

WRDC-TR-90-8007
Volume VIII
Part 17

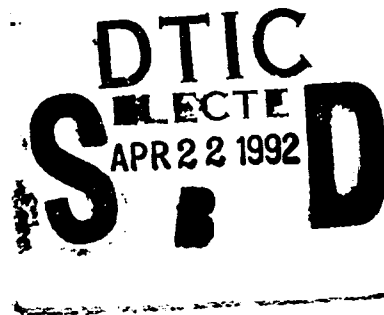
AD-A248 925



INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 17 - Forms Language Compiler Product Specification

S. Barker

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



September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

92-10289



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

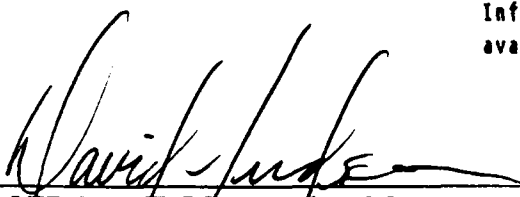
92 4 21 129

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

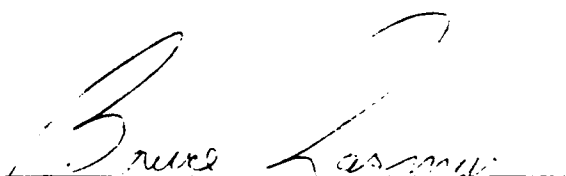


DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

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 from 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 620344401		5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR-90-8007 Vol. VIII, Part 17			
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 Forms		PROGRAM ELEMENT NO. 78011F	PROJECT NO. 595600	TASK NO. F95600	WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Structural Dynamics Research Corporation: Barker, S.					
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 135		
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 Language Compiler (FLAN) computer program. BLOCK 11: INTEGRATED INFORMATION SUPPORT SYSTEM Vol VIII - User Interface Subsystem Part 17 - Forms Language Compiler 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		

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.

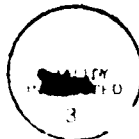
TABLE OF CONTENTS

	<u>Page</u>
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	2-2
SECTION 3.0 REQUIREMENTS	3-1
3.1 Structural Description	3-1
3.1.1 Stand Alone FLAN	3-1
3.1.2 Integrated FLAN	3-2
3.1.3 Reverse FLAN	3-2
3.1.4 MAKINC	3-2
3.2 Functional Flow	3-3
3.2.1 Stand Alone FLAN	3-3
3.2.2 Integrated FLAN	3-4
3.2.3 Reverse FLAN	3-5
3.2.4 MAKINC	3-5
3.3 Interfaces	3-6
3.3.1 Form Processor	3-6
3.3.2 Forms Driven Form Editor	3-6
3.3.3 Report Writer	3-6
3.3.4 Rapid Application Generator	3-6
3.3.5 Application Interface	3-6
3.4 Program Interrupts	3-7
3.5 Timing and Sequencing Description	3-7
3.6 Special Control Features	3-7
3.7 Storage Allocation	3-7
3.7.1 Data Base Definition	3-7
3.7.1.1 File Descriptions	3-7
3.8 Object Code Creation	3-10
3.9 Adaptation Data	3-10
3.10 Detailed Design Description	3-10
3.10.1 Main Program List	3-10
3.10.2 Module List	3-12
3.10.3 External Routines List	3-15
3.10.4 Include File List	3-18
3.10.5 Where Include File Used List	3-20
3.10.6 Where External Routine Used List	3-28
3.10.7 Main Program Parts List	3-38
3.10.8 Module Documentation	3-43
3.10.9 Include File Description	3-98
3.10.10 Hierarchy Chart	3-109
3.11 Program Listings Comments	3-124
SECTION 4.0 QUALITY ASSURANCE PROVISIONS	4-1
4.1 Introduction and Definitions	4-1
4.2 Computer Programming and Test Evaluation	4-1

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3-1	YACC Structural Description	3-1
3-2	Stand Alone FLAN Module Relationships	3-2
3-3	Integrated FLAN Module Relationships	3-2
3-4	Stand Alone FLAN Data Flow	3-3
3-5	Integrated FLAN Data Flow	3-4
3-6	Reverse FLAN Data Flow	3-5
3-7	MAKINC Data Flow	3-5
3-8	Integrated FLAN Application Interface	3-6

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



SECTION 1

SCOPE

1.1 Identification

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

1.2 Functional Summary

FLAN is a compiler which translates Form Definition Language source files into binary Form Definition File format. The binary Form Definition Files are then used as input by the Form Processor (another configuration item of the IISS UI) for display and entry of data under the control of other application programs.

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 Driven Form Editor Product Specification, PS 620144402 , 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 Language Compiler Development Specification, DS 620144401B, 1 November 1985.
- [10] Structural Dynamics Research Corporation, Forms Language Compiler Unit Test Plan, UTP620144401 , 1 November 1985.
- [11] Structural Dynamics Research Corporation, Form Editor User Manual, UM 620144400B, 1 November 1985.

2.2 Terms and Abbreviations

Application Definition Language: an extension of the Forms Definition Language that includes retrieval of database information and conditional actions. It is used to define interactive application programs.

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.

Common Data Model: (CDM), IISS subsystem that describes common data application process formats, form definitions, etc. of the IISS and includes conceptual schema, external schemas, internal schemas, and schema transformation operators.

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.

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.

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.

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.

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.

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 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.

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

SECTION 3
REQUIREMENTS

3.1 Structural Description

The detailed structure of the Forms Language Compiler is illustrated in section 3.10. The four major subsystems are:

- o Stand alone FLAN
- o Integrated FLAN
- o Reverse FLAN (REVFLAN)
- o MAKINC

3.1.1 Stand Alone FLAN

Stand alone FLAN is a batch process which resembles a normal computer language compiler in concept and use. It accepts a series of Form Definition Language statements (FDL file) as input and produces one or more Form Processor Form Definition (fd) files as output.

FLAN is partially generated by the UNIX tool YACC. The file FLAN.Y, which contains the YACC specifications for FLAN, is the input file to YACC. YACC generates the Forms Language parser, YTAB.C.

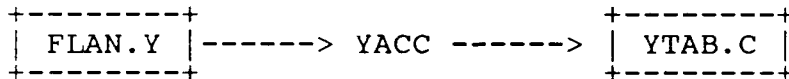


Figure 3-1 YACC Structural Description

Some support routines for the parser (FLANSP.C) were coded from scratch along with the module WRTFRM.C that writes the form definition files. Figure 3-2 describes the major module relationships for stand alone FLAN.

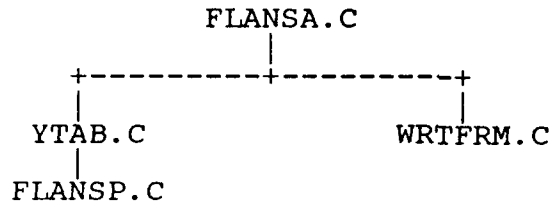


Figure 3-2 Stand Alone FLAN Module Relationships

3.1.2 Integrated FLAN

Integrated FLAN is available within the IISS environment. It makes use of the batch compiler but interfaces to the user through forms. The modules FLFRNT.C and FLUIERR.C do the interfacing with the IISS user. FLFRNT.C processes the form and FLUIERR.C performs error handling. Figure 3-3 describes the major module relationships for the integrated FLAN.

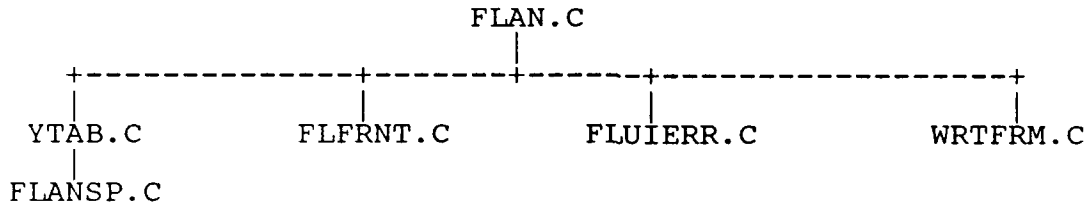


Figure 3-3 Integrated FLAN Module Relationships

3.1.3 Reverse FLAN

Reverse FLAN (REVFLAN) is a batch process that generates Form Definition Language specifications from Form Processor Form Definition (fd) files.

3.1.4 MAKINC

MAKINC is a batch process that generates a data structure corresponding to the item fields in a form and its static subforms. These data structures can be used in application programs as include files when calling the Form Processor routines GDATA and PDATA.

3.2 Functional Flow

3.2.1 Stand Alone FLAN

Figure 3-4 is a data flow diagram of Stand Alone FLAN. The Compile Language File process is almost entirely generated by YACC.

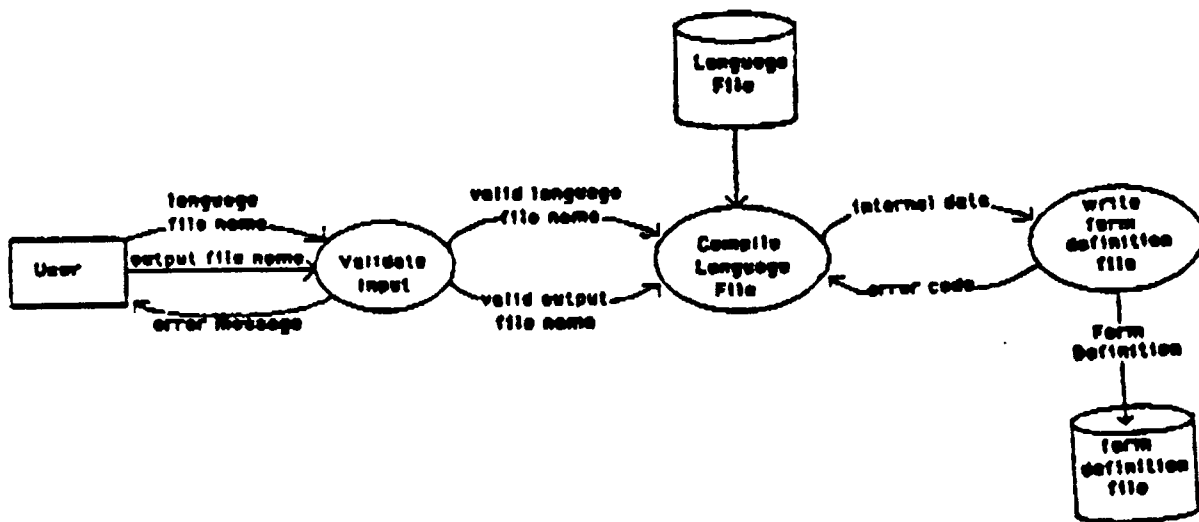


Figure 3-4 Stand Alone FLAN Data Flow

3.2.2 Integrated FLAN

Figure 3-5 is a data flow diagram of Integrated FLAN.

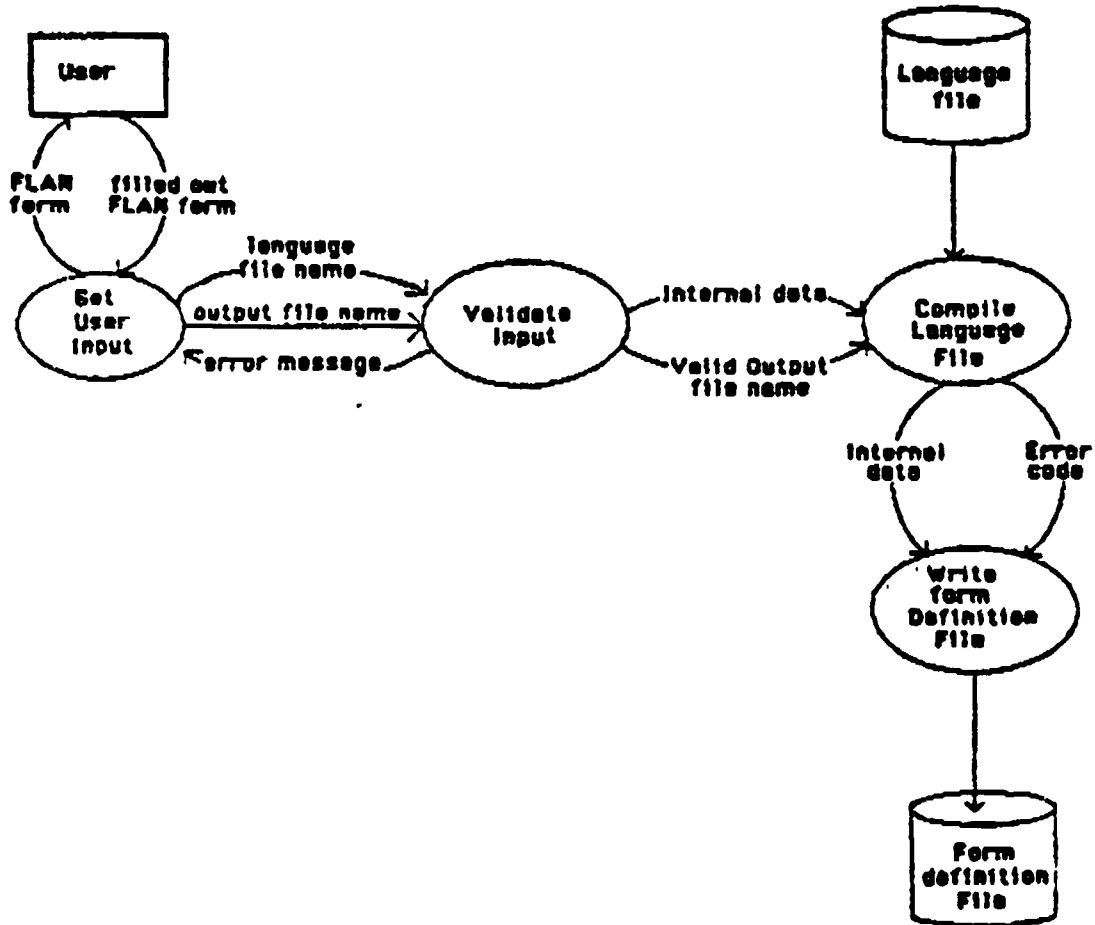


Figure 3-5 Integrated FLAN Data Flow

3.2.3 Reverse FLAN

Figure 3-6 is a data flow diagram of Reverse FLAN.



Figure 3-6 Reverse FLAN Data Flow

3.2.4 MAKINC

Figure 3-7 is a data flow diagram of MAKINC.

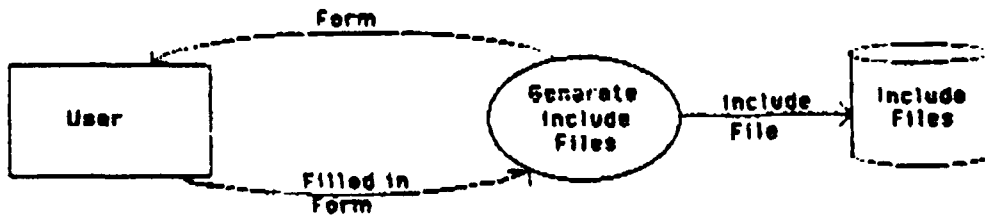


Figure 3-7 MAKINC Data Flow

3.4 Program Interrupts

This section does not apply to the detailed design of the Forms Language Compiler.

3.5 Timing and Sequencing Description

The data flow diagrams 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 Forms Language Compiler does not include any special control features as defined in the ICAM Documentation Standards manual.

3.7 Storage Allocation

The executable sizes for Stand Alone FLAN, Integrated FLAN, Reverse FLAN, and MAKINC are:

Stand Alone FLAN	138 blocks
Integrated FLAN	186 blocks
Reverse FLAN	47 blocks
MAKINC	102 blocks

3.7.1 Data Base Definition

3.7.1.1 File Descriptions

1. FILE NAME: FLAN.Y - Sequential file containing the YACC specifications for the FLAN parser.

PURPOSE: This file is used by YACC to generate the source file YTAB.C which is the parser for FLAN.

DECLARATION:

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

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

PURPOSE: This file contains the language description of one or more forms. It is compiled to produce Form Definition files.

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 620344401). 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 help_line[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;
```

4. FILE NAME: formname.inc - This include file is generated by MAKINC and contains the data declaration that corresponds to the form definition. Its name is dependent upon its corresponding form.

PURPOSE: These include files are used by application programs that use the Form Processor. Generating these include files from the form definitions eliminates the need for editing the application code whenever data definitions for form variables change.

DECLARATION: dependent on Form Definition file

3.8 Object Code Creation

The FLAN routine YTAB.C was generated by the UNIX tool YACC. It is compiled like any other C program. All other modules were compiled using a C compiler developed by Interactive Software under VAX/VMS.

3.9 Adaptation Data

The C source modules for the Forms Language Compiler can be compiled using any UNIX version 7 compatible C compiler. The YTAB.C module can be used just as any other C module, but it can only be generated with YACC.

The files FPDINI.H and GETFLS.C contain file names for the form definition and forms 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 LANGUAGE COMPILER Main Program List

<u>Module Name</u>	<u>Purpose</u>
FLAN/MAIN	FLAN MAIN PROGRAM
MAKINC/MAIN	MAKE INCLUDE FILES FOR FORMS
REVFLAN/MAIN	REVERSE FLAN

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 LANGUAGE COMPILER Module List

Module Name -----	Purpose -----
ADDCHK	ADD POSITION TO CHECK LIST
CHKARY	CHECK ARRAY
CHKFLD	CHECK FIELD
CHKFRM	CHECK FORM
COBSUB	COBOL SUBROUTINE
CSTASH	CHARACTER STASH
CSUB	C SUBROUTINE
ERROR	ISSUE ERROR MESSAGE
FATAL	ISSUE FATAL ERROR MESSAGE
FLAN/MAIN	FLAN MAIN PROGRAM
FLANCI	FLAN CALLABLE INTERFACE
FLDTYP	FIELD TYPE
FNDATT	FIND ATTRIBUTE
FRNTND	FORMS FRONT END FOR FLAN
GETFILE	GET INPUT FILENAME
GFLDPT	GET FIELD POINTER
MAKACT	MAKE ACTION LIST ELEMENT
MAKINC/INDENT	INDENT OUTPUT LINE
MAKINC/MAIN	MAKE INCLUDE FILES FOR FORMS
MAKINT	MAKE EXPRESSION INTO AN INTEGER
MAKSTR	MAKE EXPRESSION INTO A STRING

FORMS LANGUAGE COMPILER Module List

Module Name -----	Purpose -----
MKPOS	MAKE POSITION NODE
MYALLOC	MY MALLOC
PLISUB	PL/I SUBROUTINE
REVFLAN/MAIN	REVERSE FLAN
WARNING	ISSUE WARNING MESSAGE
WRTEXP	WRITE EXPRESSION
WRTFRM	WRITE FORM
WRTFRM/DBFCLOS	DEFAULT BUFFER CLOSE
WRTFRM/FORMAT	INSERT FORMAT CODES
WRTFRM/TBFCLOS	TEXT BUFFER CLOSE
WRTFRM/WRTDBF	WRITE DEFAULT BUFFER
WRTFRM/WRTFLD	WRITE FIELD
WRTFRM/WRTTBF	WRITE TEXT BUFFER
WRTFRM/WRTTXT	WRITE TEXT
YYLEX	LEXICAL ANALYZER FOR FLAN
YYPARSE	FLAN PARSER

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. See section 3.10.6 for a list of the modules that call each of these external routines.

FORMS LANGUAGE COMPILER External Routines List

Module Name -----	First User -----
ABS	CHKARY
ADDFRM	FRNTND
ATOF	YYLEX
ATOI	YYLEX
BLEN	CSUB
CALLOC	MAKINC/MAIN
CLSFRM	MAKINC/MAIN
DELFLD	FLANCI
FCLOSE	WRTRFM
FEOF	MAKINC/MAIN
FNDMSG	MAKINC/MAIN
FOPEN	MAKINC/MAIN
FPRINTF	COBSUB
FREAD	REVFLAN/MAIN
FREE	YYPARSE
FWRITE	WRTRFM/TBFCLOS
GDATA	FRNTND
GETC	REVFLAN/MAIN
GETCHAR	MAKINC/MAIN
GFPTR	MAKINC/MAIN
INITAL	FRNTND
INITFP	FRNTND
ISALNUM	YYLEX
ISALPHA	YYLEX
ISDIGIT	YYLEX
ISSPACE	YYLEX
MAKFLD	YYPARSE
MALLOC	MYALLOC
MAX	CHKFLD
MEMCMP	FRNTND
MEMCPY	YYPARSE
MEMSET	CHKFLD
OISCR	FRNTND
OPNFRM	MAKINC/MAIN
PMSGLC	FRNTND
PMSGLS	ERROR
PRINTF	MAKINC/MAIN
PUTC	MAKINC/INDENT
SCANF	REVFLAN/MAIN
SPRINTF	FATAL
STRASN	CHKARY

FORMS LANGUAGE COMPILER External Routines List

Module Name -----	First User -----
STRCAT	YYPARSE
STRCHR	FLAN/MAIN
STRCMP	GFLDPT
STRCPY	CSUB
STRLEN	WRTRFM/WRTTXT
STRLOC	CSUB
STRNCMP	REVFLAN/MAIN
STRNCPY	WRTRFM/WRTFLD
STRUPC	YYPARSE
SYSMSG	CHKFLD
TERMFP	FLAN/MAIN
TOUPPER	YYLEX
TRMNAT	FLAN/MAIN
UNGETC	YYLEX
YYERROR	YYPARSE

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 LANGUAGE COMPILER Include File List

<u>File Name</u>	<u>Purpose</u>
CTYPE	**** PURPOSE NOT FOUND BY STRIPPER ****
FFV2	FORM FILE FORMAT - VERSION 2
FLAN.Y"	**** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	FORM PROCESSOR RETURN CODES
FPD	FORM PROCESSOR DATA
FPDINI	FPD INITIALIZATION
FPPARM	FORM PROCESSOR PARAMETERS
MATH	**** PURPOSE NOT FOUND BY STRIPPER ****
NTM	NTM INTERFACE INCLUDE FILE
RW	REPORT WRITER DEFINITIONS
STDIO	**** PURPOSE NOT FOUND BY STRIPPER ****
STDYTP	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 LANGUAGE COMPILER Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
CTYPE		
	MAKACT	MAKE ACTION LIST ELEMENT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
FFFV2		
	WRTFRM	WRITE FORM
	WRTFRM/DB	DEFAULT BUFFER CLOSE
	WRTFRM/FO	INSERT FORMAT CODES
	WRTFRM/TB	TEXT BUFFER CLOSE
	WRTFRM/WR	WRITE DEFAULT BUFFER
	WRTFRM/WR	WRITE FIELD
	WRTFRM/WR	WRITE TEXT BUFFER
	WRTFRM/WR	WRITE TEXT
FLAN.Y"		
	MAKACT	MAKE ACTION LIST ELEMENT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
FPCODE		
	ADDCHK	ADD POSITION TO CHECK LIST
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	COBSUB	COBOL SUBROUTINE
	CSTASH	CHARACTER STASH
	CSUB	C SUBROUTINE
	FLANCI	FLAN CALLABLE INTERFACE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE

FORMS LANGUAGE COMPILER Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	GFLDPT	GET FIELD POINTER
	MAKINC/IN	INDENT OUTPUT LINE
	MAKINC/MA	MAKE INCLUDE FILES FOR FORMS
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MKPOS	MAKE POSITION NODE
	MYALLOC	MY MALLOC
	PLISUB	PL/I SUBROUTINE
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	WRTFRM/DB	DEFAULT BUFFER CLOSE
	WRTFRM/FO	INSERT FORMAT CODES
	WRTFRM/TB	TEXT BUFFER CLOSE
	WRTFRM/WR	WRITE DEFAULT BUFFER
	WRTFRM/WR	WRITE FIELD
	WRTFRM/WR	WRITE TEXT BUFFER
	WRTFRM/WR	WRITE TEXT

FPD

ADDCHK	ADD POSITION TO CHECK LIST
CHKARY	CHECK ARRAY
CHKFLD	CHECK FIELD
CHKFRM	CHECK FORM
COBSUB	COBOL SUBROUTINE
CSTASH	CHARACTER STASH
CSUB	C SUBROUTINE
FLAN/MAIN	FLAN MAIN PROGRAM
FLANCI	FLAN CALLABLE INTERFACE
FLDTYP	FIELD TYPE
FNDATT	FIND ATTRIBUTE
GETFILE	GET INPUT FILENAME
GFLDPT	GET FIELD POINTER
MAKACT	MAKE ACTION LIST ELEMENT
MAKINC/IN	INDENT OUTPUT LINE
MAKINC/MA	MAKE INCLUDE FILES FOR FORMS
MAKINT	MAKE EXPRESSION INTO AN INTEGER
MAKSTR	MAKE EXPRESSION INTO A STRING

FORMS LANGUAGE COMPILER Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	MKPOS	MAKE POSITION NODE
	MYALLOC	MY MALLOC
	PLISUB	PL/I SUBROUTINE
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	WRTFRM/DB	DEFAULT BUFFER CLOSE
	WRTFRM/FO	INSERT FORMAT CODES
	WRTFRM/TB	TEXT BUFFER CLOSE
	WRTFRM/WR	WRITE DEFAULT BUFFER
	WRTFRM/WR	WRITE FIELD
	WRTFRM/WR	WRITE TEXT BUFFER
	WRTFRM/WR	WRITE TEXT
	YLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
FPDINI	COBSUB	COBOL SUBROUTINE
	CSUB	C SUBROUTINE
	FLAN/MAIN	FLAN MAIN PROGRAM
	GETFILE	GET INPUT FILENAME
	MAKINC/IN	INDENT OUTPUT LINE
	MAKINC/MA	MAKE INCLUDE FILES FOR FORMS
	PLISUB	PL/I SUBROUTINE
FPPARM	COBSUB	COBOL SUBROUTINE
	CSUB	C SUBROUTINE
	FLAN/MAIN	FLAN MAIN PROGRAM
	FRNTND	FORMS FRONT END FOR FLAN
	GETFILE	GET INPUT FILENAME
	MAKACT	MAKE ACTION LIST ELEMENT
	MAKINC/IN	INDENT OUTPUT LINE
	MAKINC/MA	MAKE INCLUDE FILES FOR FORMS
	PLISUB	PL/I SUBROUTINE

FORMS LANGUAGE COMPILER Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	YYLEX YYPARSE	LEXICAL ANALYZER FOR FLAN FLAN PARSER
MATH	MAKACT YYLEX YYPARSE	MAKE ACTION LIST ELEMENT LEXICAL ANALYZER FOR FLAN FLAN PARSER
NTM	FLAN/MAIN FRNTND GETFILE	FLAN MAIN PROGRAM FORMS FRONT END FOR FLAN GET INPUT FILENAME
RW	ADDCHK CHKARY CHKFLD CHKFRM CSTASH FLANCI FLDTYP FNDATT GFLDPT MAKACT MAKINT MAKSTR MKPOS MYALLOC WRTEXP YYLEX YYPARSE	ADD POSITION TO CHECK LIST CHECK ARRAY CHECK FIELD CHECK FORM CHARACTER STASH FLAN CALLABLE INTERFACE FIELD TYPE FIND ATTRIBUTE GET FIELD POINTER MAKE ACTION LIST ELEMENT MAKE EXPRESSION INTO AN INTEGER MAKE EXPRESSION INTO A STRING MAKE POSITION NODE MY MALLOC WRITE EXPRESSION LEXICAL ANALYZER FOR FLAN FLAN PARSER

FORMS LANGUAGE COMPILER Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
STDIO		
	ADDCHK	ADD POSITION TO CHECK LIST
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	COBSUB	COBOL SUBROUTINE
	CSTASH	CHARACTER STASH
	CSUB	C SUBROUTINE
	FLAN/MAIN	FLAN MAIN PROGRAM
	FLANCI	FLAN CALLABLE INTERFACE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE
	GETFILE	GET INPUT FILENAME
	GFLDPT	GET FIELD POINTER
	MAKACT	MAKE ACTION LIST ELEMENT
	MAKINC/IN	INDENT OUTPUT LINE
	MAKINC/MA	MAKE INCLUDE FILES FOR FORMS
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MKPOS	MAKE POSITION NODE
	MYALLOC	MY MALLOC
	PLISUB	PL/I SUBROUTINE
	REVFLAN/M	REVERSE FLAN
	WRTEXP	WRITE EXPRESSION
	WRTRFM	WRITE FORM
	WRTRFM/DB	DEFAULT BUFFER CLOSE
	WRTRFM/FO	INSERT FORMAT CODES
	WRTRFM/TB	TEXT BUFFER CLOSE
	WRTRFM/WR	WRITE DEFAULT BUFFER
	WRTRFM/WR	WRITE FIELD
	WRTRFM/WR	WRITE TEXT BUFFER
	WRTRFM/WR	WRITE TEXT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER

FORMS LANGUAGE COMPILER Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
--------------------------	-------------------------	----------------------------

STDTYP

ADDCHK		ADD POSITION TO CHECK LIST
CHKARY		CHECK ARRAY
CHKFLD		CHECK FIELD
CHKFRM		CHECK FORM
COBSUB		COBOL SUBROUTINE
CSTASH		CHARACTER STASH
CSUB		C SUBROUTINE
ERROR		ISSUE ERROR MESSAGE
FATAL		ISSUE FATAL ERROR MESSAGE
FLAN/MAIN		FLAN MAIN PROGRAM
FLANCI		FLAN CALLABLE INTERFACE
FLDTYP		FIELD TYPE
FNDATT		FIND ATTRIBUTE
FRNTND		FORMS FRONT END FOR FLAN
GETFILE		GET INPUT FILENAME
GFLDPT		GET FIELD POINTER
MAKACT		MAKE ACTION LIST ELEMENT
MAKINC/IN		INDENT OUTPUT LINE
MAKINC/MA		MAKE INCLUDE FILES FOR FORMS
MAKINT		MAKE EXPRESSION INTO AN INTEGER
MAKSTR		MAKE EXPRESSION INTO A STRING
MKPOS		MAKE POSITION NODE
MYALLOC		MY MALLOC
PLISUB		PL/I SUBROUTINE
REVFLAN/M		REVERSE FLAN
WARNING		ISSUE WARNING MESSAGE
WRTEXP		WRITE EXPRESSION
WRTFRM		WRITE FORM
WRTFRM/DB		DEFAULT BUFFER CLOSE
WRTFRM/FO		INSERT FORMAT CODES
WRTFRM/TB		TEXT BUFFER CLOSE
WRTFRM/WR		WRITE DEFAULT BUFFER
WRTFRM/WR		WRITE FIELD
WRTFRM/WR		WRITE TEXT BUFFER
WRTFRM/WR		WRITE TEXT

FORMS LANGUAGE COMPILER Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER

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 LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
ABS	CHKARY CHKFRM CHKFRM	CHECK ARRAY CHECK FORM CHECK FORM
ADDFRM	FRNTND	FORMS FRONT END FOR FLAN
ATOF	YYLEX	LEXICAL ANALYZER FOR FLAN
ATOI	YYLEX	LEXICAL ANALYZER FOR FLAN
BLFN	CHKFLD COBSUB CSUB PLISUB	CHECK FIELD COBOL SUBROUTINE C SUBROUTINE PL/I SUBROUTINE
CALOC	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
CLSFRM	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS

FORMS LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
DELFLD	FLANCI	FLAN CALLABLE INTERFACE
FCLOSE	MAKINC/MAIMAKE REVFLAN/MAREVERSE WRTFRM	INCLUDE FILES FOR FORMS FLAN WRITE FORM
FEOF	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
FNDMSG	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
FOPEN	GETFILE MAKINC/MAIMAKE REVFLAN/MAREVERSE WRTFRM	GET INPUT FILENAME INCLUDE FILES FOR FORMS FLAN WRITE FORM
FPRINTF	COBSUB CSUB PLISUB REVFLAN/MAREVERSE	COBOL SUBROUTINE C SUBROUTINE PL/I SUBROUTINE FLAN
FREAD	REVFLAN/MAREVERSE	FLAN

FORMS LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
FREE	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	WRTEXP	WRITE EXPRESSION
	YYPARSE	FLAN PARSER
FWRITE	WRTFRM	WRITE FORM
	WRTFRM/DBFDEFAULT	BUFFER CLOSE
	WRTFRM/TBFTEXT	BUFFER CLOSE
	WRTFRM/WRTWRITE	DEFAULT BUFFER
	WRTFRM/WRTWRITE	FIELD
	WRTFRM/WRTWRITE	TEXT BUFFER
	WRTFRM/WRTWRITE	TEXT
GDATA	FRNTND	FORMS FRONT END FOR FLAN
GETC	REVFLAN/MAREVERSE	FLAN
	YYLEX	LEXICAL ANALYZER FOR FLAN
GETCHAR	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
GOFPTR	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS

FORMS LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
INITAL	FRNTND	FORMS FRONT END FOR FLAN
INITFP	FRNTND	FORMS FRONT END FOR FLAN
ISALNUM	YYLEX	LEXICAL ANALYZER FOR FLAN
ISALPHA	YYLEX	LEXICAL ANALYZER FOR FLAN
ISDIGIT	YYLEX	LEXICAL ANALYZER FOR FLAN
ISSPACE	YYLEX	LEXICAL ANALYZER FOR FLAN
MAKFLD	YYPARSE	FLAN PARSER
MALLOC	CSUB	C SUBROUTINE

FORMS LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	MYALLOC	MY MALLOC
	REVFLAN/MAREVERSE	FLAN
MAX	CHKFLD CHKFRM	CHECK FIELD CHECK FORM
MEMCMP	FRNTND	FORMS FRONT END FOR FLAN
MEMCPY	CHKFLD WRTEXP WRTFRM/WRTWRITE YYPARSE	CHECK FIELD WRITE EXPRESSION FIELD FLAN PARSER
MEMSET	CHKFLD	CHECK FIELD
OISCR	FLAN/MAIN FRNTND	FLAN MAIN PROGRAM FORMS FRONT END FOR FLAN
OPNFRM	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS

FORMS LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
PMSGLC	FRNTND	FORMS FRONT END FOR FLAN
PMSGLS	ERROR	ISSUE ERROR MESSAGE
	FATAL	ISSUE FATAL ERROR MESSAGE
	WARNING	ISSUE WARNING MESSAGE
PRINTF	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
	REVFLAN/MAREVERSE	FLAN
	YYPARSE	FLAN PARSER
PUTC	MAKINC/INDIDENT	OUTPUT LINE
	PLISUB	PL/I SUBROUTINE
	REVFLAN/MAREVERSE	FLAN
SCANF	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
	REVFLAN/MAREVERSE	FLAN
SPRINTF	ERROR	ISSUE ERROR MESSAGE
	FATAL	ISSUE FATAL ERROR MESSAGE
	FRNTND	FORMS FRONT END FOR FLAN
	GETFILE	GET INPUT FILENAME
	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
	REVFLAN/MAREVERSE	FLAN
	WARNING	ISSUE WARNING MESSAGE

FORMS LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	YYPARSE	FLAN PARSER
STRASN	CHKARY	CHECK ARRAY
	CHKFRM	CHECK FORM
	WRTFRM	WRITE FORM
STRCAT	YYPARSE	FLAN PARSER
STRCHR	FLAN/MAIN	FLAN MAIN PROGRAM
	FRNTND	FORMS FRONT END FOR FLAN
	YYPARSE	FLAN PARSER
STRCMP	FNDATT	FIND ATTRIBUTE
	GFLDPT	GET FIELD POINTER
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
STRCPY	CSTASH	CHARACTER STASH
	CSUB	C SUBROUTINE
	WRTFRM	WRITE FORM
	WRTFRM/WRTWRITE	FIELD
	YYPARSE	FLAN PARSER

FORMS LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
STRLEN	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CSTASH	CHARACTER STASH
	ERROR	ISSUE ERROR MESSAGE
	FATAL	ISSUE FATAL ERROR MESSAGE
	REVFLAN/MAREVERSE	FLAN
	WARNING	ISSUE WARNING MESSAGE
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	WRTFRM/WRTWRITE	TEXT
	YYPARSE	FLAN PARSER
STRLOC	CSUB	C SUBROUTINE
STRNCMP	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
	REVFLAN/MAREVERSE	FLAN
STRNCPY	WRTFRM/WRTWRITE	FIELD
	YYPARSE	FLAN PARSER
STRUPC	MAKINC/MAIMAKE	INCLUDE FILES FOR FORMS
	YYPARSE	FLAN PARSER

FORMS LANGUAGE COMPILER Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
SYSMSG	CHKFLD WRTFRM	CHECK FIELD WRITE FORM
TERMFP	FLAN/MAIN	FLAN MAIN PROGRAM
TOUPPER	YYLEX	LEXICAL ANALYZER FOR FLAN
TRMNAT	FLAN/MAIN FRNTND	FLAN MAIN PROGRAM FORMS FRONT END FOR FLAN
UNGETC	YYLEX	LEXICAL ANALYZER FOR FLAN
YYERROR	YYPARSE	FLAN PARSER

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 than 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 LANGUAGE COMPILER Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
FLAN/MAIN		Purpose-->FLAN MAIN PROGRAM
	ABS	External routine
	ADDCHK	Well-defined module
	ADDFRM	External routine
	ATOF	External routine
	ATOI	External routine
	BLFN	External routine
	CHKARY	Well-defined module
	CHKFLD	Well-defined module
	CHKFRM	Well-defined module
	CSTASH	Well-defined module
	DELFLD	External routine
	ERROR	Well-defined module
	FATAL	Well-defined module
	FCLOSE	External routine
	FLANCI	Well-defined module
	FLDTYP	Well-defined module
	FNDATT	Well-defined module
	FOPEN	External routine
	FREE	External routine
	FRNTND	Well-defined module
	FWRITE	External routine
	GDATA	External routine
	GETC	External routine
	GETFILE	Well-defined module
	GFLDPT	Well-defined module
	INITAL	External routine
	INITFP	External routine
	ISALNUM	External routine
	ISALPHA	External routine
	ISDIGIT	External routine
	ISSPACE	External routine
	MAKACT	Well-defined module
	MAKFLD	External routine
	MAKINT	Well-defined module
	MAKSTR	Well-defined module
	MALLOC	External routine
	MAX	External routine
	MEMCMP	External routine
	MEMCPY	External routine

FORMS LANGUAGE COMPILER Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	MEMSET	External routine
	MKPOS	Well-defined module
	MYALLOC	Well-defined module
	OISCR	External routine
	PMSGLC	External routine
	PMSGLS	External routine
	PRINTF	External routine
	SPRINTF	External routine
	STRASN	External routine
	STRCAT	External routine
	STRCHR	External routine
	STRCMP	External routine
	STRCPY	External routine
	STRLEN	External routine
	STRNCPY	External routine
	STRUPC	External routine
	SYSMSG	External routine
	TERMFP	External routine
	TOUPPER	External routine
	TRMNAT	External routine
	UNGETC	External routine
	WARNING	Well-defined module
	WRTEXP	Well-defined module
	WRTFRM	Well-defined module
	WRTFRM/DBFCLOS	Well-defined module
	WRTFRM/FORMAT	Well-defined module
	WRTFRM/TBFCLOS	Well-defined module
	WRTFRM/WRTDBF	Well-defined module
	WRTFRM/WRTFLD	Well-defined module
	WRTFRM/WRTTBF	Well-defined module
	WRTFRM/WRTTXT	Well-defined module
	YYERROR	External routine
	YYLEX	Well-defined module
	YYPARSE	Well-defined module

FORMS LANGUAGE COMPILER Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
MAKINC/MAIN		Purpose-->MAKE INCLUDE FILES FOR FORMS
	BLEN	External routine
	CALLOC	External routine
	CLSFRM	External routine
	COBSUB	Well-defined module
	CSUB	Well-defined module
	FCLOSE	External routine
	FEOF	External routine
	FNDMSG	External routine
	FOPEN	External routine
	FPRINTF	External routine
	GETCHAR	External routine
	GOFPTR	External routine
	MAKINC/INDENT	Well-defined module
	MALLOC	External routine
	OPNFRM	External routine
	PLISUB	Well-defined module
	PRINTF	External routine
	PUTC	External routine
	SCANF	External routine
	SPRINTF	External routine
	STRCPY	External routine
	STRLOC	External routine
	STRNCMP	External routine
	STRUPC	External routine

FORMS LANGUAGE COMPILER Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
REVFLAN/MAIN	Purpose-->	REVERSE FLAN
	FCLOSE	External routine
	FOPEN	External routine
	FPRINTF	External routine
	FREAD	External routine
	GETC	External routine
	MAI·LOC	External routine
	PRINTF	External routine
	PUTC	External routine
	SCANF	External routine
	SPRINTF	External routine
	STRLEN	External routine
	STRNCMP	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 LANGUAGE COMPILER 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:

STD TYP - 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):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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:

STD TYP - 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):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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:

STDYTP - 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

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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: FDPE/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

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: COBSUB
PURPOSE: COBOL SUBROUTINE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKINC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

COBSUB - COBOL SUBROUTINE
BLEN
FPRINTF
MAKINC/INDENT - INDENT OUTPUT LINE

CALLED DIRECTLY BY:

COBSUB - COBOL SUBROUTINE
MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS

USED IN MAIN PROGRAM(S):

MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS

FORMS LANGUAGE COMPILER 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:

STDTPY - 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

CALLED DIRECTLY BY:

YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: CSUB
PURPOSE: C SUBROUTINE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKINC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

CSUB - C SUBROUTINE
BLEN
MAKINC/INDENT - INDENT OUTPUT LINE
FPRINTF
STRLOC
STRCPY
MALLOC

CALLED DIRECTLY BY:

CSUB - C SUBROUTINE
MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS

USED IN MAIN PROGRAM(S):

MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS

FORMS LANGUAGE COMPILER Module Documentation

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

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:

```
FLAN/MAIN - FLAN MAIN PROGRAM
GETFILE - GET INPUT FILENAME
CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM
```

ADDCHK - ADD POSITION TO CHECK LIST
YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

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

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
YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: FLAN/MAIN
PURPOSE: FLAN MAIN PROGRAM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: FLAN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FLAN
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

SYNOPSIS

```
MAIN(ARGC, ARGV)
  INT ARGC;
  CHAR *ARGV[];
```

DESCRIPTION

MAIN PROGRAM. PROMPTS FOR FILE NAME IF NOT GIVEN, CALLS
PARSER, CALLS
WRITEOUT IF NO ERRORS.

ARGUMENTS:

ARGC = INT
ARGV = CHAR * []

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
NTM - NTM INTERFACE INCLUDE FILE
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

STRCHR
GETFILE - GET INPUT FILENAME
FLANCI - FLAN CALLABLE INTERFACE
WRTRFM - WRITE FORM
ERROR - ISSUE ERROR MESSAGE
OISCR
TERMPF
TRMNAT
FRNTND - FORMS FRONT END FOR FLAN

FORMS LANGUAGE COMPILER 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 - FLAN PARSER
DELFLD

CALLED DIRECTLY BY:

FLAN/MAIN - FLAN MAIN PROGRAM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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:

STD TYP - 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):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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:

STD TYP - 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:

CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: FRNTND
PURPOSE: FORMS FRONT END FOR FLAN
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: FLFRNT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FLAN
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

SYNOPSIS

CHAR *FLFRNT()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

THIS FUNCTION PRESENTS A TOP LEVEL FORM REQUESTING A FILE NAME FROM THE USER. IT RETURNS THAT FILE NAME TO GRP. THE NAME OF THE FORM IS "APFRONT.FDL" FOR THE APPLICATION GENERATOR AND "RWFRONT.FDL" FOR THE REPORT WRITER AND "FLFRONT.FDL" FOR FLAN. IT IS HARDCODED INTO THE ROUTINE. THERE IS ONE COPY OF THIS ROUTINE FOR THE AP AND ONE FOR THE RW AND ONE FOR FLAN.

ARGUMENTS:

FILNAM = CHAR [41]

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPPARM - FORM PROCESSOR PARAMETERS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

STRCHR
INITAL
MEMCMP
TRMNAT
PMSGC
INITFP
ADDFRM
GDATA
OISCR
SPRINTF

CALLED DIRECTLY BY:

FLAN/MAIN - FLAN MAIN PROGRAM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: GETFILE
PURPOSE: GET INPUT FILENAME
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FILE * ()
SOURCE FILE: FLAN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FLAN
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

ARGUMENTS:

ARGC = INT
NAMPTR = CHAR *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
NTM - NTM INTERFACE INCLUDE FILE
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

SPRINTF
FOPEN
ERROR - ISSUE ERROR MESSAGE

CALLED DIRECTLY BY:

FLAN/MAIN - FLAN MAIN PROGRAM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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:

STD TYP - 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
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: MAKACT
PURPOSE: MAKE ACTION LIST ELEMENT
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: YTAB
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

VOID MAKACT(TYPE)
CHAR TYPE;

DESCRIPTION

MAKES AN ACTLST NODE, PUTS IN VALUES AND ADDS IT TO THE
LIST

ARGUMENTS:

TYPE = CHAR

INCLUDE FILES:

FLAN.Y" - **** PURPOSE NOT FOUND BY STRIPPER ****
STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
MATH - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

MYALLOC - MY MALLOC

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: MAKINC/INDENT
PURPOSE: INDENT OUTPUT LINE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKINC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

ARGUMENTS:

M = INT
T = INT

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

PUTC

CALLED DIRECTLY BY:

CSUB - C SUBROUTINE
COBSUB - COBOL SUBROUTINE
PLISUB - PL/I SUBROUTINE

USED IN MAIN PROGRAM(S):

MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS

FORMS LANGUAGE COMPILER Module Documentation

NAME: MAKINC/MAIN
PURPOSE: MAKE INCLUDE FILES FOR FORMS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKINC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

SYNOPSIS
MAIN()

DESCRIPTION
CREATES AN INCLUDE FILE IN THE CURRENT DIRECTORY FOR THE
GIVEN FORMS.

INCLUDE FILFS:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

CALLOC
PRINTF
GFPTR
CLSFRM
FCLOSE
CSUB - C SUBROUTINE
COBSUB - COBOL SUBROUTINE
PLISUB - PL/I SUBROUTINE
SPRINTF
FOPEN
STRUPC
OPNFRM
STRNCMP
FNDMSG
GETCHAR
FEOF
SCANF

FORMS LANGUAGE COMPILER 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:

STDYTP - 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

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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:

STDYTP - 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

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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:

STDTP - 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

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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:

STD TYP - 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
MAKACT - MAKE ACTION LIST ELEMENT
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: PLISUB
PURPOSE: PL/I SUBROUTINE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKINC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

PLISUB - PL/I SUBROUTINE
BLEN
PUTC
MAKINC/INDENT - INDENT OUTPUT LINE
FPRINTF

CALLED DIRECTLY BY:

PLISUB - PL/I SUBROUTINE
MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS

USED IN MAIN PROGRAM(S):

MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS

FORMS LANGUAGE COMPILER Module Documentation

NAME: REVFLAN/MAIN
PURPOSE: REVERSE FLAN
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: REVFLAN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FLAN
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

SYNOPSIS
FDREAD()

INPUTS/OUTPUTS:

INPUTS:

A FORMS FILE WITH THE .FD EXTENSION.

OUTPUTS:

A FORMATTED DUMP OF THE FILE.

DESCRIPTION

REVERSE COMPILES A .FD FILE TO PROVIDE THE SOURCE FOR A
FORM.

THE PROGRAM PROMPTS FOR THE OUTPUT FILE NAME THEN
REPEATEDLY

ASKS FOR FORMS FROM THE IISSFLIB DIRECTORY. A 'Z
TERMINATES

THE LIST.

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRINTF
FPRINTF
FCLOSE
GETC
PUTC
MALLOC

FREAD
SCANF
STRLEN
STRNCMP
SPRINTF
FOPEN

FORMS LANGUAGE COMPILER Module Documentation

NAME: WARNING
PURPOSE: ISSUE WARNING MESSAGE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: FLUIERR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

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:

STDYTP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

PMSGLS
STRLEN
CPRINTF

CALLED DIRECTLY BY:

CHKFRM - CHECK FORM
YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER 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):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTFRM
PURPOSE: WRITE FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: WRTFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

CHAR *WRTFRM(FP)
FIELD *FP;

INPUTS:

FP - POINTER TO FORM TO WRITE OUT

DESCRIPTION

WRITES THE SPECIFIED FORM INTO A .FD FILE.

ARGUMENTS:

OPNPTR = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFFV2 - FORM FILE FORMAT - VERSION 2

ROUTINES CALLED:

SPRINTF
FOPEN
SYSMSG
FWRITE
FCLOSE
WRTFRM/WRTTXT - WRITE TEXT
WRTFRM/WRTFLD - WRITE FIELD
WRTFRM/WRTTBF - WRITE TEXT BUFFER
WRTFRM/TBFCLOS - TEXT BUFFER CLOSE
WRTFRM/WRTDBF - WRITE DEFAULT BUFFER
WRTFRM/DBFCLOS - DEFAULT BUFFER CLOSE

STRASN
STRCPY
STRLEN

CALLED DIRECTLY BY:

FLAN/MAIN - FLAN MAIN PROGRAM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTFRM/DBFCLOS
PURPOSE: DEFAULT BUFFER CLOSE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: WRTFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

```
DBFCLOS (FPTR, I, LINE)
FILE *FPTR;
INT I;
CHAR LINE[81];
```

DESCRIPTION

WRITES THE LAST LINE OF THE DEFAULT LINE BUFFER.

ARGUMENTS:

FPTR = FILE *
I = INT
LINE = CHAR [81]

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFFV2 - FORM FILE FORMAT - VERSION 2

ROUTINES CALLED:

FWRITE

CALLED DIRECTLY BY:

WRTFRM - WRITE FORM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTFRM/FORMAT
PURPOSE: INSERT FORMAT CODES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: WRTFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

```
FORMAT(FLDREC, FMT1, FMT2)
      FLDREC *FLDREC;
      CHAR   FMT1, FMT2;
```

DESCRIPTION

INSERTS THE SPECIFIED FORMAT INTO THE SPECIFIED FIELD RECORD.

ARGUMENTS:

FLDREC = FLDREC *
FMT1 = CHAR
FMT2 = CHAR

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFFV2 - FORM FILE FORMAT - VERSION 2

CALLED DIRECTLY BY:

WRTFRM/WRTFLD - WRITE FIELD

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTFRM/TBFCLOS
PURPOSE: TEXT BUFFER CLOSE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: WRTFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

```
TBFCLOS (FPTR, I, LINE)
  FILE *FPTR;
  INT I;
  CHAR LINE[];
```

DESCRIPTION

WRITES THE LAST LINE OF THE TEXT LINE BUFFER.

ARGUMENTS:

```
FPTR = FILE *
I = INT
LINE = CHAR []
```

INCLUDE FILES:

```
STDTPY - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFV2 - FORM FILE FORMAT - VERSION 2
```

ROUTINES CALLED:

FWRITE

CALLED DIRECTLY BY:

WRTFRM - WRITE FORM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTRM/WRTRDBF
PURPOSE: WRITE DEFAULT BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: WRTRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

```
INT WRTRDBF(FPTR, FLDPTR, I, LINE)
FILE *FPTR;
FIELD *FLDPTR;
INT I;
CHAR LINE[81];
```

DESCRIPTION

COPIES THE SPECIFIED FIELD DEFAULT VALUE INTO THE DEFAULT
VALUE LINE
BUFFER STARTING AT THE SPECIFIED POSITION AND WRITING THE
LINE BUFFER
WHEN FULL. RETURNS THE NEXT POSITION TO USE.

ARGUMENTS:

```
FPTR = FILE *
FLDPTR = FIELD *
I = INT
LINE = CHAR [81]
```

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFFV2 - FORM FILE FORMAT - VERSION 2
```

ROUTINES CALLED:

FWRITE

CALLED DIRECTLY BY:

WRTRM - WRITE FORM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTRM/WRTRFLD
PURPOSE: WRITE FIELD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: WRTRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

WRTRFLD(FPTR, FLDPTR)
FILE *FPTR;
FIELD *FLDPTR;

DESCRIPTION

WRITES THE FIELD RECORD FOR THE SPECIFIED FIELD STRUCTURE.

ARGUMENTS:

FPTR = FILE *
FLDPTR = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFFV2 - FORM FILE FORMAT - VERSION 2

ROUTINES CALLED:

FWRITE
STRCPY
WRTRM/FORMAT - INSERT FORMAT CODES
STRNCPY
MEMCPY

CALLED DIRECTLY BY:

WRTRM - WRITE FORM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTRM/WRTTBF
PURPOSE: WRITE TEXT BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: WRTRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

```
INT WRTTBF(FPTR, TXTPTR, I, LINE)
FILE *FPTR;
TEXT *TXTPTR;
CHAR LINE[81];
INT I;
```

DESCRIPTION

COPIES THE SPECIFIED TEXT INTO THE TEXT LINE BUFFER
STARTING AT THE
SPECIFIED POSITION AND WRITING THE LINE BUFFER WHEN FULL.
RETURNS THE
NEXT POSITION TO USE.

ARGUMENTS:

```
FPTR = FILE *
TXTPTR = TEXT *
I = INT
LINE = CHAR [81]
```

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFV2 - FORM FILE FORMAT - VERSION 2
```

ROUTINES CALLED:

FWRITE

CALLED DIRECTLY BY:

WRTRM - WRITE FORM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTRM/WRRTXT
PURPOSE: WRITE TEXT
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: WRTRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

WRRTXT(FPTR, TXTPTR)
FILE *FPTR;
TEXT *TXTPTR;

DESCRIPTION

WRITES THE TEXT RECORD FOR THE SPECIFIED TEXT STRUCTURE.

ARGUMENTS:

FPTR = FILE *
TXTPTR = TEXT *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFFV2 - FORM FILE FORMAT - VERSION 2

ROUTINES CALLED:

FWRITE
STRLEN

CALLED DIRECTLY BY:

WRTRM - WRITE FORM

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: YYLEX
PURPOSE: LEXICAL ANALYZER FOR FLAN
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: YTAB
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS
INT LEX()

OUTPUTS:
SETS YYLVAL TO THE TOKEN VALUE (IF ANY)
RETURN THE TOKEN TYPE

DESCRIPTION
RECOGNIZES TOKENS (KEYWORDS, IDENTIFIERS, NUMBERS, ETC.),
SETS YYLVAL,
AND RETURNS THE APPROPRIATE TOKEN TYPE.

INCLUDE FILES:

FLAN.Y" - **** PURPOSE NOT FOUND BY STRIPPER ****
STDYYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
MATH - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

GETC
ERROR - ISSUE ERROR MESSAGE
ISALNUM
ISDIGIT
FATAL - ISSUE FATAL ERROR MESSAGE
UNGETC
WARNING - ISSUE WARNING MESSAGE
STRCMP
CSTASH - CHARACTER STASH
ATOF
ISALPHA

TOUPPER
ATOI
ISSPACE

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

FORMS LANGUAGE COMPILER Module Documentation

NAME: YYPARSE
PURPOSE: FLAN PARSER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: YTAB
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

DESCRIPTION
DEFINITION LANGUAGE GRAMMAR.

INCLUDE FILES:

FLAN.Y" - **** PURPOSE NOT FOUND BY STRIPPER ****
STDYYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
MATH - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRINTF
STRUPC
STRNCPY
FREE
STRCAT
MYALLOC - MY MALLOC
MEMCPY
MAKACT - MAKE ACTION LIST ELEMENT
MAKINT - MAKE EXPRESSION INTO AN INTEGER
STRCMP
STRLEN
WARNING - ISSUE WARNING MESSAGE
SPRINTF
MKPOS - MAKE POSITION NODE
FATAL - ISSUE FATAL ERROR MESSAGE
STRCPY
CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM
STRCHR
ERROR - ISSUE ERROR MESSAGE

MAKSTR - MAKE EXPRESSION INTO A STRING
CSTASH - CHARACTER STASH
GFLDPT - GET FIELD POINTER
MAKFLD
FNDATT - FIND ATTRIBUTE
YYERROR
YYLEX - LEXICAL ANALYZER FOR FLAN

CALLED DIRECTLY BY:

FLANCI - FLAN CALLABLE INTERFACE

USED IN MAIN PROGRAM(S):

FLAN/MAIN - FLAN MAIN PROGRAM

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 LANGUAGE COMPILER 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 LANGUAGE COMPILER Include File Description

FILE NAME: FLAN
PURPOSE: FLAN INTERNAL STRUCTURES
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
AUXILIARY DATA STRUCTURES USED BY FLAN.

FORMS LANGUAGE COMPILER Include File Description

FILE NAME: FPCODE
PURPOSE: FORM PROCESSOR RETURN CODES
LANGUAGE: C

DESCRIPTION:

FORMS LANGUAGE COMPILER Include File Description

FILE NAME: FPD
PURPOSE: FORM PROCESSOR DATA
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
DATA DEFINITIONS FOR ALL FORM PROCESSOR (INCLUDING
MONITER) DATA.

FORMS LANGUAGE COMPILER Include File Description

FILE NAME: FPDINI
PURPOSE: FPD INITIALIZATION
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INITIALIZED VERSION OF UID FOR INCLUSION IN MAIN PROGRAM.

FORMS LANGUAGE COMPILER 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 LANGUAGE COMPILER Include File Description

FILE NAME: NTM
PURPOSE: NTM INTERFACE INCLUDE FILE
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INCLUDE FILE FOR NTM INTERFACE

FORMS LANGUAGE COMPILER Include File Description

FILE NAME: RW
PURPOSE: REPORT WRITER DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION

FORMS LANGUAGE COMPILER 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 LANGUAGE COMPILER 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:

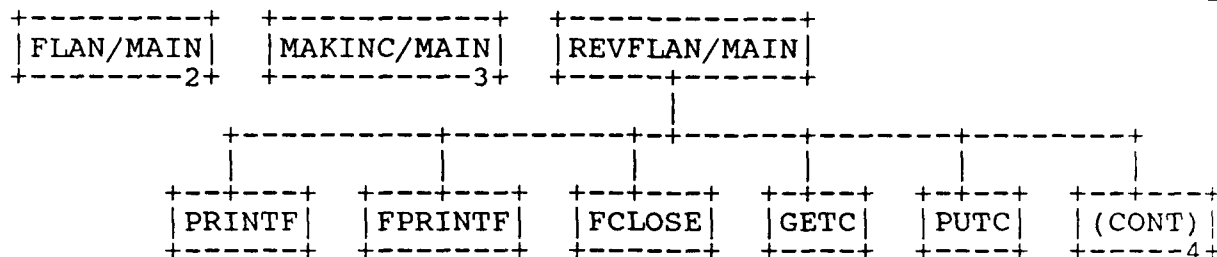
USHORT - COMPILER SUPPORTS UNSIGNED SHORT
TINY - COMPILER TREATS CHAR AS SIGNED
UTINY - CHAR IS SIGNED AND COMPILER SUPPORTS
UNSIGNED CHAR
VOID - COMPILER SUPPORTS VOID
FORTRAN - COMPILER SUPPORTS FORTRAN
STRASN - DEFINE APPROPRIATE MACRO
SUCCESS - DEFINE APPROPRIATE VALUE IF NOT 0
FAILURE - DEFINE APPROPRIATE VALUE IF NOT 1

3.10.10 Hierarchy Chart

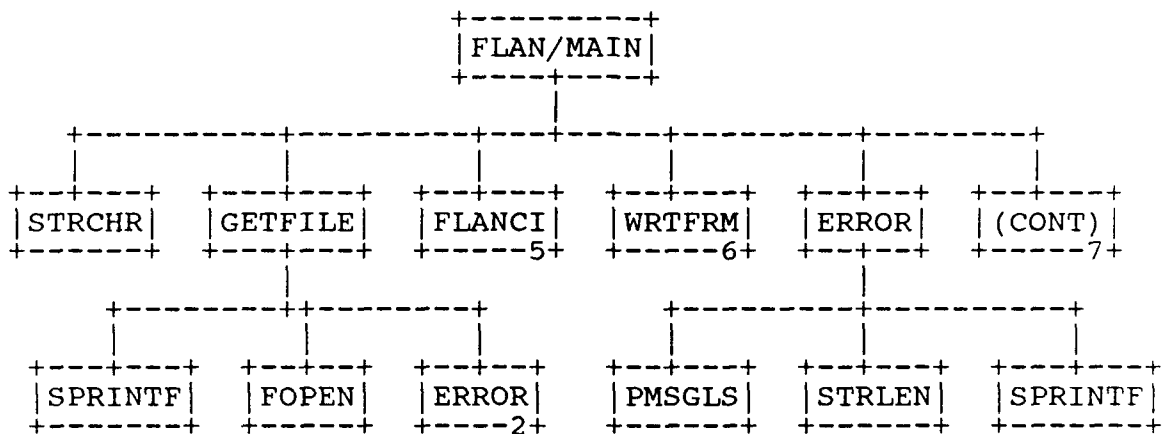
The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the 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.

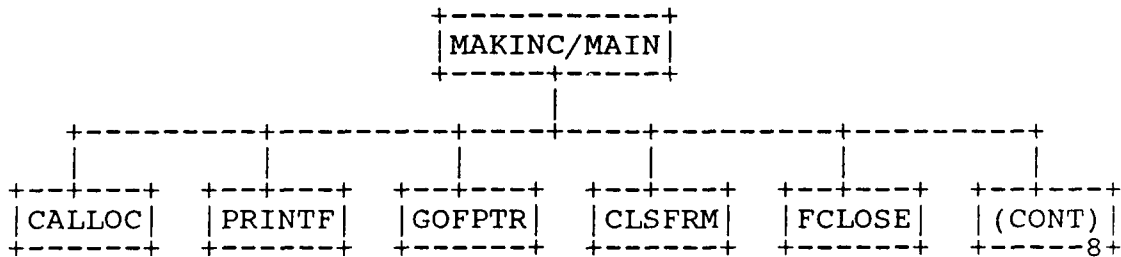
1



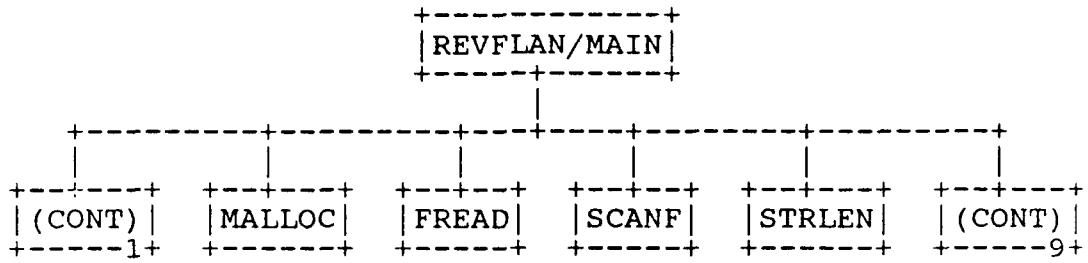
2



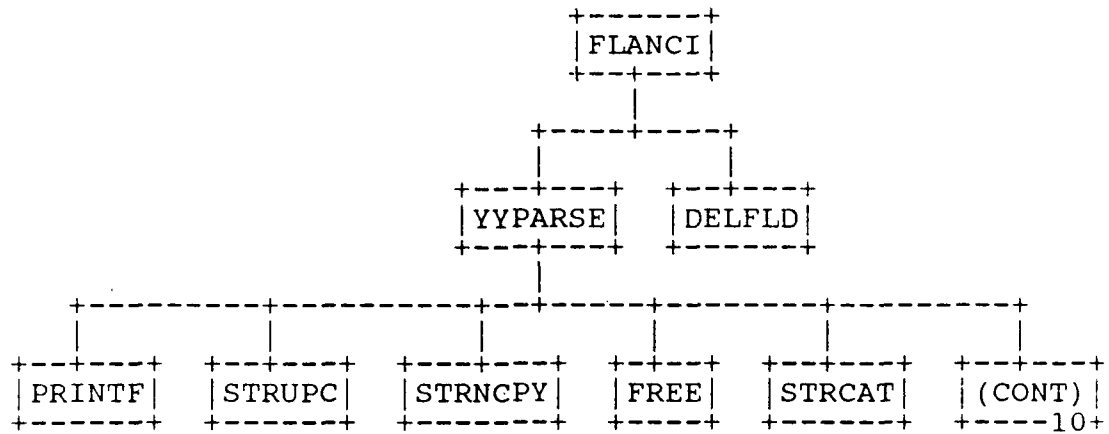
3



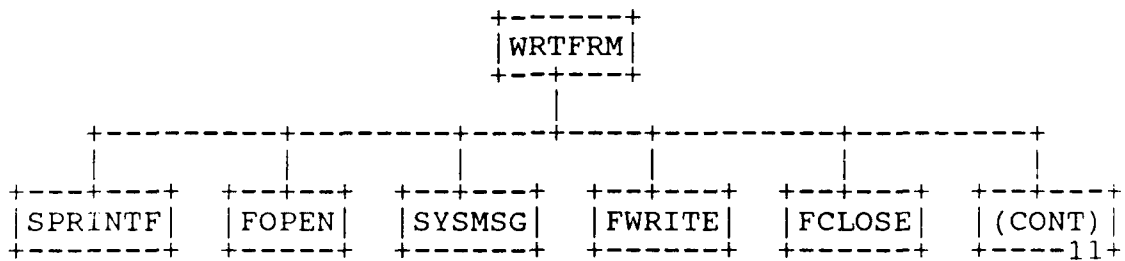
4



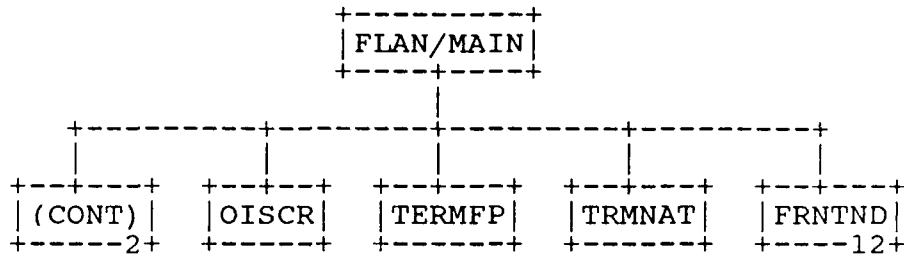
5



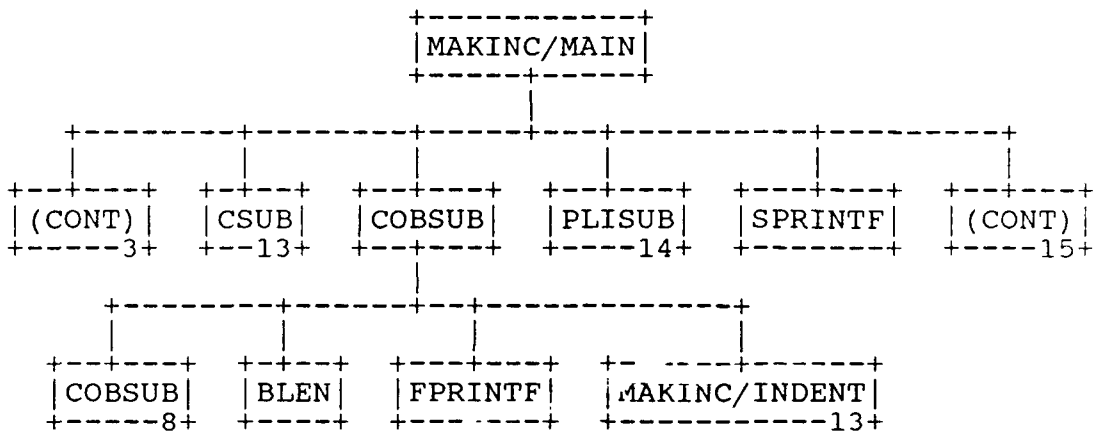
6



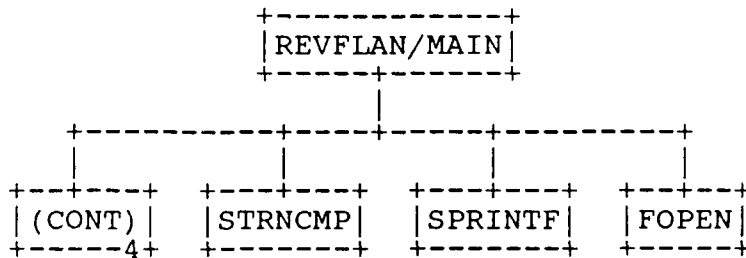
7



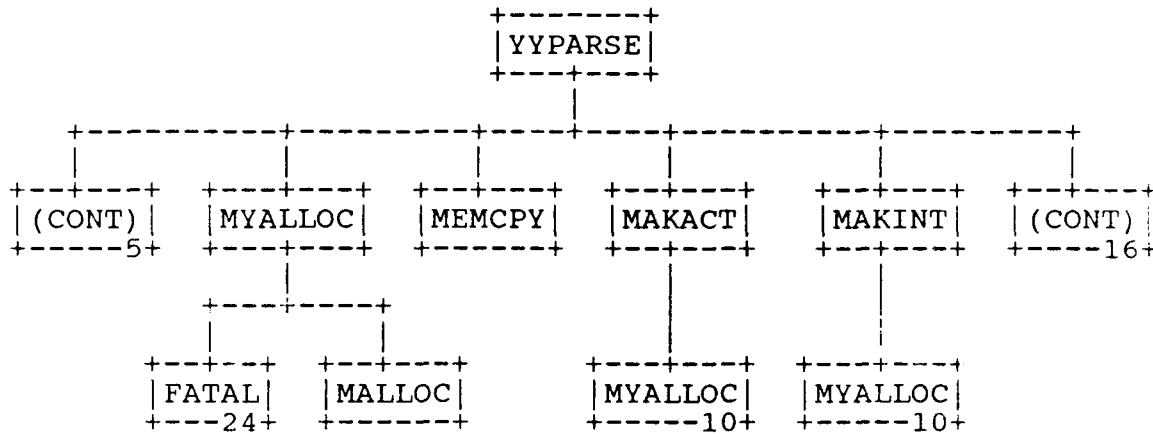
8



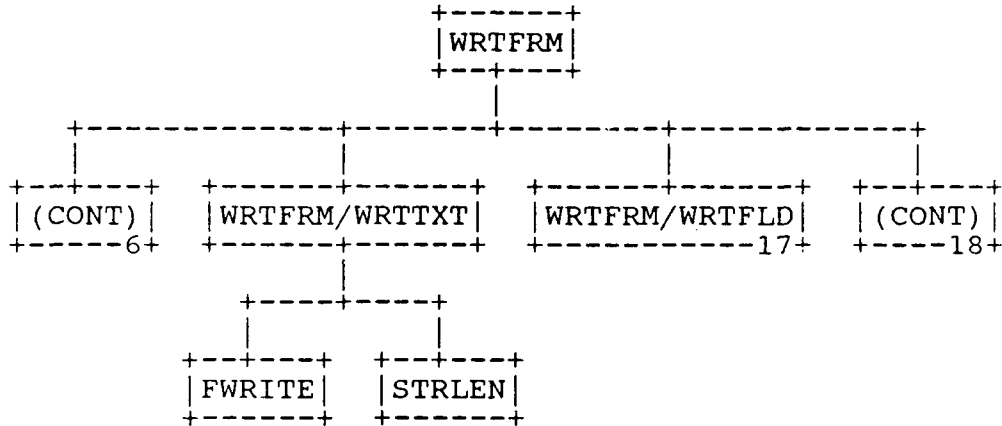
9



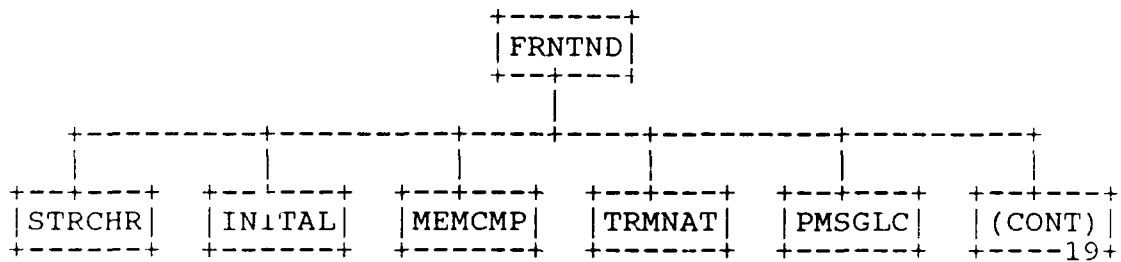
10



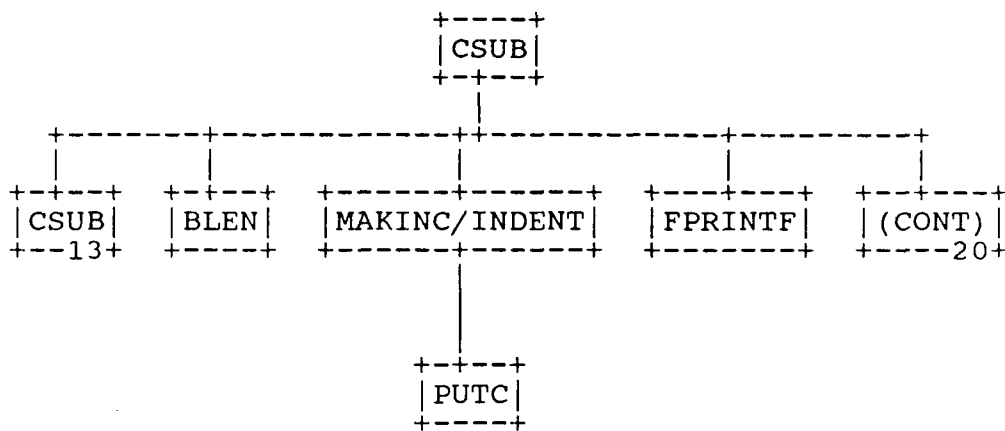
11



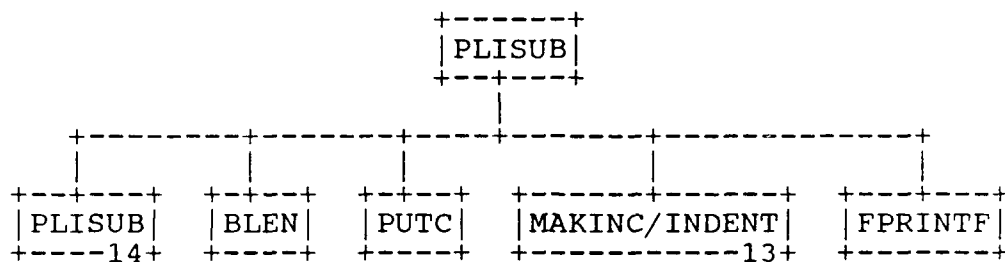
12



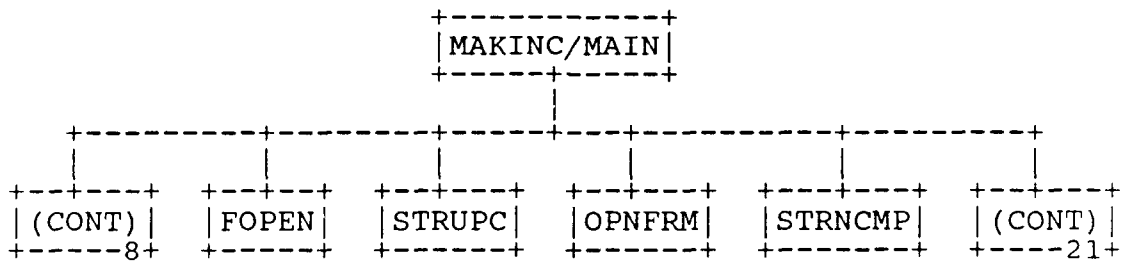
13



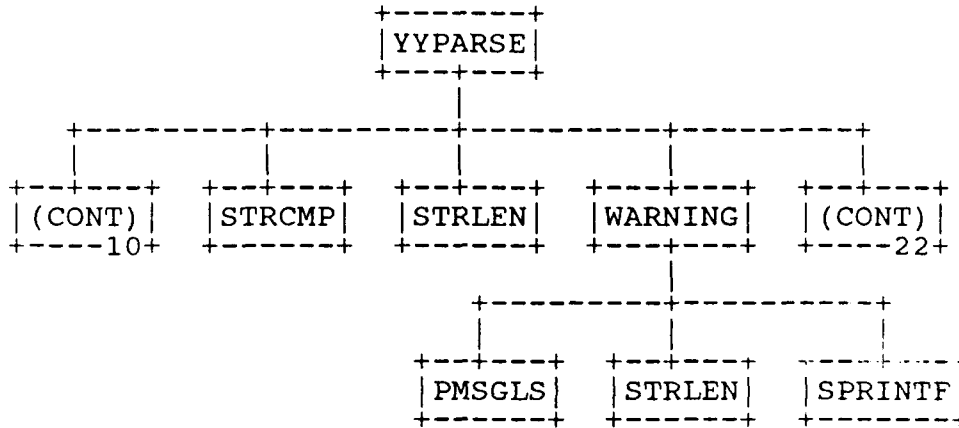
14



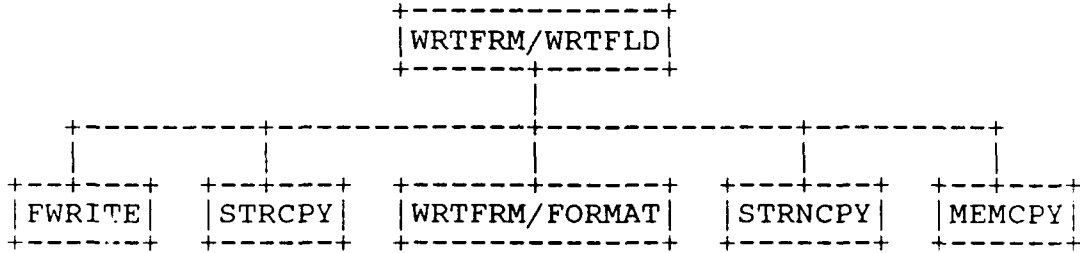
15



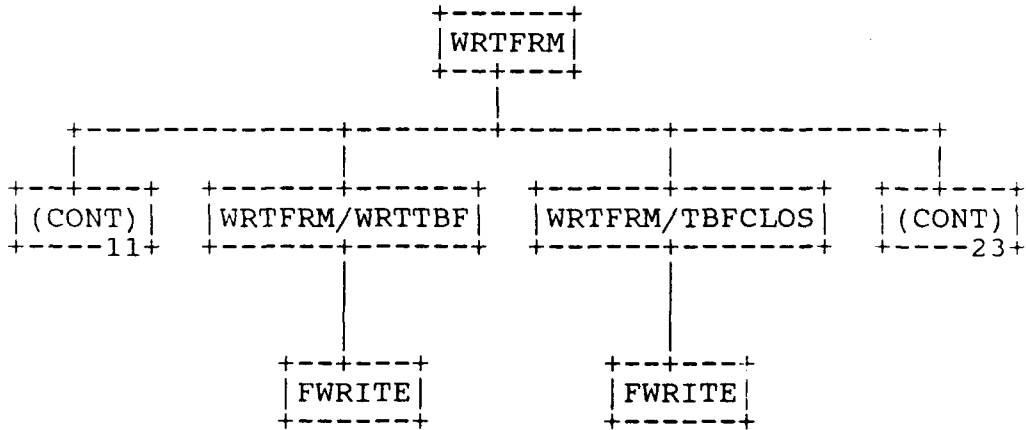
16



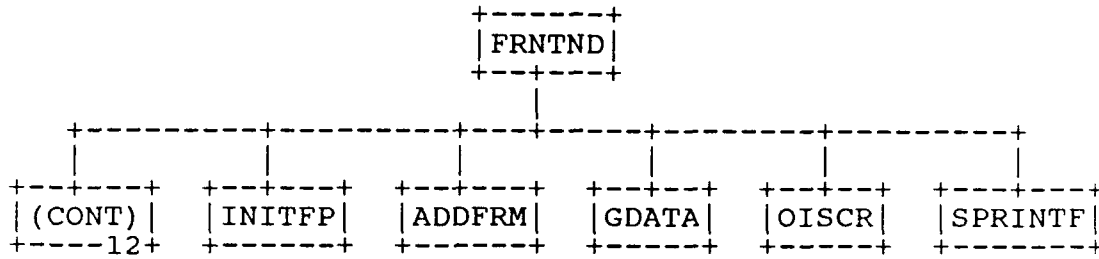
17



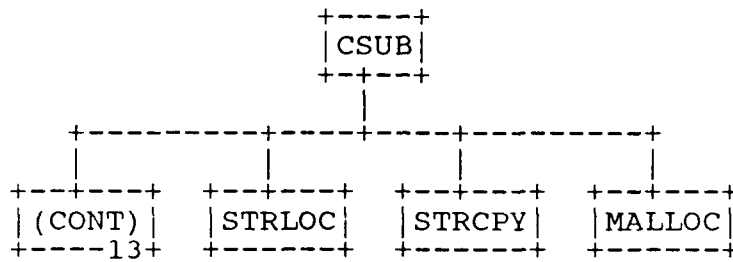
18



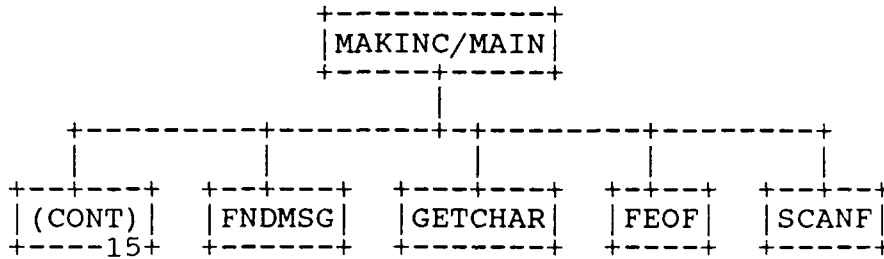
19



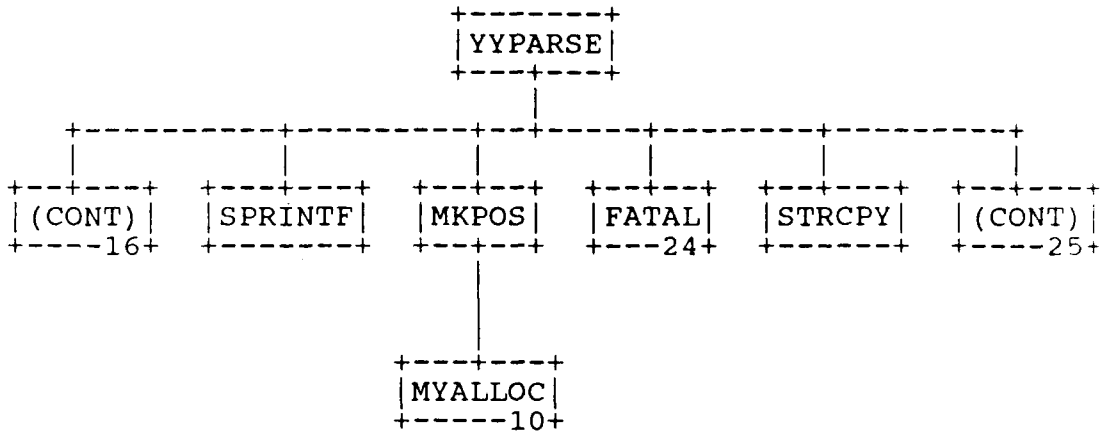
20



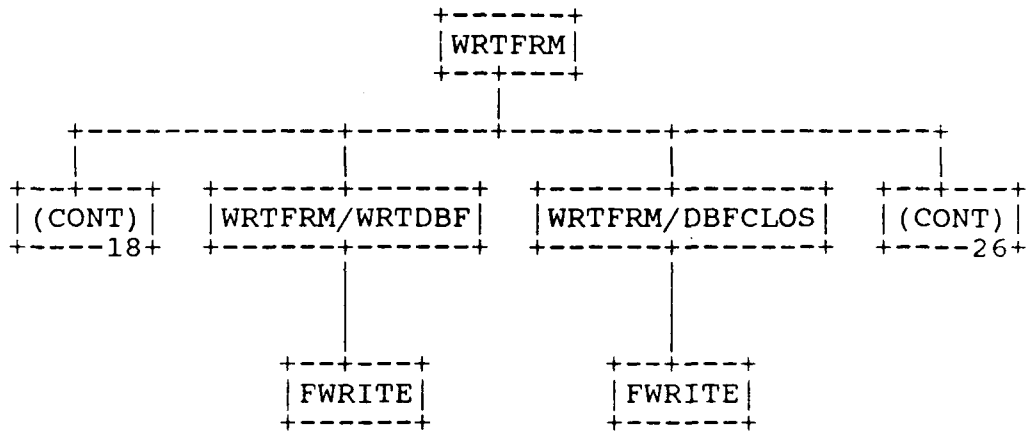
21



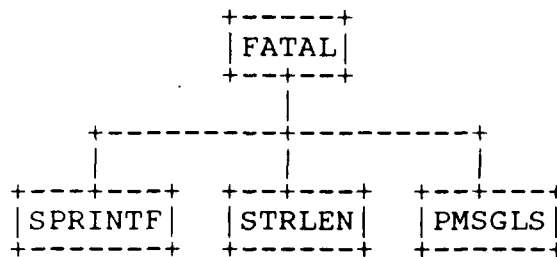
22



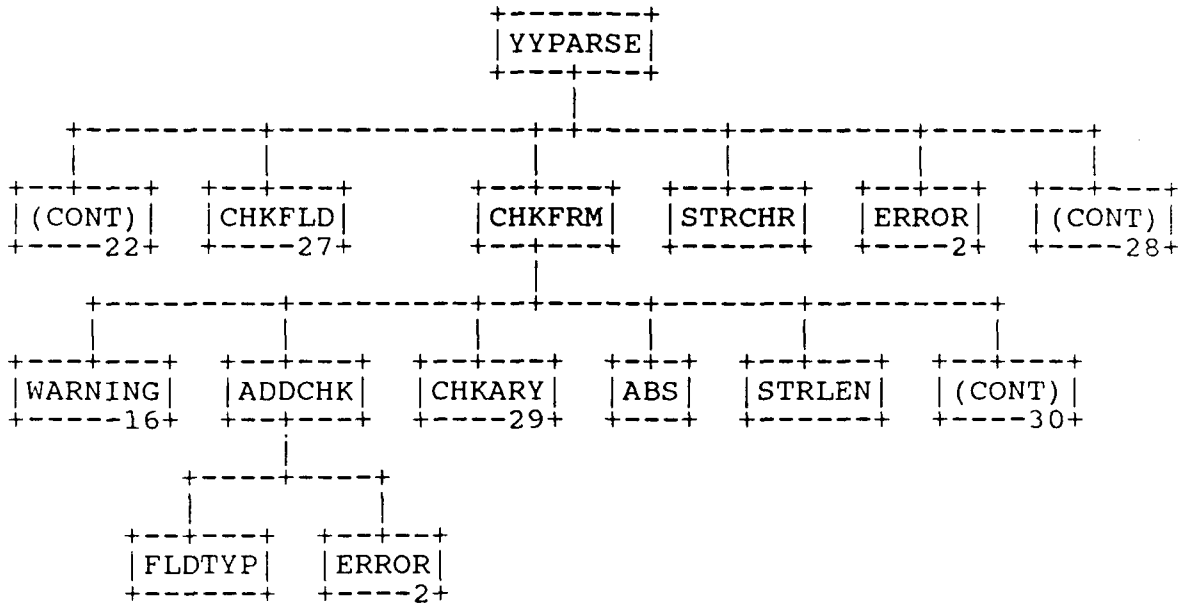
23



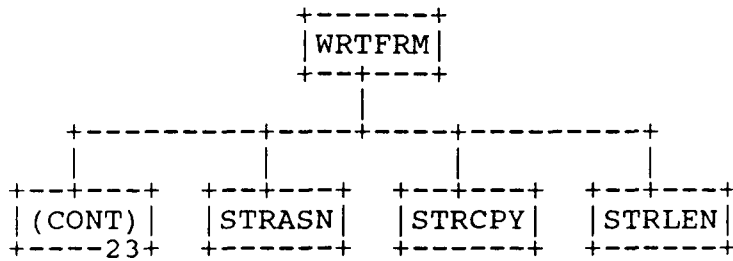
24



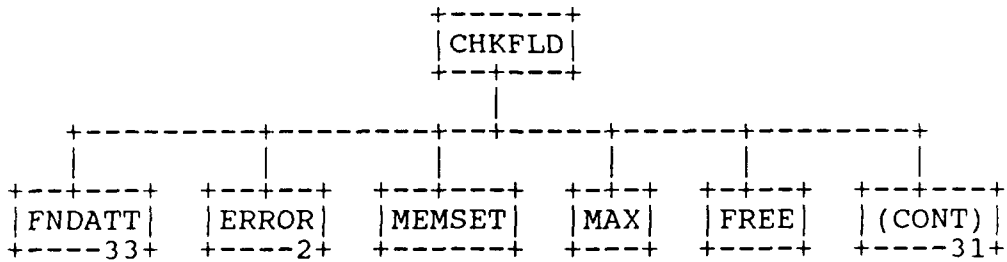
25



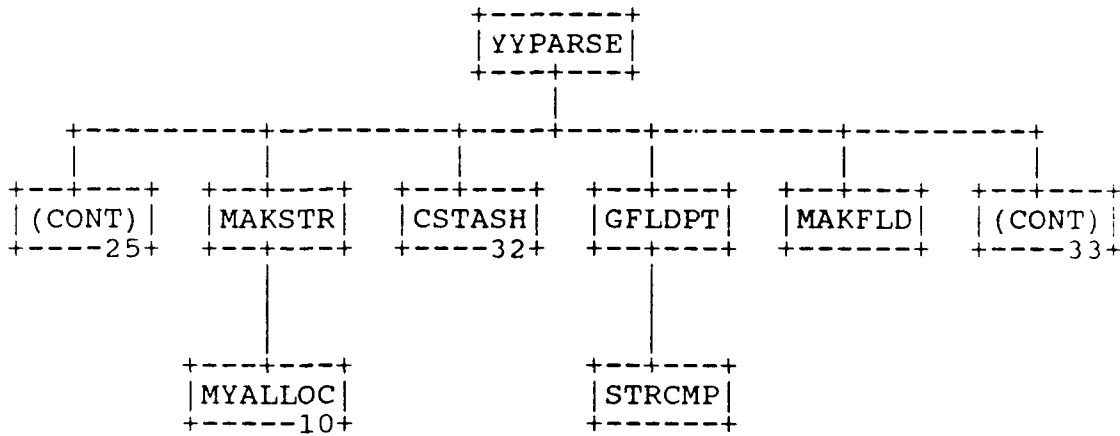
26



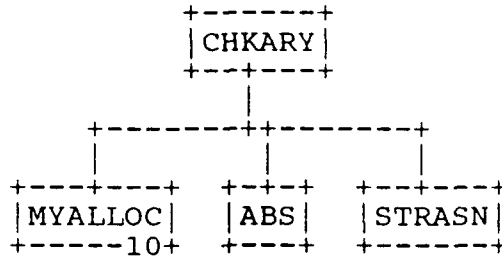
27



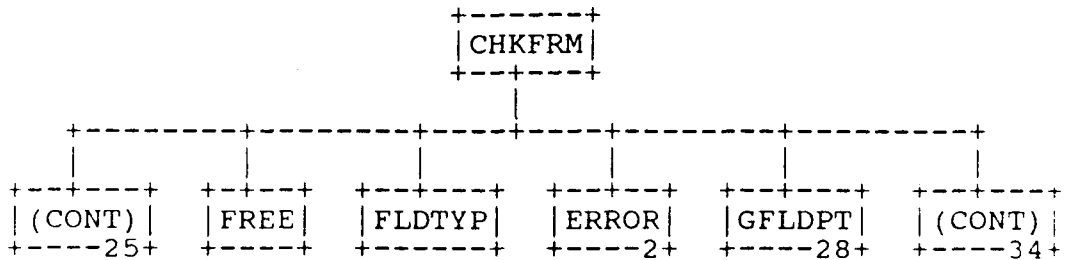
28



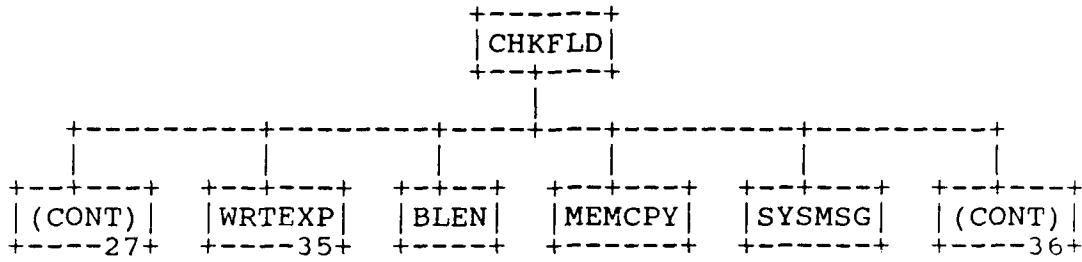
29



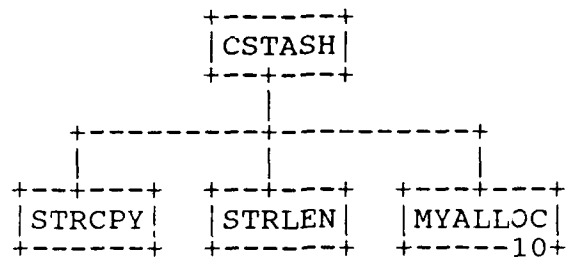
30



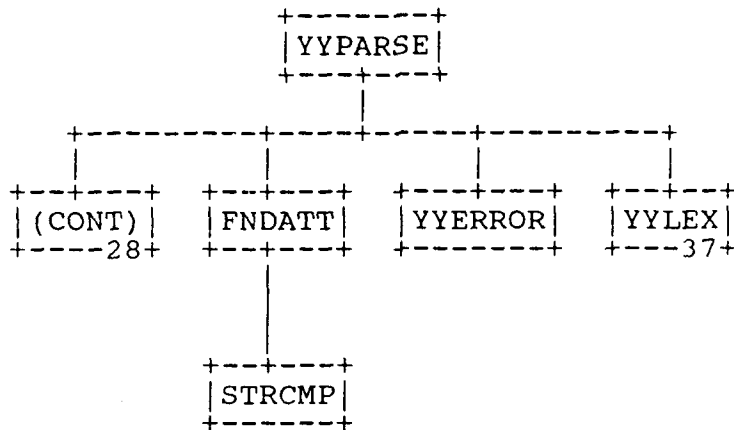
31



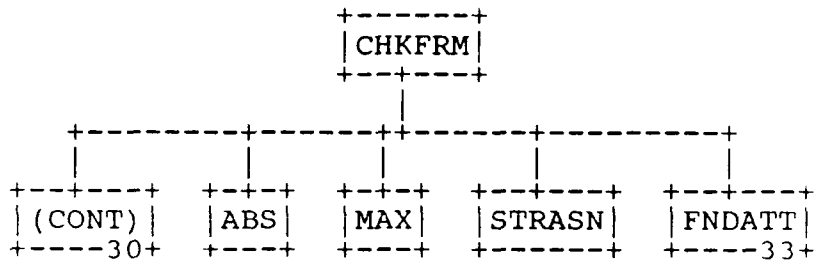
32



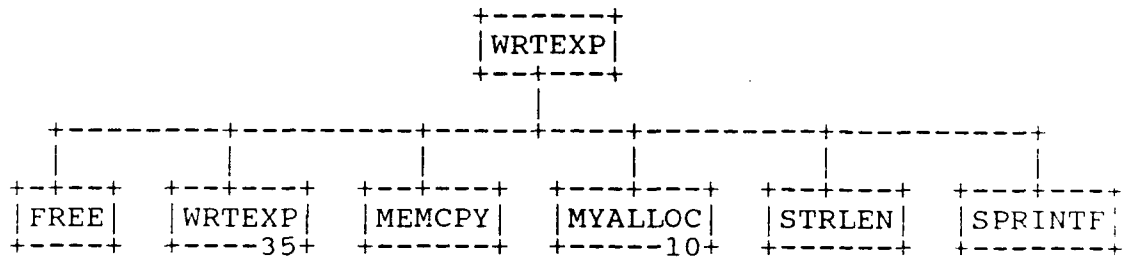
33



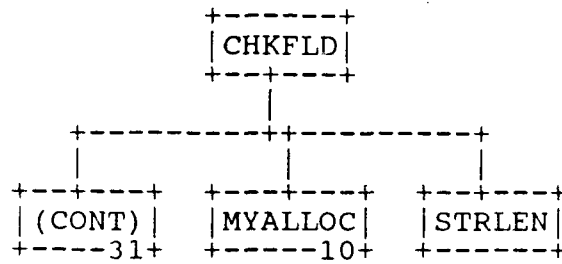
34



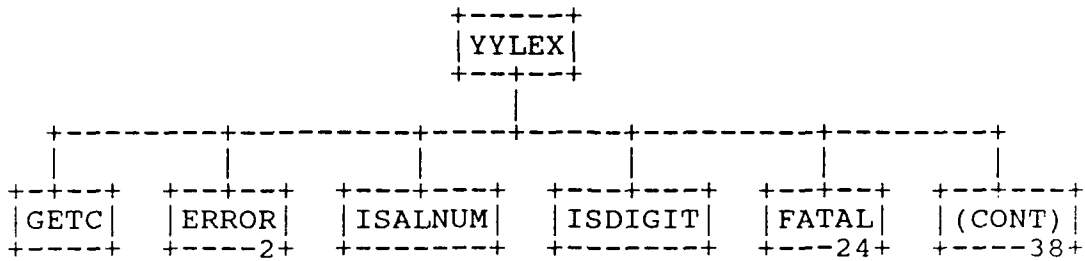
35



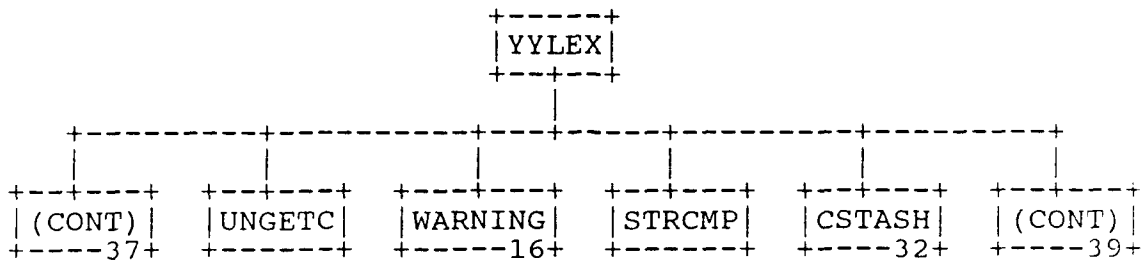
36



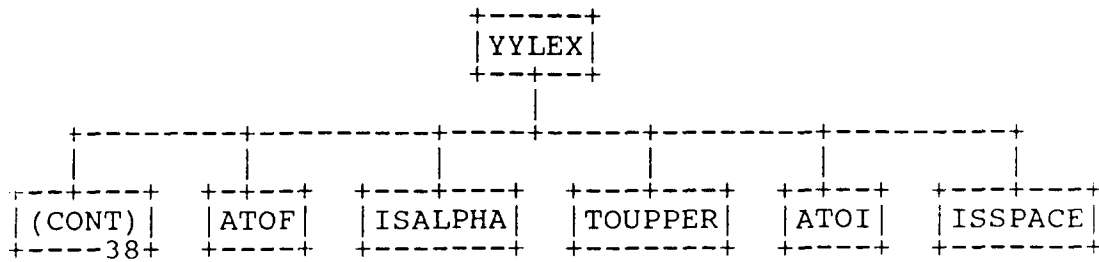
37



38



39



ABS		MAKSTR.....	28
ADDCHK.....	25	MALLOC	
ADDFRM		MAX	
ATOF		MEMCMP	
ATOI		MEMCPY	
BLFN		MEMSET	
CALLOC		MKPOS.....	22
CHKARY.....	29	MYALLOC.....	10
CHKFLD.....	27	OISCR	
CHKFRM.....	25	OPNFRM	
CLSFRM		PLISUB.....	14
COBSUB.....	8	PMSGLC	
CSTASH.....	32	PMSGLS	
CSUB.....	13	PRINTF	
DELFLD		PUTC	
ERROR.....	2	REVFLAN/MAIN.....	1
FATAL.....	24	SCANF	
FCLOSE		SPRINTF	
FEOF		STRASN	
FLAN/MAIN.....	2	STRCAT	
FLANCI.....	5	STRCHR	
FLDTYP		STRCMP	
FNDATT.....	33	STRCPY	
FNDMSG		STRLEN	
FOPEN		STRLOC	
FPRINTF		STRNCMP	
FREAD		STRNCPY	
FREE		STRUPC	
FRNTND.....	12	SYSMSG	
FWRITE		TERMF	
GDATA		TOUPPER	
GETC		TRMNAT	
GETCHAR		UNGETC	
GETFILE.....	2	WARNING.....	16
GFLDPT.....	28	WRTEXP.....	35
GOFPTR		WRTFRM.....	6
INITAL		WRTFRM/DBFCLOS...	23
INITFP		WRTFRM/FORMAT	
ISALNUM		WRTFRM/TBFCLOS...	18
ISALPHA		WRTFRM/WRTDBF...	23
ISDIGIT		WRTFRM/WRTFLD...	17
ISSPACE		WRTFRM/WRTTBF...	18
MAKACT.....	10	WRTFRM/WRTTXT...	11
MAKFLD		YYERROR	
MAKINC/INDENT...	13	YLEX.....	37
MAKINC/MAIN.....	3	YYPARSE.....	5
MAKINT.....	10		

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."