INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 15 - Forms Language Compiler Product Specification

General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345

November 1985

Approved for public release; distribution is unlimited.

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533
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, and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

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

David L. Judson  
PROJECT MANAGER  
AFWAL/MLTC  
WRIGHT PATTERSON AFB OH 45433

DATE  
5 Aug 1986

FOR THE COMMANDER:

Gerald C. Shumaker  
BRANCH CHIEF  
AFWAL/MLTC  
WRIGHT PATTERSON AFB OH 45433

DATE  
7 Aug 86

"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 AFWAL/MLTC, W-PAFB, OH 45433 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.
Title: Forms Language Compiler

This specification establishes the detailed design of a computer program identified as the Forms Language Compiler (FLAX). FLAX 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 for display and entry of data under the control of other application programs.

FLAX

General Electric Company
Production Resources Consulting
1 River Road
Schenectady, NY 12345

Wright-Patterson AFB, Ohio 45433

AFVAL-TR-86-4006 Vol VIII, Part 15
11. Title

Integrated Information Support System (IISS)
Vol VIII - User Interface Subsystem
Part 15 - Forms Language Compiler Product Specification

A S D 86 1487
17 Jul 1986
This product specification covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Allan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FT820100001, Project Overview.

The subcontractors and their contributing activities were as follows:

**TASK 4.2**

<table>
<thead>
<tr>
<th>Subcontractors</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boeing Military Aircraft Company (BMAC)</td>
<td>Reviewer</td>
</tr>
<tr>
<td>D. Appleton Company (DACOM)</td>
<td>Responsible for IDEF support, state-of-the-art literature search</td>
</tr>
<tr>
<td>General Dynamics/ Ft. Worth</td>
<td>Responsible for factory view function and information models</td>
</tr>
</tbody>
</table>
Subcontractors

Illinois Institute of Technology

Role
Responsible for factory view function research (IITRI) and information models of small and medium-size business

North American Rockwell

Reviewer

Northrop Corporation

Responsible for factory view function and information models

Pritsker and Associates

Responsible for IDEF2 support

SofTech

Responsible for IDEFO support

**TASKS 4.3 - 4.9 (TEST BED)**

Subcontractors

Boeing Military Aircraft Company (BMAC)

Role
Responsible for consultation on applications of the technology and on IBM computer technology.

Computer Technology Associates (CTA)

Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager.

Control Data Corporation (CDC)

Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM).

D. Appleton Company (DACOM)

Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems.
<table>
<thead>
<tr>
<th>Subcontractors</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Equipment Corporation (DEC)</td>
<td>Consulting and support of the performance testing and on DEC software and computer systems operation.</td>
</tr>
<tr>
<td>McDonnell Douglas Automation Company (McAuto)</td>
<td>Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.</td>
</tr>
<tr>
<td>On-Line Software International (OSI)</td>
<td>Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.</td>
</tr>
<tr>
<td>Rath and Strong Systems Products (RSSP) (In 1985 became McCormack &amp; Dodge)</td>
<td>Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.</td>
</tr>
<tr>
<td>SofTech, Inc.</td>
<td>Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.</td>
</tr>
<tr>
<td>Software Performance Engineering (SPE)</td>
<td>Responsible for directing the work on performance evaluation and analysis.</td>
</tr>
<tr>
<td>Structural Dynamics Research Corporation (SDRC)</td>
<td>Responsible for the User Interface and Virtual Terminal Interface Subsystems.</td>
</tr>
</tbody>
</table>

Subcontractors and other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

<table>
<thead>
<tr>
<th>Subcontractors</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Dynamics/ Ft. Worth</td>
<td>Responsible for factory view</td>
</tr>
<tr>
<td>Contractors</td>
<td>ICAM Project</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Boeing Military Aircraft Company</td>
<td>1701, 2201, 2202</td>
</tr>
<tr>
<td>(BMAC)</td>
<td></td>
</tr>
<tr>
<td>Control Data Corporation (CDC)</td>
<td>1502, 1701</td>
</tr>
<tr>
<td>D. Appleton Company (DACOM)</td>
<td>1502</td>
</tr>
<tr>
<td>General Electric</td>
<td>1502</td>
</tr>
<tr>
<td>Hughes Aircraft Company (HAC)</td>
<td>1701</td>
</tr>
<tr>
<td>Structural Dynamics Research Corp.</td>
<td>1502, 1701, 1703</td>
</tr>
<tr>
<td>(SDRC)</td>
<td>1703</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>DESCRIPTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>SCOPE</td>
<td>1-1</td>
</tr>
<tr>
<td>1.1</td>
<td>Identification</td>
<td>1-1</td>
</tr>
<tr>
<td>1.2</td>
<td>Functional Summary</td>
<td>1-1</td>
</tr>
<tr>
<td>2.0</td>
<td>DOCUMENTS</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1</td>
<td>Reference Documents</td>
<td>2-1</td>
</tr>
<tr>
<td>2.2</td>
<td>Terms and Abbreviations</td>
<td>2-2</td>
</tr>
<tr>
<td>3.0</td>
<td>REQUIREMENTS</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>Structural Description</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Stand Alone FLAN</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Integrated FLAN</td>
<td>3-2</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Reverse FLAN</td>
<td>3-2</td>
</tr>
<tr>
<td>3.1.4</td>
<td>MAKINC</td>
<td>3-2</td>
</tr>
<tr>
<td>3.2</td>
<td>Functional Flow</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Stand Alone FLAN</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Integrated FLAN</td>
<td>3-4</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Reverse FLAN</td>
<td>3-5</td>
</tr>
<tr>
<td>3.2.4</td>
<td>MAKINC</td>
<td>3-5</td>
</tr>
<tr>
<td>3.3</td>
<td>Interfaces</td>
<td>3-6</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Form Processor</td>
<td>3-6</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Forms Driven Form Editor</td>
<td>3-6</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Report Writer</td>
<td>3-6</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Rapid Application Generator</td>
<td>3-6</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Application Interface</td>
<td>3-6</td>
</tr>
<tr>
<td>3.4</td>
<td>Program Interrupts</td>
<td>3-7</td>
</tr>
<tr>
<td>3.5</td>
<td>Timing and Sequencing Description</td>
<td>3-7</td>
</tr>
<tr>
<td>3.6</td>
<td>Special Control Features</td>
<td>3-7</td>
</tr>
<tr>
<td>3.7</td>
<td>Storage Allocation</td>
<td>3-7</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Data Base Definition</td>
<td>3-7</td>
</tr>
<tr>
<td>3.7.1.1</td>
<td>File Descriptions</td>
<td>3-7</td>
</tr>
<tr>
<td>3.8</td>
<td>Object Code Creation</td>
<td>3-10</td>
</tr>
<tr>
<td>3.9</td>
<td>Adaptation Data</td>
<td>3-10</td>
</tr>
<tr>
<td>3.10</td>
<td>Detailed Design Description</td>
<td>3-10</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Main Program List</td>
<td>3-10</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Module List</td>
<td>3-12</td>
</tr>
<tr>
<td>3.10.3</td>
<td>External Routines List</td>
<td>3-15</td>
</tr>
<tr>
<td>3.10.4</td>
<td>Include File List</td>
<td>3-18</td>
</tr>
<tr>
<td>3.10.5</td>
<td>Where Include File Used List</td>
<td>3-20</td>
</tr>
<tr>
<td>3.10.6</td>
<td>Where External Routine Used List</td>
<td>3-28</td>
</tr>
<tr>
<td>3.10.7</td>
<td>Main Program Parts List</td>
<td>3-38</td>
</tr>
</tbody>
</table>
3.10.8 Module Documentation ....................... 3-43
3.10.9 Include File Description .................... 3-119
3.10.10 Hierarchy Chart ........................... 3-131
3.11 Program Listings Comments ................... 3-172

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

FIGURES
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
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


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:

- Stand alone FLAN
- Integrated FLAN
- Reverse FLAN (REVFLAN)
- 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.

```
+----------+     +----------+
| FLAN.Y   | ----->| YACC ----->| 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.
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 FLFRMT.C and FLUIERR.C do the interfacing with the IISS user. FLFRMT.C processes the form and FLUIERR.C performs error handling. Figure 3-3 describes the major module relationships for the integrated FLAN.

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.
3.2.2 Integrated FLAN

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

![Data Flow Diagram of Integrated FLAN](image)
3.2.3 Reverse FLAN

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

![Reverse FLAN Data Flow Diagram]

3.2.4 MAXINC

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

![MAXINC Data Flow Diagram]
3.3 Interfaces

3.3.1 Form Processor

The Form Definition files generated by FLAN are used as input to the Form Processor. The format of these files is described in section 3.7.1.1.

3.3.2 Forms Driven Form Editor

The Forms Driven Form Editor (FDFE) generates Form Definition Language (fdl) files and compiles them using code that it shares with FLAN. The FLAN modules YTAB.C, FLANSP.C, WRTFRM.C, and FLUIERR.C are used by the FDFE.

3.3.3 Report Writer

The Report Writer (RW) uses FLAN to parse its input language file and to generate its fd files. The FLAN modules YTAB.C, FLANSP.C, WRTFRM.C, and FLUIERR.C are used by the RW.

3.3.4 Rapid Application Generator

The Rapid Application Generator uses FLAN to parse its input language file and to generate its fd files. The FLAN modules YTAB.C, FLANSP.C, WRTFRM.C, and FLUIERR.C are used by the Rapid Application Generator.

3.3.5 Application Interface

Integrated FLAN is an application that uses forms to communicate with the terminal user. Integrated FLAN accomplishes this communication by calling Application Interface routines. Figure 3-8 describes this interface.

```
+-----------------------+
| Integrated            |
| FLAN | AI <-----messages to and from the Form Processor
|      |       |     via the Network Transaction Manager
+-----------------------+
```

Figure 3-8 Integrated FLAN Application Interface
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:

<table>
<thead>
<tr>
<th>Language</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand Alone FLAN</td>
<td>138</td>
</tr>
<tr>
<td>Integrated FLAN</td>
<td>186</td>
</tr>
<tr>
<td>Reverse FLAN</td>
<td>47</td>
</tr>
<tr>
<td>MAKINC</td>
<td>102</td>
</tr>
</tbody>
</table>

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:**

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

3. **FILE NAME:** formname.FD - Form Definition file. 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:**

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

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

typedef struct /* field record */
{
    char fld_name[10];    /* field name */
    char fld_type;        /* field type (F, I, W, A) */
    short row;            /* starting row */
    short col;            /* starting col */
    short width;          /* field width */
    short depth;          /* field depth */
    int min_value;        /* minimum value (if any) */
    int max_value;        /* maximum value (if any) */
    char helpline[80];    /* help text */
    char disp_att[10];    /* display attribute */
    short n_formats;      /* number of formats */
    char format[12][2];   /* format strings */
    short n_arydefs;      /* number of dimensions */
    struct /* dimension specification */
    {
        char dir;    /* repeat direction (H, V) */
        short cnt;   /* actual repeat count */
        short sp;    /* number of spaces between repetitions */
        short dsp_size; /* display repeat count */
    } array_def[3];
    char linefeed;
} FLDREC;
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.
<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAN/MAIN</td>
<td>FLAN MAIN PROGRAM</td>
</tr>
<tr>
<td>MAKINC/MAIN</td>
<td>MAKE INCLUDE FILES FOR FORMS</td>
</tr>
<tr>
<td>REVFLAN/MAIN</td>
<td>REVERSE FLAN</td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDCHK</td>
<td>ADD POSITION TO CHECK LIST</td>
</tr>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td>COBSUB</td>
<td>COBOL SUBROUTINE</td>
</tr>
<tr>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td>CSUB</td>
<td>C SUBROUTINE</td>
</tr>
<tr>
<td>ERROR</td>
<td>ISSUE ERROR MESSAGE</td>
</tr>
<tr>
<td>FATAL</td>
<td>ISSUE FATAL ERROR MESSAGE</td>
</tr>
<tr>
<td>FLAN/Main</td>
<td>FLAN MAIN PROGRAM</td>
</tr>
<tr>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
</tr>
<tr>
<td>FLDTYP</td>
<td>FIELD TYPE</td>
</tr>
<tr>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
</tr>
<tr>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
</tr>
<tr>
<td>GETFILE</td>
<td>GET INPUT FILENAME</td>
</tr>
<tr>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
</tr>
<tr>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
</tr>
<tr>
<td>MAKING/INDENT</td>
<td>INDENT OUTPUT LINE</td>
</tr>
<tr>
<td>MAKING/MAIN</td>
<td>MAKE INCLUDE FILES FOR FORMS</td>
</tr>
<tr>
<td>MAKINT</td>
<td>MAKE EXPRESSION INTO AN INTEGER</td>
</tr>
<tr>
<td>MAKSTR</td>
<td>MAKE EXPRESSION INTO A STRING</td>
</tr>
</tbody>
</table>
**FORMS LANGUAGE COMPILER Module List**

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKPOS</td>
<td>MAKE POSITION NODE</td>
</tr>
<tr>
<td>MYALLOC</td>
<td>MY MALLOC</td>
</tr>
<tr>
<td>PLISUB</td>
<td>PL/I SUBROUTINE</td>
</tr>
<tr>
<td>REVFLAN/MAIN</td>
<td>REVERSE FLAN</td>
</tr>
<tr>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
</tr>
<tr>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
<tr>
<td>WRTFRM/DBFCLOS</td>
<td>DEFAULT BUFFER CLOSE</td>
</tr>
<tr>
<td>WRTFRM/FORMAT</td>
<td>INSERT FORMAT CODES</td>
</tr>
<tr>
<td>WRTFRM/TBFCLOS</td>
<td>TEXT BUFFER CLOSE</td>
</tr>
<tr>
<td>WRTFRM/WRTDBF</td>
<td>WRITE DEFAULT BUFFER</td>
</tr>
<tr>
<td>WRTFRM/WRTFLD</td>
<td>WRITE FIELD</td>
</tr>
<tr>
<td>WRTFRM/WRTTBF</td>
<td>WRITE TEXT BUFFER</td>
</tr>
<tr>
<td>WRTFRM/WRTTXT</td>
<td>WRITE TEXT</td>
</tr>
<tr>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>
3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. The specification in which any module is documented may be found in the Module Documentation Index (Document Number CM 620100001). See section 3.10.6 for a list of the modules that call each of these external routines.
<table>
<thead>
<tr>
<th>Module Name</th>
<th>First User</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>CHKARY</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>FRMTND</td>
</tr>
<tr>
<td>ATOF</td>
<td>YYLEX</td>
</tr>
<tr>
<td>ATOI</td>
<td>YYLEX</td>
</tr>
<tr>
<td>BLEN</td>
<td>CSUB</td>
</tr>
<tr>
<td>CALLOC</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>CLSFRM</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>DELFLD</td>
<td>FLANDCI</td>
</tr>
<tr>
<td>FCLOSE</td>
<td>WRTFRM</td>
</tr>
<tr>
<td>FEOF</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>FNDMSG</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>FOPEN</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>FPRINTF</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>FREE</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>FWRITE</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>GDATA</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>GETC</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>GETCHAR</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>GOFPTR</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>INITIAL</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>INITFP</td>
<td>MAKINC/MAIN</td>
</tr>
<tr>
<td>ISALNUM</td>
<td>YYLEX</td>
</tr>
<tr>
<td>ISALPHA</td>
<td>YYLEX</td>
</tr>
<tr>
<td>ISDIGIT</td>
<td>YYLEX</td>
</tr>
<tr>
<td>ISSPACE</td>
<td>YYLEX</td>
</tr>
<tr>
<td>MAKFLD</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>MALLOC</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>MAX</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>MEMCMP</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>MEMCPY</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>MEMSET</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>OISCR</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>OPMFRM</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>PMSGLC</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>PMSGLS</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>PRINTF</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>PUTC</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>SCANF</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>SPRINTF</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>STRASN</td>
<td>YYPARSE</td>
</tr>
</tbody>
</table>

3-16
## FORMS LANGUAGE COMPILER External Routines List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>First User</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRCAT</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>STRCHR</td>
<td>FLAN/MAIN</td>
</tr>
<tr>
<td>STRCMP</td>
<td>GFLDPT</td>
</tr>
<tr>
<td>STRCPY</td>
<td>CSUB</td>
</tr>
<tr>
<td>STRLEN</td>
<td>WRTFRM/WRTTXT</td>
</tr>
<tr>
<td>STRLOC</td>
<td>CSUB</td>
</tr>
<tr>
<td>STRNCMP</td>
<td>REVFLAN/MAIN</td>
</tr>
<tr>
<td>STRNCPY</td>
<td>WRTFRM/WRTFLD</td>
</tr>
<tr>
<td>STRUPC</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>SYSMSG</td>
<td>CHKFLD</td>
</tr>
<tr>
<td>TERMFP</td>
<td>FLAN/MAIN</td>
</tr>
<tr>
<td>TOUPPER</td>
<td>YYLEX</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>FLAN/MAIN</td>
</tr>
<tr>
<td>UNGETC</td>
<td>YYLEX</td>
</tr>
<tr>
<td>YYERROR</td>
<td>YYPARSE</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>File Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTYP</td>
<td>**** PURPOSE NOT FOUND BY STRIPPER ****</td>
</tr>
<tr>
<td>FFFV2</td>
<td>FORM FILE FORMAT - VERSION 2</td>
</tr>
<tr>
<td>FLAN.Y</td>
<td>**** PURPOSE NOT FOUND BY STRIPPER ****</td>
</tr>
<tr>
<td>FPCODE</td>
<td>FORM PROCESSOR RETURN CODES</td>
</tr>
<tr>
<td>FPD</td>
<td>FORM PROCESSOR DATA</td>
</tr>
<tr>
<td>FPDINI</td>
<td>FPD INITIALIZATION</td>
</tr>
<tr>
<td>FPPARM</td>
<td>FORM PROCESSOR PARAMETERS</td>
</tr>
<tr>
<td>MATH</td>
<td>**** PURPOSE NOT FOUND BY STRIPPER ****</td>
</tr>
<tr>
<td>NTM</td>
<td>NTM INTERFACE INCLUDE FILE</td>
</tr>
<tr>
<td>RW</td>
<td>REPORT WRITER DEFINITIONS</td>
</tr>
<tr>
<td>STDIO</td>
<td>**** PURPOSE NOT FOUND BY STRIPPER ****</td>
</tr>
<tr>
<td>STDYP</td>
<td>STANDARD TYPE DEFINITIONS</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CTYPE**

<table>
<thead>
<tr>
<th></th>
<th>Make Action List Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKACT</td>
<td></td>
</tr>
<tr>
<td>YYLEX</td>
<td>Lexical Analyzer For FLAN</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN Parser</td>
</tr>
</tbody>
</table>

**FFFV2**

<table>
<thead>
<tr>
<th></th>
<th>Write Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRTFRM</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/DB</td>
<td>Default Buffer Close</td>
</tr>
<tr>
<td>WRTFRM/FO</td>
<td>Insert Format Codes</td>
</tr>
<tr>
<td>WRTFRM/TB</td>
<td>Text Buffer Close</td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>Write Default Buffer</td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>Write Field</td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>Write Text Buffer</td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>Write Text</td>
</tr>
</tbody>
</table>

**FLAN.Y**

<table>
<thead>
<tr>
<th></th>
<th>Make Action List Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKACT</td>
<td></td>
</tr>
<tr>
<td>YYLEX</td>
<td>Lexical Analyzer For FLAN</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN Parser</td>
</tr>
</tbody>
</table>

**FPCODE**

<table>
<thead>
<tr>
<th></th>
<th>Add Position To Check List</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDCHK</td>
<td></td>
</tr>
<tr>
<td>CHKARY</td>
<td>Check Array</td>
</tr>
<tr>
<td>CHKFLD</td>
<td>Check Field</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>Check Form</td>
</tr>
<tr>
<td>COBSUB</td>
<td>COBOL Subroutine</td>
</tr>
<tr>
<td>CSTASH</td>
<td>Character Stash</td>
</tr>
<tr>
<td>CSUB</td>
<td>C Subroutine</td>
</tr>
<tr>
<td>FLANCI</td>
<td>FLAN Callable Interface</td>
</tr>
<tr>
<td>FLDTYPE</td>
<td>Field Type</td>
</tr>
<tr>
<td>FNDATT</td>
<td>Find Attribute</td>
</tr>
</tbody>
</table>

3-21
**FORMS LANGUAGE COMPILER** Where-include-file-used List

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

**FPD**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDCHK</td>
<td>ADD POSITION TO CHECK LIST</td>
</tr>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td>COBSUB</td>
<td>COBOL SUBROUTINE</td>
</tr>
<tr>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td>CSUB</td>
<td>C SUBROUTINE</td>
</tr>
<tr>
<td>FLAN/MAIN</td>
<td>FLAN MAIN PROGRAM</td>
</tr>
<tr>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
</tr>
<tr>
<td>FLDTYP</td>
<td>FIELD TYPE</td>
</tr>
<tr>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
</tr>
<tr>
<td>GETFILE</td>
<td>GET INPUT FILENAME</td>
</tr>
<tr>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
</tr>
<tr>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
</tr>
<tr>
<td>MAKINC/IN</td>
<td>INDENT OUTPUT LINE</td>
</tr>
<tr>
<td>MAKINC/MA</td>
<td>MAKE INCLUDE FILES FOR FORMS</td>
</tr>
<tr>
<td>MAKINT</td>
<td>MAKE EXPRESSION INTO AN INTEGER</td>
</tr>
<tr>
<td>MAKSTR</td>
<td>MAKE EXPRESSION INTO A STRING</td>
</tr>
</tbody>
</table>
## FORMS LANGUAGE COMPILER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MKPOS</td>
<td>MAKE POSITION NODE</td>
</tr>
<tr>
<td></td>
<td>MYALLOC</td>
<td>MY MALLOC</td>
</tr>
<tr>
<td></td>
<td>PLISUB</td>
<td>PL/I SUBROUTINE</td>
</tr>
<tr>
<td></td>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td></td>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/DB</td>
<td>DEFAULT BUFFER CLOSE</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/FO</td>
<td>INSERT FORMAT CODES</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/TB</td>
<td>TEXT BUFFER CLOSE</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WR</td>
<td>WRITE DEFAULT BUFFER</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WR</td>
<td>WRITE FIELD</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WR</td>
<td>WRITE TEXT BUFFER</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WR</td>
<td>WRITE TEXT</td>
</tr>
<tr>
<td></td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>

**FPDINI**

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBSUB</td>
<td>COBOL SUBROUTINE</td>
</tr>
<tr>
<td>CSUB</td>
<td>C SUBROUTINE</td>
</tr>
<tr>
<td>FLAN/MAIN</td>
<td>FLAN MAIN PROGRAM</td>
</tr>
<tr>
<td>GETFILE</td>
<td>GET INPUT FILENAME</td>
</tr>
<tr>
<td>MAKINC/IN</td>
<td>INDENT OUTPUT LINE</td>
</tr>
<tr>
<td>MAKINC/MA</td>
<td>MAKE INCLUDE FILES FOR FORMS</td>
</tr>
<tr>
<td>PLISUB</td>
<td>PL/I SUBROUTINE</td>
</tr>
</tbody>
</table>

**FPFARM**

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBSUB</td>
<td>COBOL SUBROUTINE</td>
</tr>
<tr>
<td>CSUB</td>
<td>C SUBROUTINE</td>
</tr>
<tr>
<td>FLAN/MAIN</td>
<td>FLAN MAIN PROGRAM</td>
</tr>
<tr>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
</tr>
<tr>
<td>GETFILE</td>
<td>GET INPUT FILENAME</td>
</tr>
<tr>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
</tr>
<tr>
<td>MAKINC/IN</td>
<td>INDENT OUTPUT LINE</td>
</tr>
<tr>
<td>MAKINC/MA</td>
<td>MAKE INCLUDE FILES FOR FORMS</td>
</tr>
<tr>
<td>PLISUB</td>
<td>PL/I SUBROUTINE</td>
</tr>
</tbody>
</table>
**FORMS LANGUAGE COMPILER Where-include-file-used List**

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
</tbody>
</table>

**MATH**

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
</tr>
<tr>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>

**NTM**

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAN MAIN</td>
<td>FLAN MAIN PROGRAM</td>
</tr>
<tr>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
</tr>
<tr>
<td>GETFILE</td>
<td>GET INPUT FILENAME</td>
</tr>
</tbody>
</table>

**RW**

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDCHK</td>
<td>ADD POSITION TO CHECK LIST</td>
</tr>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
</tr>
<tr>
<td>FLDTYP</td>
<td>FIELD TYPE</td>
</tr>
<tr>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
</tr>
<tr>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
</tr>
<tr>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
</tr>
<tr>
<td>MAKINT</td>
<td>MAKE EXPRESSION INTO AN INTEGER</td>
</tr>
<tr>
<td>MAKSTR</td>
<td>MAKE EXPRESSION INTO A STRING</td>
</tr>
<tr>
<td>MKPOS</td>
<td>MAKE POSITION NODE</td>
</tr>
<tr>
<td>MYALLOC</td>
<td>MY MALLOC</td>
</tr>
<tr>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>
FORMS LANGUAGE COMPILER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
</table>

**STDDIO**

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

<table>
<thead>
<tr>
<th>Include</th>
<th>Module</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Name</td>
<td>Purpose</td>
</tr>
</tbody>
</table>

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

3-26
FORMS LANGUAGE COMPILER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th>System</th>
<th>Module</th>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Module</td>
<td>Name</td>
<td>Purpose</td>
</tr>
<tr>
<td></td>
<td></td>
<td>--</td>
<td>----</td>
</tr>
<tr>
<td>ABS</td>
<td>CHKARY</td>
<td>CHECK</td>
<td>ARRAY</td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK</td>
<td>FORM</td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK</td>
<td>FORM</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>FRNTND</td>
<td>FORMS</td>
<td>FRONT END FOR FLAN</td>
</tr>
<tr>
<td>ATOF</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td>ATOI</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td>BLEN</td>
<td>CHKFLD</td>
<td>CHECK</td>
<td>FIELD</td>
</tr>
<tr>
<td></td>
<td>COBSUB</td>
<td>COBOL</td>
<td>SUBROUTINE</td>
</tr>
<tr>
<td></td>
<td>CSUB</td>
<td>C</td>
<td>SUBROUTINE</td>
</tr>
<tr>
<td></td>
<td>PLISUB</td>
<td>PL/I</td>
<td>SUBROUTINE</td>
</tr>
<tr>
<td>CALLOC</td>
<td></td>
<td>MAKINC/MAIMAKE INCLUDE FILES FOR FORMS</td>
<td></td>
</tr>
<tr>
<td>CLSFRM</td>
<td></td>
<td>MAKINC/MAIMAKE INCLUDE FILES FOR FORMS</td>
<td></td>
</tr>
</tbody>
</table>
FORMS LANGUAGE COMPILER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELFLD</td>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
</tr>
</tbody>
</table>

**FCLOSE**
- MAKINC/MAIMAKE INCLUDE FILES FOR FORMS
- REVFLAN/MAREVERSE FLAN
- WRTFRM WRITE FORM

**FEOF**
- MAKINC/MAIMAKE INCLUDE FILES FOR FORMS

**FNDMSG**
- MAKINC/MAIMAKE INCLUDE FILES FOR FORMS

**FOPEN**
- GETFILE GET INPUT FILENAME
- MAKINC/MAIMAKE INCLUDE FILES FOR FORMS
- REVFLAN/MAREVERSE FLAN
- WRTFRM WRITE FORM

**PRINTF**
- COBSUB COBOL SUBROUTINE
- CSUB C SUBROUTINE
- PLISUB PL/I SUBROUTINE
- REVFLAN/MAREVERSE FLAN

**FREAD**
- REVFLAN/MAREVERSE FLAN
FORMS LANGUAGE COMPILER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREE</td>
<td>CHXFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td></td>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>

| 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       | MAKING/MAIMAKE INCLUDE FILES FOR FORMS |

| GOPFTR        | MAKING/MAIMAKE INCLUDE FILES FOR FORMS |
FORMS LANGUAGE COMPILER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITAL</td>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
</tr>
<tr>
<td>INITFP</td>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
</tr>
<tr>
<td>ISALNUM</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>ISALPHA</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>ISDIGIT</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>ISSPACE</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>MAKFLD</td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
<tr>
<td>MALLOC</td>
<td>CSUB</td>
<td>C SUBROUTINE</td>
</tr>
<tr>
<td>System</td>
<td>Module</td>
<td>Module Name</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MYALLOC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REVFLAN/MAREVERSE</td>
</tr>
<tr>
<td>MAX</td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td>MEMCMP</td>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
</tr>
<tr>
<td>MEMCPY</td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WRTWRITE</td>
<td>FIELD</td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
<tr>
<td>MEMSET</td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td>OISCR</td>
<td></td>
<td>FLAN/MAIN</td>
</tr>
<tr>
<td></td>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
</tr>
<tr>
<td>OPWFRM</td>
<td></td>
<td>MAKINC/MAIMAKE</td>
</tr>
</tbody>
</table>
# FORMS LANGUAGE COMPILER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMSGLC</td>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
<td></td>
</tr>
<tr>
<td>PMSGLS</td>
<td>ERROR</td>
<td>ISSUE ERROR MESSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FATAL</td>
<td>ISSUE FATAL ERROR MESSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
<td></td>
</tr>
<tr>
<td>PRINTF</td>
<td>MAKINC/MAIMAKE INCLUDE FILES FOR FORMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REVFLAN/MAREVERSE FLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
<tr>
<td>PUTC</td>
<td>MAKINC/INDINDENT OUTPUT LINE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLISUB</td>
<td>PL/I SUBROUTINE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REVFLAN/MAREVERSE FLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCANNF</td>
<td>MAKINC/MAIMAKE INCLUDE FILES FOR FORMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REVFLAN/MAREVERSE FLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRINTF</td>
<td>ERROR</td>
<td>ISSUE ERROR MESSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FATAL</td>
<td>ISSUE FATAL ERROR MESSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FRNTND</td>
<td>FORMS FRONT END FOR FLAN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GETFILE</td>
<td>GET INPUT FILENAME</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKINC/MAIMAKE INCLUDE FILES FOR FORMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REVFLAN/MAREVERSE FLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
<td></td>
</tr>
<tr>
<td>System Module Name</td>
<td>Module Name</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STRASN</td>
<td>CHECK ARRAYS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STRCAT</td>
<td>YYPARSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STRCHR</td>
<td>FLAN MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FRMTND</td>
<td>FORMS FRONT END FOR FLAN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STRCMP</td>
<td>FIND ATTRIBUTE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
</tbody>
</table>

STRCPY
<table>
<thead>
<tr>
<th>CSTASH</th>
<th>CHARACTER STASH</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSUB</td>
<td>C SUBROUTINE</td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
<tr>
<td>WRTFRM/WRTWRITE</td>
<td>FIELD</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>
**FORMS LANGUAGE COMPILER Where-external-routine-used List**

<table>
<thead>
<tr>
<th>System</th>
<th>Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRLEN</td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ERROR</td>
<td>ISSUE ERROR MESSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FATAL</td>
<td>ISSUE FATAL ERROR MESSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REVFLAN/MAREVERSE FLAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WRTWRITE TEXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STRLOC</th>
<th>CSUB</th>
<th>C SUBROUTINE</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>STRNCMP</th>
<th>MAKING/MAIMAKE INCLUDE FILES FOR FORMS</th>
<th>REVFLAN/MAREVERSE FLAN</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>STRNCPY</th>
<th>WRTFRM/WRTWRITE FIELD</th>
<th>YYPARSE</th>
<th>FLAN PARSER</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>STRUPC</th>
<th>MAKING/MAIMAKE INCLUDE FILES FOR FORMS</th>
<th>YYPARSE</th>
<th>FLAN PARSER</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Module</td>
<td>Module Name</td>
<td>Module Purpose</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSMSG</td>
<td>CHECK FIELD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TERMFP</td>
<td>FLAN MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOUPPER</td>
<td>YYLEX LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRMNAT</td>
<td>FLAN MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FORMS FRONT END FOR FLAN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UNGETC</td>
<td>YYLEX LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>YYERROR</td>
<td>YYPARSE FLAN PARSER</td>
<td></td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAN/MAIN</td>
<td>Purpose---FLAN MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td>ABS</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ADDCHK</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ADDFRM</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ATOF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ATOI</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>BLEN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>CHKARY</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CHKFLD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CHKFPRM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CSTASH</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DELFLD</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ERROR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FATAL</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FCLOSE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FLANCI</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FLDTYPE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FNDATT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FOPEN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FREE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FRNTND</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FWRITE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GDATA</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETFILE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GFLDPT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>INITIAL</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>INITFP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ISALNUM</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ISALPHA</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ISDIGIT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ISSPACE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAKACT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKFLD</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAKINT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKSTR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MALLOC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MEMCMP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>

3-39
FORMS LANGUAGE COMPILER Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MEMSET</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>MKPOS</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MYALLOC</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>OISCR</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>PMSGLC</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>PMSGLS</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>PRINTF</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>SPRINTF</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRASN</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRCAT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRCHR</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRCMP</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRLEN</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRNCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRUPC</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>SYSMSG</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>TERMFP</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>TOUPPER</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>TRHNAT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>UNGETC</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>WARNING</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTEXP</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTF RM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTF RM/DBF CLOS</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTF RM/FORMAT</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTF RM/TBF CLOS</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTF RM/WRTDBF</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTF RM/WRTFLD</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTF RM/WRTTBF</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>WRTF RM/WRTTXT</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>YYERROR</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>YYLEX</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>Well-defined module</td>
</tr>
<tr>
<td>Main Pgm Name</td>
<td>Module Name</td>
<td>Module Type</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>MAKINC/MAIN</td>
<td>Purpose--MAKE INCLUDE FILES FOR FORMS</td>
<td></td>
</tr>
<tr>
<td>BLEN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>CALLOC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>CLSFRM</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>COBSUB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CSUB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FCLOSE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FEOF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FNDMSG</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FOPEN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FPRINTF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETCHAR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GOFPTR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAKINC/INDENT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MALLOC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>OPMFRM</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PLISUB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PRINTF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PUTC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>SCANF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>SPRINTF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRLOC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRNCMP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRUPC</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
FORMS LANGUAGE COMPILER Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>REVFLAN/MAIN</td>
<td>FCLOSE</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>FOPEN</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>FPRINTF</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>FREAD</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETC</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>MALLOC</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>PRINTF</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>PUTC</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>SCANF</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>SPRINTF</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRLEN</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>STRNCSMP</td>
<td>External routine</td>
</tr>
</tbody>
</table>
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
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.
NAME: ADDCHK
PURPOSE: ADD POSITION TO CHECK LIST
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: FLANSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS
VOID ADDCHK(POSPTR)
    POS *POSPTR;

DESCRIPTION
ADDS THE SPECIFIED POSITION TO THE OVERLAP CHECK LIST

ARGUMENTS:

POSPTR = POS *

INCLUDE FILES:

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

ROUTINES CALLED:

FLDTYP - FIELD TYPE
ERROR - ISSUE ERROR MESSAGE

CALLED DIRECTLY BY:

3-45
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:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS
FP.CODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

MYALLOC - MY MALLOC
ABS
STRASN

3-47
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:

STDTYPE - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS
FPFCODE - 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
STLEN

3-49
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: FDFE/FLAN

DESCRIPTION:

----------
SYNOPSIS
VOID CHKFRM()

DESCRIPTION
CHECKS THE CURRENT FORM FOR COMPLETENESS AND CONSISTENCY

INCLUDE FILES:

---------
STDTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS
FPFDC - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

---------
WARNING - ISSUE WARNING MESSAGE
ADDCCH - ADD POSITION TO CHECK LIST
CHKKARY - CHECK ARRAY
ABS
STRLEN
FREE
FLDTP - FIELD TYPE
ERROR - ISSUE ERROR MESSAGE
GFLDPT - GET FIELD POINTER
ABS
MAX

3-51
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:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

COBSUB - COBOL SUBROUTINE
MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS
USED IN MAIN PROGRAM(S):
------------------------
MAKING/MAI - MAKE INCLUDE FILES FOR FORMS
NAME: CSTASH
PURPOSE: CHARACTER STASH
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR *()
SOURCE FILE: FLANSP
SOURCE FILE TYPE: .C
HOST: UI
SUBSYSTEM: FE
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

---

SYNOPSIS

CHAR *CSTASH(S)
CHAR *S;

DESCRIPTION

SAVES THE SPECIFIED CHARACTER STRING AND RETURNS A
POINTER TO IT

ARGUMENTS:

---

S = CHAR *

INCLUDE FILES:

---

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

ROUTINES CALLED:

---

STRCPY
STRLEN
MYALLOC  - MY MALLOC
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:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

3-57
CSUB - C SUBROUTINE
MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS

USED IN MAIN PROGRAM(S):
---------------------
MAKINC/MAI - MAKE INCLUDE FILES FOR FORMS
NAME: ERROF
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)

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

3-59
CALLED DIRECTLY BY:

----------------------
FLAN/MAIN - FLAN MAIN PROGRAM
GETFILE - GET INPUT FILENAME
CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM
ADDCRK - ADD POSITION TO CHECK LIST
YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):
----------------------
FLAN/MAIN - FLAN MAIN PROGRAM
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)

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:

PRINTF
STRLEN
PMSGCLS
CALLED DIRECTLY BY:

- MYALLOC - MY MALLOC
- YYLEX - LEXICAL ANALYZER FOR FLAN
- YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

- FLAN/MAIN - FLAN MAIN PROGRAM
# FLAN/MAIN 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:**

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FP - FORM PROCESSOR DATA
FPDINI - FPL INITIALIZATION
NTM - NTM INTERFACE INCLUDE FILE
FPARM - FORM PROCESSOR PARAMETERS

---

**ROUTINES CALLED:**

STRCHR

---

3-63
GETFILE - GET INPUT FILENAME
FLANCI - FLAN CALLABLE INTERFACE
WRTFRM - WRITE FORM
ERROR - ISSUE ERROR MESSAGE
OISCR
TERMFP
TRMNAT
FRNTND - FORMS FRONT END FOR FLAN
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
STDLIO - "**** 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
NAME: FLDTYPE
PURPOSE: FIELD TYPE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR *
SOURCE FILE: FLANSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

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

DESCRIPTION
RETURNS A STRING OF THE SPECIFIED FIELD TYPE

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

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

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

USEL IN MAIN PROGRAM(S):
-------------------------

3-67
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:

STDTPY - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS
FPICODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

STRCMP

CALLED DIRECTLY BY:

3-69
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:

3-71
FILNAM = CHAR [41]

INCLUDE FILES:
-------------------
STDTYP      - STANDARD TYPE DEFINITIONS
FPPARM      - FORM PROCESSOR PARAMETERS
NTH         - NTH INTERFACE INCLUDE FILE

ROUTINES CALLED:
-------------------
STRCHR
INITAL
MEMCMP
TRMNAT
PMSGLC
INITFP
ADDFRM
GDATA
OISCR
PRINTF

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: UI
SUBSYSTEM: FLAN
SUBDIRECTORY: FLAN
DOCUMENTATION GROUP: FLAN

DESCRIPTION:

ARGUMENTS:

ARGC = INT
NAMPTR = CHAR *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
STDDO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FPD INTERFACE INCLUDE FILE
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
NAME: GFLDPT
PURPOSE: GET FIELD POINTER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD * ()
SOURCE FILE: FLANSP
SOURCE FILE TYPE: .C
HOST: UI
SUBSYSTEM: SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

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

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

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

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

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

CALLED DIRECTLY BY:
3-75
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 ****
STD_TYPED - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPFARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
MATH - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

---------
MYALLOC - MY MALLOC

3-77
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
STIO - ***** PURPOSE NOT FOUND BY STRIPPER *****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPFARM - 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:     MAKING/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 FILES:

---------
STDTPY  - STANDARD TYPE definitions
STDIO    - PURPOSE NOT FOUND BY STRIPPER ****
FPD      - FORM PROCESSOR DATA
FPDINI   - FPD INITIALIZATION
FPAPRM   - FORM PROCESSOR PARAMETERS
FPARCD   - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

---------
CALLOC
PRINTF
GOFPTR
CLSFRM
FCLOSE
CSUB     - C SUBROUTINE
COBSSUB   - COBOL SUBROUTINE
PLISUB   - PL SUBROUTINE
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A
PS 620144401
1 November 1985

PRINTF
FOPEN
STRUPC
OPNFRM
STRNCMP
FNDMSG
GETCHAR
FEOF
SCANF
NAME: MAKINT
PURPOSE: MAKE EXPRESSION INTO AN INTEGER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: ENODE * (
SOURCE FILE: FLANSP
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

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

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

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

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

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

CALLED DIRECTLY BY:
3-83
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:
---------------
STDDTYP       - 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:

3-85
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

FLAN/Main - FLAN MAIN PROGRAM
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
CREATE 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
STDDO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS
FPICODE - 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
NAME: MYALLOC
PURPOSE: MY MALLOC
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: FLANSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FE
DOCUMENTATION GROUP: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

CHAR *MYALLOC(SIZE)
UNSIGNED SIZE:

DESCRIPTION

ALLOCATE THE SPECIFIED MEMORY IF POSSIBLE. ELSE ISSUE
FATAL ERROR

ARGUMENTS:

SIZE = UNSIGNED

INCLUDE FILES:

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

ROUTINES CALLED:

FATAL - ISSUE FATAL ERROR MESSAGE
MALLOC

CALLED DIRECTLY BY:

3-89
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: PLAN

DESCRIPTION:

ARGUMENTS:

DP - FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

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

3-91
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:

STDTYPE - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
ROUTINES CALLED:

PRINTF
FPRINTF
FCLOSE
GETC
PUTC
MALLOC
FREAD
SCANF
STRLEN
STRNCMP
SPRINTF
FOPEN
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)

DESCRIPTION
PRINTS A WARNING MESSAGE ON STDERR

ARGUMENTS:

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

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

PMSGLS
STRLEN
SPRINTF
CALLED DIRECTLY BY:

- chkfrm - check form
- yylex - lexical analyzer for flan
- yyparse - flan parser

USED IN MAIN PROGRAM(S):

- flan/main - flan main program
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

3-97
ROUTINES CALLED:

FREE
WRTEXP - WRITE EXPRESSION
MEMCPY
MYALLOC - MY MALLOC
STRLEN
PRINTF

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:
-------------
STDTYP - 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:
-----------------
PRINTF
FOPEN

3-99
SYMSG
WRITE
FCLOSE
WRTFRM/WRTTXT - WRITE TEXT
WRTFRM/WRTFLD - WRITE FIELD
WRTFRM/WRTTBF - WRITE TEXT BUFFER
WRTFRM/TBFCLOSE - TEXT BUFFER CLOSE
WRTFRM/WRTDBF - WRITE DEFAULT BUFFER
WRTFRM/DBFCLOSE - DEFAULT BUFFER CLOSE
STRASH
STRCPY
STRLEN

CALLED DIRECTLY BY:

------------------
| FLAN/MAIN - FLAN MAIN PROGRAM |
------------------

USED IN MAIN PROGRAM(S):

------------------
| FLAN/MAIN - FLAN MAIN PROGRAM |
------------------
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:

STDTYP - 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 COMPILERS 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:
----------------
STDTYP - 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:
-----------------
STDTYP - STANDARD TYPE DEFINITIONS
STDBO - **** 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/WRTDBF
PURPOSE: WRITE DEFAULT BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: WRTFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FDPE/FLAN

DESCRIPTION:

SYNOPSIS

```c
INT WRTDBF(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:

```
STDTYP - STANDARD TYPE DEFINITIONS
STDIO - ***** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFFFV2 - 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/WRTFLD
PURPOSE: WRITE FIELD
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

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

DESCRIPTION
WRITES THE FIELD RECORD FOR THE SPECIFIED FIELD STRUCTURE.

ARGUMENTS:

FPTR = FILE *
FLDPTR = FIELD *

INCLUDE FILES:

STDTYPE - STANDARD TYPE DEFINITIONS
STUDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDCODE - FORM PROCESSOR RETURN CODES
FFFV2 - FORM FILE FORMAT - VERSION 2

ROUTINES CALLED:

FWRITE
STRCPY
WRTFRM/FORMAT - INSERT FORMAT CODES
STRNCPY

3-109
MEMCPY

CALLED DIRECTLY BY:

---------------------
WRTFRM - WRITE FORM

USED IN MAIN PROGRAM(S):

-----------------------
FLAN/MAIN - FLAN MAIN PROGRAM
FORMS LANGUAGE COMPILER Module Documentation

NAME: WRTFRM/WRTTBF
PURPOSE: WRITE TEXT BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ( )
SOURCE FILE: WRTFRM
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:

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

3-111
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: WRTFRMWRTTXT
PURPOSE: WRITE TEXT
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

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

DESCRIPTION

WRITES THE TEXT RECORD FOR THE SPECIFIED TEXT STRUCTURE.

ARGUMENTS:

FPTR - FILE *
TXTPTR - TEXT *

INCLUDE FILES:

STDTYPE - STANDARD TYPE DEFINITIONS
STDI0 - ***** 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:

-----------------
WRTFRM - 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 RETURN THE APPROPRIATE TOKEN TYPE.

INCLUDE FILES:

-----------------
FLAN.Y* - ***** PURPOSE NOT FOUND BY STRIPPER ****
STDTYP - 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

3-115
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: FDFF/FLAN

DESCRIPTION:

DESCRIPTION DEFINITION LANGUAGE GRAMMAR.

INCLUDE FILES:

- FLAN.Y* - **** PURPOSE NOT FOUND BY STRIPPER ****
- STDYP - 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
PRINTF
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 COMPILE 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

3-127
FORMS LANGUAGE COMPILER Include File Description

FILE NAME: STDYP
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

3-128
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 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.
PS 620144401
1 November 1985

+--------+
| MAKINC/MAIN |
+--------+

+--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+
| (CONT) | CSUB | COBSUB | PLISUB | SPRINTF | (CONT) |
| +--------+ +--------+ +--------+ +--------+ +--------+ +--------+
| +--------+ +--------+ +--------+ +--------+ +--------+ +--------+
| +--------+ +--------+ +--------+ +--------+ +--------+ +--------+
| COBSUB | BLN | FPRINTF | MAKINC/INDENT |
| +--------+ +--------+ +--------+ +--------+ +--------+ +--------+

3-139
PS 620144401
1 November 1985

31

+--------+
| CHKFLD |
+--------+

+--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+

| (CONT) | | WRTEXP | | BLEN | | MEMCPY | | SYMSG | | (CONT) |
+--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+ +--------+

3-162
<table>
<thead>
<tr>
<th>Function</th>
<th>Line No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>1</td>
</tr>
<tr>
<td>ADDCHK</td>
<td>25</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>26</td>
</tr>
<tr>
<td>ATOF</td>
<td>27</td>
</tr>
<tr>
<td>ATOI</td>
<td>28</td>
</tr>
<tr>
<td>BLEN</td>
<td>29</td>
</tr>
<tr>
<td>CALLOC</td>
<td>30</td>
</tr>
<tr>
<td>CHKARY</td>
<td>31</td>
</tr>
<tr>
<td>CHKLD</td>
<td>32</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>33</td>
</tr>
<tr>
<td>CLSFRM</td>
<td>34</td>
</tr>
<tr>
<td>COBSUB</td>
<td>35</td>
</tr>
<tr>
<td>CSTASH</td>
<td>36</td>
</tr>
<tr>
<td>CSUB</td>
<td>37</td>
</tr>
<tr>
<td>DELFLD</td>
<td>38</td>
</tr>
<tr>
<td>ERROR</td>
<td>39</td>
</tr>
<tr>
<td>FATAL</td>
<td>40</td>
</tr>
<tr>
<td>FCLOSE</td>
<td>41</td>
</tr>
<tr>
<td>FEOF</td>
<td>42</td>
</tr>
<tr>
<td>FLAN/MAIN</td>
<td>43</td>
</tr>
<tr>
<td>FLANCI</td>
<td>44</td>
</tr>
<tr>
<td>PLDTYP</td>
<td>45</td>
</tr>
<tr>
<td>FNDATT</td>
<td>46</td>
</tr>
<tr>
<td>FNDMSG</td>
<td>47</td>
</tr>
<tr>
<td>FOOPEN</td>
<td>48</td>
</tr>
<tr>
<td>FPRINTF</td>
<td>49</td>
</tr>
<tr>
<td>FREE</td>
<td>50</td>
</tr>
<tr>
<td>FRNTND</td>
<td>51</td>
</tr>
<tr>
<td>FWRITE</td>
<td>52</td>
</tr>
<tr>
<td>GDATA</td>
<td>53</td>
</tr>
<tr>
<td>GETC</td>
<td>54</td>
</tr>
<tr>
<td>GETCHAR</td>
<td>55</td>
</tr>
<tr>
<td>GETFILE</td>
<td>56</td>
</tr>
<tr>
<td>GFLDPT</td>
<td>57</td>
</tr>
<tr>
<td>GOPPTR</td>
<td>58</td>
</tr>
<tr>
<td>INITIAL</td>
<td>59</td>
</tr>
<tr>
<td>INITFP</td>
<td>60</td>
</tr>
<tr>
<td>ISALNUM</td>
<td>61</td>
</tr>
<tr>
<td>ISALPHA</td>
<td>62</td>
</tr>
<tr>
<td>ISDIGIT</td>
<td>63</td>
</tr>
<tr>
<td>ISSPACE</td>
<td>64</td>
</tr>
<tr>
<td>MAKACT</td>
<td>65</td>
</tr>
<tr>
<td>MAKFLD</td>
<td>66</td>
</tr>
<tr>
<td>MAKINC/INDENT</td>
<td>67</td>
</tr>
<tr>
<td>MAKINC/MAIN</td>
<td>68</td>
</tr>
<tr>
<td>MAKINT</td>
<td>69</td>
</tr>
<tr>
<td>MAKSTR</td>
<td>70</td>
</tr>
<tr>
<td>MALLOCMAX</td>
<td>71</td>
</tr>
<tr>
<td>MALLOC</td>
<td>72</td>
</tr>
<tr>
<td>MEMCMP</td>
<td>73</td>
</tr>
<tr>
<td>MEMCPY</td>
<td>74</td>
</tr>
<tr>
<td>MEMSET</td>
<td>75</td>
</tr>
<tr>
<td>MKPOS</td>
<td>76</td>
</tr>
<tr>
<td>MYALLOC</td>
<td>77</td>
</tr>
<tr>
<td>OISCR</td>
<td>78</td>
</tr>
<tr>
<td>OPNFRM</td>
<td>79</td>
</tr>
<tr>
<td>PLISUB</td>
<td>80</td>
</tr>
<tr>
<td>PMSGLC</td>
<td>81</td>
</tr>
<tr>
<td>PMSGLS</td>
<td>82</td>
</tr>
<tr>
<td>PRINTF</td>
<td>83</td>
</tr>
<tr>
<td>PUTC</td>
<td>84</td>
</tr>
<tr>
<td>REVFLAN/MAIN</td>
<td>85</td>
</tr>
<tr>
<td>SCAINF</td>
<td>86</td>
</tr>
<tr>
<td>SPRINTF</td>
<td>87</td>
</tr>
<tr>
<td>STRASN</td>
<td>88</td>
</tr>
<tr>
<td>STRCAT</td>
<td>89</td>
</tr>
<tr>
<td>STRCHR</td>
<td>90</td>
</tr>
<tr>
<td>STRCMP</td>
<td>91</td>
</tr>
<tr>
<td>STRCPY</td>
<td>92</td>
</tr>
<tr>
<td>STRLEN</td>
<td>93</td>
</tr>
<tr>
<td>STRLOC</td>
<td>94</td>
</tr>
<tr>
<td>STRNCMP</td>
<td>95</td>
</tr>
<tr>
<td>STRNCPY</td>
<td>96</td>
</tr>
<tr>
<td>STRUPC</td>
<td>97</td>
</tr>
<tr>
<td>SYSMSG</td>
<td>98</td>
</tr>
<tr>
<td>TERMFP</td>
<td>99</td>
</tr>
<tr>
<td>TOUPPER</td>
<td>100</td>
</tr>
<tr>
<td>TRMNat</td>
<td>101</td>
</tr>
<tr>
<td>UNGETC</td>
<td>102</td>
</tr>
<tr>
<td>WARNING</td>
<td>103</td>
</tr>
<tr>
<td>WRTEXP</td>
<td>104</td>
</tr>
<tr>
<td>WRTFRM</td>
<td>105</td>
</tr>
<tr>
<td>WRTFRM/DBFCLOS</td>
<td>106</td>
</tr>
<tr>
<td>WRTFRM/FORMAT</td>
<td>107</td>
</tr>
<tr>
<td>WRTFRM/TBF CLOS</td>
<td>108</td>
</tr>
<tr>
<td>WRTFRM/WRT DBF</td>
<td>109</td>
</tr>
<tr>
<td>WRTFRM/WRTFLD</td>
<td>110</td>
</tr>
<tr>
<td>WRTFRM/WRT TBF</td>
<td>111</td>
</tr>
<tr>
<td>WRTFRM/WRT TXT</td>
<td>112</td>
</tr>
<tr>
<td>YYERROR</td>
<td>113</td>
</tr>
<tr>
<td>YYLEX</td>
<td>114</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>115</td>
</tr>
</tbody>
</table>
### 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."
END
8-87
DTIC