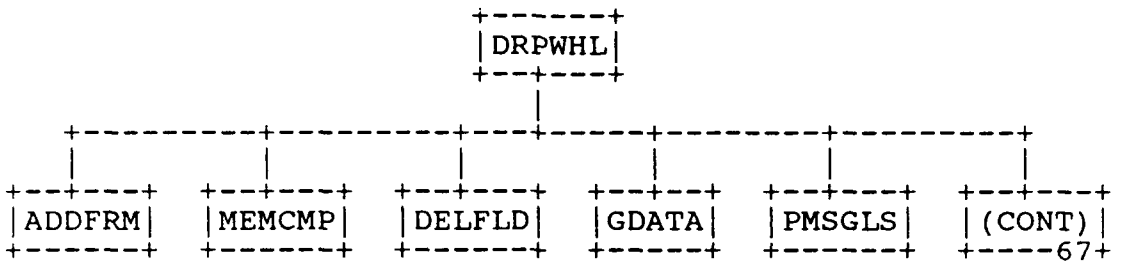
52


SCRMAN
(CONT)
41
GDATA
OISCR
TRNSTR
PMSGLS
(CONT)
61
MEMSET
TRNSTR/FLPRMPT
TRNSTR/FLFLD
62
MIN

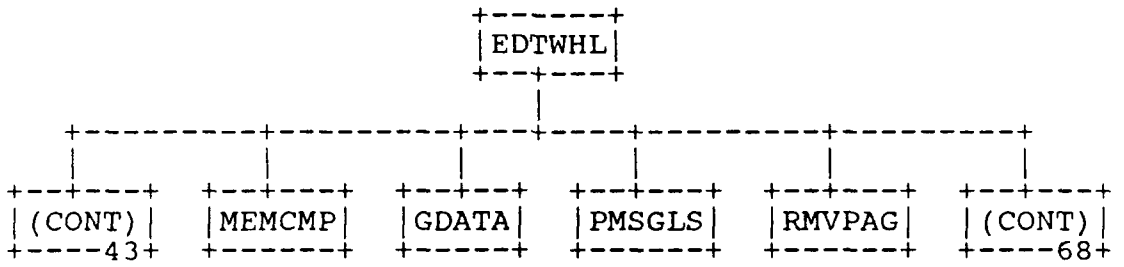
53


EDTFLD
(CONT)
42
VALINP
63
PMSGLC
CPYFRM
64
GNXTFD
65
(CONT)
66

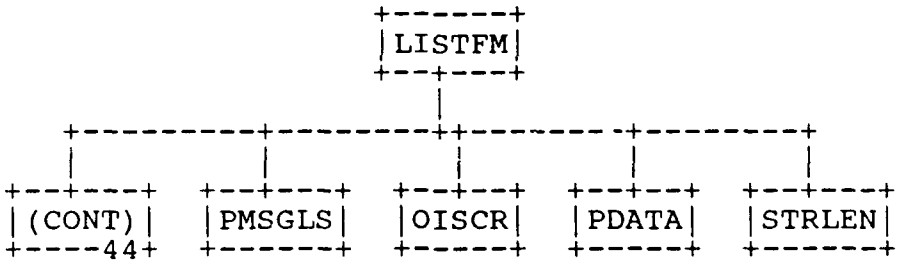
54


DRPWHL
ADDFRM
MEMCMP
DELFLD
GDATA
PMSGLS
(CONT)
67

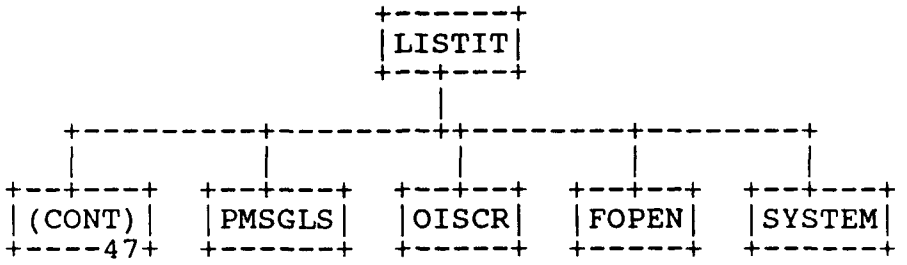
55


EDTWHL
(CONT)
43
MEMCMP
GDATA
PMSGLS
RMVPAG
(CONT)
68

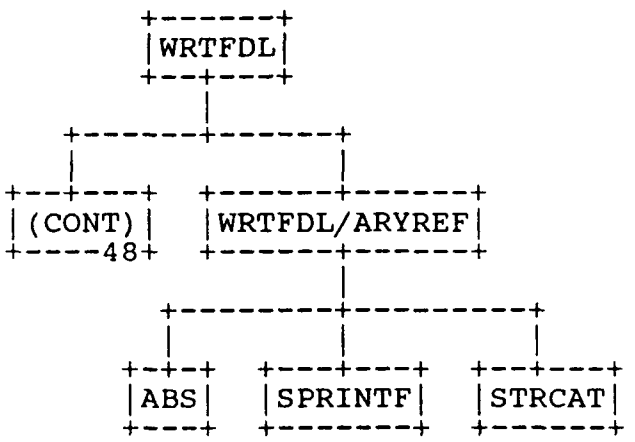
56


LISTFM
(CONT)
44
PMSGLS
OISCR
PDATA
STRLEN

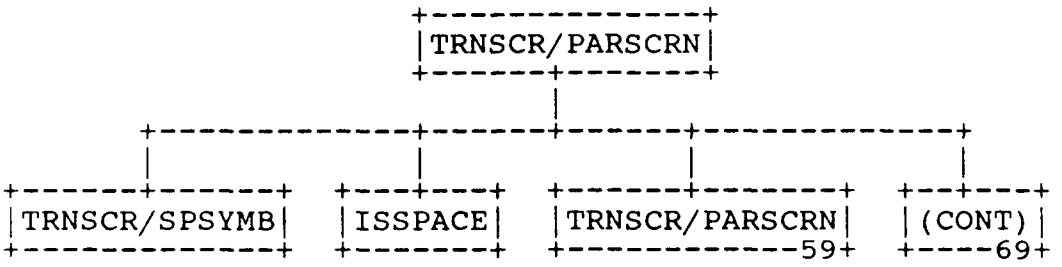
57


LISTIT
(CONT)
47
PMSGLS
OISCR
FOPEN
SYSTEM

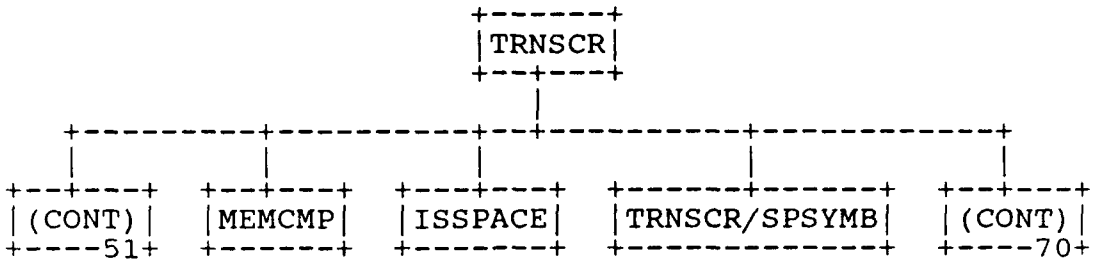
58


WRTFDL
(CONT)
48
WRTFDL/ARYREF
ABS
SPRINTF
STRCAT

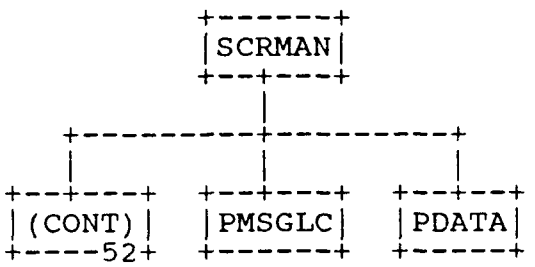
59


TRNSCR/PARSCRN
TRNSCR/SPSYMB
ISSPACE
TRNSCR/PARSCRN
59
(CONT)
69

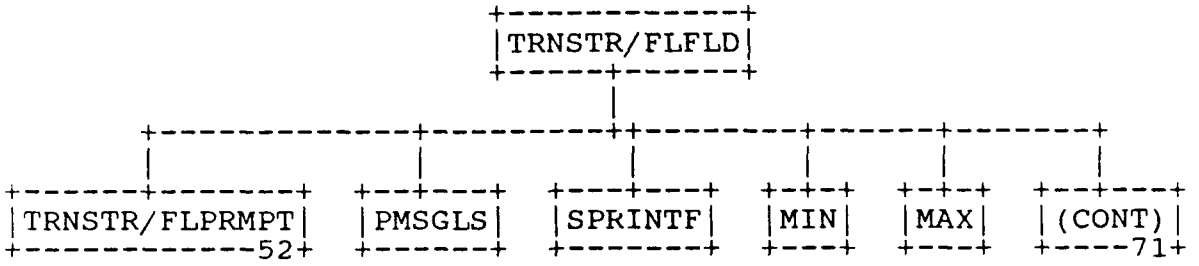
60


TRNSCR
(CONT)
51
MEMCMP
ISSPACE
TRNSCR/SPSYMB
(CONT)
70

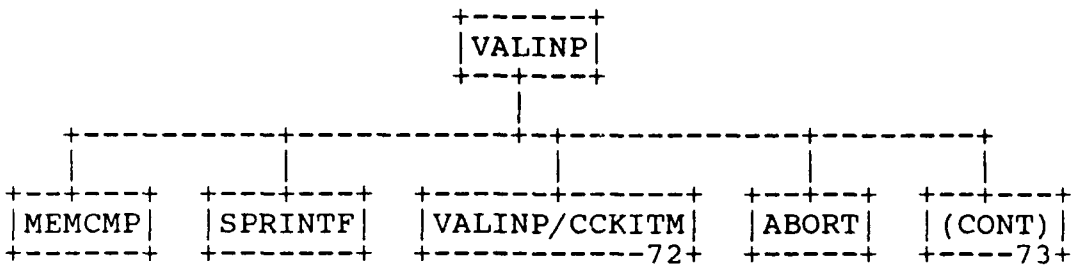
61


SCRMAN
(CONT)
52
PMSGLC
PDATA

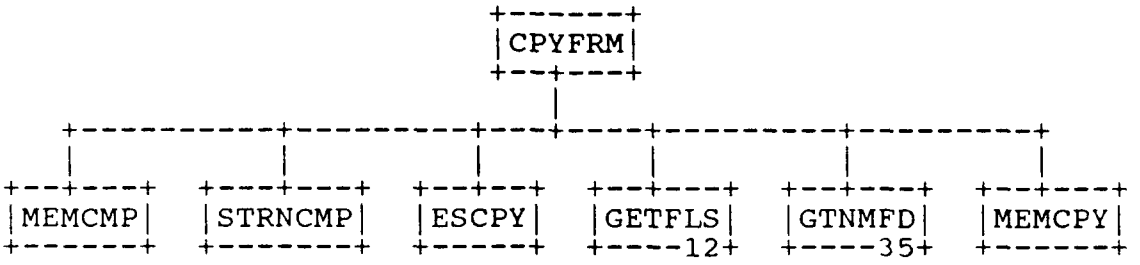
62


TRNSTR/FLFLD
TRNSTR/FLPRMPT
52
PMSGLS
SPRINTF
MIN
MAX
(CONT)
71

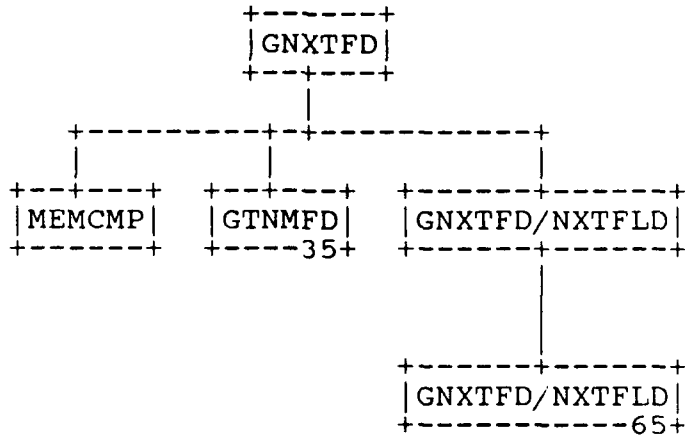
63


VALINP
MEMCMP
SPRINTF
VALINP/CCKITM
72
ABORT
(CONT)
73

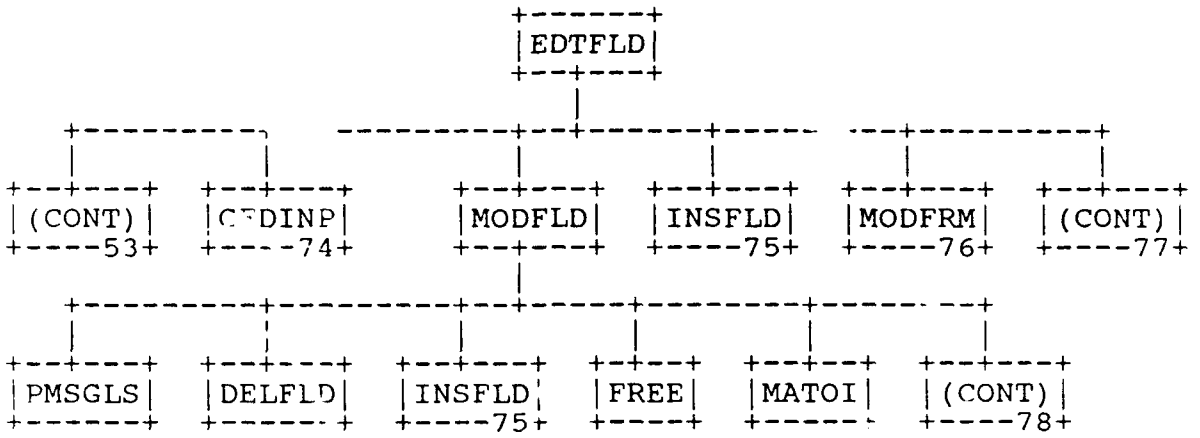
64


CPYFRM
MEMCMP
STRNCMP
ESCPY
GETFLS
12
GTNMFD
35
MEMCPY

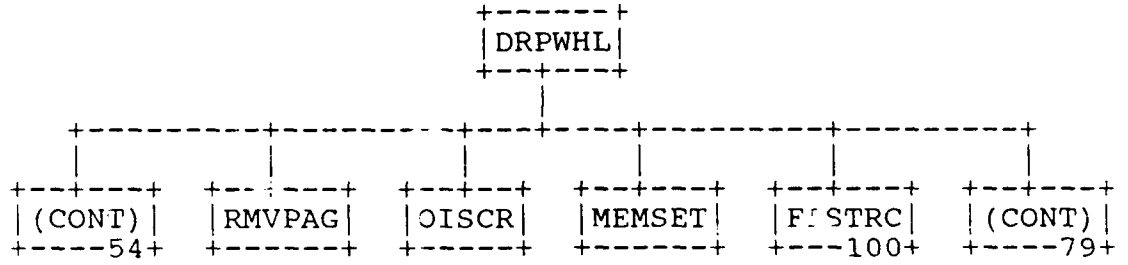
65


GNXTFD
MEMCMP
GTNMFD
35
GNXTFD/NXTFLD
GNXTFD/NXTFLD
65

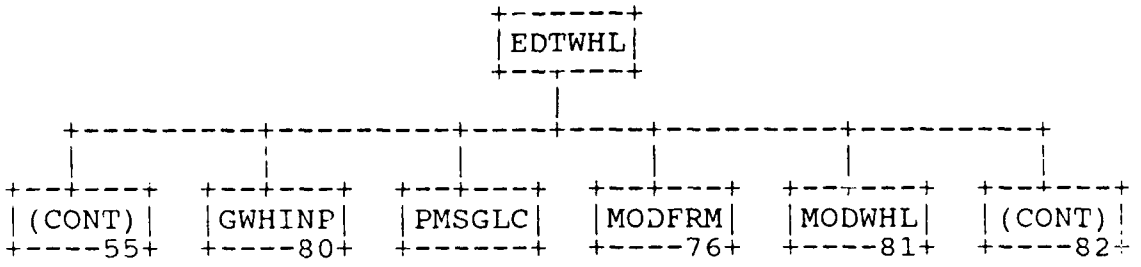
66


EDTFLD
(CONT)
53
CRDINP
74
MODFLD
INSFLD
75
MODFRM
76
(CONT)
77
PMSGLS
DELFLD
INSFLD
75
FREE
MATOI
(CONT)
78

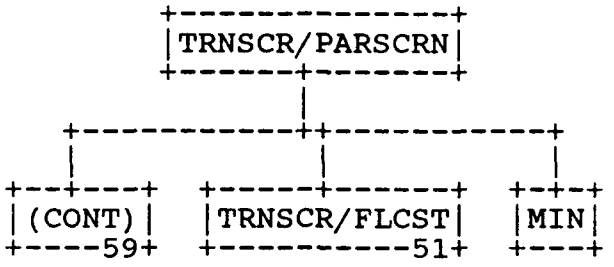
67


DRPWHL
(CONT)
54
RMVPAG
OISCR
MEMSET
STRC
100
(CONT)
79

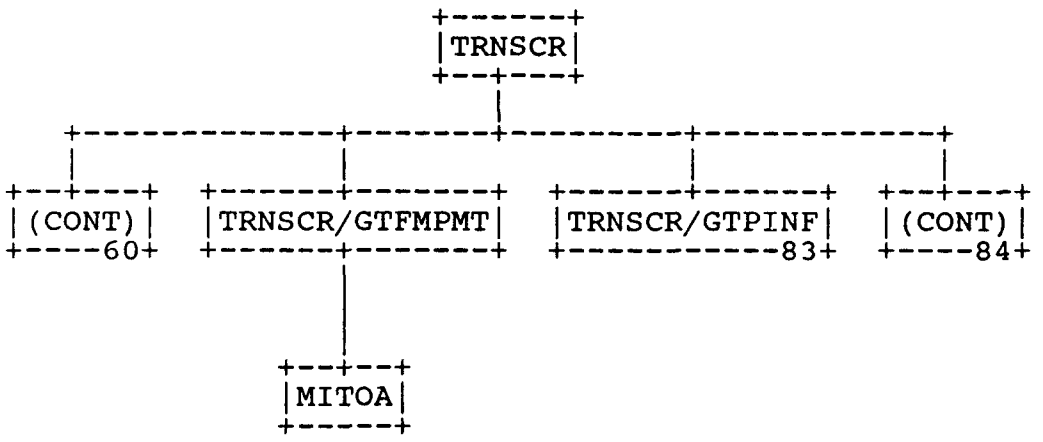
68


EDTWHL
(CONT)
55
GWHINP
80
PMSGLC
MODFRM
76
MODWHL
81
(CONT)
82

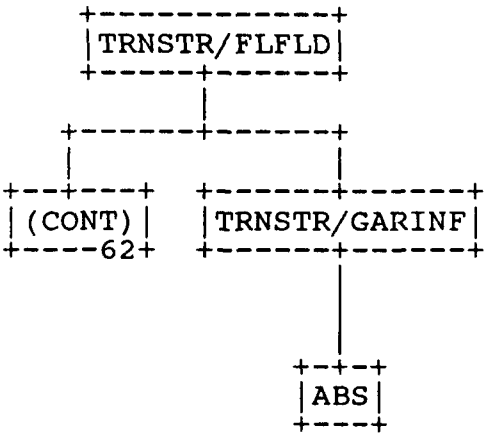
69


TRNSCR/PARSCRN
(CONT)
59
TRNSCR/FLCST
51
MIN

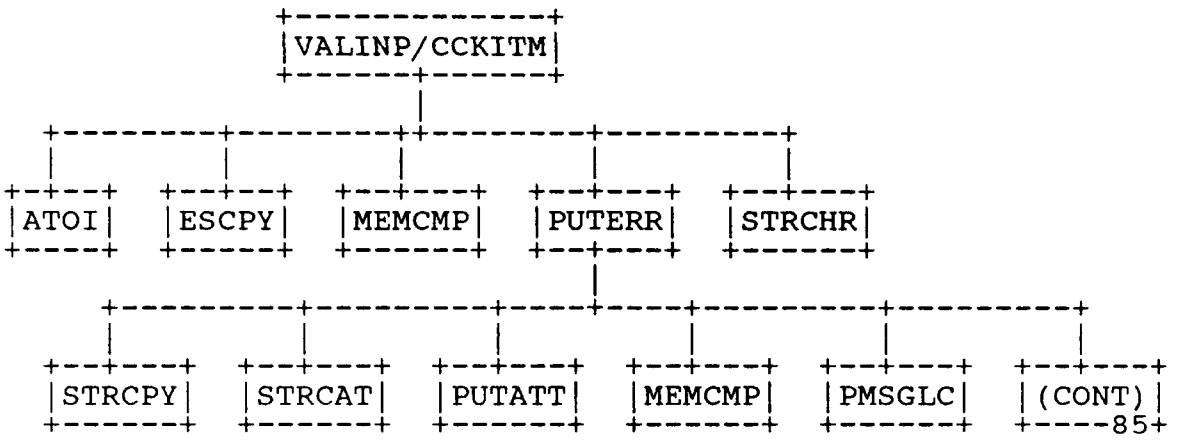
70


TRNSCR
(CONT)
60
TRNSCR/GTFMPMT
MITOA
TRNSCR/GTPINF
83
(CONT)
84

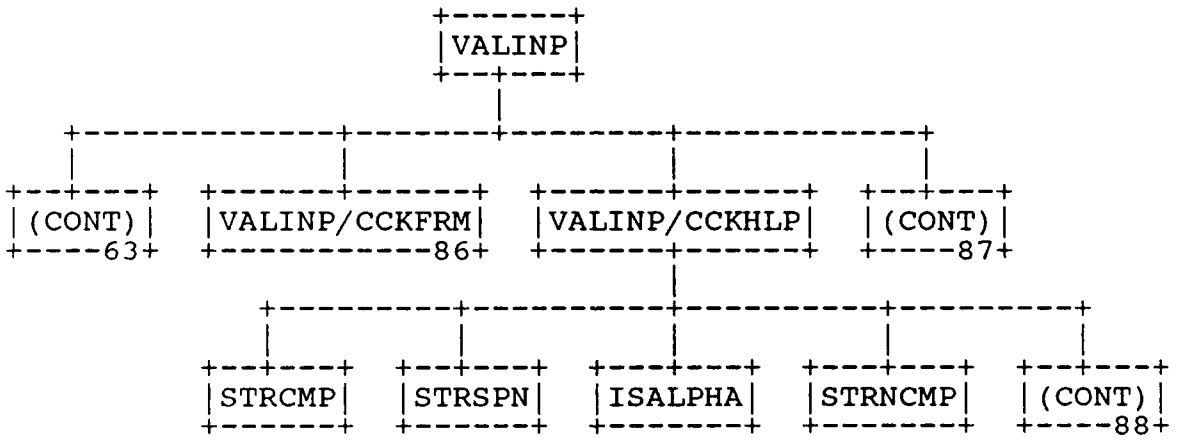
71


TRNSTR/FLFLD
(CONT)
62
TRNSTR/GARINF
ABS

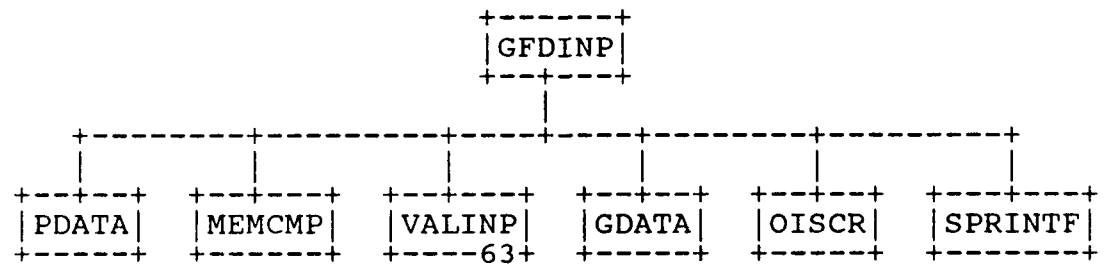
72


VALINP/CCKITM
ATOI
ESCPY
MEMCMP
PUTERR
STRCHR
STRCPY
STRCAT
PUTATT
MEMCMP
PMSGLC
(CONT)
85

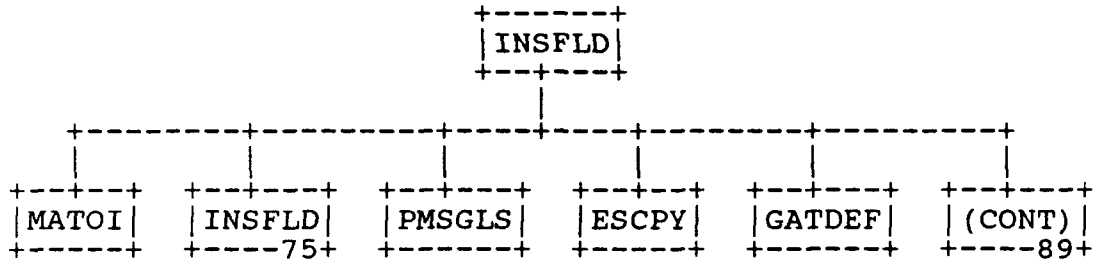
73


VALINP
(CONT)
63
VALINP/CCKFRM
86
VALINP/CCKHLP
(CONT)
87
STRCMP
STRSPN
ISALPHA
STRNCMP
(CONT)
88

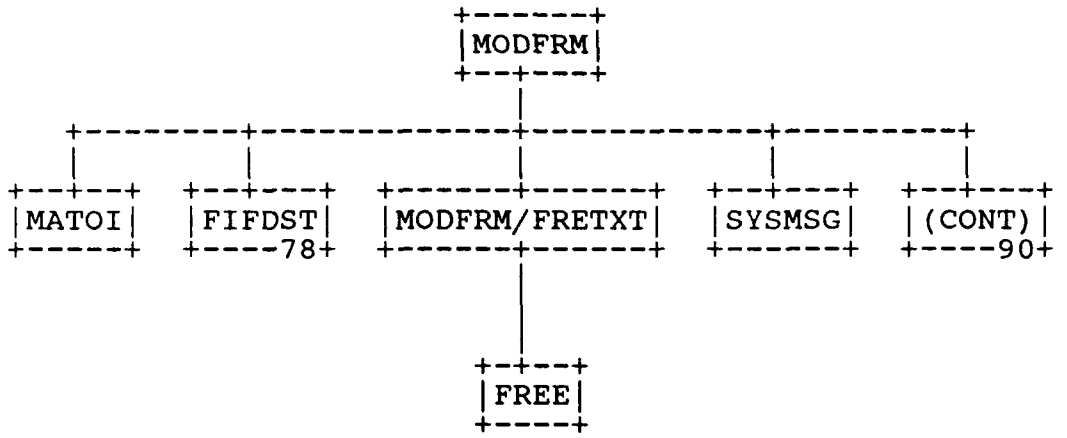
74


GFDINP
PDATA
MEMCMP
VALINP
63
GDATA
OISCR
SPRINTF

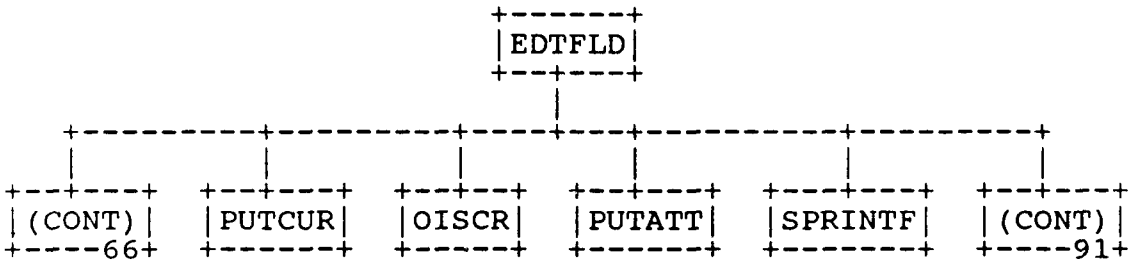
75


INSFLD
MATOI
INSFLD
75
PMSGLS
ESCPY
GATDEF
(CONT)
89

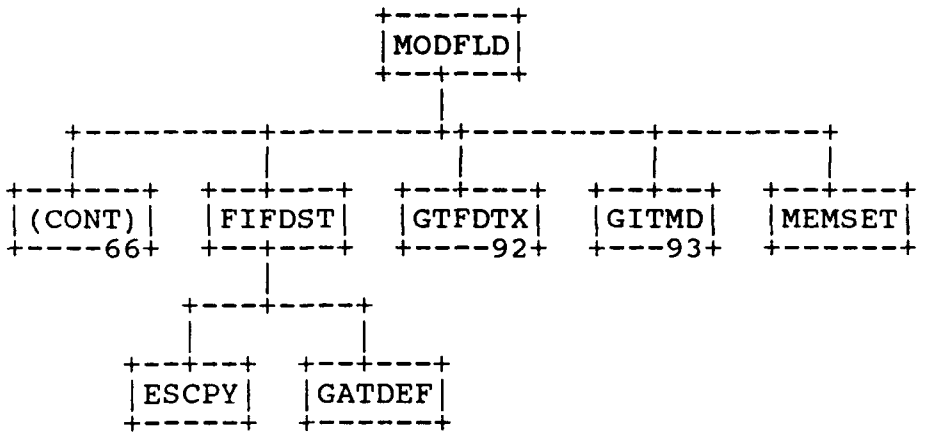
76


MODFRM
MATOI
FIFDST
78
MODFRM/FRETXT
SYSMSG
(CONT)
90
FREE

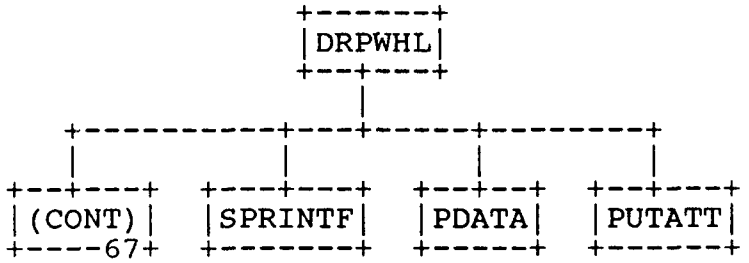
77


EDTFLD
(CONT)
66
PUTCUR
OISCR
PUTATT
SPRINTF
(CONT)
91

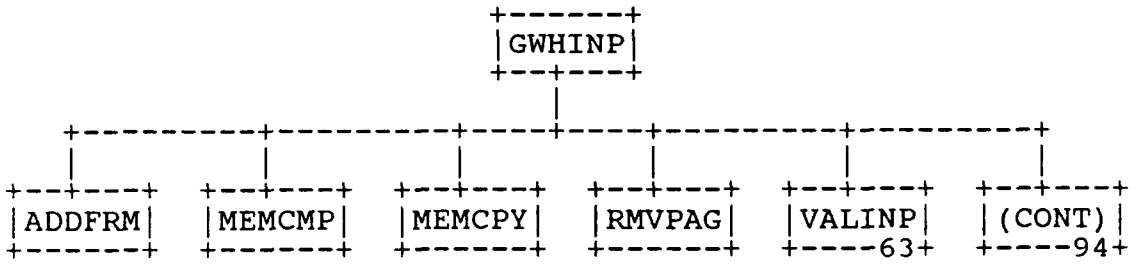
78


MODFLD
(CONT)
66
FIFDST
GTFDTX
92
GITMD
93
MEMSET
ESCPY
GATDEF

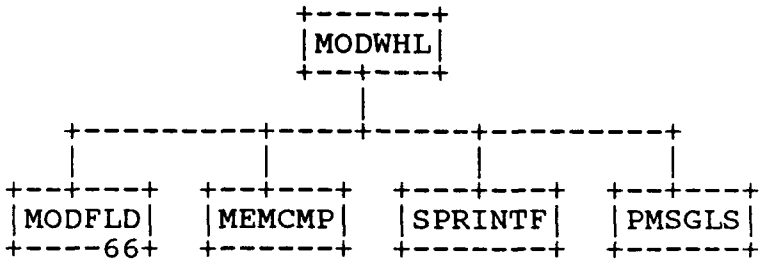
79


DRPWHL
(CONT)
67
SPRINTF
PDATA
PUTATT

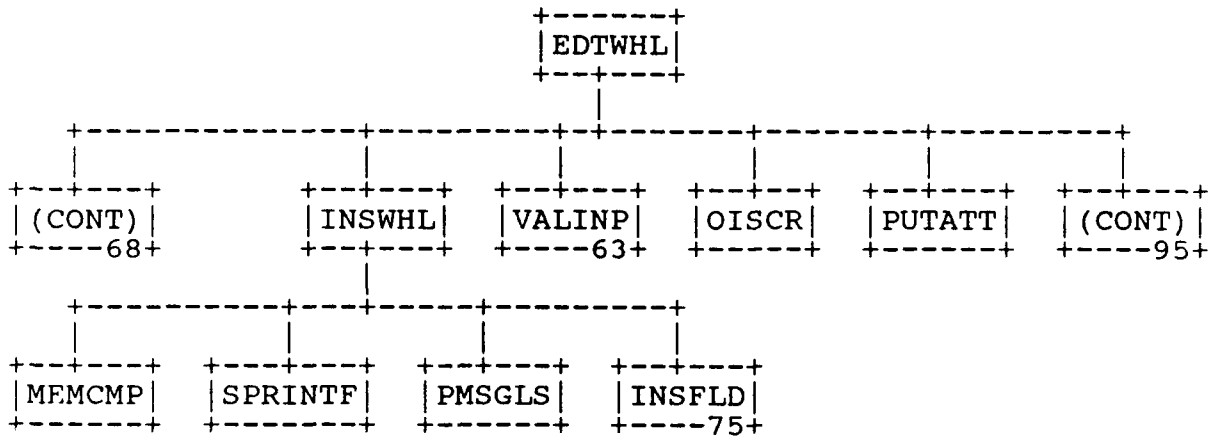
80


GWHINP
ADDFRM
MEMCMP
MEMCPY
RMVPAG
VALINP
63
(CONT)
94

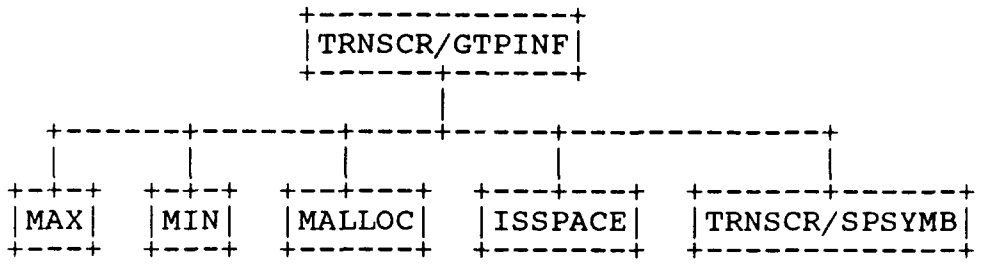
81


MODWHL
MODFLD
66
MEMCMP
SPRINTF
PMSGLS

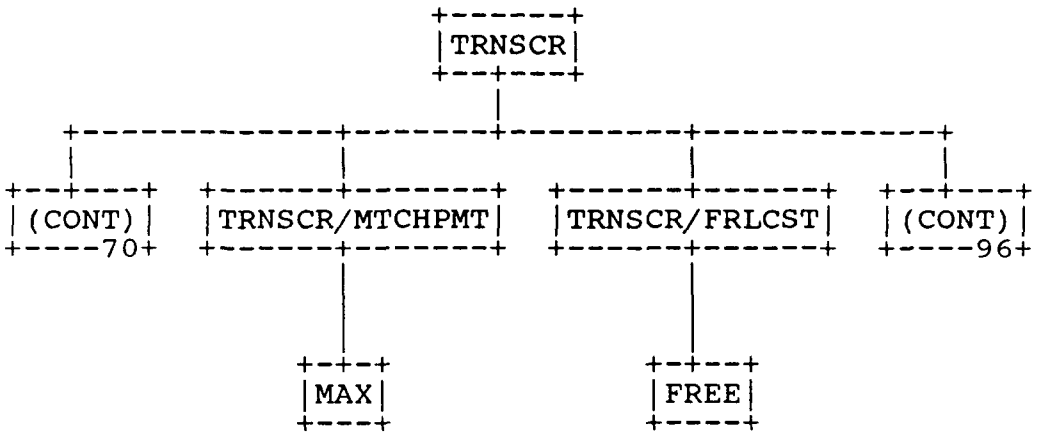
82


EDTWHL
(CONT)
68
INSWHL
VALINP
63
OISCR
PUTATT
(CONT)
95
MEMCMP
SPRINTF
PMSGLS
INSFLD
75

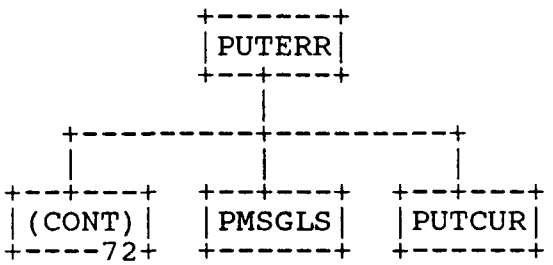
83


TRNSCR/GTPINF
MAX
MIN
MALLOC
ISSPACE
TRNSCR/SPSYMB

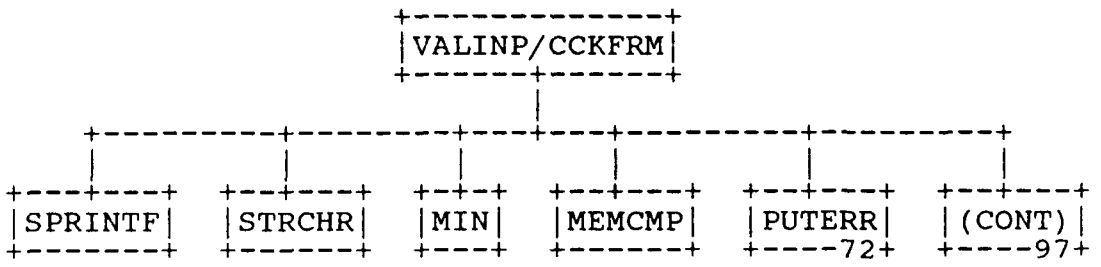
84


TRNSCR
(CONT)
70
TRNSCR/MTCHPMT
TRNSCR/FRLCST
(CONT)
96
MAX
FREE

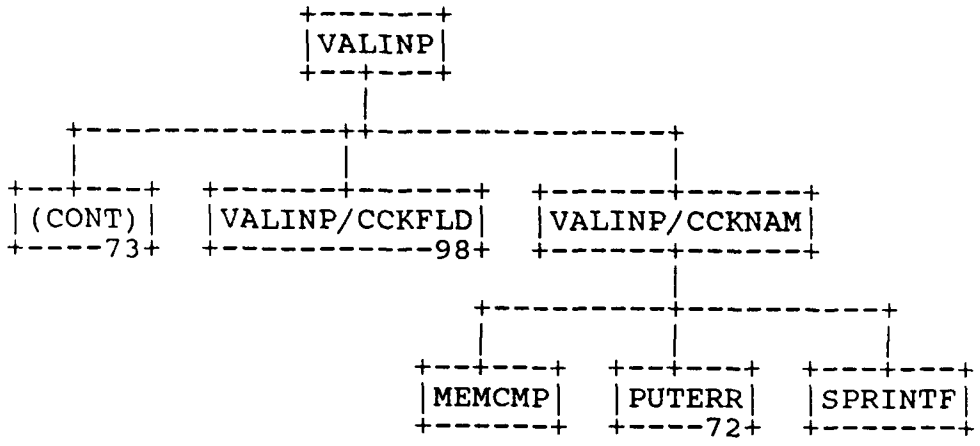
85


PUTERR
(CONT)
72
PMSGLS
PUTCUR

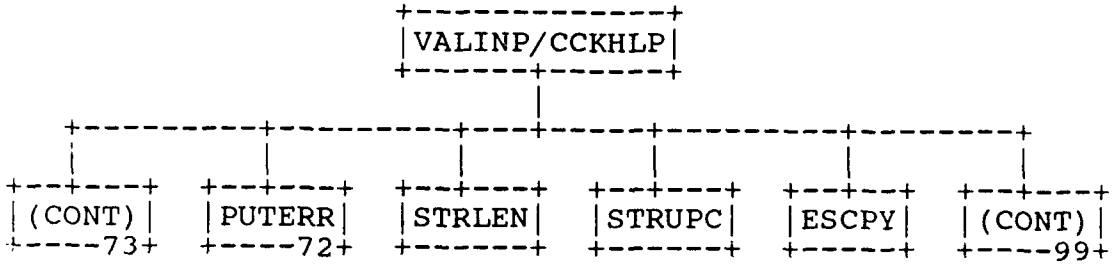
86


VALINP/CCKFRM
SPRINTF
STRCHR
MIN
MEMCMP
PUTERR
72
(CONT)
97

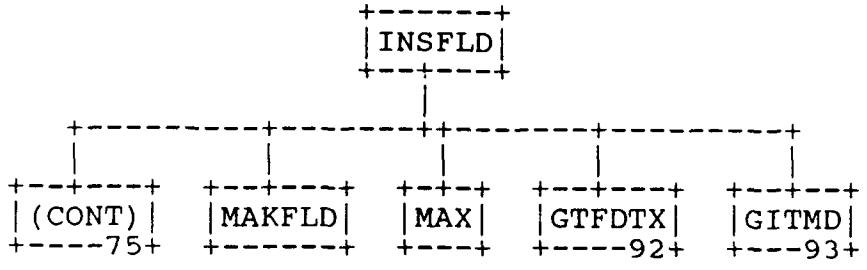
87


VALINP
(CONT)
73
VALINP/CCKFLD
98
VALINP/CCKNAM
MEMCMP
PUTERR
72
SPRINTF

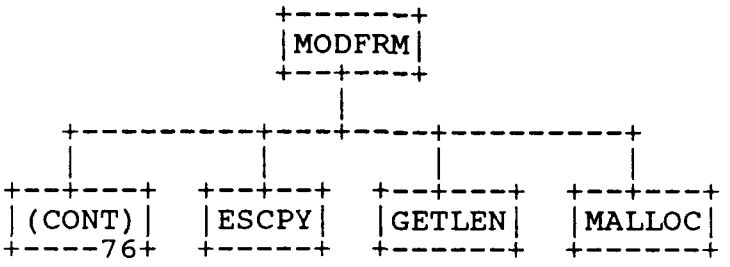
88


VALINP/CCKHLP
(CONT)
73
PUTERR
72
STRLEN
STRUPC
ESCPY
(CONT)
99

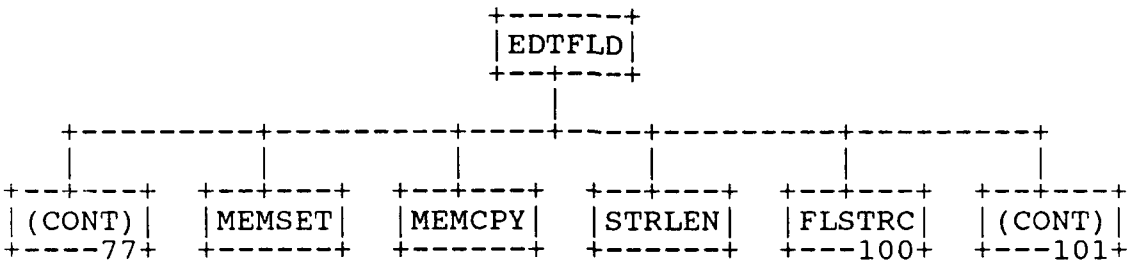
89


INSFLD
(CONT)
75
MAKFLD
MAX
GTFDTX
92
GITMD
93

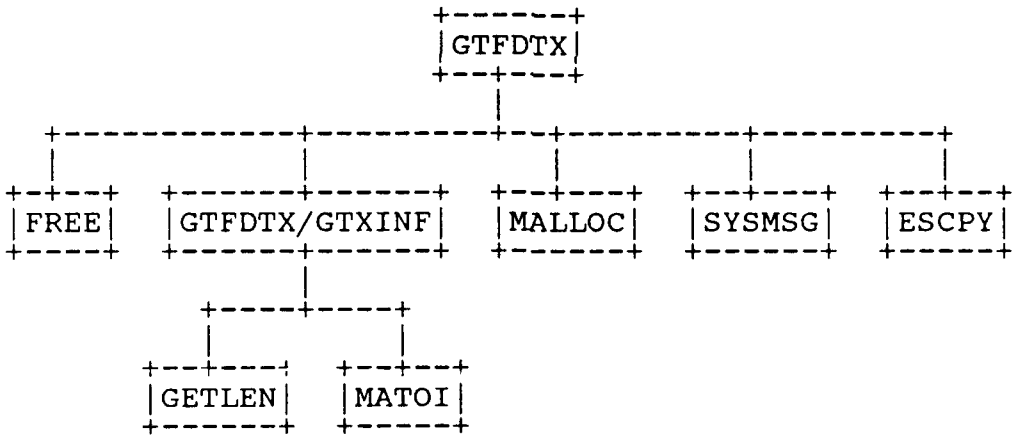
90


MODFRM
(CONT)
76
ESCPY
GETLEN
MALLOC

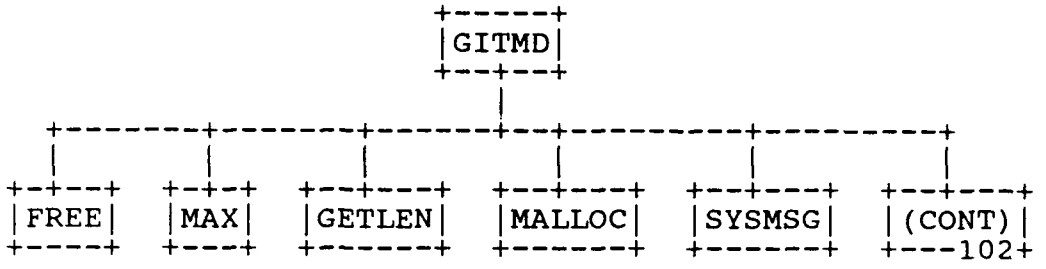
91


EDTFLD
(CONT)
77
MEMSET
MEMCPY
STRLEN
FLSTRC
100
(CONT)
101

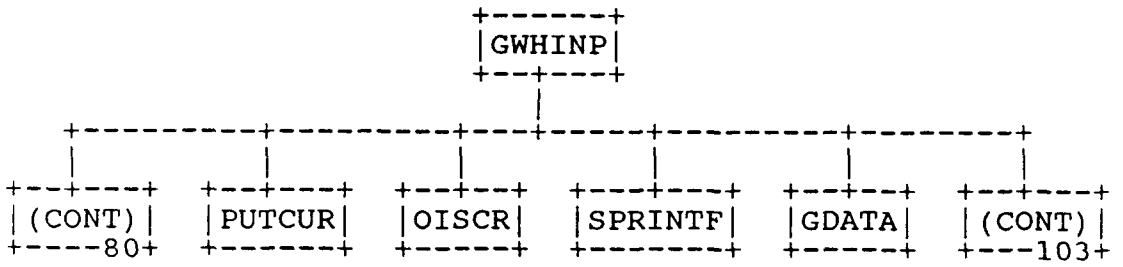
92


GTFDTX
FREE
GTFDTX/GTXINF
MALLOC
SYSMSG
ESCPY
GETLEN
MATOI

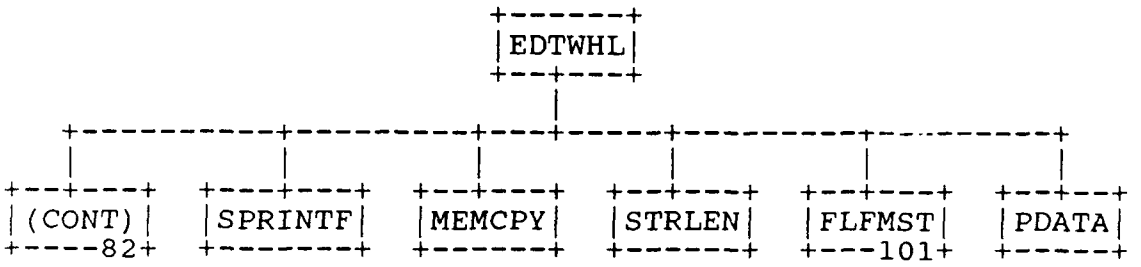
93


GITMD
FREE
MAX
GETLEN
MALLOC
SYSMSG
(CONT)
102

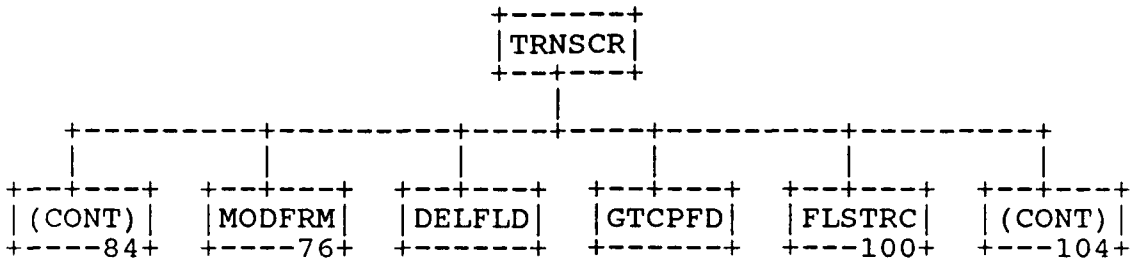
94


GWHINP
(CONT)
80
PUTCUR
OISCR
SPRINTF
GDATA
(CONT)
103

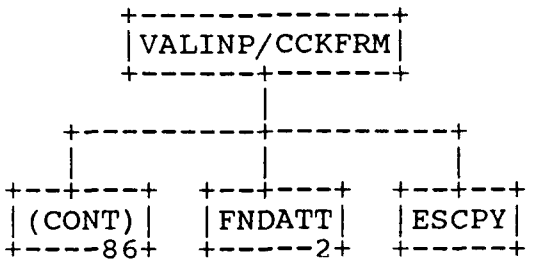
95


EDTWHL
(CONT)
82
SPRINTF
MEMCPY
STRLEN
FLFMST
101
PDATA

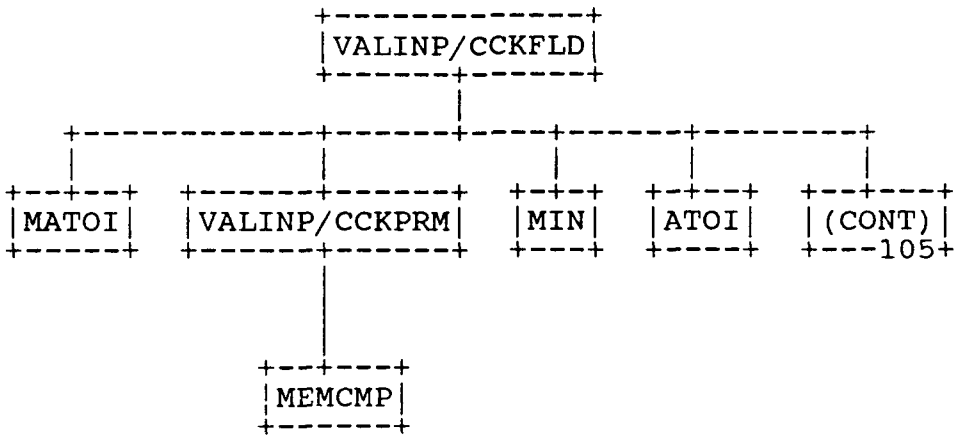
96


TRNSCR
(CONT)
84
MODFRM
76
DELFLD
GTCPFD
FLSTRC
100
(CONT)
104

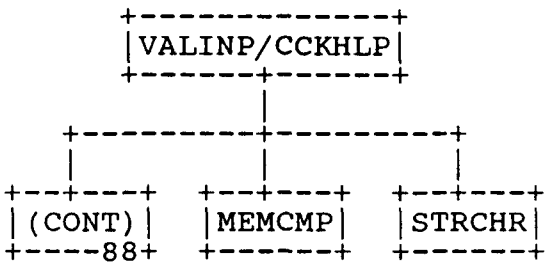
97


VALINP/CCKFRM
(CONT)
86
FNDATT
2
ESCPY

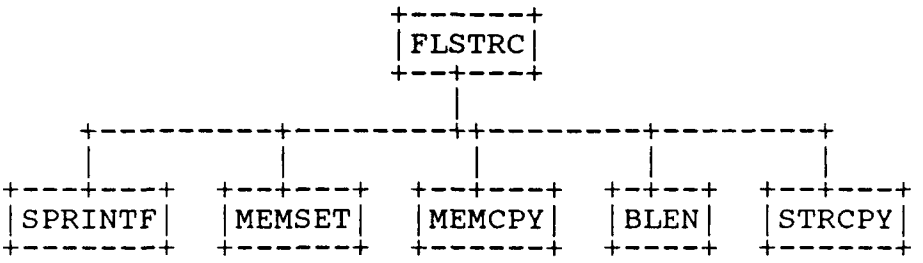
98


VALINP/CCKFLD
MATOI
VALINP/CCKPRM
MEMCMP
MIN
ATOI
(CONT)
105

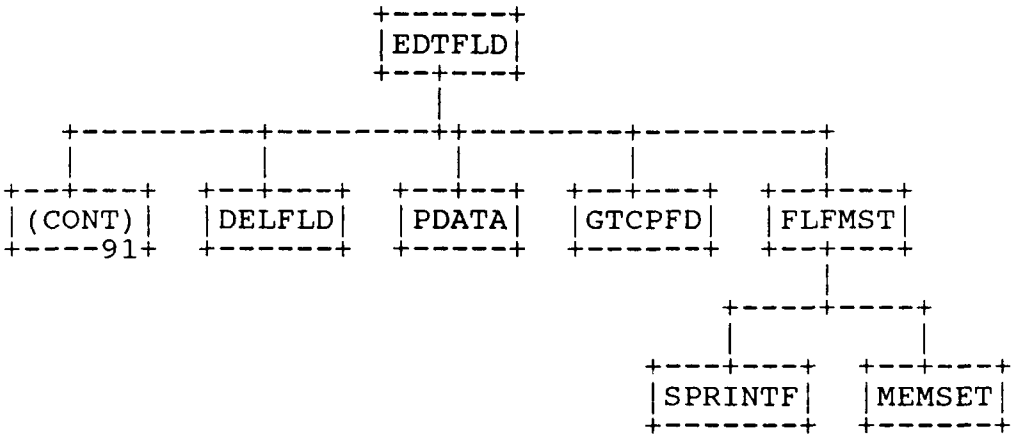
99


VALINP/CCKHLP
(CONT)
88
MEMCMP
STRCHR

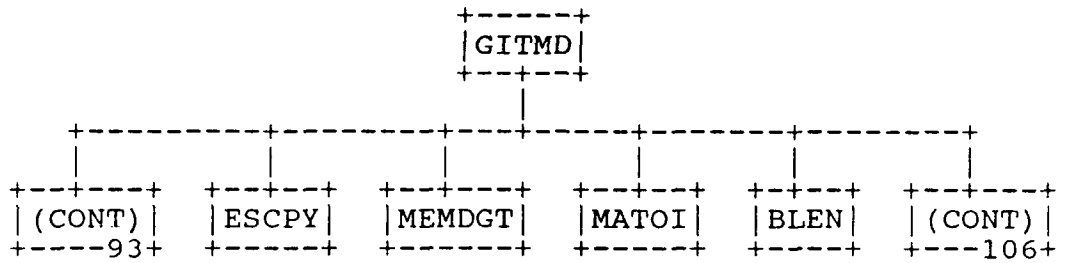
100


FLSTRC
SPRINTF
MEMSET
MEMCPY
BLEN
STRCPY

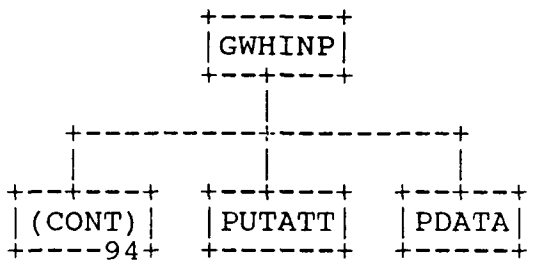
101


EDTFLD
(CONT)
91
DELFLD
PDATA
GTCPFD
FLFMST
SPRINTF
MEMSET

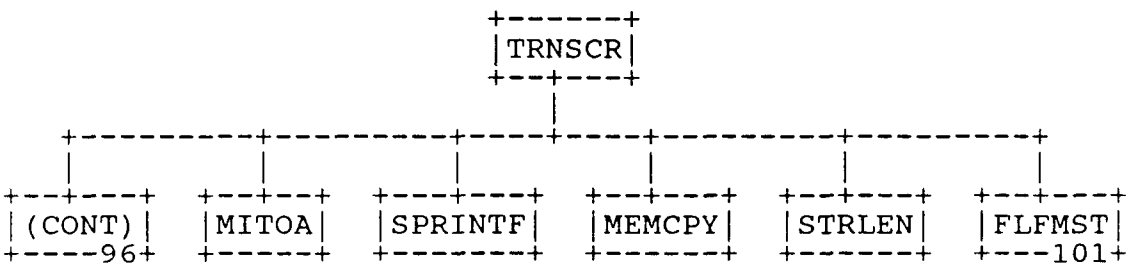
102


GITMD
(CONT)
93
ESCPY
MEMDGT
MATOI
BLEN
(CONT)
106

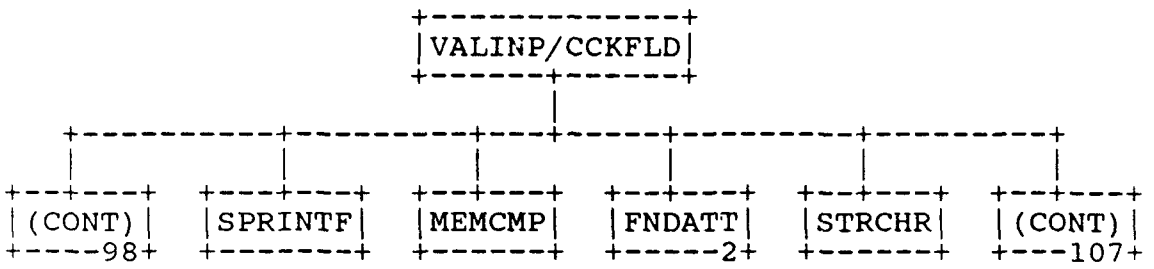
103


GWHINP
(CONT)
94
PUTATT
PDATA

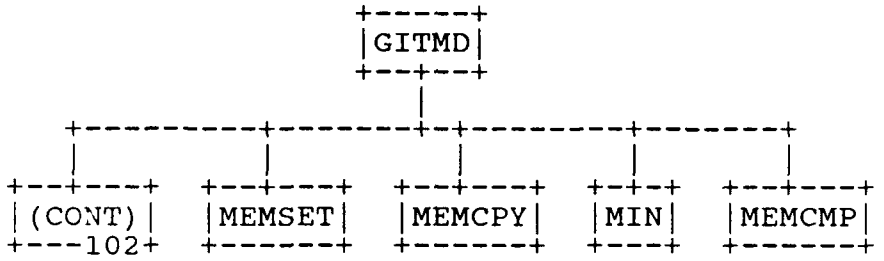
104


TRNSCR
(CONT)
96
MITOA
SPRINTF
MEMCPY
STRLEN
FLFMST
101

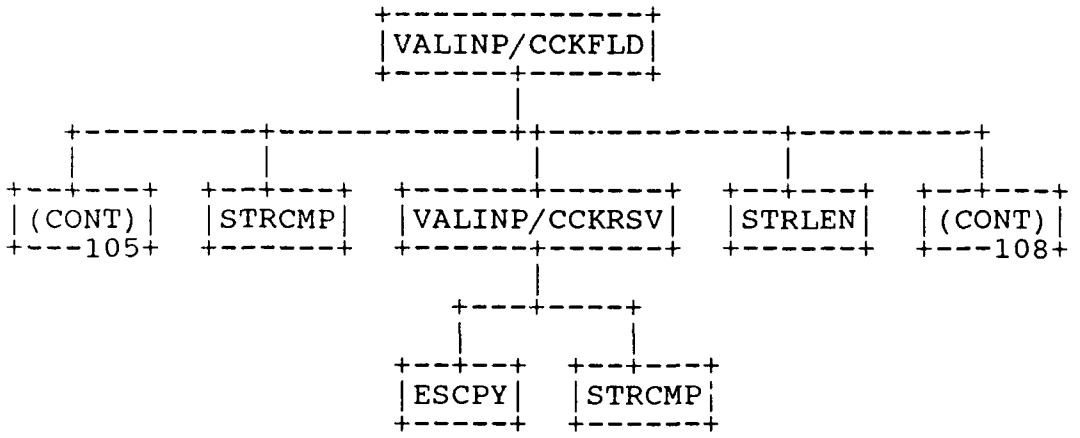
105


VALINP/CCKFLD
(CONT)
98
SPRINTF
MEMCMP
FNDATT
2
STRCHR
(CONT)
107

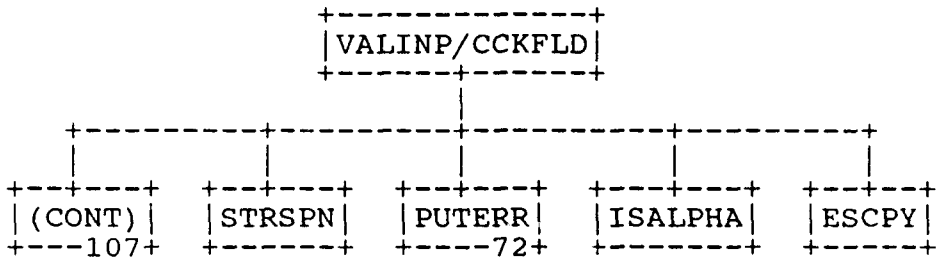
106


GITMD
(CONT)
102
MEMSET
MEMCPY
MIN
MEMCMP

107


VALINP/CCKFLD
(CONT)
105
STRCMP
VALINP/CCKRSV
STRLEN
(CONT)
108
ESCPY
STRCMP

108



ABORT
ABS
ACCESS
ADDCHK.....3
ADDEXT.....25
ADDFRM
ATOI
BLEN
CHKARY.....9
CHKFLD.....2
CHKFRM.....3
CHKPRM.....26
CLSFRM
COPFLD
COPY
CPYFRM.....64
CSTASH.....4
DELFLD
DRPFRM.....28
DRPWHL.....54
EDTFLD.....42
EDTMOD.....20
EDTWHL.....43
ERROR.....2
ESCPY
EXPAND.....14
EXPAND/FIXFRM....22
FATAL.....16
FCLOSE
FDFE.....5
FDFE/MAIN.....5
FEOF
FERROR
FGETS
FIFDST.....78
FLANCI.....23
FLDTYP
FLFMST.....101
FLSTRC.....100
FLWHST.....43
FNDATT.....2
FOPEN
FPRINTF
FREBUF.....12
FREE
GATDEF
GDATA
GETCUR
GETFLS.....12
GETFLS/TREEXP....27
GETLEN
GFDINP.....74
GFLDPT.....10
GITMD.....93
GNXTFD.....65
GNXTFD/NXTFLD....65
GTCPFD
GTFDTX.....92
GTFDTX/GTXINF....92
GTNMFD.....35
GWHINP.....80
GWINDO
INITAL
INITFP
INSFLD.....75
INSFRM.....33
INSWHL.....82
ISALPHA
ISSPACE
LAYOUT.....34
LISTFM.....44
LISTIT.....29
MAKFLD
MAKINT.....6
MAKSTR.....6
MALLOC
MATOI
MAX
MEMCMP
MEMCPY
MEMDGT
MEMSET
MIN
MITOA
MKPOS.....6
MKTEMP
MODFLD.....66
MODFRM.....76
MODFRM/FRETXT....76
MODWHL.....81
MYALLOC.....16
OISCR
PDATA
PMSGLC

PMSGLS
PRCFIL.....46
PREC.....27
PRSCMD.....30
PUTATT
PUTCUR
PUTERR.....72
RENAME
REWIND
RMVPAG
RSVATT
SAVFLS.....32
SCRMAN.....34
SCRMAN/CHGPOS....39
SCRMAN/GETROW....34
SPRINTF
STRASN
STRCAT
STRCHR
STRCMP
STRCPY
STRLEN
STRNCMP
STRNCPY
STRRCHR
STRSPN
STRUPC
SYSMSG
SYSTEM
TERMFP
TRMNAT
TRNSCR.....40
TRNSCR/FLCST.....51
TRNSCR/FRLCST....84
TRNSCR/GTFMPMT...70
TRNSCR/GTPINF....83
TRNSCR/LDPMINF...40
TRNSCR/MTCHPMT...84
TRNSCR/PARSCRN...59
TRNSCR/SPSYMB
TRNSTR.....52
TRNSTR/FLFLD.....62
TRNSTR/FLPRMPT...52
TRNSTR/GARINF....71
UNLINK
VALINP.....63
VALINP/CCKFLD....98
VALINP/CCKFRM....86
VALINP/CCKHLP....73
VALINP/CCKITM....72
VALINP/CCKNAM....87
VALINP/CCKPRM....98
VALINP/CCKRSV...107
VIEW.....11
WARNING.....8
WRTEXP.....15
WRTFDL.....48
WRTFDL/ARYREF....58
WRTFRM
YYPARSE

3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."

☆U.S. Government Printing Office:1992—648-127/62225