

WRDC-TR-90-8007
Volume VIII
Part 27



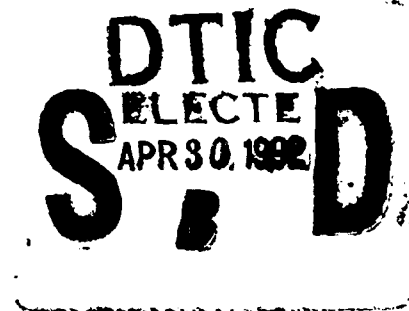
AD-A250 484



INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 27 - Rapid Application Generator Product Specification

F. Glandorf

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



September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

92-11383



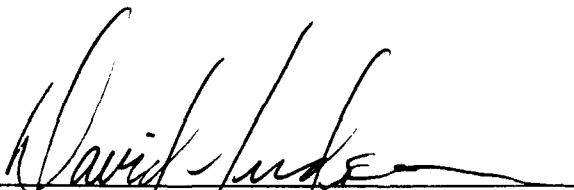
MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

92 4 23 018

NOTICE

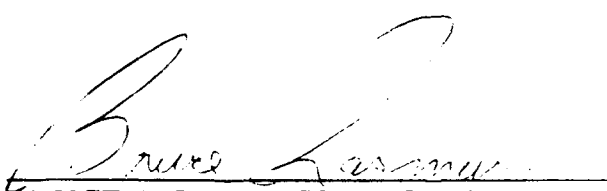
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may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

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


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

25 July 91
DATE

FOR THE COMMANDER:


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

25 July 91
DATE

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

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

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) DS 620344600		5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR-90-8007 Vol. VIII, Part 27	
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services	6b. OFFICE SYMBOL (if applicable) WRDC/MTI	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI	
6c. ADDRESS (City, State, and ZIP Code) 2970 Presidential Drive Fairborn, OH 45324-6209		7b. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6533	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF	8b. OFFICE SYMBOL (if applicable) WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464	
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, Ohio 45433-6533		10. SOURCE OF FUNDING NOS.	
11. TITLE (Include Security Classification) F See block 19 tion		PROGRAM ELEMENT NO. 78011F	PROJECT NO. 595600
		TASK NO. F95600	WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Structural Dynamics Research Corporation: Glandorf, F.			
13a. TYPE OF REPORT Final Report	13b. TIME COVERED 4 / 1 / 87 - 12 / 30 / 90	14. DATE OF REPORT (Yr., Mo., Day) 1990 September 30	15. PAGE COUNT 371
16. SUPPLEMENTARY NOTATION WRDC/MTI Project Priority 6203			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify block no.)	
FIELD	GROUP	SUB GR.	
1338	0905		
19. ABSTRACT (Continue on reverse if necessary and identify block number) This specification establishes the detailed design of the Rapid Application Generator computer program. BLOCK 11: INTEGRATED INFORMATION SUPPORT SYSTEM Vol VIII -User Interface Subsystem Part 27 - Rapid Application Generator Product Specification			
20. DISTRIBUTION AVAILABILITY OF ABSTRACT UNCLASSIFIED-UNLIMITED x SAME AS RPT. DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. (Include Area Code) (513) 255-7371	22c. OFFICE SYMBOL WRDC MTI	

FOREWORD

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

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

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

SUBCONTRACTOR

ROLE

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

TABLE OF CONTENTS

Page

SECTION 1.0	SCOPE	1-1
1.1	Identification	1-1
1.2	Functional Summary	1-1
SECTION 2.0	DOCUMENTS	2-1
2.1	Reference Documents	2-1
2.2	Terms and Abbreviations	2-1
SECTION 3.0	REQUIREMENTS	3-1
3.1	Structural Description	3-1
3.2	Functional Flow	3-1
3.3	Interfaces	3-2
3.3.1	Forms Language Compiler	3-2
3.3.2	CDM Data Dictionary	3-2
3.3.3	Generated Application Interfaces	3-3
3.4	Program Interrupts	3-3
3.5	Timing and Sequencing Description ..	3-3
3.6	Special Control Features	3-3
3.7	Storage Allocation	3-3
3.7.1	Data Base Definition	3-3
3.7.1.1	File Descriptions	3-3
3.7.1.2	Table Description	3-6
3.8	Object Code Creation	3-6
3.9	Adaptation Data	3-6
3.10	Detailed Design Description	3-6
3.10.1	Main Program List	3-6
3.10.2	Module List	3-8
3.10.3	External Routines List	3-17
3.10.4	Include File List	3-20
3.10.5	Where Include File Used List	3-22
3.10.6	Where External Routine Used List..	3-47
3.10.7	Main Program Parts List	3-61
3.10.8	Module Documentation	3-68
3.10.9	Include File Description	3-303
3.10.10	Hierarchy Chart	3-317
3.11	Program Listings Comments	3-359
SECTION 4.0	QUALITY ASSURANCE PROVISIONS	4-1
4.1	Introduction and Definitions	4-1
4.2	Computer Programming and Test Evaluation	4-1

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3-1	Rapid Application Generator Data Flow	3-2

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

SECTION 1

SCOPE

1.1 Identification

This specification establishes the detailed design of a computer program identified as the Rapid Application Generator hereinafter referred to as RAP. RAP is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Functional Summary

RAP is used to facilitate development of screen driven interactive programs accessing databases through the Common Data Model (CDM).

The major functions of the Rapid Application Generator are:

- 1) To provide a screen driven interface to database application programs;
- 2) To provide a screen based means of displaying the contents of a database;
- 3) To provide a context within which a single application program can switch between modes of database access: update, query, deletion, and insertion; *AND*
- 4) To allow the application developer to apply human engineering to the means by which the user dialogues with the database program.

SECTION 2
DOCUMENTS

2.1 Reference Documents

- [1] Structural Dynamics Research Corporation, Application Interface Product Specification, PS 620144700 , 1 November 1985.
- [2] Structural Dynamics Research Corporation, Forms Driven Form Editor Product Specification, PS 620144402 , 1 November 1985.
- [3] Structural Dynamics Research Corporation, Form Processor Product Specification, PS 620144200 , 1 November 1985.
- [4] Structural Dynamics Research Corporation, Forms Language Compiler Product Specification, PS 620144401 , 1 November 1985.
- [5] Structural Dynamics Research Corporation, Report Writer Product Specification, PS 620144501 , 1 November 1985.
- [6] Structural Dynamics Research Corporation, Text Editor Product Specification, PS 620144600 , 1 November 1985.
- [7] Structural Dynamics Research Corporation, User Interface Services Product Specification, PS 620144100 , 1 November 1985.
- [8] Structural Dynamics Research Corporation, Virtual Terminal Product Specification, PS 620144300, 1 November 1985.
- [9] Structural Dynamics Research Corporation, Rapid Application Generator Development Specification, DS 620144502, 1 November 1985.
- [10] Structural Dynamics Research Corporation, Rapid Application Generator Unit Test Plan, UTP620144502 , 1 November 1985.
- [11] Structural Dynamics Research Corporation, Rapid Application Generator User Manual, UM 620144502, 1 November 1985.

2.2 Terms and Abbreviations

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

Application Generator: (AG), subset of the IISS User Interface that consists of software modules that generate IISS application code and associated form definitions based on a language input. The part of the AG that generates report programs is called the Report Writer. The part of the AG that generates interactive applications is called the Rapid Application Generator.

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

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

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

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.

Communication Services: allows on host interprocess communication and inter-host communication between the various Test Bed subsystems.

Communication Subsystem: (COMM), IISS subsystem that provides communication services to the Test Bed and subsystems.

Computer Program Configuration Item: (CPCI), an aggregation of computer programs or any of their discrete portions which satisfies an end-use function.

Conceptual Schema: (CS), the standard definition used for all data in the CDM. It is based on IDEF1 information modelling.

Cursor Position: the position of the cursor after any command is issued.

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

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

External Schema: (ES), an application's view of the CDM's conceptual schema.

Field: two dimensional space on a terminal screen.

Field Pointer: indicates the ITEM which contains the current cursor position.

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

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

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

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

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

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

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

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

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

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

Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on

heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

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

Logical Device: a conceptual device which is indistinguishable from a physical device to an application and is then mapped to part or all of a physical device.

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

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

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

Neutral Data Manipulation Language: (NDML), the command language by which the CDM is accessed for the purpose of extracting, deleting, adding, or modifying data.

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

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

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

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

Physical Device: a hardware terminal.

Presentation Schema: (PS), may be equivalent to a form. It is the view presented to the user of the application.

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

Rapid Application Generator: (RAP), part of the Application Generator that generates source code for interactive programs based on a language input.

Subform: a form that is used within another form.

Text Editor: (TE), subset of the IISS User Interface that consists of a file editor that is based on the text editing functions built into the Form Processor.

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

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

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

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

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

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

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

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

Virtual Terminal Interface: (VTI), the callable interface to the VT.

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

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

SECTION 3

REQUIREMENTS

3.1 Structural Description

The RAP is used to translate interactive database applications defined using the Application Definition Language (ADL) into programs that access databases via the CDM. Conceptually, the RAP is a compiler that takes ADL as input and generates:

- o Binary form definition files that determine the layout of the interactive screens.
- o A database operation program for each type of NDML statement that maps the CDM External Schema to the Presentation Schema (forms defined by the FD files) for the NDML operations. The module NDMLGEN.C calls the COBOL module CDMESQY.PRC to get meta data about the NDML operations from the CDM data dictionary to check for legal schema mappings. Illegal mappings are recorded in a warning file. This program also generates the appropriate NDML to do the operations and must be precompiled using the NDML precompiler.
- o A control flow program based on the specified conditions. This is the main module of the generated application that uses the Application Interface to display the interactive screens and process the information entered by the user.

3.2 Functional Flow

Figure 3-1 is a data flow diagram of the Rapid Application Generator environment.

where '*' = X - Select
I - Insert

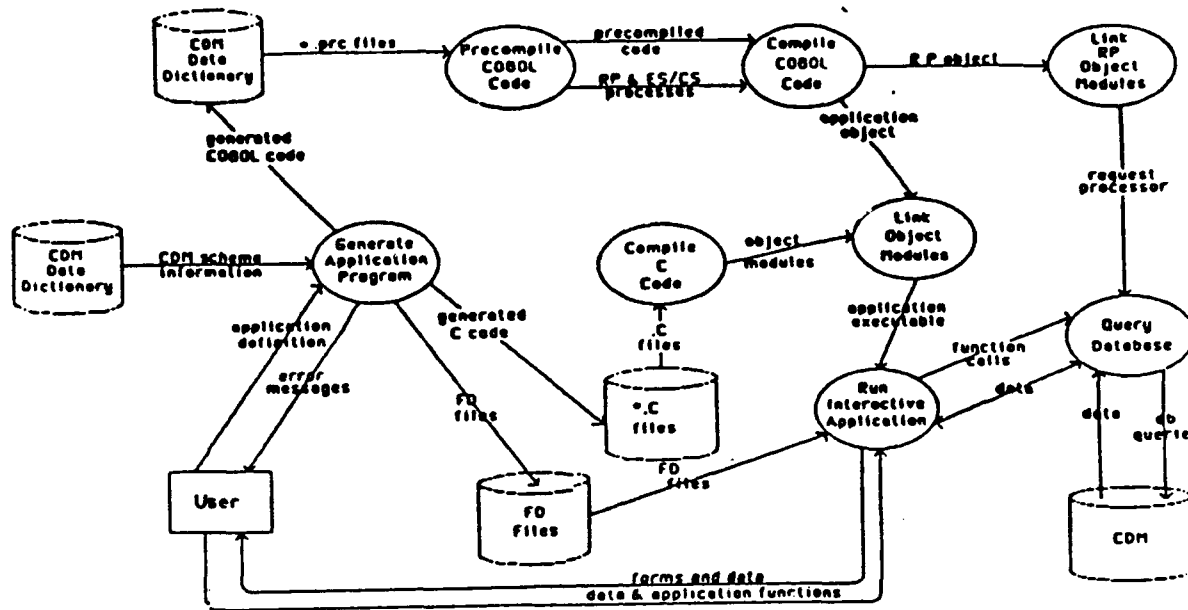


Figure 3-1 Rapid Application Generator Data Flow

3.3 Interfaces

3.3.1 Forms Language Compiler

The RAP uses the modules YTAB.C, FLANSP.C, and WRTFRM.C of the Forms Language Compiler (FLAN) to generate binary form definition files from its ADL input file. FLAN also produces the internal data structures used to generate the control flow program.

3.3.2 CDM Data Dictionary

The data base operation programs extract meta data about the NDML operations from the CDM Data Dictionary to check for legal External Schema to Presentation Schema mappings.

3.3.3 Generated Application Interfaces

The generated application is like any other IISS application. It interfaces with the User Interface via the Application Interface and the CDM via the CDMP calls generated by the NDML precompiler. All applications that use the CDMP or the Application Interface of the UI also interface with the NTM.

3.4 Program Interrupts

This section does not apply to the detailed design of the Rapid Application Generator.

3.5 Timing and Sequencing Description

This section does not apply to the detailed design of the Rapid Application Generator.

3.6 Special Control Features

This section does not apply to the detailed design of the Rapid Application Generator.

3.7 Storage Allocation

The Application Generator executable size is 422 blocks.

3.7.1 Data Base Definition

3.7.1.1 File Descriptions

1. 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:

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

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

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

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

2. FILE NAME: generated using the CDM file namer program with a TMP extention - the generated COBOL program processes the results of the NDML select and creates this Presentation Schema format file of the selected data.

PURPOSE: This file is a temporary file that pertains to the current NDML SELECT. It is input to the generated control flow program and processed to display the selected data on the user's terminal screen. If the interactive application terminates abnormally, this file may be examined to help determine the cause.

DECLARATION: The module GENDB.C generates a character type declaration based on the Presentation Schema sizes of the selected columns.

3. FILE NAME: *C.C - where * is the application name as given on the CREATE APPLICATION of the ADL file - generated code.

PURPOSE: This is the control flow program generated by the RAP that uses the Application Interface to display the interactive screens and process the information entered by the user.

DECLARATION: Character (i.e., PIC X(80). in COBOL).

4. FILE NAME: *z.PRC where * is the application name as given on the CREATE APPLICATION statement of the ADL file and 'z' indicates the NDML operation to be performed. A separate file for each NDML operation is generated with the following values for 'z':

X for SELECT
I for INSERT
D for DELETE
M for MODIFY

These files are COBOL code that contain:

- External Schema COBOL record structures
- Presentation Schema COBOL record structures
- Machine Representation Conversion code

PURPOSE: This code contains the CDM operation procedures to do the database operations specified by the NDML operations and map the External Schema to the Presentation Schema.

DECLARATION: Character (i.e., PIC X(80). in COBOL)

5. FILE NAME: *.WRN where * is the application name as given on the CREATE APPLICATION line of the ADL file - generated error file listing any inconsistencies in the External to Presentation Schema mapping.

PURPOSE: This file should be examined by the developer to verify inconsistencies in form item sizes and External Schema data.

DECLARATION: Character (i.e., PIC X(80). in COBOL)

3.7.1.2 Table Description

The database tables accessed by the RAP are predefined and under the control of the CDM.

3.8 Object Code Creation

The RAP routines were compiled using a C compiler developed by Interactive Software under VAX/VMS. The generated C programs can be compiled using the same compiler. The generated COBOL programs can be compiled using any ANSI COBOL compiler.

3.9 Adaptation Data

The C source modules for the RAP can be compiled using any UNIX version 7 compatible C compiler. The generated COBOL code must be precompiled using the NDML precompiler before being compiled by the COBOL compiler.

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.

APPLICATION GENERATOR Main Program List

Module Name -----	Purpose -----
GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM

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.

APPLICATION GENERATOR Module List

Module Name -----	Purpose -----
ACTRSV	ACTION RESOLVE
ADDCHK	ADD POSITION TO CHECK LIST
ASSIGN	ASSIGN FILE SECTION
BLDSUB	BUILD SUBROUTINES
BSCODE	BUILD SUBROUTINE CODE
CALCSTAT	CALCULATE STATISTIC
CCONV	C CONVERSIONS
CDMESQY	PROGRAM NAME CDMESQY
CES	C ES
CESPS	C ES TO PS
CHKARY	CHECK ARRAY
CHKFLD	CHECK FIELD
CHKFRM	CHECK FORM
CHKGRP	CHECK FOR GROUP SEPERATORS OR END OF FILE
CHKSIZE	CHECK SIZE OF ITEMS DOING CONVERSIONS ON
CLRNDP	CLEAR NODUPLICATE FIELDS
CLSFIL	CLOSE FILES
COBCONV	COBOL CONVERSIONS
COBES	COBOL ES RECORD
COBESPS	COBOL ES TO PS
COBPE	COBOL PE

APPLICATION GENERATOR Module List

Module Name -----	Purpose -----
CPE	C PE
CSTASH	CHARACTER STASH
CTLRSV	CONTROL RESOLVE
DASH	WRITE DASH '-'
DATAGEN	DATA DIVISION GENERATE
DBFREAD	GENERATE DATA BASE FREAD
DCLINDX	DECLARE INDEX VARIABLES
ENDGEN	END GERNERATE
ERROR	ISSUE ERROR MESSAGE
ESPSMAP	THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING
ESPSMAP/INDENT	INDENT
FATAL	ISSUE FATAL ERROR MESSAGE
FD	FD SECTION DECLARATIONS
FILELNK	FILE LINKAGE SECTION GENERATE
FLANCI	FLAN CALLABLE INTERFACE
FLDRSV	FIELD RESOLVE
FLDTYP	FIELD TYPE
FNDATT	FIND ATTRIBUTE
FNDFRM	FIND FORM
FRMPDAT	FORM PDATA

APPLICATION GENERATOR Module List

<u>Module Name</u> -----	<u>Purpose</u> -----
FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
GEN	GENERATE A LINE OF CODE
GENAA	GENERATE PROCEDURE "ADDACT" ADD AN ACTION
GENAAL	GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
GENACT	GENERATE ACTIONS
GENAE	GENERATE ACTION EXIT
GENAH	GENERATE ACTION HELP
GENAI	GENERATE ACTION INSERT
GENAL	GENERATE ACTION LIST
GENAP	GENERATE ACTION PAGE
GENAQ	GENERATE ACTION QUERY (SELECT)
GENAR	GENERATE ACTION PRESENT
GENAS	GENERATE ACTION SET
GENAT	GENERATE ACTION SIGNAL
GENBEG	GENERATE BEGINNING OF APPLICATION OR REPORT
GENCHG	GENERATE CHANGE DECLARATIONS
GENDB	GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
GENDOA	GENERATE PROCEDURE "DOACT" DO ACTION
GENDS	GENERATE DATA DATA STRUCTURES
GENFP	GENERATE FORM PATH

APPLICATION GENERATOR Module List

Module Name -----	Purpose -----
GENFS	GENERATE FORM DATA STRUCTURES
GENFSD	GENERATE FORM STRUCTURE DATA INITIALIZATION
GENINS	GENERATE INSERT DECLARATIONS
GENMAIN	GENERATE MAIN PROGRAM
GENNDP	GENERATE NODUPLICATE DECLARATIONS
GENPAG	GENERATE NEWPAG PROCEDURE
GENTAL	GENERATE TRIGGER ACTION LIST
GENTRG	GENERATE TRIGGERS
GETCOL	GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER
GETPTH	GET PATH
GETTBL	GET A TABLE NAME
GFLDPT	GET FIELD POINTER
GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
HASDATA	DETERMINE IF THERE ARE ANY SELECT STATEMENTS
HASITEM	THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
HASLOWER	HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
INDENT	INDENT A LINE OF GENERATED CODE

APPLICATION GENERATOR Module List

<u>Module Name</u> -----	<u>Purpose</u> -----
INSERT	INSERT PROCEDURE
INSRSV	INSERT RESOLVE
INWS	INSERT WORKING STORAGE SECTION
ISOPNE	DETERMINE IF THIS FIELD IS OPEN ENDED
MAKACT	MAKE ACTION LIST ELEMENT
MAKES	MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
MAKES/CNUMPIC	C NUMBERS
MAKES/INDENT	INDENT
MAKES/NUMPIC	NUMBER PICTURE CLAUSE
MAKINS	MAKE INSERT
MAKINT	MAKE EXPRESSION INTO AN INTEGER
MAKPS	MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
MAKQR	MAKE QUALIFIED REFERENCE
MAKSTR	MAKE EXPRESSION INTO A STRING
MAKWH	MAKE WHERE
MAKWHEs	MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
MAKWHEs/COBWHEs	COBOL WHERE ES
MAKWHEs/CWHEs	C WHERE ES
MAKWHEs/NUMPIC	NUMBER PICTURE CLAUSE
MAPDB	MAP DATABASE

APPLICATION GENERATOR Module List

Module Name -----	Purpose -----
MKINC	MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
MKPOS	MAKE POSITION NODE
MLPFRM	MAKE A LIST OF PRESENTED FORMS
MYALLOC	MY MALLOC
NDMLGEN	NDML COBOL APPLICATION GENERATOR
NDMLLAB	GENERATE LABELS
NDMLLNK	LINKAGE SECTION
NULBLK	BLANK FILL A STRING
OPNFIL	GENERATE OPEN FILE SECTION
PEMAP	THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING
PROCGEN	PROCEDURE DIVISION GENERATE
PSSTRC/COBSUB	COB L SUBSTITUTE
PSSTRC/CSUB	C SUBSTITUTE
PSSTRC/INDENT	INDENT
READDB	READ DATA BASE
RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
RSETSTAT	RESET STATISTIC
RWEXPD	REPORT WRITER EXPAND ARRAYS
RWOPN	REPORT WRITER OPEN FORMS

APPLICATION GENERATOR Module List

Module Name -----	Purpose -----
RWSP/FIXFRM	FIX UP A FORM
SAVEES	SAVE ES INFORMATION
SELECT	GENERATE SELECT CODE
SELGEN	SELECT GENERATE
SELLEN	COMPUTE LENGTH OF SELECT PS RECORD
SELMAP	MAP SELECTED DATA TO OUTPUT RECORD
SELOPN	SELECT OPEN
SELRSV	SELECT RESOLVE
SELWHR	SELECT WHERE
SELWS	SELECT WORKING STORAGE SECTION
SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
STATRSV	STATISTIC RESOLVE
STDCODE	STANDARD COBOL CODE
TRGRSV	TRIGGER RESOLVE
UQFOR	UNIVERSAL QUALIFIER FOR LOOP
UQPTH	UNIVERSAL QUALIFIER PATH
USING	GENERATE USING SECTION
VISITA	VISIT ARRAYS ON THIS FORM
WARNING	ISSUE WARNING MESSAGE
WINRSV	WINDOW RESOLVE

APPLICATION GENERATOR Module List

Module Name -----	Purpose -----
WRTEXP	WRITE EXPRESSION
WRTFRM	WRITE FORM
WRTFRM/DBFCLOS	DEFAULT BUFFER CLOSE
WRTFRM/FORMAT	INSERT FORMAT CODES
WRTFRM/TBFCLOS	TEXT BUFFER CLOSE
WRTFRM/WRTDBF	WRITE DEFAULT BUFFER
WRTFRM/WRTFLD	WRITE FIELD
WRTFRM/WRTTBF	WRITE TEXT BUFFER
WRTFRM/WRTTXT	WRITE TEXT
YYLEX	LEXICAL ANALYZER FOR FLAN
YYPARSE	FLAN PARSER

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. See section 3.10.6 for a list of the modules that call each of these external routines.

APPLICATION GENERATOR External Routines List

Module Name -----	First User -----
ABS	CHKFRM
ADDFRM	FRNTND
ATOF	YYLEX
ATOI	YYLEX
BLFN	CHKFLD
CALLOC	GRP/MAIN
COPFLD	RWEXPD
DELFLD	FLANCI
ERRPRO	CDMESQY
ESCPY	CCONV
FCLOSE	NDMLGEN
FOPEN	WRTFRM
FPRINTF	INSERT
FREE	CHKFLD
FWRITE	WRTFRM/WRTDBF
GDATA	FRNTND
GETC	YYLEX
INITAL	FRNTND
INITFP	FRNTND
INSMAP	PROCGEN
ISALNUM	YYLEX
ISALPHA	YYLEX
ISDIGIT	YYLEX
ISSPACE	YYLEX
MAKFLD	YYPARSE
MALLOC	MLPFRM
MAP	PROCGEN
MAX	CHKFRM
MEMCMP	FRNTND
MEMCPY	WRTFRM/WRTFLD
MEMSET	CHKFLD
OISCR	FRNTND
PMSGLC	FRNTND
PMSGLS	FATAL
PRINTF	YYPARSE
PSESMAP	PROCGEN
PTHPTR	GETPTH
PUTC	ESPSMAP/INDENT
SPRINTF	GENDOA
STRASN	WRTFRM
STRCAT	DCLINDX

APPLICATION GENERATOR External Routines List

Module Name -----	First User -----
STRCHR	GETPTH
STRCMP	RWSP/FIXFRM
STRCPY	INSWS
STRLEN	CSTASH
STRNCMP	SAVEES
STRNCPY	YYPARSE
STRSPN	GENAS
STRUPC	GETPTH
SYSMSG	CHKFLD
TERMFP	GRP/MAIN
TOUPPER	YYLEX
TRMNAT	FRNTND
TRMNDML	GRP/MAIN
UNGETC	YYLEX
YYERROR	YYPARSE

3.10.4 Include File List

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

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

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

APPLICATION GENERATOR Include File List

File Name -----	Purpose -----
CTLCHR	CONTROL CHARACTERS
CTYPE	**** PURPOSE NOT FOUND BY STRIPPER ****
ERRPRO	PROCESS ERROR INCLUDE FILE
FFFV2	FORM FILE FORMAT - VERSION 2
FLAN.Y"	**** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE	FORM PROCESSOR RETURN CODES
FPD	FORM PROCESSOR DATA
FPDINI	FPD INITIALIZATION
FPPARM	FORM PROCESSOR PARAMETERS
MATH	**** PURPOSE NOT FOUND BY STRIPPER ****
NTM	NTM INTERFACE INCLUDE FILE
RW	REPORT WRITER DEFINITIONS
SRVRET	AS THE RETURN GIVEN A TABLE-FULL ERROR
STDIO	**** PURPOSE NOT FOUND BY STRIPPER ****
STDTP	STANDARD TYPE DEFINITIONS

3.10.5 Where Include File Used List

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

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
CTLCHR	ASSIGN	ASSIGN FILE SECTION
	CLSFIL	CLOSE FILES
	DATAGEN	DATA DIVISION GENERATE
	ENDGEN	END GERNERATE
	FD	FD SECTION DECLARATIONS
	FILELNK	FILE LINKAGE SECTION GENERATE
	INDENT	INDENT A LINE OF GENERATED CODE
	INSERT	INSERT PROCEDURE
	INSWC	INSERT WORKING STORAGE SECTION
	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	NDMLLAB	GENERATE LABELS
	NDMLLNK	LINKAGE SECTION
	NULBLK	BLANK FILL A STRING
	OPNFIL	GENERATE OPEN FILE SECTION
	PROCGEN	PROCEDURE DIVISION GENERATE
	SAVEES	SAVE ES INFORMATION
	SELECT	GENERATE SELECT CODE
	SELGEN	SELECT GENERATE
	SELLEN	COMPUTE LENGTH OF SELECT PS RECORD
	SELMAP	MAP SELECTED DATA TO OUTPUT RECORD
	SELWS	SELECT WORKING STORAGE SECTION
	STDCODE	STANDARD COBOL CODE
	USING	GENERATE USING SECTION
CTYPE	MAKACT	MAKE ACTION LIST ELEMENT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
ERRPRO	CDMESQY	PROGRAM NAME CDMESQY

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
FFV2		
	WRTFRM	WRITE FORM
	WRTFRM/DB	DEFAULT BUFFER CLOSE
	WRTFRM/FO	INSERT FORMAT CODES
	WRTFRM/TB	TFXT BUFFER CLOSE
	WRTFRM/WR	WRITE DEFAULT BUFFER
	WRTFRM/WR	WRITE FIELD
	WRTFRM/WR	WRITE TEXT BUFFER
	WRTFRM/WR	WRITE TEXT
FLAN.Y"		
	MAKACT	MAKE ACTION LIST ELEMENT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
FPCODE		
	ACTRSV	ACTION RESOLVE
	ADDCHK	ADD POSITION TO CHECK LIST
	ASSIGN	ASSIGN FILE SECTION
	CALCSTAT	CALCULATE STATISTIC
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CLSFIL	CLOSE FILES
	CSTASH	CHARACTER STASH
	CTLRSV	CONTROL RESOLVE
	DATAGEN	DATA DIVISION GENERATE
	ENDGEN	END GERNERATE
	FD	FD SECTION DECLARATIONS
	FILELNK	FILE LINKAGE SECTION GENERATE

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	FLANCI	FLAN CALLABLE INTERFACE
	FLDRSV	FIELD RESOLVE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE
	FNDFRM	FIND FORM
	GETCOL	GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
	GETPTH	GET PATH
	GETTBL	GET A TABLE NAME
	GFLDPT	GET FIELD POINTER
	INDENT	INDENT A LINE OF GENERATED CODE
	INSERT	INSERT PROCEDURE
	INSRSV	INSERT RESOLVE
	INSWS	INSERT WORKING STORAGE SECTION
	MAKINS	MAKE INSERT
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKPS	MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MAKWH	MAKE WHERE
	MAKWHES	MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
	MAKWHES/C	COBOL WHERE ES
	MAKWHES/C	C WHERE ES
	MAKWHES/N	NUMBER PICTURE CLAUSE
	MKPOS	MAKE POSITION NODE
	MLPFRM	MAKE A LIST OF PRESENTED FORMS
	MYALLOC	MY MALLOC
	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	NDMLLAB	GENERATE LABELS
	NDMLLNK	LINKAGE SECTION
	NULBLK	BLANK FILL A STRING
	OPNFIL	GENERATE OPEN FILE SECTION
	PROCGEN	PROCEDURE DIVISION GENERATE
	PSSTRC/CO	COBOL SUBSTITUTE
	PSSTRC/CS	C SUBSTITUTE
	PSSTRC/IN	INDENT
	RSETSTAT	RESET STATISTIC
	RWEXPD	REPORT WRITER EXPAND ARRAYS
	RWOPN	REPORT WRITER OPEN FORMS

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	RWSP/FIXF	FIX UP A FORM
	SAVEES	SAVE ES INFORMATION
	SELECT	GENERATE SELECT CODE
	SELGEN	SELECT GENERATE
	SELLEN	COMPUTE LENGTH OF SELECT PS RECORD
	SELMAP	MAP SELECTED DATA TO OUTPUT RECORD
	SELRSV	SELECT RESOLVE
	SELWS	SELECT WORKING STORAGE SECTION
	STATRSV	STATISTIC RESOLVE
	STDCODE	STANDARD COBOL CODE
	TRGRSV	TRIGGER RESOLVE
	UQPTH	UNIVERSAL QUALIFIER PATH
	USING	GENERATE USING SECTION
	WINRSV	WINDOW RESOLVE
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	WRTFRM/DB	DEFAULT BUFFER CLOSE
	WRTFRM/FO	INSERT FORMAT CODES
	WRTFRM/TB	TEXT BUFFER CLOSE
	WRTFRM/WR	WRITE DEFAULT BUFFER
	WRTFRM/WR	WRITE FIELD
	WRTFRM/WR	WRITE TEXT BUFFER
	WRTFRM/WR	WRITE TEXT

FPD

ACTRSV	ACTION RESOLVE
ADDCHK	ADD POSITION TO CHECK LIST
ASSIGN	ASSIGN FILE SECTION
BLDSUB	BUILD SUBROUTINES
BSCODE	BUILD SUBROUTINE CODE
CALCSTAT	CALCULATE STATISTIC
CCONV	C CONVERSIONS
CES	C ES
CESPS	C ES TO PS
CHKARY	CHECK ARRAY
CHKFLD	CHECK FIELD
CHKFRM	CHECK FORM

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	CHKGRP	CHECK FOR GROUP SEPERATORS OR END OF FILE
	CHKSIZE	CHECK SIZE OF ITEMS DOING CONVERSIONS ON
	CLRNDP	CLEAR NODUPLICATE FIELDS
	CLSFIL	CLOSE FILES
	COBCONV	COBOL CONVERSIONS
	COBES	COBOL ES RECORD
	COBESPS	COBOL ES TO PS
	COBPE	COBOL PE
	CPE	C PE
	CSTASH	CHARACTER STASH
	CTLRSV	CONTROL RESOLVE
	DASH	WRITE DASH '-'
	DATAGEN	DATA DIVISION GENERATE
	DBFREAD	GENERATE DATA BASE FREAD
	DCLINDX	DECLARE INDEX VARIABLES
	ENDGEN	END GERNERATE
	ESPSMAP	THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING
	ESPSMAP/I	INDENT
	FD	FD SECTION DECLARATIONS
	FILELNK	FILE LINKAGE SECTION GENERATE
	FLANCI	FLAN CALLABLE INTERFACE
	FLDRSV	FIELD RESOLVE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE
	FNDFRM	FIND FORM
	FRMPDAT	FORM PDATA
	GEN	GENERATE A LINE OF CODE
	GENAA	GENERATE PROCEDURE "ADDACT" ADD AN ACTION
	GENAAL	GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
	GENACT	GENERATE ACTIONS
	GENAE	GENERATE ACTION EXIT
	GENAH	GENERATE ACTION HELP
	GENAI	GENERATE ACTION INSERT
	GENAL	GENERATE ACTION LIST
	GENAP	GENERATE ACTION PAGE
	GENAQ	GENERATE ACTION QUERY (SELECT)
	GENAR	GENERATE ACTION PRESENT
	GENAS	GENERATE ACTION SET
	GENAT	GENERATE ACTION SIGNAL

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	GENBEG	GENERATE BEGINNING OF APPLICATION OR REPORT
	GENCHG	GENERATE CHANGE DECLARATIONS
	GENDB	GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
	GENDOA	GENERATE PROCEDURE "DOACT" DO ACTION
	GENDS	GENERATE DATA DATA STRUCTURES
	GENFP	GENERATE FORM PATH
	GENFS	GENERATE FORM DATA STRUCTURES
	GENFSD	GENERATE FORM STRUCTURE DATA INITIALIZATION
	GENINS	GENERATE INSERT DECLARATIONS
	GENMAIN	GENERATE MAIN PROGRAM
	GENNDP	GENERATE NODUPLICATE DECLARATIONS
	GENPAG	GENERATE NEWPAG PROCEDURE
	GENTAL	GENERATE TRIGGER ACTION LIST
	GENTRG	GENERATE TRIGGERS
	GETCOL	GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
	GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER
	GETPTH	GET PATH
	GETTBL	GET A TABLE NAME
	GFLDPT	GET FIELD POINTER
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	HASDATA	DETERMINE IF THERE ARE ANY SELECT STATEMENTS
	HASITEM	THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
	HASLOWER	HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
	INDENT	INDENT A LINE OF GENERATED CODE
	INSERT	INSERT PROCEDURE
	INSRSV	INSERT RESOLVE
	INSWS	INSERT WORKING STORAGE SECTION
	ISOPNE	DETERMINE IF THIS FIELD IS OPEN ENDED
	MAKACT	MAKE ACTION LIST ELEMENT
	MAKES	MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
	MAKES/CNU	C NUMBERS
	MAKES/IND	INDENT

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	MAKES/NUM	NUMBER PICTURE CLAUSE
	MAKINS	MAKE INSERT
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKPS	MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
	MAKQR	MAKE QUALIFIED REFERENCE
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MAKWH	MAKE WHERE
	MAKWHE	MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
	MAKWHE/C	COBOL WHERE ES
	MAKWHE/C	C WHERE ES
	MAKWHE/N	NUMBER PICTURE CLAUSE
	MAPDB	MAP DATABASE
	MKINC	MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
	MKPOS	MAKE POSITION NODE
	MLPFRM	MAKE A LIST OF PRESENTED FORMS
	MYALLO	MY MALLOC
	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	NDMLLAB	GENERATE LABELS
	NDMLLNK	LINKAGE SECTION
	NULBLK	BLANK FILL A STRING
	OPNFIL	GENERATE OPEN FILE SECTION
	PEMAP	THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING
	PROCEN	PROCEDURE DIVISION GENERATE
	PSSTRC/CO	COBOL SUBSTITUTE
	PSSTRC/CS	C SUBSTITUTE
	PSSTRC/IN	INDENT
	READDB	READ DATA BASE
	RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
	RSETSTAT	RESET STATISTIC
	RWEXPD	REPORT WRITER EXPAND ARRAYS
	RWOPN	REPORT WRITER OPEN FORMS
	RWSP/FIXF	FIX UP A FORM
	SAVEES	SAVE ES INFORMATION
	SELECT	GENERATE SELECT CODE
	SELGEN	SELECT GENERATE

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	SELLEN	COMPUTE LENGTH OF SELECT PS RECORD
	SELMAP	MAP SELECTED DATA TO OUTPUT RECORD
	SELOPN	SELECT OPEN
	SELRSV	SELECT RESOLVE
	SELWHR	SELECT WHERE
	SELWS	SELECT WORKING STORAGE SECTION
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	STATRSV	STATISTIC RESOLVE
	STDCODE	STANDARD COBOL CODE
	TRGRSV	TRIGGER RESOLVE
	UQFOR	UNIVERSAL QUALIFIER FOR LOOP
	UQPTH	UNIVERSAL QUALIFIER PATH
	USING	GENERATE USING SECTION
	VISITA	VISIT ARRAYS ON THIS FORM
	WINRSV	WINDOW RESOLVE
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	WRTFRM/DB	DEFAULT BUFFER CLOSE
	WRTFRM/FO	INSERT FORMAT CODES
	WRTFRM/TB	TEXT BUFFER CLOSE
	WRTFRM/WR	WRITE DEFAULT BUFFER
	WRTFRM/WR	WRITE FIELD
	WRTFRM/WR	WRITE TEXT BUFFER
	WRTFRM/WR	WRITE TEXT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
FPDINI	BLDSUB	BUILD SUBROUTINES
	BSCODE	BUILD SUBROUTINE CODE
	CHKGRP	CHECK FOR GROUP SEPERATORS OR END OF FILE
	CLRNDP	CLEAR NODUPLICATE FIELDS
	DBFREAD	GENERATE DATA BASE FREAD
	GEN	GENERATE A LINE OF CODE
	GENPAG	GENERATE NEWPAG PROCEDURE

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	HASDATA	DETERMINE IF THERE ARE ANY SELECT STATEMENTS
	HASITEM	THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
	HASLOWER	HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
	ISOPNE	DETERMINE IF THIS FIELD IS OPEN ENDED
	MAKQR	MAKE QUALIFIED REFERENCE
	MAPDB	MAP DATABASE
	READDB	READ DATA BASE
	RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	VISITA	VISIT ARRAYS ON THIS FORM

FPPARM

ASSIGN	ASSIGN FILE SECTION
BLDSUB	BUILD SUBROUTINES
BSCODE	BUILD SUBROUTINE CODE
CHKGRP	CHECK FOR GROUP SEPERATORS OR END OF FILE
CLRNDP	CLEAR NODUPLICATE FIELDS
CLSFIL	CLOSE FILES
DATAGEN	DATA DIVISION GENERATE
DBFREAD	GENERATE DATA BASE FREAD
ENDGEN	END GERNERATE
FD	FD SECTION DECLARATIONS
FILELNK	FILE LINKAGE SECTION GENERATE
FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
GEN	GENERATE A LINE OF CODE
GENMAIN	GENERATE MAIN PROGRAM
GENPAG	GENERATE NEWPAG PROCEDURE
GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
GRP/MAIN	HASDATA	GENERATE APPLICATION/REPORT PROGRAM DETERMINE IF THERE ARE ANY SELECT STATEMENTS
HASITEM		THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
HASLOWER		HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
INDENT		INDENT A LINE OF GENERATED CODE
INSERT		INSERT PROCEDURE
INSWS		INSERT WORKING STORAGE SECTION
ISOPNE		DETERMINE IF THIS FIELD IS OPEN ENDED
MAKACT		MAKE ACTION LIST ELEMENT
MAKINS		MAKE INSERT
MAKPS		MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
MAKQR		MAKE QUALIFIED REFERENCE
MAKWH		MAKE WHERE
MAKWHE		MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
MAKWHE/C		COBOL WHERE ES
MAKWHE/C		C WHERE ES
MAKWHE/N		NUMBER PICTURE CLAUSE
MAPDB		MAP DATABASE
NDMLGEN		NDML COBOL APPLICATION GENERATOR
NDMLLAB		GENERATE LABELS
NDMLLNK		LINKAGE SECTION
NULBLK		BLANK FILL A STRING
OPNFIL		GENERATE OPEN FILE SECTION
PROCGEN		PROCEDURE DIVISION GENERATE
PSSTRC/CO		COBOL SUBSTITUTE
PSSTRC/CS		C SUBSTITUTE
PSSTRC/IN		INDENT
READDB		READ DATA BASE
RSETNDP		RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
SAVEES		SAVE ES INFORMATION
SELECT		GENERATE SELECT CODE
SELGEN		SELECT GENERATE
SELLEN		COMPUTE LENGTH OF SELECT PS RECORD
SELMAP		MAP SELECTED DATA TO OUTPUT RECORD

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	SELWS	SELECT WORKING STORAGE SECTION
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	STDCODE	STANDARD COBOL CODE
	USING	GENERATE USING SECTION
	VISITA	VISIT ARRAYS ON THIS FORM
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
MATH	MAKACT	MAKE ACTION LIST ELEMENT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
NTM	ASSIGN	ASSIGN FILE SECTION
	BLDSUB	BUILD SUBROUTINES
	BSCODE	BUILD SUBROUTINE CODE
	CHKGRP	CHECK FOR GROUP SEPERATORS OR END OF FILE
	CLRNDP	CLEAR NODUPLICATE FIELDS
	CLSFIL	CLOSE FILES
	DATAGEN	DATA DIVISION GENERATE
	DBFREAD	GENERATE DATA BASE FREAD
	ENDGEN	END GERNERATE
	FD	FD SECTION DECLARATIONS
	FILELNK	FILE LINKAGE SECTION GENERATE
	FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
	GEN	GENERATE A LINE OF CODE
	GENPAG	GENERATE NEWPAG PROCEDURE
	GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	HASDATA	DETERMINE IF THERE ARE ANY SELECT STATEMENTS

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	HASITEM	THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
	HASLOWER	HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
	INDENT	INDENT A LINE OF GENERATED CODE
	INSERT	INSERT PROCEDURE
	INWS	INSERT WORKING STORAGE SECTION
	ISOPNE	DETERMINE IF THIS FIELD IS OPEN ENDED
	MAKQR	MAKE QUALIFIED REFERENCE
	MAPDB	MAP DATABASE
	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	NDMLLAB	GENERATE LABELS
	NDMLLNK	LINKAGE SECTION
	NULBLK	BLANK FILL A STRING
	OPNFIL	GENERATE OPEN FILE SECTION
	PROCGEN	PROCEDURE DIVISION GENERATE
	READDB	READ DATA BASE
	RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
	SAVEES	SAVE ES INFORMATION
	SELECT	GENERATE SELECT CODE
	SELGEN	SELECT GENERATE
	SELLEN	COMPUTE LENGTH OF SELECT PS RECORD
	SELMAP	MAP SELECTED DATA TO OUTPUT RECORD
	SELWS	SELECT WORKING STORAGE SECTION
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	STDCODE	STANDARD COBOL CODE
	USING	GENERATE USING SECTION
	VISITA	VISIT ARRAYS ON THIS FORM
RW		
	ACTRSV	ACTION RESOLVE
	ADDCHK	ADD POSITION TO CHECK LIST
	ASSIGN	ASSIGN FILE SECTION
	BLDSUB	BUILD SUBROUTINES
	BSCODE	BUILD SUBROUTINE CODE

APPLICATION GENERATOR Where Include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	CALCSTAT	CALCULATE STATISTIC
	CCONV	C CONVERSIONS
	CES	C ES
	CESPS	C ES TO PS
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CHKGRP	CHECK FOR GROUP SEPERATORS OR END OF FILE
	CHKSIZE	CHECK SIZE OF ITEMS DOING CONVERSIONS ON
	CLRNDP	CLEAR NODUPLICATE FIELDS
	CLSFIL	CLOSE FILES
	COBCONV	COBOL CONVERSIONS
	COBES	COBOL ES RECORD
	COBESPS	COBOL ES TO PS
	COBPE	COBOL PE
	CPE	C PE
	CSTASH	CHARACTER STASH
	CTLRSV	CONTROL RESOLVE
	DASH	WRITE DASH '-'
	DATAGEN	DATA DIVISION GENERATE
	DBFREAD	GENERATE DATA BASE FREAD
	DCLINDX	DECLARE INDEX VARIABLES
	ENDGEN	END GERNERATE
	ESPSMAP	THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING
	ESPSMAP/I	INDENT
	FD	FD SECTION DECLARATIONS
	FILELNK	FILE LINKAGE SECTION GENERATE
	FLANCI	FLAN CALLABLE INTERFACE
	FLDRSV	FIELD RESOLVE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE
	FNDFRM	FIND FORM
	FRMPDAT	FORM PDATA
	GEN	GENERATE A LINE OF CODE
	GENAA	GENERATE PROCEDURE "ADDACT" ADD AN ACTION
	GENAAL	GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
	GENACT	GENERATE ACTIONS
	GENAE	GENERATE ACTION EXIT
	GENAH	GENERATE ACTION HELP

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	GENAI	GENERATE ACTION INSERT
	GENAL	GENERATE ACTION LIST
	GENAP	GENERATE ACTION PAGE
	GENAQ	GENERATE ACTION QUERY (SELECT)
	GENAR	GENERATE ACTION PRESENT
	GENAS	GENERATE ACTION SET
	GENAT	GENERATE ACTION SIGNAL
	GENBEG	GENERATE BEGINNING OF APPLICATION OR REPORT
	GENCHG	GENERATE CHANGE DECLARATIONS
	GENDB	GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
	GENDOA	GENERATE PROCEDURE "DOACT" DO ACTION
	GENDS	GENERATE DATA DATA STRUCTURES
	GENFP	GENERATE FORM PATH
	GENFS	GENERATE FORM DATA STRUCTURES
	GENFSD	GENERATE FORM STRUCTURE DATA INITIALIZATION
	GENINS	GENERATE INSERT DECLARATIONS
	GENMAIN	GENERATE MAIN PROGRAM
	GENNDP	GENERATE NODUPLICATE DECLARATIONS
	GENPAG	GENERATE NEWPAG PROCEDURE
	GENTAL	GENERATE TRIGGER ACTION LIST
	GENTRG	GENERATE TRIGGERS
	GETCOL	GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
	GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER
	GETPTH	GET PATH
	GETTBL	GET A TABLE NAME
	GFLDPT	GET FIELD POINTER
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	HASDATA	DETERMINE IF THERE ARE ANY SELECT STATEMENTS
	HASITEM	THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
	HASLOWER	HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
	INDENT	INDENT A LINE OF GENERATED CODE
	INSERT	INSERT PROCEDURE

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	INSRSV	INSERT RESOLVE
	INSWS	INSERT WORKING STORAGE SECTION
	ISOPNE	DETERMINE IF THIS FIELD IS OPEN ENDED
	MAKACT	MAKE ACTION LIST ELEMENT
	MAKES	MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
	MAKES/CNU	C NUMBERS
	MAKES/IND	INDENT
	MAKES/NUM	NUMBER PICTURE CLAUSE
	MAKINS	MAKE INSERT
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKPS	MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
	MAKQR	MAKE QUALIFIED REFERENCE
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MAKWH	MAKE WHERE
	MAKWHEs	MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
	MAKWHEs/C	COBOL WHERE ES
	MAKWHEs/C	C WHERE ES
	MAKWHEs/N	NUMBER PICTURE CLAUSE
	MAPDB	MAP DATABASE
	MKINC	MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
	MKPOS	MAKE POSITION NODE
	MLPFRM	MAKE A LIST OF PRESENTED FORMS
	MYALLOc	MY MALLOC
	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	NDMLLAB	GENERATE LABELS
	NDMLLNK	LINKAGE SECTION
	NULBLK	BLANK FILL A STRING
	OPNFIL	GENERATE OPEN FILE SECTION
	PEMAP	THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING
	PROCEN	PROCEDURE DIVISION GENERATE
	PSSTRC/CO	COBOL SUBSTITUTE
	PSSTRC/CS	C SUBSTITUTE
	PSSTRC/IN	INDENT
	READDB	READ DATA BASE
	RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	RSETSTAT	RESET STATISTIC
	RWEXPD	REPORT WRITER EXPAND ARRAYS
	RWOPN	REPORT WRITER OPEN FORMS
	RWSP/FIXF	FIX UP A FORM
	SAVEES	SAVE ES INFORMATION
	SELECT	GENERATE SELECT CODE
	SELGEN	SELECT GENERATE
	SELLEN	COMPUTE LENGTH OF SELECT PS RECORD
	SELMAP	MAP SELECTED DATA TO OUTPUT RECORD
	SELOPN	SELECT OPEN
	SELRSV	SELECT RESOLVE
	SELWHR	SELECT WHERE
	SELWS	SELECT WORKING STORAGE SECTION
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	STATRSV	STATISTIC RESOLVE
	STDCODE	STANDARD COBOL CODE
	TRGRSV	TRIGGER RESOLVE
	UQFOR	UNIVERSAL QUALIFIER FOR LOOP
	UQPTH	UNIVERSAL QUALIFIER PATH
	USING	GENERATE USING SECTION
	VISITA	VISIT ARRAYS ON THIS FORM
	WINRSV	WINDOW RESOLVE
	WRTEXP	WRITE EXPRESSION
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
SRVRET	CDMESQY	PROGRAM NAME CDMESQY
STDIO	ADDCHK	ADD POSITION TO CHECK LIST
	ASSIGN	ASSIGN FILE SECTION
	BLDSUB	BUILD SUBROUTINES

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	BSCODE	BUILD SUBROUTINE CODE
	CCONV	C CONVERSIONS
	CES	C ES
	CESPS	C ES TO PS
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CHKGRP	CHECK FOR GROUP SEPERATORS OR END OF FILE
	CHKSIZE	CHECK SIZE OF ITEMS DOING CONVERSIONS ON
	CLRNDP	CLEAR NODUPLICATE FIELDS
	CLSFIL	CLOSE FILES
	COBCONV	COBOL CONVERSIONS
	COBES	COBOL ES RECORD
	COBESPS	COBOL ES TO PS
	COBPE	COBOL PE
	CPE	C PE
	CSTASH	CHARACTER STASH
	DASH	WRITE DASH '-'
	DATAGEN	DATA DIVISION GENERATE
	DBFREAD	GENERATE DATA BASE FREAD
	ENDGEN	END GERNERATE
	ESPSMAP	THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING
	ESPSMAP/I	INDENT
	FD	FD SECTION DECLARATIONS
	FILELNK	FILE LINKAGE SECTION GENERATE
	FLANCI	FLAN CALLABLE INTERFACE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE
	GEN	GENERATE A LINE OF CODE
	GENBEG	GENERATE BEGINNING OF APPLICATION OR REPORT
	GENCHG	GENERATE CHANGE DECLARATIONS
	GENDB	GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
	GENDS	GENERATE DATA DATA STRUCTURES
	GENFP	GENERATE FORM PATH
	GENFS	GENERATE FORM DATA STRUCTURES
	GENFSD	GENERATE FORM STRUCTURE DATA INITIALIZATION

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	GENINS	GENERATE INSERT DECLARATIONS
	GENMAIN	GENERATE MAIN PROGRAM
	GENNDP	GENERATE NODUPLICATE DECLARATIONS
	GENPAG	GENERATE NEWPAG PROCEDURE
	GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER
	GFLDPT	GET FIELD POINTER
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	HASDATA	DETERMINE IF THERE ARE ANY SELECT STATEMENTS
	HASITEM	THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
	HASLOWER	HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
	INDENT	INDENT A LINE OF GENERATED CODE
	INSERT	INSERT PROCEDURE
	INSWS	INSERT WORKING STORAGE SECTION
	ISOPNE	DETERMINE IF THIS FIELD IS OPEN ENDED
	MAKACT	MAKE ACTION LIST ELEMENT
	MAKES	MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
	MAKES/CNU	C NUMBERS
	MAKES/IND	INDENT
	MAKES/NUM	NUMBER PICTURE CLAUSE
	MAKINS	MAKE INSERT
	MAKINT	MAKE EXPRESSION INTO AN INTEGER
	MAKPS	MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
	MAKQR	MAKE QUALIFIED REFERENCE
	MAKSTR	MAKE EXPRESSION INTO A STRING
	MAKWH	MAKE WHERE
	MAKWHE	MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
	MAKWHE/C	COBOL WHERE ES
	MAKWHE/C	C WHERE ES
	MAKWHE/N	NUMBER PICTURE CLAUSE
	MAPDB	MAP DATABASE
	MKINC	MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
	MKPOS	MAKE POSITION NODE
	MYALLOC	MY MALLOC

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	NDMLLAB	GENERATE LABELS
	NDMLLNK	LINKAGE SECTION
	NULBLK	BLANK FILL A STRING
	OPNFIL	GENERATE OPEN FILE SECTION
	FEMAP	THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING
	PROCGEN	PROCEDURE DIVISION GENERATE
	PSSTRC/CO	COBOL SUBSTITUTE
	PSSTRC/CS	C SUBSTITUTE
	PSSTRC/IN	INDENT
	READDB	READ DATA BASE
	RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
	SAVEES	SAVE ES INFORMATION
	SELECT	GENERATE SELECT CODE
	SELGEN	SELECT GENERATE
	SELLEN	COMPUTE LENGTH OF SELECT PS RECORD
	SELMAP	MAP SELECTED DATA TO OUTPUT RECORD
	SELWS	SELECT WORKING STORAGE SECTION
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	STDCODE	STANDARD COBOL CODE
	USING	GENERATE USING SECTION
	VISITA	VISIT ARRAYS ON THIS FORM
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	WRTFRM/DB	DEFAULT BUFFER CLOSE
	WRTFRM/FO	INSERT FORMAT CODES
	WRTFRM/TB	TEXT BUFFER CLOSE
	WRTFRM/WR	WRITE DEFAULT BUFFER
	WRTFRM/WR	WRITE FIELD
	WRTFRM/WR	WRITE TEXT BUFFER
	WRTFRM/WR	WRITE TEXT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
STD TYP		
	ACTRSV	ACTION RESOLVE
	ADDCHK	ADD POSITION TO CHECK LIST
	ASSIGN	ASSIGN FILE SECTION
	BLDSUB	BUILD SUBROUTINES
	BSCODE	BUILD SUBROUTINE CODE
	CALCSTAT	CALCULATE STATISTIC
	CCONV	C CONVERSIONS
	CES	C ES
	CESPS	C ES TO PS
	CHKARY	CHECK ARRAY
	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CHKGRP	CHECK FOR GROUP SEPERATORS OR END OF FILE
	CHKSIZE	CHECK SIZE OF ITEMS DOING CONVERSIONS ON
	CLRNDP	CLEAR NODUPLICATE FIELDS
	CLSFIL	CLOSE FILES
	COBCONV	COBOL CONVERSIONS
	COBES	COBOL ES RECORD
	COBESPS	COBOL ES TO PS
	COBPE	COBOL PE
	CPE	C PE
	CSTASH	CHARACTER STASH
	CTLRSV	CONTROL RESOLVE
	DASH	WRITE DASH '-'
	DATAGEN	DATA DIVISION GENERATE
	DBFREAD	GENERATE DATA BASE FREAD
	DCLINDX	DECLARE INDEX VARIABLES
	ENDGEN	END GERNERATE
	ERROR	ISSUE ERROR MESSAGE
	ESPSMAP	THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING
	ESPSMAP/I	INDENT
	FATAL	ISSUE FATAL ERROR MESSAGE
	FD	FD SECTION DECLARATIONS
	FILELNK	FILE LINKAGE SECTION GENERATE
	FLANCI	FLAN CALLABLE INTERFACE
	FLDRSV	FIELD RESOLVE
	FLDTYP	FIELD TYPE
	FNDATT	FIND ATTRIBUTE

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	FNDFRM	FIND FORM
	FRMPDAT	FORM PDATA
	FRNTND	FORMS FRONT END TO APPLICATIC GENERATOR
	GEN	GENERATE A LINE OF CODE
	GENAA	GENERATE PROCEDURE "ADDACT" ADD AN ACTION
	GENAAL	GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
	GENACT	GENERATE ACTIONS
	GENAE	GENERATE ACTION EXIT
	GENAH	GENERATE ACTION HELP
	GENAI	GENERATE ACTION INSERT
	GENAL	GENERATE ACTION LIST
	GENAP	GENERATE ACTION PAGE
	GENAQ	GENERATE ACTION QUERY (SELECT)
	GENAR	GENERATE ACTION PRESENT
	GENAS	GENERATE ACTION SET
	GENAT	GENERATE ACTION SIGNAL
	GENBEG	GENERATE BEGINNING OF APPLICATION OR REPORT
	GENCHG	GENERATE CHANGE DECLARATIONS
	GENDB	GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
	GENDOA	GENERATE PROCEDURE "DOACT" DO ACTION
	GENDS	GENERATE DATA DATA STRUCTURES
	GENFP	GENERATE FORM PATH
	GENFS	GENERATE FORM DATA STRUCTURES
	GENFSD	GENERATE FORM STRUCTURE DATA INITIALIZATION
	GENINS	GENERATE INSERT DECLARATIONS
	GENMAIN	GENERATE MAIN PROGRAM
	GENNDP	GENERATE NODUPLICATE DECLARATIONS
	GENPAG	GENERATE NEWPAG PROCEDURE
	GENTAL	GENERATE TRIGGER ACTION LIST
	GENTRG	GENERATE TRIGGERS
	GETCOL	GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
	GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER
	GETPTH	GET PATH
	GETTBL	GET A TABLE NAME
	GFLDPT	GET FIELD POINTER

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
GRP/MAIN		GENERATE APPLICATION/REPORT PROGRAM
HASDATA		DETERMINE IF THERE ARE ANY SELECT STATEMENTS
HASITEM		THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
HASLOWER		HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
INDENT		INDENT A LINE OF GENERATED CODE
INSERT		INSERT PROCEDURE
INSRSV		INSERT RESOLVE
INSWS		INSERT WORKING STORAGE SECTION
ISOPNE		DETERMINE IF THIS FIELD IS OPEN ENDED
MAKACT		MAKE ACTION LIST ELEMENT
MAKES		MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
MAKES/CNU		C NUMBERS
MAKES/IND		INDENT
MAKES/NUM		NUMBER PICTURE CLAUSE
MAKINS		MAKE INSERT
MAKINT		MAKE EXPRESSION INTO AN INTEGER
MAKPS		MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
MAKQR		MAKE QUALIFIED REFERENCE
MAKSTR		MAKE EXPRESSION INTO A STRING
MAKWH		MAKE WHERE
MAKWHE		MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
MAKWHE/C		COBOL WHERE ES
MAKWHE/C		C WHERE ES
MAKWHE/N		NUMBER PICTURE CLAUSE
MAPDB		MAP DATABASE
MKINC		MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
MKPOS		MAKE POSITION NODE
MLPFRM		MAKE A LIST OF PRESENTED FORMS
MYALLOC		MY MALLOC
NDMLGEN		NDML COBOL APPLICATION GENERATOR
NDMLLAB		GENERATE LABELS
NDMLLNK		LINKAGE SECTION
NULBLK		BLANK FILL A STRING
OPNFIL		GENERATE OPEN FILE SECTION

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	PEMAP	THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING
	PROCGEN	PROCEDURE DIVISION GENERATE
	PSSTRC/CO	COBOL SUBSTITUTE
	PSSTRC/CS	C SUBSTITUTE
	PSSTRC/IN	INDENT
	READDB	READ DATA BASE
	RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
	RSETSTAT	RESET STATISTIC
	RWEXPD	REPORT WRITER EXPAND ARRAYS
	RWOPN	REPORT WRITER OPEN FORMS
	RWSP/FIXF	FIX UP A FORM
	SAVEES	SAVE ES INFORMATION
	SELECT	GENERATE SELECT CODE
	SELGEN	SELECT GENERATE
	SELLEN	COMPUTE LENGTH OF SELECT PS RECORD
	SELMAP	MAP SELECTED DATA TO OUTPUT RECORD
	SELOPN	SELECT OPEN
	SELRSV	SELECT RESOLVE
	SELWHR	SELECT WHERE
	SELWS	SELECT WORKING STORAGE SECTION
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	STATRSV	STATISTIC RESOLVE
	STDCODE	STANDARD COBOL CODE
	TRGRSV	TRIGGER RESOLVE
	UQFOR	UNIVERSAL QUALIFIER FOR LOOP
	UQPTH	UNIVERSAL QUALIFIER PATH
	USING	GENERATE USING SECTION
	VISITA	VISIT ARRAYS ON THIS FORM
	WARNING	ISSUE WARNING MESSAGE
	WINRSV	WINDOW RESOLVE
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	WRTFRM/DB	DEFAULT BUFFER CLOSE
	WRTFRM/FO	INSERT FORMAT CODES
	WRTFRM/TB	TEXT BUFFER CLOSE
	WRTFRM/WR	WRITE DEFAULT BUFFER
	WRTFRM/WR	WRITE FIELD

APPLICATION GENERATOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	WRTFRM/WR	WRITE TEXT BUFFER
	WRTFRM/WR	WRITE TEXT
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER

3.10.6 Where External Routine Used List

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

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
ABS	CHKARY CHKFRM CHKFRM RWEXPD	CHECK ARRAY CHECK FORM CHECK FORM REPORT WRITER. EXPAND ARRAYS
ADDFRM	FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
ATOF	YYLEX	LEXICAL ANALYZER FOR FLAN
ATOI	CCONV CES COBCONV COBES SAVEES YYLEX	C CONVERSIONS C ES COBOL CONVERSIONS COBOL ES RECORD SAVE ES INFORMATION LEXICAL ANALYZER FOR FLAN
BLFN	CHKFLD CHKSIZE MKINC PSSTRC/COBCOBOL PSSTRC/CSUC SELLEN	CHECK FIELD CHECK SIZE OF ITEMS DOING CONVERSIONS ON MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC) SUBSTITUTE SUBSTITUTE COMPUTE LENGTH OF SELECT PS RECORD
CALLOC		

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
COPFLD	RWEXPD	REPORT WRITER EXPAND ARRAYS
	RWSP/FIXFR	FIX UP A FORM
	WINRSV	WINDOW RESOLVE
DELFLD	FLANCI	FLAN CALLABLE INTERFACE
ERRPRO	CDMESQY	PROGRAM NAME CDMESQY
ESCPY	CCONV	C CONVERSIONS
	CES	C ES
	COBCONV	COBOL CONVERSIONS
	COBES	COBOL ES RECORD
	GETTBL	GET A TABLE NAME
	SAVEES	SAVE ES INFORMATION
FCLOSE	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	WRTFRM	WRITE FORM
FOPEN	GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	WRTFRM	WRITE FORM

FPRINTF

ASSIGN	ASSIGN FILE SECTION
CCONV	C CONVERSIONS
CES	C ES
CHKSIZE	CHECK SIZE OF ITEMS DOING CONVERSIONS ON
CLSFIL	CLOSE FILES
COBCONV	COBOL CONVERSIONS
COBES	COBOL ES RECORD
COBESPS	COBOL ES TO PS
CORPE	COBOL PE
DATAGEN	DATA DIVISION GENERATE
ENDGEN	END GERNERATE
FD	FD SECTION DECLARATIONS
FILELNK	FILE LINKAGE SECTION GENERATE
GEN	GENERATE A LINE OF CODE
INSERT	INSERT PROCEDURE
INSWS	INSERT WORKING STORAGE SECTION
MAKES/CNUMC	NUMBERS
MAKES/NUMPNUMBER	PICTURE CLAUSE
MAKINS	MAKE INSERT
MAKWH	MAKE WHERE
MAKWHE	MAKE THE WHERE CLAUSE EXTERNAL SCHEMA
	VARIABLES
MAKWHE/COCOBOL	WHERE ES
MAKWHE/NUNUMBER	PICTURE CLAUSE
NDMLLAB	GENERATE LABELS
OPNFIL	GENERATE OPEN FILE SECTION
PROCEN	PROCEDURE DIVISION GENERATE
PSSTRC/COBCOBOL	SUBSTITUTE
PSSTRC/CSUC	SUBSTITUTE
SELGEN	SELECT GENERATE
SELWS	SELECT WORKING STORAGE SECTION
STDCODE	STANDARD COBOL CODE
USING	GENERATE USING SECTION

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
FREE	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	WINRSV	WINDOW RESOLVE
	WRTEXP	WRITE EXPRESSION
	YYPARSE	FLAN PARSE
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 TO APPLICATION GENERATOR
GETC	YYLEX	LEXICAL ANALYZER FOR FLAN
INITAL	FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
INITFP	FRNTND	FORMS FRONT END TO APPLICATION GENERATOR

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
INSMAP	PROCGEN	PROCEDURE DIVISION GENERATE
ISALNUM	YYLEX	LEXICAL ANALYZER FOR FLAN
ISALPHA	YYLEX	LEXICAL ANALYZER FOR FLAN
ISDIGIT	YYLEX	LEXICAL ANALYZER FOR FLAN
ISSPACE	YYLEX	LEXICAL ANALYZER FOR FLAN
MAKFLD	YYPARSE	FLAN PARSER
MALLOC	MLPFRM MYALLOC UQPTH WINRSV	MAKE A LIST OF PRESENTED FORMS MY MALLOC UNIVERSAL QUALIFIER PATH WINDOW RESOLVE

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
MAP	PROCGEN	PROCEDURE DIVISION GENERATE
MAX	CHKFLD CHKFRM DCLINDX	CHECK FIELD CHECK FORM DECLARE INDEX VARIABLES
MEMCMP	FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
MEMCPY	CHKFLD WRTEXP WRTFRM/WRT YYPARSE	CHECK FIELD WRITE EXPRESSION WRITE FIELD FLAN PARSE
MEMSET	CHKFLD	CHECK FIELD
OISCR	FRNTND GRP/MAIN	FORMS FRONT END TO APPLICATION GENERATOR GENERATE APPLICATION/REPORT PROGRAM
PMSGLC		

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	FRNTND GRP/MAIN	FORMS FRONT END TO APPLICATION GENERATOR GENERATE APPLICATION/REPORT PROGRAM
PMSGLS	ERROR FATAL WARNING	ISSUE ERROR MESSAGE ISSUE FATAL ERROR MESSAGE ISSUE WARNING MESSAGE
PRINTF	YYPARSE	FLAN PARSER
PSESMAP	PROCEN	PROCEDURE DIVISION GENERATE
PTHPTR	GETPTH UQPTH	GET PATH UNIVERSAL QUALIFIER PATH
PUTC	ESPSMAP/ININDENT INDENT MAKES/INDEINDENT PSSTRC/INDINDENT	INDENT A LINE OF GENERATED CODE
SPRINTF	BSCODE CALCSTAT CHKGRP	BUILD SUBROUTINE CODE CALCULATE STATISTIC CHECK FOR GROUP SEPERATORS OR END OF FILE

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	CLRNDP	CLEAR NODUPLICATE FIELDS
	DBFREAD	GENERATE DATA BASE FREAD
	DCLINDX	DECLARE INDEX VARIABLES
	ERROR	ISSUE ERROR MESSAGE
	FATAL	ISSUE FATAL ERROR MESSAGE
	FRMPDAT	FORM PDATA
	FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
	GENAAL	GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
	GENAH	GENERATE ACTION HELP
	GENAI	GENERATE ACTION INSERT
	GENAP	GENERATE ACTION PAGE
	GENAQ	GENERATE ACTION QUERY (SELECT)
	GENAR	GENERATE ACTION PRESENT
	GENAS	GENERATE ACTION SET
	GENAT	GENERATE ACTION SIGNAL
	GENBEG	GENERATE BEGINNING OF APPLICATION OR REPORT
	GENCHG	GENERATE CHANGE DECLARATIONS
	GENDB	GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
	GENDOA	GENERATE PROCEDURE "DOACT" DO ACTION
	GENDS	GENERATE DATA DATA STRUCTURES
	GENFP	GENERATE FORM PATH
	GENFS	GENERATE FORM DATA STRUCTURES
	GENFSD	GENERATE FORM STRUCTURE DATA INITIALIZATION
	GENINS	GENERATE INSERT DECLARATIONS
	GENMAIN	GENERATE MAIN PROGRAM
	GENNDP	GENERATE NODUPLICATE DECLARATIONS
	GENTRG	GENERATE TRIGGERS
	GETFILE	RETURN A FILE POINTER BASED ON INPUT FROM THE USER
	MAKQR	MAKE QUALIFIED REFERENCE
	MAPDB	MAP DATABASE
	MKINC	MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
	NDMLGEN	NDML COBOL APPLICATION GENERATOR
	RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
	RSETSTAT	RESET STATISTIC

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	SELGEN	SELECT GENERATE
	SELOPN	SELECT OPEN
	SELWHR	SELECT WHERE
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	STDCODE	STANDARD COBOL CODE
	UQFOR	UNIVERSAL QUALIFIER FOR LOOP
	VISITA	VISIT ARRAYS ON THIS FORM
	WARNING	ISSUE WARNING MESSAGE
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	YYPARSE	FLAN PARSER
STRASN	CHKARY	CHECK ARRAY
	CHKFRM	CHECK FORM
	RWEXPD	REPORT WRITER EXPAND ARRAYS
	WRTFRM	WRITE FORM
STRCAT	DCLINDX	DECLARE INDEX VARIABLES
	GENTRG	GENERATE TRIGGERS
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	MAKES/CNUMC	NUMBERS
	MAKQR	MAKE QUALIFIED REFERENCE
	YYPARSE	FLAN PARSER
STRCHR	DASH	WRITE DASH '-'
	FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
	GENDOA	GENERATE PROCEDURE "DOACT" DO ACTION
	GENMAIN	GENERATE MAIN PROGRAM
	GENTRG	GENERATE TRIGGERS

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	GETCOL	GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
	GETPTH	GET PATH
	GETTBL	GET A TABLE NAME
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	MAPDB	MAP DATABASE
	NULBLK	BLANK FILL A STRING
	UQPTH	UNIVERSAL QUALIFIER PATH
	YYPARSE	FLAN PARSER
STRCMP	FNDATT	FIND ATTRIBUTE
	FNDFRM	FIND FORM
	GENAR	GENERATE ACTION PRESENT
	GENTRG	GENERATE TRIGGERS
	GETTBL	GET A TABLE NAME
	GFLDPT	GET FIELD POINTER
	RWSP/FIXFR	FIX UP A FORM
	SELWS	SELECT WORKING STORAGE SECTION
	YYLEX	LEXICAL ANALYZER FOR FLAN
	YYPARSE	FLAN PARSER
SIRCPY	CSTASH	CHARACTER STASH
	DCLINDX	DECLARE INDEX VARIABLES
	GENAS	GENERATE ACTION SET
	GENTRG	GENERATE TRIGGERS
	GETCOL	GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
	GETPTH	GET PATH
	GETTBL	GET A TABLE NAME
	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
	INSWS	INSERT WORKING STORAGE SECTION
	MAKES	MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
	MAKQR	MAKE QUALIFIED REFERENCE
	NULBLK	BLANK FILL A STRING

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	SELGEN	SELECT GENERATE
	SELWS	SELECT WORKING STORAGE SECTION
	UQPTH	UNIVERSAL QUALIFIER PATH
	WRTFRM	WRITE FORM
	WRTFRM/WRTWRITE	FIELD
	YYPARSE	FLAN PARSER
STRLEN	CHKFLD	CHECK FIELD
	CHKFRM	CHECK FORM
	CSTASH	CHARACTER STASH
	DCLINDX	DECLARE INDEX VARIABLES
	ERROR	ISSUE ERROR MESSAGE
	FATAL	ISSUE FATAL ERROR MESSAGE
	GENAS	GENERATE ACTION SET
	GENFSD	GENERATE FORM STRUCTURE DATA INITIALIZATION
	GENTRG	GENERATE TRIGGERS
	MAKES	MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
	MAPDB	MAP DATABASE
	MKINC	MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
	RSETNDP	RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
	SAVEES	SAVE ES INFORMATION
	SETNDP	SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
	VISITA	VISIT ARRAYS ON THIS FORM
	WARNING	ISSUE WARNING MESSAGE
	WRTEXP	WRITE EXPRESSION
	WRTFRM	WRITE FORM
	WRTFRM/WRTWRITE	TEXT
	YYPARSE	FLAN PARSER
STRNCMP	CCONV	C CONVERSIONS

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	SAVEES	SAVE ES INFORMATION
STRNCPY	MAKES NDMLGEN WRTFRM/WRTWRITE FIELD YYPARSE	MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE NDML COBOL APPLICATION GENERATOR FLAN PARSE
STRSPN	GENAS	GENERATE ACTION SET
STRUPC	GETPTH STDCODE YYPARSE	GET PATH STANDARD COBOL CODE FLAN PARSE
SYSMSG	CHKFLD NDMLGEN WRTFRM	CHECK FIELD NDML COBOL APPLICATION GENERATOR WRITE FORM
TERMFP	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
TOUPPER	YYLEX	LEXICAL ANALYZER FOR FLAN

APPLICATION GENERATOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
TRMNAT	FRNTND	FORMS FRONT END TO APPLICATION GENERATOR
TRMNDML	GRP/MAIN	GENERATE APPLICATION/REPORT PROGRAM
UNGETC	YYLEX	LEXICAL ANALYZER FOR FLAN
YYERROR	YYPARSE	FLAN PARSER

3.10.7 Main Program Parts List

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

APPLICATION GENERATOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
GRP/MAIN	Purpose-->	GENERATE APPLICATION/REPORT PROGRAM
	ABS	External routine
	ACTRSV	Well-defined module
	ADDCHK	Well-defined module
	ADDFRM	External routine
	ASSIGN	Well-defined module
	ATOF	External routine
	ATOI	External routine
	BLDSUB	Well-defined module
	BLFN	External routine
	BSCODE	Well-defined module
	CALCSTAT	Well-defined module
	CALLOC	External routine
	CCONV	Well-defined module
	CDMESQY	Well-defined module
	CES	Well-defined module
	CESPS	Well-defined module
	CHKARY	Well-defined module
	CHKFLD	Well-defined module
	CHKFRM	Well-defined module
	CHKGRP	Well-defined module
	CHKSIZE	Well-defined module
	CLRNDP	Well-defined module
	CLSFIL	Well-defined module
	COBCONV	Well-defined module
	COBES	Well-defined module
	COBESPS	Well-defined module
	COBPE	Well-defined module
	COPFLD	External routine
	CPE	Well-defined module
	CSTASH	Well-defined module
	CTLRSV	Well-defined module
	DASH	Well-defined module
	DATAGEN	Well-defined module
	DBFREAR	Well-defined module
	DCLINDX	Well-defined module
	DELFLD	External routine
	ENDGEN	Well-defined module
	ERROR	Well-defined module

APPLICATION GENERATOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	ERRPRO	External routine
	ESCPY	External routine
	ESPSMAP	Well-defined module
	ESPSMAP/INDENT	Well-defined module
	FATAL	Well-defined module
	FCLOSE	External routine
	FD	Well-defined module
	FILELNK	Well-defined module
	FLANCI	Well-defined module
	FLDRSV	Well-defined module
	FLDTYP	Well-defined module
	FNDATT	Well-defined module
	FNDFRM	Well-defined module
	FOPEN	External routine
	FPRINTF	External routine
	FREE	External routine
	FRMPDAT	Well-defined module
	FRNTND	Well-defined module
	FWRITE	External routine
	GDATA	External routine
	GEN	Well-defined module
	GENAA	Well-defined module
	GENAAL	Well-defined module
	GENACT	Well-defined module
	GENAE	Well-defined module
	GENAH	Well-defined module
	GENAI	Well-defined module
	GENAL	Well-defined module
	GENAP	Well-defined module
	GENAQ	Well-defined module
	GENAR	Well-defined module
	GENAS	Well-defined module
	GENAT	Well-defined module
	GENBEG	Well-defined module
	GENCHG	Well-defined module
	GENDB	Well-defined module
	GENDOA	Well-defined module
	GENDS	Well-defined module
	GENFP	Well-defined module
	GENFS	Well-defined module

APPLICATION GENERATOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	GENFSD	Well-defined module
	GENINS	Well-defined module
	GENMAIN	Well-defined module
	GENNDP	Well-defined module
	GENPAG	Well-defined module
	GENTAL	Well-defined module
	GENTRG	Well-defined module
	GETC	External routine
	GETCOL	Well-defined module
	GETFILE	Well-defined module
	GETPTH	Well-defined module
	GETTBL	Well-defined module
	GFLDPT	Well-defined module
	HASDATA	Well-defined module
	HASITEM	Well-defined module
	HASLOWER	Well-defined module
	INDENT	Well-defined module
	INITAL	External routine
	INITFP	External routine
	INSERT	Well-defined module
	INSMAP	External routine
	INSRSV	Well-defined module
	INSWS	Well-defined module
	ISALNUM	External routine
	ISALPHA	External routine
	ISDIGIT	External routine
	ISOPNE	Well-defined module
	ISSPACE	External routine
	MAKACT	Well-defined module
	MAKES	Well-defined module
	MAKES/CNUMPIC	Well-defined module
	MAKES/INDENT	Well-defined module
	MAKES/NUMPIC	Well-defined module
	MAKFLD	External routine
	MAKINS	Well-defined module
	MAKINT	Well-defined module
	MAKPS	Well-defined module
	MAKQR	Well-defined module
	MAKSTR	Well-defined module
	MAKWH	Well-defined module

APPLICATION GENERATOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	MAKWHES	Well-defined module
	MAKWHES/COBWHES	Well-defined module
	MAKWHES/CWHES	Well-defined module
	MAKWHES/NUMPIC	Well-defined module
	MALLOC	External routine
	MAP	External routine
	MAPDB	Well-defined module
	MAX	External routine
	MEMCMP	External routine
	MEMCPY	External routine
	MEMSET	External routine
	MKINC	Well-defined module
	MKPOS	Well-defined module
	MLPFRM	Well-defined module
	MYALLOC	Well-defined module
	NDMLGEN	Well-defined module
	NDMLLAB	Well-defined module
	NDMLLNK	Well-defined module
	NULBLK	Well-defined module
	OISCR	External routine
	OPNFIL	Well-defined module
	PEMAP	Well-defined module
	PMSGLC	External routine
	PMSGLS	External routine
	PRINTF	External routine
	PROCGEN	Well-defined module
	PSESMAP	External routine
	PSSTRC/COBSUB	Well-defined module
	PSSTRC/CSUB	Well-defined module
	PSSTRC/INDENT	Well-defined module
	PTHPTR	External routine
	PUTC	External routine
	READDB	Well-defined module
	RSETNDP	Well-defined module
	RSETSTAT	Well-defined module
	RWEXPD	Well-defined module
	RWOPN	Well-defined module
	RWSP/FIXFRM	Well-defined module
	SAVEES	Well-defined module
	SELECT	Well-defined module

APPLICATION GENERATOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	SELGEN	Well-defined module
	SELLEN	Well-defined module
	SELMAP	Well-defined module
	SELOPN	Well-defined module
	SELRSV	Well-defined module
	SELWHR	Well-defined module
	SELWS	Well-defined module
	SETNDP	Well-defined module
	SPRINTF	External routine
	STATRSV	Well-defined module
	STDCODE	Well-defined module
	STRASN	External routine
	STRCAT	External routine
	STRCHR	External routine
	STRCMP	External routine
	STRCPY	External routine
	STRLEN	External routine
	STRNCMP	External routine
	STRNCPY	External routine
	STRSPN	External routine
	STRUPC	External routine
	SYSMSG	External routine
	TERMFP	External routine
	TOUPPER	External routine
	TRGRSV	Well-defined module
	TRMNAT	External routine
	TRMNDML	External routine
	UNGETC	External routine
	UQFOR	Well-defined module
	UQPTH	Well-defined module
	USING	Well-defined module
	VISITA	Well-defined module
	WARNING	Well-defined module
	WINRSV	Well-defined module
	WRTEXP	Well-defined module
	WRTFRM	Well-defined module
	WRTFRM/DBFCLOS	Well-defined module
	WRTFRM/FORMAT	Well-defined module
	WRTFRM/TBFCLOS	Well-defined module
	WRTFRM/WRTDBF	Well-defined module

APPLICATION GENERATOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	WRTFRM/WRTFLD	Well-defined module
	WRTFRM/WRTTBF	Well-defined module
	WRTFRM/WRTTXT	Well-defined module
	YYERROR	External routine
	YYLEX	Well-defined module
	YYPARSE	Well-defined module

3.10.8 Module Documentation

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

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

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

ROUTINES CALLED: Subroutines or Functions, either documented or external, called by this module, if any.

CALLED DIRECTLY BY: The documented routines which call this module, if any.

USED IN MAIN PROGRAM(S): The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.

APPLICATION GENERATOR Module Documentation

NAME: ACTRSV
PURPOSE: ACTION RESOLVE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

ACTRSV(ACTPTR, TRGPTR)
 ACTLST *ACTPTR;
 TRGLST *TRGPTR;

INPUTS:

ACTPTR - ACTION LIST FROM WHICH TO LOOK FOR PATHS.
TRGPTR - TRIGGER ASSOCIATED WITH THIS ACTION.

DESCRIPTION

RESOLVES ALL QUALIFIED NAMES INTO FIELD POINTERS FOR ALL
 NAMES
WHICH ARE ROOTED IN ACTLST (ACTION LIST).

ARGUMENTS:

ACTPTR = ACTLST *
TRGPTR = TRGLST *

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GEIPTH - GET PATH
ERROR - ISSUE ERROR MESSAGE
SELRSV - SELECT RESOLVE
INSRSV - INSERT RESOLVE
UQPTH - UNIVERSAL QUALIFIER PATH

CALLFD DIRECTLY BY:

TRGRSV - TRIGGER RESOLVE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

----- SYNOPSIS

VOID ADDCHK(POSPTR)
POS *POSPTR;

DESCRIPTION

ADDS THE SPECIFIED POSITION TO THE OVERLAP CHECK LIST

ARGUMENTS:

POSPTR = POS *

INCLUDE FILES:

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

ROUTINES CALLED:

FLDTyp - FIELD TYPE
ERROR - ISSUE ERROR MESSAGE

CALLED DIRECTLY BY:

CHKFRM - CHECK FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: ASSIGN
PURPOSE: ASSIGN FILE SECTION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SPTR = SELECT *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

FPRINTF
INDENT - INDENT A LINE OF GENERATED CODE

CALLED DIRECTLY BY:

DATAGEN - DATA DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: BLDSUB
PURPOSE: BUILD SUBROUTINES
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

BLDSUB (DP)
FIELD *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
(DP) - FIELD POINTER

OUTPUTS:
NONE

DESCRIPTION

THIS ROUTINE TRAVERSES THE FORMS HIERARCHY LOOKING FOR
FORMS
WHICH HAVE A SELECT WHICH TARGETS TO ITEMS ON THE FORM OR
ONE
OF ITS SUBFORMS. WHEN IT FINDS ONE IT CALLS BSCODE WHICH
GENERATES A FORM PROCEDURE.

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
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

HASDATA	- DETERMINE IF THERE ARE ANY SELECT STATEMENTS
BSCODE	- BUILD SUBROUTINE CODE
BLDSUB	- BUILD SUBROUTINES

CALLED DIRECTLY BY:

GRP/MAIN	- GENERATE APPLICATION/REPORT PROGRAM
BLDSUB	- BUILD SUBROUTINES

USED IN MAIN PROGRAM(S):

GRP/MAIN	- GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: BSCODE
PURPOSE: BUILD SUBROUTINE CODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

BSCODE(DP)
FIELD *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
(DP) - FIELD POINTER

OUTPUTS:
NONE

DESCRIPTION

THE FIELD POINTER WHICH IS PASSED TO THIS ROUTINE IS
POINTING
TO A FIELD WHOSE CONTENTS ARE A FORM. THIS ROUTINE
GENERATES THE
CODE FOR A SUBROUTINE THAT CORRESPONDS TO THE NAME OF THAT
FORM.
THIS PROCEDURE IMPLEMENTS THE "INSTANTIATION RULES". THE
FORM
PROCEDURES ARE OF THE FORM:

```
FORMNAME(FORMPTR, FORMPATH)
  STRUCT FRM%D *FORMPTR;          POINTER TO DATA
                                  STRUCTURE OF FORM.
  CHAR *FORMPATH;                 PATH IN FORM PROCESSOR
                                  TO FORM.
```

```
!
<DECLARE SOME VARIABLES>
```

```
                                  "VISIT ALL ITEMS ON
FORM".
                                  COPY DATA VALUES TO
ITEMS ON FORM.
```

```
MEMCPY(FORMPTR->FIELD, DBR%D.FIELD, SIZE);  
. . .  
                                "VISIT ALL ARRAYS ON  
                                FORM".  
FOR (I = 0; !DONE; I++)  
{  
    <CHECK FOR GROUP SEPERATOR OR END OF FILE F  
    DATA RECORDS WHICH TARGET TO THESE SUBFORMS.>  
  
    <CHECK FOR OVERFLOW ON THIS ARRAY.>  
  
    <CALL THE SUBFORM'S PROCEDURE.>  
}  
. . .  
  
<READ NEXT DATA RECORD AND CHECK FOR CHANGE  
    CONDITIONS.>  
  
RETURN <TRUE IF ANY CONDITIONS TRIPPED OR READ END  
    OF FILE.>  
}
```

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE
READDB - READ DATA BASE
RESETNDP - RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
MAPDB - MAP DATABASE
VISITA - VISIT ARRAYS ON THIS FORM
HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
 WITHIN
SETNDP - SET NODUPLICATE FIELDS TO BLANK IF THEY ARE
 DUPLICATED

CALLED DIRECTLY BY:

BIDSUB - BUILD SUBROUTINES

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CALCSTAT
PURPOSE: CALCULATE STATISTIC
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

FP = FIELD *
DP = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

CALCSTAT - CALCULATE STATISTIC
MAKQR - MAKE QUALIFIED REFERENCE
SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

FORMPDAT - FORM PDATA
CALCSTAT - CALCULATE STATISTIC

USED IN MAIN PROGRAM(S):

GRP MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CCONV
PURPOSE: C CONVERSIONS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

ES = ESDTYPE *
TBLSTR = CHAR *
SELNO = INT

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
SIDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

atoi
escpy
strncmp
fprintf
MAKES/INDENT - INDENT

CALLED DIRECTLY BY:

MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CDMESQY
PURPOSE: PROGRAM NAME CDMESQY
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CDMESQY
SOURCE FILE TYPE: .PRC
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SELECTS ALL THE DATA ITEMS FOR A GIVEN VIEW AND RETRIEVES EACH ES DATA ITEM'S MACHINE TYPE, SIZE, AND NUMBER OF DECIMAL DIGITS.

THIS INFORMATION IS RETURNED TO THE CALLING PROGRAM IN AN ARRAY STRUCTURE.

THIS ROUTINE WILL CHANGE WHEN DOMAINS AND DATA TYPES ARE COMPLETELY DEFINED FOR THE CDM.

ARGUMENTS:

VIEW = DSPLY [X(30)]

INCLUDE FILES:

SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR
ERRPRO - PROCESS ERROR INCLUDE FILE

ROUTINES CALLED:

ERRPRO

CALLED DIRECTLY BY:

MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE

USED IN MAIN PROGRAM(S):

GRP:MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CES
PURPOSE: C ES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

ES = ESDTYPE *
SELNO = INT
TBLNUM = INT
REC_CNT = INT

INCLUDE FILES:

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

ROUTINES CALLED:

FPRINTF
ATOI
ESCPY
MAKES/INDENT - INDENT

CALLED DIRECTLY BY:

MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CESPS
PURPOSE: C ES TO PS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: ESPSMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SELPTR = SELECT *

INCLUDE FILES:

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

CALLED DIRECTLY BY:

ESPSMAP - THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA
MAPPING

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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:

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

ROUTINES CALLED:

MYALLOC - MY MALLOC
ABS
STRASN

CALLED DIRECTLY BY:

CHKFRM - CHECK FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

VOID CHKFLD()

DESCRIPTION

CHECKS THE CURRENT FIELD FOR COMPLETENESS AND CONSISTENCY

INCLUDE FILES:

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

ROUTINES CALLED:

FNDATT - FIND ATTRIBUTE
ERROR - ISSUE ERROR MESSAGE
MEMSET
MAX
FREE
WRTEXP - WRITE EXPRESSION
BLEN
MEMCPY
SYMSG
MYALLOC - MY MALLOC
STRLEN

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRF/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CHKGRP
PURPOSE: CHECK FOR GROUP SEPERATORS OR END OF FILE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

CHKGRP(FP)
FIELD *FP;

INPUTS/OUTPUTS:
NONE

INPUTS:
FP - FIELD POINTER

• OUTPUTS:
NONE

DESCRIPTION

CHECKS IF THE DATA RECORD WHICH TARGETS TO THE FORM (FP)
HAS A
GROUP SEPERATOR OR END OF FILE. IF SO IT CLEARS THE
NODUP%D FIELDS
AND READS ANOTHER RECORD.

ARGUMENTS:

FP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

HASLOWER - HAS A LOWER FORM WHICH READS THE SAME DATA
 RECORD?
SPRINTF
GEN - GENERATE A LINE OF CODE
CLRNDP - CLEAR NODUPLICATE FIELDS
DBFREAD - GENERATE DATA BASE FREAD

CALLED DIRECTLY BY:

VISITA - VISIT ARRAYS ON THIS FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CHKSIZE
PURPOSE: CHECK SIZE OF ITEMS DOING CONVERSIONS ON
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: CHKSIZE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
CHKSIZE(DPTR, FPTR, DIR)
CDMDTYPE *DPTR;
FIELD *FPTR;
CHAR DIR;
```

DESCRIPTION

CHECK THE SIZE OF THE CDM DATA TYPE TO THE PRESENTATION
ITEM SIZE
ONLY PUT OUT A WARNING MESSAGE IF TRUNCATION WILL OCCUR
ON CONVERSION

ARGUMENTS:

```
DPTR = CDMDTYPE *
FPTR = FIELD *
DIR = CHAR
```

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

```
BLN
FPRINTF
```

CALLED DIRECTLY BY:

```
COBESPS - COBOL ES TO PS
COBPE - COBOL PE
```

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CLRNDP
PURPOSE: CLEAR NODUPLICATE FIELDS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

CLRNDP(SP)
SELECT *SP;

INPUTS/OUTPUTS:
NONE

INPUTS:
SP - SELECT POINTER

OUTPUTS:
NONE

DESCRIPTION

CLEARs ALL THE NODUP%D FIELDS WHICH THIS SELECT AND ALL
ITS CHILDREN
TARGET TO.

ARGUMENTS:

SP = SELECT *

INCLUDE FILES:

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

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE
CLRNDP - CLEAR NODUPLICATE FIELDS

CALLED DIRECTLY BY:

GENAQ - GENERATE ACTION QUERY (SELECT)
CHKGRP - CHECK FOR GROUP SEPERATORS OR END OF FILE
CLRNDP - CLEAR NODUPLICATE FIELDS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CLSFIL
PURPOSE: CLOSE FILES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SPTR = SELECT *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

CLSFIL - CLOSE FILES
FPRINTF
INDENT - INDENT A LINE OF GENERATED CODE

CALLED DIRECTLY BY:

CLSFIL - CLOSE FILES
PROCEN - PROCEDURE DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: COBCONV
PURPOSE: COBOL CONVERSIONS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

ES = ESDTYPE *
TBLSTR = CHAR *
SELNO = INT
REC_CNT = INT

INCLUDE FILES:

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

ROUTINES CALLED:

MAKES/NUMPIC - NUMBER PICTURE CLAUSE
MAKES/CNUMPIC - C NUMBERS
ATOI
DASH - WRITE DASH '-'
ESCPY
FPRINTF
MAKES/INDENT - INDENT

CALLED DIRECTLY BY:

MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: COBES
PURPOSE: COBOL ES RECORD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

ES = ESDTYPE *
SELNO = INT
TBLNUM = INT
REC_CNT = INT

INCLUDE FILES:

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

ROUTINES CALLED:

MAKES/NUMPIC - NUMBER PICTURE CLAUSE
FPRINTF
MAKES/INDENT - INDENT
ATOI
DASH - WRITE DASH '-'
ESCPY

CALLED DIRECTLY BY:

MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: COBESPS
PURPOSE: COBOL ES TO PS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: ESPSMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SELPTR = SELECT *

INCLUDE FILES:

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

ROUTINES CALLED:

DASH - WRITE DASH '-'
GETTBL - GET A TABLE NAME
GETCOL - GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN
STRING
CHKSIZE - CHECK SIZE OF ITEMS DOING CONVERSIONS ON
FPRINTF
ESPSMAP/INDENT - INDENT

CALLED DIRECTLY BY:

ESPSMAP - THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA
MAPPING

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: COBPE
PURPOSE: COBOL PE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PEMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

STR1 = CHAR *
STR2 = CHAR *
FPTR = FIELD *
DPTR = CDMDTYPE *

INCLUDE FILES:

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

ROUTINES CALLED:

FPRINTF
CHKSIZE - CHECK SIZE OF ITEMS DOING CONVERSIONS ON
INDENT - INDENT A LINE OF GENERATED CODE

CALLED DIRECTLY BY:

PEMAP - THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA
AND MAPPING

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CPE
PURPOSE: C PE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PEMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

STR1 = CHAR *
STR2 = CHAR *
FPTR = FIELD *
DPTR = CDMDTYPE *

INCLUDE FILES:

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

CALLED DIRECTLY BY:

PEMAP - THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA
AND MAPPING

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

CHAR *CSTASH(S)
CHAR *S;

DESCRIPTION

SAVES THE SPECIFIED CHARACTER STRING AND RETURNS A
POINTER TO IT

ARGUMENTS:

S = CHAR *

INCLUDE FILES:

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

ROUTINES CALLED:

STRCPY
STRLEN
MYALLOC - MY MALLOC

CALLED DIRECTLY BY:

YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: CTLRSV
PURPOSE: CONTROL RESOLVE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

CTLRSV(CTLPTR)
CTLLST *CTLPTR;

INPUTS:

CTLPTR - CONTROL LIST FROM WHICH TO LOOK FOR PATHS.

DESCRIPTION

RESOLVES ALL QUALIFIED NAMES INTO FIELD POINTERS FOR ALL
NAMES
WHICH ARE ROOTED IN CTLLST'S (CONTROL LISTS).

ARGUMENTS:

CTLPTR = CTLLST *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GETPTH - GET PATH
ERROR - ISSUE ERROR MESSAGE

CALLED DIRECTLY BY:

FLDRSV - FIELD RESOLVE

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: DASH
PURPOSE: WRITE DASH '-'
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

STR = CHAR []

INCLUDE FILES:

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

ROUTINES CALLED:

STRCHR

CALLED DIRECTLY BY:

COBESPS - COBOL ES TO PS
COBES - COBOL ES RECORD
COBCONV - COBOL CONVERSIONS
MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
MAKWHEs/COBWHEs - COBOL WHERE ES
SELGEN - SELECT GENERATE
SELWS - SELECT WORKING STORAGE SECTION
INSWS - INSERT WORKING STORAGE SECTION
INSERT - INSERT PROCEDURE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: DATAGEN
PURPOSE: DATA DIVISION GENERATE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT
APNAME = CHAR *
TYPE = CHAR

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

NDMLLNK - LINKAGE SECTION
FILELNK - FILE LINKAGE SECTION GENERATE
INDENT - INDENT A LINE OF GENERATED CODE
FPRINTF
SELWS - SELECT WORKING STORAGE SECTION
INSWS - INSERT WORKING STORAGE SECTION
FD - FD SECTION DECLARATIONS
ASSIGN - ASSIGN FILE SECTION

CALLED DIRECTLY BY:

STD CODE - STANDARD COBOL CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: DBFREAD
PURPOSE: GENERATE DATA BASE FREAD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
DBFREAD(SP, GENCHG)
  SELECT *SP;
  BOOL GENCHG;
```

INPUTS/OUTPUTS:
NONE

INPUTS:
SP - SELECT POINTER INDICATES DATA RECORD TO READ.
GENCHG - IF TRUE THEN ALSO GENERATE THE CHECK CHANGE
CONDITION CODE.

OUTPUTS:
NONE

DESCRIPTION

GENERATES THE FREAD TO READ THE DATA RECORD ASSOCIATED
WITH A SELECT.
SETS THE DBCODE TO INDICATE STATUS (TRUE INDICATES AN EOF
OR GROUP
SEPERATOR WAS READ). IF GENCHG IS TRUE THEN ALSO GENERATE
THE CODE
TO CHECK CHANGE CONDITIONS.

ARGUMENTS:

```
SP =      SELECT *
GENCHG =   BOOL
```

INCLUDE FILES:

```
STD TYP      - STANDARD TYPE DEFINITIONS
STD IO       - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD          - FORM PROCESSOR DATA
FPD INI      - FPD INITIALIZATION
```

FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

SELOPN - SELECT OPEN
READDB - READ DATA BASE
CHKGRP - CHECK FOR GROUP SEPERATORS OR END OF FILE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: DCLINDX
PURPOSE: DECLARE INDEX VARIABLES
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

DCLINDX(TP)
TRGLST *TP;

INPUTS:

TP - CONDITION WHICH REQUIRES INDEX VARIABLES.

DESCRIPTION

THIS PROCEDURE DECLARES THE INDEX VARIABLES USED IN
CONDITIONS AND
ACTIONS WHICH MAKE USE OF UNIVERSAL QUALIFICATION. THESE
DECLARATIONS
FOR CONDITIONS AND ACTIONS RESPECTIVELY ARE:

INT TINDX%D,...;
INT AINDX%D,...;

ARGUMENTS:

TP = TRGLST *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

MAX
STRCPY
SPRINTF
STRLEN
STRCAT
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENTRG - GENERATE TRIGGERS
GENDOA - GENERATE PROCEDURE "DOACT" DO ACTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: ENDGEN
PURPOSE: END GERNERATE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

GENERATE THE ENDING CODE WHICH CLOSES THE FILES AND DOES THE
NDML ERROR PROCESSING

ARGUMENTS:

LANG = INT

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

FPRINTF
INDENT - INDENT A LINE OF GENERATED CODE

CALLED DIRECTLY BY:

STDCODE - STANDARD COBOL CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

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

DESCRIPTION

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

ARGUMENTS:

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

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

PMSGLS
STRLEN
SPRINTF

CALLED DIRECTLY BY:

```
GENTRG - GENERATE TRIGGERS
CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM
ADDCHK - ADD POSITION TO CHECK LIST
YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER
```

MKINC	- MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
GETFILE	- RETURN A FILE POINTER BASED ON INPUT FROM THE USER
MAKES	- MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
INSRSV	- INSERT RESOLVE
SELRSV	- SELECT RESOLVE
CTLRSV	- CONTROL RESOLVE
STATRSV	- STATISTIC RESOLVE
TRGRSV	- TRIGGER RESOLVE
ACTRSV	- ACTION RESOLVE
MLPFRM	- MAKE A LIST OF PRESENTED FORMS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: ESPSMAP
PURPOSE: THE EXTERNAL SCHEMA AND PRESENTATION
SCHEMA MAPPING
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: ESPSMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
ESPSMAP (LANG, SELPTR)  
INT LANG;  
SELECT *SELPTR;
```

DESCRIPTION

GENERATES THE CODE TO TRANSFORM AN EXTERNAL SCHEMA DATA
ITEM INTO
A PRESENTATION SCHEMA FORM ITEMS AND VICE VERSA.

ARGUMENTS:

```
LANG =      INT  
SELPTR =    SELECT *
```

INCLUDE FILES:

```
STDTyp      - STANDARD TYPE DEFINITIONS  
STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****  
FPD         - FORM PROCESSOR DATA  
RW          - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

```
CESPS       - C ES TO PS  
COBESPS     - COBOL ES TO PS
```

CALLED DIRECTLY BY:

```
SELMAP      - MAP SELECTED DATA TO OUTPUT RECORD
```

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: ESPSMAP/INDENT
PURPOSE: INDENT
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: ESPSMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

M = INT
T = INT

INCLUDE FILES:

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

ROUTINES CALLED:

PUTC

CALLED DIRECTLY BY:

COBESPS - COBOL ES TO PS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

----- SYNOPSIS

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

DESCRIPTION

PRINTS A FATAL MESSAGE ON STDERR AND EXITS

ARGUMENTS:

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

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

SPRINTF
STRLEN
PMSGLS

CALLED DIRECTLY BY:

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

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: FD
PURPOSE: FD SECTION DECLARATIONS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SPTR = SELECT *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

SELLEN - COMPUTE LENGTH OF SELECT PS RECORD
FPRINTF
INDENT - INDENT A LINE OF GENERATED CODE

CALLED DIRECTLY BY:

DATAGEN - DATA DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: FILELNK
PURPOSE: FILE LINKAGE SECTION GENERATE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SPTR = SELECT *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

INDENT - INDENT A LINE OF GENERATED CODE
FPRINTF

CALLED DIRECTLY BY:

DATAGEN - DATA DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

----- SYNOPSIS

CHAR *FLANCI(FPTR)
FILE *FPTR;

INPUTS:

FPTR - FILE TO BE COMPILED

DESCRIPTION

COMPILES THE SPECIFIED FILE INTO THE LOCAL OPEN LIST.

ARGUMENTS:

FPTR = FILE *

INCLUDE FILES:

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

ROUTINES CALLED:

YYPARSE - FLAN PARSER
DELFLD

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: FLDRSV
PURPOSE: FIELD RESOLVE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

FLDRSV(DP)
FIELD *DP;

INPUTS:

DP - FIELD FROM WHICH TO BEGIN SEARCH.

DESCRIPTION

RESOLVES ALL QUALIFIED NAMES INTO FIELD POINTERS FOR ALL
NAMES
WHICH ARE ROOTED IN FIELDS.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

FLDRSV - FIELD RESOLVE
CTLRSV - CONTROL RESOLVE
STATRSV - STATISTIC RESOLVE

CALLED DIRECTLY BY:

RWOPN - REPORT WRITER OPEN FORMS
FLDRSV - FIELD RESOLVE

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: FLDTYP
PURPOSE: FIELD TYPE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: FLANSF
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:

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

CALLED DIRECTLY BY:

CHKFRM - CHECK FORM
ADDCHK - ADD POSITION TO CHECK LIST

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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:

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

ROUTINES CALLED:

STRCMP

CALLED DIRECTLY BY:

CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM
YYPARSE - FLAN PARSER
RWEXPD - REPORT WRITER EXPAND ARRAYS
RWSP/FIXFR - FIX UP A FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: FNDFRM
PURPOSE: FIND FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD * ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

FIELD *FNDFRM(STR)
CHAR STR[];

INPUTS:

STR - NAME OF FORM TO FIND

DESCRIPTION

FINDS THE NAMED FORM ON THE OPNLST AND RETURNS A POINTER
TO IT.
RETURNS A NULL IF THE FORM CAN NOT BE FOUND.

ARGUMENTS:

STR = CHAR []

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

STRCMP

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM
GENAR - GENERATE ACTION PRESENT
MLPFRM - MAKE A LIST OF PRESENTED FORMS
WINRSV - WINDOW RESOLVE

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: FRMPDAT
PURPOSE: FORM PDATA
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

FRMPDAT(FDP)
FIELD *FDP;

INPUTS:

FDP - POINTER TO A FORM.

DESCRIPTION

GENERATES A PDATA FOR THE FORM POINTED TO BY FDP IF THERE
ARE ANY
ITEMS ON IT.

ARGUMENTS:

FDP = FIELD *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
WITHIN
CALCSTAT - CALCULATE STATISTIC
SPRINTF
GEN - GENERATE A LINE OF CODE
RSETSTAT - RESET STATISTIC

CALLED DIRECTLY BY:

GENAP - GENERATE ACTION PAGE
GENAR - GENERATE ACTION PRESENT

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: FRNTND
PURPOSE: FORMS FRONT END TO APPLICATION GENERATOR
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: APFRNT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: AP
DOCUMENTATION GROUP: APGEN

DESCRIPTION:

SYNOPSIS

CHAR *FRNTND()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

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

ARGUMENTS:

FILNAM = CHAR [41]

INCLUDE FILES:

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

ROUTINES CALLED:

STRCHR
INITAL
MEMCMP
TRMNAT
PMSGLC
INITFP
ADDFRM
GDATA
OISCR
SPRINTF

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GEN
PURPOSE: GENERATE A LINE OF CODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

GEN (STRING)
CHAR STRING[];

INPUTS/OUTPUTS:
NONE

INPUTS:
STRING - THIS IS THE LINE OF CODE TO GENERATE

OUTPUTS:
NONE

DESCRIPTION

THIS ROUTINE WILL MOVE A LINE OF CODE TO THE OUTPUT FILE,
IT ALSO TAKES CARE OF BALANCING RIGHT AND LEFT BRACKETS
AS WELL AS ALIGNING # TYPE STATEMENTS.

ARGUMENTS:

STRING = CHAR []

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

FPRINTF

CALLED DIRECTLY BY:

GENMAIN	- GENERATE MAIN PROGRAM
GENTRG	- GENERATE TRIGGERS
GENAAL	- GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
GENAA	- GENERATE PROCEDURE "ADDACT" ADD AN ACTION
GENDOA	- GENERATE PROCEDURE "DOACT" DO ACTION
DCLINDX	- DECLARE INDEX VARIABLES
GENAL	- GENERATE ACTION LIST
UQFOR	- UNIVERSAL QUALIFIER FOR LOOP
GENAP	- GENERATE ACTION PAGE
GENAR	- GENERATE ACTION PRESENT
GENAQ	- GENERATE ACTION QUERY (SELECT)
GENAS	- GENERATE ACTION SET
GENAE	- GENERATE ACTION EXIT
GENAH	- GENERATE ACTION HELP
GENAT	- GENERATE ACTION SIGNAL
GENAI	- GENERATE ACTION INSERT
SELWHR	- SELECT WHERE
SELOPN	- SELECT OPEN
FRMPDAT	- FORM PDATA
GENBEG	- GENERATE BEGINNING OF APPLICATION OR REPORT
MKINC	- MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
GENDB	- GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
GENFS	- GENERATE FORM DATA STRUCTURES
GENDS	- GENERATE DATA DATA STRUCTURES
GENFSD	- GENERATE FORM STRUCTURE DATA INITIALIZATION
GENFP	- GENERATE FORM PATH
GENNDP	- GENERATE NODUPLICATE DECLARATIONS
GENCHG	- GENERATE CHANGE DECLARATIONS
GENINS	- GENERATE INSERT DECLARATIONS
BSCODE	- BUILD SUBROUTINE CODE
MAPDB	- MAP DATABASE
VISITA	- VISIT ARRAYS ON THIS FORM
CHKGRP	- CHECK FOR GROUP SEPERATORS OR END OF FILE
CLRNDP	- CLEAR NODUPLICATE FIELDS
GENPAG	- GENERATE NEWPAG PROCEDURE
DBFREAD	- GENERATE DATA BASE FREAD
SETNDP	- SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
RSETNDP	- RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
CALCSTAT	- CALCULATE STATISTIC
RSETSTAT	- RESET STATISTIC

USED IN MAIN PROGRAM(S):

GRP/MAIN	- GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAA
PURPOSE: GENERATE PROCEDURE "ADDACT" ADD AN ACTION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENAA()

DESCRIPTION

THIS ROUTINE GENERATES A PROGRAM THAT WILL ADD AN ACTION TO THE ACTION LIST AT RUN TIME. THE PROGRAM THAT IS GENERATED BY THIS ROUTINE IS FIXED AND IS NOT CHANGED FOR ANY REPORT, IT IS ALWAYS THE SAME PROGRAM.

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENACT - GENERATE ACTIONS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAAL
PURPOSE: GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENAAL()

DESCRIPTION

THIS ROUTINE TRAVERSES THE TRIGGER DATA STRUCTURE FOR EACH TRIGGER IT GENERATES A CALL TO ADD EACH ACTION. PRIORITIES FOR ACTIONS ARE DETERMINED BY THIS ROUTINE.

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE
SPRINTF

CALLED DIRECTLY BY:

GENACT - GENERATE ACTIONS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENACT
PURPOSE: GENERATE ACTIONS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENACT()

DESCRIPTION

THIS ROUTINE GENERATES THE CODE TO IMPLEMENT TRIGGERS AND ACTIONS. IT GENERATES CODE TO ADD A LIST OF ACTIONS TO ADD AN ACTION AND GENERATES THE CODE NECESSARY TO PERFORM AN ACTION.

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GENAAL - GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
GENAA - GENERATE PROCEDURE "ADDACT" ADD AN ACTION
GENDOA - GENERATE PROCEDURE "DOACT" DO ACTION

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAE
PURPOSE: GENERATE ACTION EXIT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

GENAE(TP, AP)
 TRGLST *TP;
 ACTLST *AP;

INPUTS:

TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES THE EXIT ACTION

ARGUMENTS:

TP = TRGLST *
AP = ACTLST *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAH!
PURPOSE: GENERATE ACTION HELP
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

GENAH(TP, AP)
TRGLST *TP;
ACTLST *AP;

INPUTS:

TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES THE HELP ACTION

ARGUMENTS:

TP = TRGLST *
AP = ACTLST *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAI
PURPOSE: GENERATE ACTION INSERT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

GENAI(TP, AP)
TRGLST *TP;
ACTLST *AP;

INPUTS:

TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES THE INSERT ACTION

ARGUMENTS:

TP = TRGLST *
AP = ACTLST *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

MAKQR - MAKE QUALIFIED REFERENCE
SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAL
PURPOSE: GENERATE ACTION LIST
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

GENAL(TP, AP)
TRGLST *TP;
ACTLST *AP;

INPUTS:

TP - CONDITION TO WHICH THIS ACTION BELONGS.
AP - ACTION TO GENERATE CODE FOR.

DESCRIPTION

CALL THE PROCEDURE WHICH GENERATES THE CODE TO IMPLEMENT AN ACTION. ALSO CALLS PROCEDURE TO GENERATE FOR LOOPS FOR UNIVERSAL QUALIFICATION.

ARGUMENTS:

TP = TRGLST *
AP = ACTLST *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

UQFOR - UNIVERSAL QUALIFIER FOR LOOP
GEN - GENERATE A LINE OF CODE
GENAP - GENERATE ACTION PAGE
GENAR - GENERATE ACTION PRESENT
GENAQ - GENERATE ACTION QUERY (SELECT)
GENAS - GENERATE ACTION SET
GENAE - GENERATE ACTION EXIT

GENAH - GENERATE ACTION HELP
GENAT - GENERATE ACTION SIGNAL
GENAI - GENERATE ACTION INSERT

CALLED DIRECTLY BY:

GENTAL - GENERATE TRIGGER ACTION LIST
GENDOA - GENERATE PROCEDURE "DOACT" DO ACTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAP
PURPOSE: GENERATE ACTION PAGE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

GENAP(TP, AP)
 TRGLST *TP;
 ACTLST *AP;

INPUTS:

TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES THE PAGE ACTION

ARGUMENTS:

TP = TRGLST *
AP = ACTLST *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE
FRMPDAT - FORM PDATA
SPRINTF

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAQ
PURPOSE: GENERATE ACTION QUERY (SELECT)
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

GENAQ(TP, AP)
TRGLST *TP;
ACTLST *AP;

INPUTS:

TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES THE SELECT ACTION

ARGUMENTS:

TP = TRGLST *
AP = ACTLST *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

SELWHR - SELECT WHERE
SPRINTF
GEN - GENERATE A LINE OF CODE
CLRNDP - CLEAR NODUPLICATE FIELDS
SELOPN - SELECT OPEN

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAR
PURPOSE: GENERATE ACTION PRESENT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

GENAR(TP, AP)
TRGLST *TP;
ACTLST *AP;

INPUTS:

TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES THE PRESENT ACTION

ARGUMENTS:

TP = TRGLST *
AP = ACTLST *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

FNDFRM - FIND FORM
GEN - GENERATE A LINE OF CODE
ISOPNE - DETERMINE IF THIS FIELD IS OPEN ENDED
HASDATA - DETERMINE IF THERE ARE ANY SELECT STATEMENTS
HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
 WITHIN
STRCMP
FRMPDAT - FORM PDATA
SPRINTF

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAS
PURPOSE: GENERATE ACTION SET
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

GENAS (TP, AP)
TRGLST *TP;
ACTLST *AP;

INPUTS:

TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES THE SET ACTION

ARGUMENTS:

TP = TRGLST *
AP = ACTLST *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

ISOPNE - DETERMINE IF THIS FIELD IS OPEN ENDED
MAKQR - MAKE QUALIFIED REFERENCE
STRCPY
SPRINTF
GEN - GENERATE A LINE OF CODE
STRSPN
STRLEN

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENAT
PURPOSE: GENERATE ACTION SIGNAL
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
GENAT(TP, AP)
  TRGLST *TP;
  ACTLST *AP;
```

INPUTS:

TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES THE SIGNAL ACTION

ARGUMENTS:

```
TP =      TRGLST *
AP =      ACTLST *
```

INCLUDE FILES:

```
STDYTP    - STANDARD TYPE DEFINITIONS
FPD        - FORM PROCESSOR DATA
RW         - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

```
WARNING    - ISSUE WARNING MESSAGE
SPRINTF     - GENERATE A LINE OF CODE
GEN
```

CALLED DIRECTLY BY:

```
GENAL      - GENERATE ACTION LIST
```

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENBEG
PURPOSE: GENERATE BEGINNING OF APPLICATION OR REPORT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

GENBEG(NAME)
CHAR NAME[];

INPUTS:

NAME - NAME OF THE APPLICATION OR REPORT

OUTPUTS:

NONE

DESCRIPTION

THIS ROUTINE GENERATES THE PROLOG FOR AN APPLICATION OR A REPORT.

IT CONSISTS OF THE #INCLUDE'S, THE ACTION STRUCTURE AND POINTERS,
AND DECLARATIONS FOR SEVERAL OTHER FIXED SIZE VARIABLES.

ARGUMENTS:

NAME = CHAR []

INCLUDE FILES:

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

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE
SPRINTF

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENCHG
PURPOSE: GENERATE CHANGE DECLARATIONS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENCHG()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

THIS ROUTINE GENERATES THE DECLARATION TO HOLD THE LAST
VALUE OF
AN ITEM WHICH HAS A CHANGE CONDITION ON IT. THE FORM OF THE
DECLARATION IS:

CHAR CHG%D[SIZE]; %D - NUMBER OF FIELD, SIZE OF
FIELD.

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENDB
PURPOSE: GENERATE DATA BASE RECORDS AND FILE
DECLARATIONS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

GENDB(COUNT)
INT COUNT;

INPUTS:

COUNT - THE NUMBER OF SELECTS IN THIS LIST

OUTPUTS:

NONE

DESCRIPTION

GENERATES DECLARATIONS FOR SELECT FILES: FILE POINTERS,
NAMES AND
STATUS CODES.

ARGUMENTS:

COUNT = INT

INCLUDE FILES:

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

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENDOA
PURPOSE: GENERATE PROCEDURE "DOACT" DO ACTION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENDOA()

DESCRIPTION

THIS ROUTINE GENERATES THE CODE NECESSARY TO PERFORM AN
ACTION AT RUN TIME. IT GENERATES CODE FOR EACH ACTION FOR
EACH
TRIGGER.

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE
DCLINDX - DECLARE INDEX VARIABLES
GENAL - GENERATE ACTION LIST
STRCHR
SPRINTF

CALLED DIRECTLY BY:

GENACT - GENERATE ACTIONS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENDS
PURPOSE: GENERATE DATA DATA STRUCTURES
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENDS()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

THIS ROUTINE GENERATES A DATA STRUCTURE FOR EACH SELECT STATEMENT. THESE ARE OF THE FORM:

```
STRUCT
{
  CHAR DBNAME[20];           FIELDS TO GET DATA.
  CHAR DBID[4];
  CHAR HOSTID[3];
  CHAR DBMSNAME[30];
  CHAR CR;                   CARRIAGE RETURN PAD.
} DBR%D;                     %D - NUMBER OF SELECT (0 IS
                             FIRST).
```

INCLUDE FILES:

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

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE
SPRINTF

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENFP
PURPOSE: GENERATE FORM PATH
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENFP()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

THIS ROUTINE GENERATES THE DECLARATION TO HOLD A PATH NAME FOR OPEN ENDED FORMS AND THE TOP FORM(S). THE FORM OF THE DECLARATIONS IS:

CHAR PATH%D[120] = "FORMNAME"; %D IS THE NUMBER OF THE
FORM AND FORMNAME IS THE NAME OF THE
FORM.

INCLUDE FILES:

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

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENFS
PURPOSE: GENERATE FORM DATA STRUCTURES
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

GENFS(DP)
FIELD *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
(DP) - FIELD POINTER

OUTPUTS:
NONE

DESCRIPTION

THIS ROUTINE USES THE STRUCTURE TAGS TO ALLOCATE SPACE FOR FORM DATA FOR CURRENT AND PREVIOUS IT GENERATES THIS CODE FOR OPEN ENDED FORMS AND FOR OPEN ENDED ARRAYS. THESE ARE DECLARED AS FOLLOWS:

STRUCT FRM%D FRM%DC, FRM%DP; %D - NUMBER OF THE FORM.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

ISOPNE - DETERMINE IF THIS FIELD IS OPEN ENDED
HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM

WITHIN

SPRINTF
GEN - GENERATE A LINE OF CODE
GENFS - GENERATE FORM DATA STRUCTURES

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM
GENFS - GENERATE FORM DATA STRUCTURES

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENFSD
PURPOSE: GENERATE FORM STRUCTURE DATA
INITIALIZATION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENFSD(DP)
FIELD *DP;

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

THIS ROUTINE INITIALIZES THE FORM CURRENT AND PREVIOUS
BUFFERS TO BLANK FOR BOTH OPEN ENDED FORMS AND OPEN
ENDED ITEMS. THESE ARE OF THE FORM:

MEMSET(&FRM%DC, ' ', SIZEOF FRM%DC); %D IS THE NUMBER OF
THE FORM.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

ISOPNE - DETERMINE IF THIS FIELD IS OPEN ENDED

HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
 WITHIN
SPRINTF
STRLEN
GEN - GENERATE A LINE OF CODE
GENFSD - GENERATE FORM STRUCTURE DATA INITIALIZATION
MAKQR - MAKE QUALIFIED REFERENCE

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM
GENFSD - GENERATE FORM STRUCTURE DATA INITIALIZATION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENINS
PURPOSE: GENERATE INSERT DECLARATIONS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENINS()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

THIS ROUTINE GENERATES THE DECLARATIONS FOR THE NDML
INSERT ACTION.
THE FORM OF THIS DECLARATION IS:

```
STRUCT
{
  STRUCT
  {
    CHAR DBID[4];
    CHAR HOSTID[3];
  } INSERT%D;           %D - NUMBER OF INSERT (0
                        IS FIRST).
  STRUCT
  {
    CHAR DBID[4];        FIELDS OF FORM TO BE
    INSERTED.
    CHAR HOSTID[3];
  } INSERT1;            ONE PRESENTATION SCHEME
    RECORD.
  CHAR DUMMY;           DUMMY FIELD IF THERE ARE
    NO INSERTS.
  } INSERTPS;           NAME OF INSERT STRUCTURE .
```


INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE
SPRINTF

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENMAIN
PURPOSE: GENERATE MAIN PROGRAM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: APMAIN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: AP
DOCUMENTATION GROUP: APGEN

DESCRIPTION:

SYNOPSIS

GENMAIN()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

THIS ROUTINE GENERATES THE FOLLOWING:

1. INCLUDE STATEMENTS
2. MAKINC IS USED TO GENERATE FORM DEFINITION
FUNCTION TAGS
3. GENCS IS CALLED TO INITIALIZE CONDITION FLAGS
4. GENFS IS USED TO GENERATE CURRENT AND PREVIOUS
BUFFERS FOR FORMS
5. GENFP IS USED TO GENERATE PATH DECLARATIONS
PER FORM
6. GENDS IS USED TO GENERATE DATA STRUCTURES FOR
EACH SELECT STATEMENT
7. TYPEDEF AND ACTION LIST POINTERS ARE GENERATED
8. THE FILE POINTERS, A DATABASE CODE AND FILE NAME
DATA STRUCTURES ARE GENERATED FOR EACH SELECT
STATEMENT
9. THE GLOBAL VARIABLE I IS DECLARED MAIN AND
DECLARATIONS FOR VARIABLES USED IN MAIN ARE
DECLARED
10. CURRENT FORM BUFFERS ARE INITIALIZED TO BLANKS
11. THE CALL TO INITFP
12. THE CALL TO THE COBOL PROGRAM TO DO THE SELECT
QUERIES
13. EACH FILE ASSOCIATED WITH A SELECT IS OPENED
AND THE FIRST RECORD IS READ THE VALUES ARE THEN

- MAPPED TO THE CURRENT FORM BUFFERS AND THEN THE
CURRENT FORM BUFFERS ARE COPIED TO THE PREVIOUS
SO THAT BOTH ARE THE SAME
14. THE FIRST ACTION FOR THE ACTION LIST IS PUT ON
AND THEN THE CODE TO PROCESS THE ACTION LIST IS
GENERATED

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

FND FRM - FIND FORM
HAS ITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
WITHIN
GEN BEG - GENERATE BEGINNING OF APPLICATION OR REPORT
MK INC - MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA
MAK INC)
GEN - GENERATE A LINE OF CODE
GEN TRG - GENERATE TRIGGERS
STR CHR
GEN FSD - GENERATE FORM STRUCTURE DATA INITIALIZATION
SPR INTF
GEN FS - GENERATE FORM DATA STRUCTURES
GEN FP - GENERATE FORM PATH
GEN NDP - GENERATE NODUPLICATE DECLARATIONS
GEN CHG - GENERATE CHANGE DECLARATIONS
GEN DS - GENERATE DATA DATA STRUCTURES
GEN DB - GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
GEN INS - GENERATE INSERT DECLARATIONS

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENNDP
PURPOSE: GENERATE NODUPPLICATE DECLARATIONS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENNDP()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

GENERATES DECLARATIONS FOR THE NODUP OPTION ON ITEMS. THE
DECLARATIONS
ARE OF THE FORM:

CHAR NODUP%D[SIZE]; %D - IS THE NUMBER OF THE FIELD,
SIZE OF FIELD.

INCLUDE FILES:

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

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENPAG
PURPOSE: GENERATE NEWPAG PROCEDURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
GENPAG()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

GENERATES THE PROCEDURE NEWPAG WHICH INCREMENTS THE FIELD
' . PAGENO; '
AND THEN DOES AN OUTSCR.

INCLUDE FILES:

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

ROUTINES CALLED:

GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENTAL
PURPOSE: GENERATE TRIGGER ACTION LIST
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENTRG
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: AP
DOCUMENTATION GROUP: APGEN

DESCRIPTION:

----- SYNOPSIS

GENTAL(TP)
TRGLST *TP;

INPUTS:

TP - CONDITION TO GENERATE CODE FOR ALL OF ITS ACTIONS.

DESCRIPTION

GENERATES CODE FOR ALL THE ACTIONS ASSOCIATED WITH THE
CONDITION
INDICATED BY TP.

ARGUMENTS:

TP = TRGLST *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

GENAL - GENERATE ACTION LIST

CALLED DIRECTLY BY:

GENTRG - GENERATE TRIGGERS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GENTRG
PURPOSE: GENERATE TRIGGERS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENTRG
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: AP
DOCUMENTATION GROUP: APGEN

DESCRIPTION:

SYNOPSIS
GENTRG()

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION
THIS ROUTINE GENERATES "TRIGN()" FUNCTIONS WHICH
DETERMINE
WHETHER OR NOT A GIVEN CONDITION HAS BEEN TRIGGERED.

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

STRCHR
SPRINTF
GEN - GENERATE A LINE OF CODE
MAQQR - MAKE QUALIFIED REFERENCE
STRCPY
STRLEN
STRCAT
UQFOR - UNIVERSAL QUALIFIER FOR LOOP
GENTAL - GENERATE TRIGGER ACTION LIST

DCLINDX - DECLARE INDEX VARIABLES
STRCMP
ERROR - ISSUE ERROR MESSAGE

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GETCOL
PURPOSE: GET THE COLUMN NAME OF A TABLE.COLUMN OR
COLUMN STRING
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RWSF
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

OUTSTR = CHAR []
COLNAM = CHAR []

INCLUDE FILES:

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

ROUTINES CALLED:

STRCHR
STRCPY

CALLED DIRECTLY BY:

COBESPS - COBOL ES TO PS
MAKWHEs/COBWHEs - COBOL WHERE ES
SELGEN - SELECT GENERATE
SELWS - SELECT WORKING STORAGE SECTION
INSERT - INSERT PROCEDURE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GETFILE
PURPOSE: RETURN A FILE POINTER BASED ON INPUT FROM
THE USER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FILE * ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
FILE *GETFILE(NAMPTR)  
CHAR *NAMPTR;
```

INPUTS/OUTPUTS:
NONE

INPUTS:
NAMPTR - STRING WITH NAME OF FILE.

OUTPUTS:
FILE POINTER IS RETURNED THROUGH THE FUNCTION REFERENCE

DESCRIPTION

GETFILE OPENS THE FILE NAMED BY THE INPUT PARAMETER. IF
THE USER
DOES NOT SPECIFY THE .FDL SUFFIX IT IS
AUTOMATICALLY APPENDED. THE FILE IS THEN OPENED.

ARGUMENTS:

NAMPTR = CHAR *

INCLUDE FILES:

```
STD TYP - STANDARD TYPE DEFINITIONS  
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****  
FPD - FORM PROCESSOR DATA  
FPDINI - FPD INITIALIZATION  
FPPARM - FORM PROCESSOR PARAMETERS  
RW - REPORT WRITER DEFINITIONS  
NTM - NTM INTERFACE INCLUDE FILE
```

ROUTINES CALLED:

ERROR - ISSUE ERROR MESSAGE
FOPEN
SPRINTF

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GETPTH
PURPOSE: GET PATH
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

```
CHAR *GETPTH(PATH, DPP, FLDLST)
  CHAR PATH[];
  FIELD **DPP, *FLDLST;
```

INPUTS:

PATH - PATH TO BE RESOLVED INTO A POINTER.
FLDLST - FIELD HIERARCHY TO SEARCH FOR A PATH.

OUTPUTS:

DPP - POINTER TO POINTER TO FIELD INDICATED BY PATH.

DESCRIPTION

RESOLVES A QUALIFIED NAME INTO A FIELD POINTER. REPEATEDLY
CALLS
PTHPTR WITH FORMS IN THE TOPLST (SEE MLPFRM()).

ARGUMENTS:

PATH = CHAR []
DPP = FIELD **
FLDLST = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

PTHPTR
STRCPY
STRUPC
STRCHR

CALLED DIRECTLY BY:

INSRSV - INSERT RESOLVE
SELRSV - SELECT RESOLVE
CTLRSV - CONTROL RESOLVE
STATRSV - STATISTIC RESOLVE
TRGRSV - TRIGGER RESOLVE
ACTRSV - ACTION RESOLVE
WINRSV - WINDOW RESOLVE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GETTBL
PURPOSE: GET A TABLE NAME
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

OUTSTR = CHAR []
TNUM = INT *
COLNAM = CHAR []
SELPTR = SELECT *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

STRCHR
ESCPY
NULBLK - BLANK FILL A STRING
STRCMP
STRCPY

CALLED DIRECTLY BY:

COBESPS - COBOL ES TO PS
MAKWHE/COBWHE - COBOL WHERE ES
SELGEN - SELECT GENERATE
SELWS - SELECT WORKING STORAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

FIELD *GFLDPT(FLDPTR, S)
FIELD *FLDPTR;
CHAR *S;

DESCRIPTION

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

ARGUMENTS:

FLDPTR = FIELD *
S = CHAR *

INCLUDE FILES:

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

ROUTINES CALLED:

STRCMP

CALLED DIRECTLY BY:

CHKFRM - CHECK FORM
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: GRP/MAIN
PURPOSE: GENERATE APPLICATION/REPORT PROGRAM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

MAIN() ! THE EXECUTABLE IS NAMED "GRP" OR "GAP".

INPUTS/OUTPUTS:
NONE

INPUTS:
NONE.

OUTPUTS:
NONE

DESCRIPTION

THIS IS THE MAIN ROUTINE FOR THE APPLICATION/REPORT
GENERATING PROGRAM.
IT PROMPTS THE USER FOR HIS .FDL DEFINITION FILE, CALLS
FLAN TO
PARSE THE APPLICATION OR REPORT DEFINITION, WRITES OUT THE
FD FILES,
GENERATES THE SPECIFIC DATA STRUCTURES, AND ESTABLISHES THE
HIERARCHICAL RELATIONSHIP BETWEEN THE SELECT STATEMENTS
AND THE
FORM HIERARCHY. IT THEN GENERATES THE C CODE IN THE
FOLLOWING STEPS:

1. GENERATES THE MAIN PROGRAM
2. GENERATES THE CODE FOR EACH SUB-ROUTINE WHERE THESE
SUB-ROUTINES
CORRESPOND TO FORMS IN THE HIERARCHY
3. GENERATES THE CODE TO PROCESS ON CONDITIONS AND ACTIONS
4. GENERATES THE COBOL CODE TO PROCESS THE SELECT
STATEMENTS

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****

FPD	- FORM PROCESSOR DATA
FPDINI	- FPD INITIALIZATION
FPPARM	- FORM PROCESSOR PARAMETERS
RW	- REPORT WRITER DEFINITIONS
NTM	- NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

FRNTND	- FORMS FRONT END TO APPLICATION GENERATOR
GETFILE	- RETURN A FILE POINTER BASED ON INPUT FROM THE USER
CALLOC	
FLANCI	- FLAN CALLABLE INTERFACE
FOPEN	
STRCAT	
STRCPY	
WRTFRM	- WRITE FORM
RWOPN	- REPORT WRITER OPEN FORMS
GENMAIN	- GENERATE MAIN PROGRAM
BLDSUB	- BUILD SUBROUTINES
GENACT	- GENERATE ACTIONS
GENPAG	- GENERATE NEWPAG PROCEDURE
NDMLGEN	- NDML COBOL APPLICATION GENERATOR
PMSGLC	
OISCR	
TERMFP	
TRMNDML	
STRCHR	

APPLICATION GENERATOR Module Documentation

NAME: HASDATA
PURPOSE: DETERMINE IF THERE ARE ANY SELECT STATEMENTS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

THAT TARGET TO THE SCOPE OF THIS FORM.

SYNOPSIS

HASDATA(DP)
FIELD *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
(DP) - FIELD POINTER

OUTPUTS:
HASDATA RETURNS A TRUE OR A FALSE VALUE DEPENDING ON
WHETHER
ANY DATA WERE FOUND.

DESCRIPTION

THIS ROUTINE TRAVERSES THE FORM PROCESSOR DATA HIERARCHY TO
DETERMINE IF ANY SELECT STATEMENT TARGETS TO AN ITEM
WITHIN THE
SCOPE INDICATED BY THE FIELD POINTER WHICH IS PASSED IN AS
AN
INPUT PARAMETER. THE SCOPE IS DETERMINED BY NOT PROCESSING
PAST WINDOWS.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION

FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

HASDATA - DETERMINE IF THERE ARE ANY SELECT STATEMENTS

CALLED DIRECTLY BY:

GENAR - GENERATE ACTION PRESENT
BLDSUB - BUILD SUBROUTINES
HASDATA - DETERMINE IF THERE ARE ANY SELECT STATEMENTS
VISITA - VISIT ARRAYS ON THIS FORM
SETNDP - SET NODUPLICATE FIELDS TO BLANK IF THEY ARE
 DUPLICATED
RSETNDP - RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: HASITEM
PURPOSE: THIS ROUTINE DETERMINES IF THERE IS AN
ITEM WITHIN
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

THE SCOPE OF REFERENCE.

SYNOPSIS

HASITEM(DP)
FIELD *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
(DP) - FIELD POINTER

OUTPUTS:
RETURNS TRUE IF AN ITEM IS WITHIN THE SCOPE OF REFERENCE.

DESCRIPTION

THIS ROUTINE TRAVERSES THE FORMS HIERARCHY LOOKING FOR
ITEMS.
THE SCOPE OF REFERENCE IS DETERMINED BY NOT TRAVERSING PAST
OPEN ENDED ARRAYS OR WINDOWS. THE ROUTINE STOPS WHEN AN
ITEM
IS FOUND.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
 WITHIN

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM
GENAR - GENERATE ACTION PRESENT
FRMPDAT - FORM PDATA
MKINC - MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA
 MAKINC)
GENFS - GENERATE FORM DATA STRUCTURES
GENFSD - GENERATE FORM STRUCTURE DATA INITIALIZATION
BSCODE - BUILD SUBROUTINE CODE
VISITA - VISIT ARRAYS ON THIS FORM
HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
 WITHIN

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: HASLOWER
PURPOSE: HAS A LOWER FORM WHICH READS THE SAME
DATA RECORD?
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

HASLOWER(FP, VP)
FIELD *FP;
VARLST *VP;

INPUTS/OUTPUTS:
NONE

INPUTS:
FP - FIELD POINTER
VP - VARIABLE LIST FROM A SELECT

OUTPUTS:
NONE

DESCRIPTION

CHECKS THE FORM FP TO SEE IF ANY OF THE VARIABLES IN THE
SELECT LIST
TARGET TO A FORM WHICH IS LOWER IN THE HIERARCHY THAN FP.
USED BY
CHKGRP AND READDB.

ARGUMENTS:

FP = FIELD *
VP = VARLST *

INCLUDE FILES:

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

CALLED DIRECTLY BY:

READDB - READ DATA BASE
CHKGRP - CHECK FOR GROUP SEPERATORS OR END OF FILE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: INDENT
PURPOSE: INDENT A LINE OF GENERATED CODE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

M = INT

INCLUDE FILES:

STDTPP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

PUTC

CALLED DIRECTLY BY:

MAKWHE/COBWHE - COBOL WHERE ES
MAKWHE - MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
SELGEN - SELECT GENERATE
ASSIGN - ASSIGN FILE SECTION
FD - FD SECTION DECLARATIONS
CLSFIL - CLOSE FILES
ENDGEN - END GENERATE
PROCEN - PROCEDURE DIVISION GENERATE
DATAGEN - DATA DIVISION GENERATE
FILELNK - FILE LINKAGE SECTION GENERATE
OPNFIL - GENERATE OPEN FILE SECTION
USING - GENERATE USING SECTION
SELWS - SELECT WORKING STORAGE SECTION
INSWS - INSERT WORKING STORAGE SECTION

INSERT	- INSERT PROCEDURE
NDMLLAB	- GENERATE LABELS
COBPE	- COBOL PE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: INSERT
PURPOSE: INSERT PROCEDURE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

INDENT - INDENT A LINE OF GENERATED CODE
FPRINTF
GETCOL - GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN
STRING
DASH - WRITE DASH '-'

CALLED DIRECTLY BY:

STDCODE - STANDARD COBOL CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: INSRSV
PURPOSE: INSERT RESOLVE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
VOID INSRSV(INSPTR, TRGPTR, ACTPTR)
    INSERT *INSPTR;
    TRGLST *TRGPTR;
    ACTLST *ACTPTR;
```

INPUTS:

INSPTR - INSERT FROM WHICH TO LOOK FOR PATH.
TRGPTR - CONDITION THIS INSERT IS ASSOCIATED WITH.
ACTPTR - ACTION THIS INSERT IS ASSOCIATED WITH.

DESCRIPTION

RESOLVES ALL QUALIFIED NAMES INTO FIELD POINTERS FOR ALL
NAMES
WHICH ARE ROOTED IN INSERT (SELECT, VALUE LIST).

ARGUMENTS:

```
-----
INSPTR =      INSERT *
TRGPTR =      TRGLST *
ACTPTR =      ACTLST *
```

INCLUDE FILES:

```
-----
STDTPY - STANDARD TYPE DEFINITIONS
FPD    - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW     - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

```
-----
UQPTH - UNIVERSAL QUALIFIER PATH
ERROR - ISSUE ERROR MESSAGE
GETPTH - GET PATH
```

CALLED DIRECTLY BY:

ACTRSV - ACTION RESOLVE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: INSWS
PURPOSE: INSERT WORKING STORAGE SECTION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT

INCLUDE FILES:

STDTP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
FPPARM	- FORM PROCESSOR PARAMETERS
FPCODE	- FORM PROCESSOR RETURN CODES
RW	- REPORT WRITER DEFINITIONS
NTM	- NTM INTERFACE INCLUDE FILE
CTLCHR	- CONTROL CHARACTERS

ROUTINES CALLED:

NULBLK	- BLANK FILL A STRING
STRCPY	
DASH	- WRITE DASH '-'
INDENT	- INDENT A LINE OF GENERATED CODE
FPRINTF	
MAKES	- MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
SAVEES	- SAVE ES INFORMATION

CALLED DIRECTLY BY:

DATAGEN - DATA DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: ISOPNE
PURPOSE: DETERMINE IF THIS FIELD IS OPEN ENDED
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

ISOPNE (DP)
FIELD *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
(DP) - FIELD POINTER

OUTPUTS:
THIS ROUTINE RETURNS TRUE IF THIS FIELD
IS OPEN ENDED.

DESCRIPTION

THIS LOOKS UP THE FORMS HIERARCHY TREE TO DETERMINE IF ITS
AN FORM OF AN OPEN ENDED ARRAY.

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
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

CALLED DIRECTLY BY:

GENAR - GENERATE ACTION PRESENT
GENAS - GENERATE ACTION SET

GENFS	- GENERATE FORM DATA STRUCTURES
GENFSD	- GENERATE FORM STRUCTURE DATA INITIALIZATION
MAKQR	- MAKE QUALIFIED REFERENCE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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 ****
STDYYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
MATH - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

MYALLOC - MY MALLOC

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKES
PURPOSE: MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
MAKES (LANG, SPTR, REC_CNT_PTR)
INT LANG;
TBLLST *TPTR;
INT SELNO;
INT *REC_CNT_PTR;
```

DESCRIPTION

WRITES A RECORD STRUCTURE ON A FILE IN THE CURRENT
DIRECTORY FOR THE GIVEN TABLE OR VIEWNAME.
ALSO CREATES A EDIT CONVERSION RECORD STRUCTURE FOR THE
EACH
EXTERNAL SCHEMA DATA ITEM

ARGUMENTS:

```
LANG = INT
TBLNAM = CHAR *
TBLNUM = INT
SELNO = INT
REC_CNT_PTR = INT *
```

INCLUDE FILES:

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

ROUTINES CALLED:

```
CDMESQY - PROGRAM NAME CDMESQY
ERROR - ISSUE ERROR MESSAGE
NULBLK - BLANK FILL A STRING
DASH - WRITE DASH '-'
CES - C ES
```

COBES	- COBOL ES RECORD
CCONV	- C CONVERSIONS
COBCONV	- COBOL CONVERSIONS
STRCPY	
STRNCPY	
STRLEN	

CALLED DIRECTLY BY:

SELWS	- SELECT WORKING STORAGE SECTION
INWS	- INSERT WORKING STORAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN	- GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKES/CNUMPIC
PURPOSE: C NUMBERS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

M = INT
T = INT

INCLUDE FILES:

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

ROUTINES CALLED:

FPRINTF
STRCAT

CALLED DIRECTLY BY:

COBCONV - COBOL CONVERSIONS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKES/INDENT
PURPOSE: INDENT
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

M = INT
T = INT

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITE . DEFINITIONS

ROUTINES CALLED:

PUTC

CALLED DIRECTLY BY:

CES - C ES
COBES - COBOL ES RECORD
CCONV - C CONVERSIONS
COBCONV - COBOL CONVERSIONS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKES/NUMPIC
PURPOSE: NUMBER PICTURE CLAUSE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

M = INT
T = INT

INCLUDE FILES:

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

ROUTINES CALLED:

FPRINTF

CALLED DIRECTLY BY:

COBES - COBOL ES RECORD
COBCONV - COBOL CONVERSIONS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKINS
PURPOSE: MAKE INSERT
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PSSTRC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT
IPTR = INSERT *

INCLUDE FILES:

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

ROUTINES CALLED:

PSSTRC/INDENT - INDENT
FPRINTF
PSSTRC/CSUB - C SUBSTITUTE
PSSTRC/COBSUB - COBOL SUBSTITUTE

CALLED DIRECTLY BY:

NDMLLNK - LINKAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

----- SYNOPSIS

ENODE *MAKINT(EP)
ENODE *EP;

DESCRIPTION

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

ARGUMENTS:

EP = ENODE *

INCLUDE FILES:

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

ROUTINES CALLED:

MYALLOC - MY MALLOC

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKPS
PURPOSE: MAKES THE PRESENTATION SCHEMA RECORD
STRUCTURE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PSSTRC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
MAKPS(LANG, SPTR)
INT LANG;
SELECT *SPTR;
```

DESCRIPTION

WRITES A RECORD STRUCTURE ON A FILE IN THE CURRENT
DIRECTORY FOR THE GIVEN SELECT.
THE RECORD STRUCTURE INCLUDES ALL THE FIELDS ON THE FORM
THAT THE
SELECT IS SELECTING INTO.

ARGUMENTS:

```
LANG =      INT
SPTR =      SELECT *
```

INCLUDE FILES:

```
STD TYP      - STANDARD TYPE DEFINITIONS
STD IO      - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD         - FORM PROCESSOR DATA
FPPARM      - FORM PROCESSOR PARAMETERS
FPCODE      - FORM PROCESSOR RETURN CODES
RW          - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

```
PSSTRC/CSUB - C SUBSTITUTE
PSSTRC/COBSUB - COBOL SUBSTITUTE
```

CALLED DIRECTLY BY:

```
SELWS      - SELECT WORKING STORAGE SECTION
```

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKQR
PURPOSE: MAKE QUALIFIED REFERENCE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

```
CHAR *MAKQR(DP, SUFFIX, S1, TFLDP, AFLDP)
    FIELD *DP;
    CHAR SUFFIX;
    CHAR S1[];
    FLDLST *TFLDP, *AFLDP;
```

INPUTS/OUTPUTS:
NONE

INPUTS:

DP - FIELD POINTER TO AN ITEM.
SUFFIX - THIS IS A CHARACTER VALUE OF EITHER C OR P
TO REPRESENT CURRENT OR PREVIOUS.
TFLDP - LIST OF FIELDS TO GENERATE A "TINDX%D" INDEX
REFERENCE.
AFLDP - LIST OF FIELDS TO GENERATE A "AINDX%D" INDEX
REFERENCE.

OUTPUTS:

S1 - THIS IS THE QUALIFIED REFERENCE CHARACTER STRING

DESCRIPTION

THIS ROUTINE STARTS AT THE ITEM POINTER LOOKING UP THE FORM
PROCESSOR HIERARCHY TO GENERATE A FULLY QUALIFIED
REFERENCE WHICH
CORRESPONDS TO THOSE CURRENT AND PREVIOUS DATA STRUCTURES
GENERATED
BY MAKINC.

ARGUMENTS:

DP = FIELD *
SUFFIX = CHAR
S1 = CHAR []
TFLDP = FLDLST *
AFLDP = FLDLST *

INCLUDE FILES:

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

ROUTINES CALLED:

STRCAT
STRCPY
SPRINTF
ISOPNE - DETERMINE IF THIS FIELD IS OPEN ENDED

CALLED DIRECTLY BY:

GENTRG - GENERATE TRIGGERS
GENAS - GENERATE ACTION SET
GENAI - GENERATE ACTION INSERT
SELWHR - SELECT WHERE
GENFSD - GENERATE FORM STRUCTURE DATA INITIALIZATION
MAPDB - MAP DATABASE
VISITA - VISIT ARRAYS ON THIS FORM
SETNDP - SET NODUPLICATE FIELDS TO BLANK IF THEY ARE
DUPLICATED
RSETNDP - RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
CALCSTAT - CALCULATE STATISTIC
RSETSTAT - RESET STATISTIC

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

----- SYNOPSIS

ENODE *MAKSTR(EP)
ENODE *EP;

DESCRIPTION

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

ARGUMENTS:

EP = ENODE *

INCLUDE FILES:

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

ROUTINES CALLED:

MYALLOC - MY MALLOC

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKWH
PURPOSE: MAKE WHERE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PSSTRC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT
SPTR = SELECT *

INCLUDE FILES:

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

ROUTINES CALLED:

PSSTRC/INDENT - INDENT
FPRINTF
PSSTRC/CSUB - C SUBSTITUTE
PSSTRC/COBSUB - COBOL SUBSTITUTE

CALLED DIRECTLY BY:

NDMLLNK - LINKAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKWHES
PURPOSE: MAKE THE WHERE CLAUSE EXTERNAL SCHEMA
VARIABLES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKWHES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
MAKWHES(LANG, SPTR)
INT LANG;
SELECT *SPTR;
```

DESCRIPTION

WRITES A WHERE CLAUSE EXTERNAL SCHEMA RECORD STRUCTURE
FOR ALL
EXTERNAL SCHEMA COLUMNS THAT MAP TO PRESENTATION ITEMS IN
THE
WHERE CLAUSE OF THE SELECT. IT IS ALLOWABLE FOR ONE ES
ITEM TO
MAP TO MORE THAN ONE PS ITEM

ARGUMENTS:

```
LANG = INT
SPTR = SELECT *
```

INCLUDE FILES:

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

ROUTINES CALLED:

```
INDENT - INDENT A LINE OF GENERATED CODE
FPRINTF
MAKWHES/CWHES - C WHERE ES
MAKWHES/COBWHES - COBOL WHERE ES
```


CALLED DIRECTLY BY:

SELWS - SELECT WORKING STORAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKWHES/COBWHES
PURPOSE: COBOL WHERE ES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKWHES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

ESWH_PTR = PREDOPER *
COLWH_PTR = PREDOPER *
SPTR = SELECT *
LOOPCNT = INT

INCLUDE FILES:

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

ROUTINES CALLED:

MAKWHES/NUMPIC - NUMBER PICTURE CLAUSE
FPRINTF
INDENT - INDENT A LINE OF GENERATED CODE
DASH - WRITE DASH '-'
GETTBL - GET A TABLE NAME
GETCOL - GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN
STRING

CALLED DIRECTLY BY:

MAKWHES - MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKWHES/CWHES
PURPOSE: C WHERE ES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKWHES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

ESWH_PTR = PREDOPER *
COLWH_PTR = PREDOPER *
SPTR = SELECT *
LOOPCNT = INT

INCLUDE FILES:

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

CALLED DIRECTLY BY:

MAKWHES - MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAKWHES/NUMPIC
PURPOSE: NUMBER PICTURE CLAUSE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MAKWHES
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

M = INT
T = INT

INCLUDE FILES:

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

ROUTINES CALLED:

FPRINTF

CALLED DIRECTLY BY:

MAKWHES/COBWHES - COBOL WHERE ES

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MAPDB
PURPOSE: MAP DATABASE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

INPUTS/OUTPUTS:
NONE

INPUTS:
(DP) - FIELD POINTER

OUTPUTS:
NONE

DESCRIPTION

TRAVERSES ALL SELECTS LOOKING FOR ONES THAT TARGET TO THE
SCOPE
OF THE FORM INDICATED BY THE INPUT PARAMETER. IT GENERATES
STATEMENTS OF THE FORM:

MEMCPY(FRMPTR->FIELD, DBR%D.FIELD, SIZE); %D - NUMBER OF
SELECT.

ARGUMENTS:

FP = FIELD *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

STRCHR
MAKQR - MAKE QUALIFIED REFERENCE
SPRINTF
STRLEN
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

BSCODE - BUILD SUBROUTINE CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: MKINC
PURPOSE: MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENMN2
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

MKINC(FP)
FIELD *FP;

INPUTS/OUTPUTS:
NONE

INPUTS:
FP - FORM POINTER

OUTPUTS:
NONE

DESCRIPTION

GENERATES THE STRUCTURE TAGS FOR ALL THE FORMS USED IN AN APPLICATION OR REPORT. THIS UPPER LEVEL PROCEDURE TRAVERSES ALL FORMS WHICH ARE PRESENTED IN WINDOWS. THE DATA STRUCTURES ARE OF THE FORM:

```
#IFNDEF FRM7                                MAKE SURE THE FORM IS
                                           DECLARED ONCE ONLY.
STRUCT FRM7                                STRUCTURE TAG.
{
    CHAR DBID[4];                          DATA FIELDS (ITEMS) ON
                                           THE FORM.
    CHAR DBNAME[20];
    CHAR HOSTID[3];
    CHAR DBMSNAME[30];
} ; (* INSRT *)
#define FRM7                                DEFINE A SYMBOL.
#endif
#IFNDEF FRM3
STRUCT FRM3
{
    CHAR PDATE[10];
    STRUCT FRM7 FRM7[10]; (* INSRT *)    A SUBFORM OF
```

```
                FORM 3.  
            } ; (* SEL45 *)  
#DEFINE FRM3  
#ENDIF
```

ARGUMENTS:

FP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

MKINC - MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA
MAKINC)
HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
WITHIN
SPRINTF
STRLEN
BLEN
GEN - GENERATE A LINE OF CODE
ERROR - ISSUE ERROR MESSAGE

CALLED DIRECTLY BY:

GENMAIN - GENERATE MAIN PROGRAM
MKINC - MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA
MAKINC)

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

```
POS *MKPOS(HPOS, HMIN, HLOC, HREF, VPOS, VMIN, VLOC, VREF)
INT HPOS, HMIN, HLOC;
CHAR *HREF;
INT VPOS, VMIN, VLOC;
CHAR *VREF;
```

DESCRIPTION

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

ARGUMENTS:

```
HPOS = INT
HMIN = INT
HLOC = INT
HREF = CHAR *
VPOS = INT
VMIN = INT
VLOC = INT
VREF = CHAR *
```

INCLUDE FILES:

```
STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
```

ROUTINES CALLED:

MYALLOC - MY MALLOC

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Modul documentation

NAME: MLPFRM
PURPOSE: MAKE A LIST OF PRESENTED FORMS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
MLPFRM()

DESCRIPTION

MAKES TWO LISTS OF PRESENTED FORMS. ONE LIST POINTED TO BY
PRSFRM,
CONTAINS ALL PRESENTED FORMS. THE SECOND LIST POINTED TO
BY TOPFRM,
CONTAINS ALL FORMS PRESENTED IN THE WINDOW SCREEN.

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

FNDFRM - FIND FORM
ERROR - ISSUE ERROR MESSAGE
MALLOC

CALLED DIRECTLY BY:

RWOPN - REPORT WRITER OPEN FORMS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

```
CHAR *MYALLOC(SIZE)  
    UNSIGNED SIZE;
```

DESCRIPTION

ALLOCATE THE SPECIFIED MEMORY IF POSSIBLE, ELSE ISSUE
FATAL ERROR

ARGUMENTS:

SIZE = UNSIGNED

INCLUDE FILES:

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

ROUTINES CALLED:

```
FATAL - ISSUE FATAL ERROR MESSAGE  
MALLOC
```

CALLED DIRECTLY BY:

```
CHKFLD - CHECK FIELD  
CHKARY - CHECK ARRAY  
CSTASH - CHARACTER STASH  
WRTEXP - WRITE EXPRESSION  
MKPOS - MAKE POSITION NODE  
MAKINT - MAKE EXPRESSION INTO AN INTEGER  
MAKSTR - MAKE EXPRESSION INTO A STRING  
MAKACT - MAKE ACTION LIST ELEMENT  
YYPARSE - FLAN PARSER
```

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: NDMLGEN
PURPOSE: NDML COBOL APPLICATION GENERATOR
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
NDMLGEN()

DESCRIPTION

CALLS THE APPROPRIATE ROUTINES TO GENERATE THE
PRESENTATION
SCHEMA RECORD STRUCTURE, THE EXTERNAL SCHEMA RECORD
STRUCTURE
AND THE CONVERSION CODE TO GO FROM ONE CDM DATA TYPE TO
ANOTHER.
AND THE NDML COMMANDS SPECIFIED.

ARGUMENTS:

LANG = INT
APNAME = CHAR []

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

STRNCPY
SPRINTF
FOPEN
SYSMSG
STDCODE - STANDARD COBOL CODE
FCLOSE

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: NDMLLAB
PURPOSE: GENERATE LABELS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

TYPE = CHAR

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

INDENT - INDENT A LINE OF GENERATED CODE
FPRINTF

CALLED DIRECTLY BY:

PROCGEN - PROCEDURE DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: NDMLLNK
PURPOSE: LINKAGE SECTION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT
TYPE = CHAR

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

MAKWH - MAKE WHERE
MAKINS - MAKE INSERT

CALLED DIRECTLY BY:

DATAGEN - DATA DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: NULBLK
PURPOSE: BLANK FILL A STRING
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

TMPSTR = CHAR []
INSTR = CHAR []

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

STRCHR
STRCPY

CALLED DIRECTLY BY:

MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
SELGEN - SELECT GENERATE
SAVEES - SAVE ES INFORMATION
SELWS - SELECT WORKING STORAGE SECTION
INSWS - INSERT WORKING STORAGE SECTION
GETTBL - GET A TABLE NAME

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: OPNFIL
PURPOSE: GENERATE OPEN FILE SECTION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SPTR = SELECT *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

INDENT - INDENT A LINE OF GENERATED CODE
FPRINTF
OPNFIL - GENERATE OPEN FILE SECTION

CALLED DIRECTLY BY:

SELGEN - SELECT GENERATE
PROCGEN - PROCEDURE DIVISION GENERATE
OPNFIL - GENERATE OPEN FILE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: PEMAP
PURPOSE: THE PRESENTATION SCHEMA AND THE EXTERNAL
SCHEMA AND MAPPING
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PEMAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
PEMAP(LANG, STR1,STR2,FPTR, DPTR)
INT LANG;
CHAR *STR1;
CHAR *STR2;
FIELD *FPTR;
STRUCT DTYPE *DPTR;
```

DESCRIPTION

GENERATES THE CODE TO TRANSFORM AN PRESENTATION SCHEMA
DATA ITEM INTO
A EXTERNAL SCHEMA ITEM. THIS IS DONE ON A PER ITEM BASIS
AND THE
SOURCE AND DESTINATION STRINGS OF CODE (STR1, STR2) ARE
PASSED IN SO
THE RESULTING CODE MAY USE THE CORRECT VARIABLES.

ARGUMENTS:

```
-----
LANG =      INT
STR1 =      CHAR *
STR2 =      CHAR *
FPTR =      FIELD
DPTR =      CDMDTYPE *
```

INCLUDE FILES:

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

ROUTINES CALLED:

CPE - C PE
COBPE - COBOL PE

CALLED DIRECTLY BY:

SELGEN - SELECT GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: PROCGEN
PURPOSE: PROCEDURE DIVISION GENERATE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT
TYPE = CHAR

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

MAP
USING - GENERATE USING SECTION
OPNFIL - GENERATE OPEN FILE SECTION
NDMLLAB - GENERATE LABELS
CLSFIL - CLOSE FILES
INSMAP
SELMAP - MAP SELECTED DATA TO OUTPUT RECORD
INDENT - INDENT A LINE OF GENERATED CODE
FPRINTF
PSESMAP

CALLED DIRECTLY BY:

STDCODE - STANDARD COBOL CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: PSSTRC/COBSUB
PURPOSE: COBOL SUBSTITUTE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PSSTRC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

BLN
FPRINTF
PSSTRC/INDENT - INDENT

CALLED DIRECTLY BY:

MAKPS - MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
MAKWH - MAKE WHERE
MAKINS - MAKE INSERT

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: PSSTRC/CSUB
PURPOSE: C SUBSTITUTE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PSSTRC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

BLFN
FPRINTF
PSSTRC/INDENT - INDENT

CALLED DIRECTLY BY:

MAKPS - MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
MAKWH - MAKE WHERE
MAKINS - MAKE INSERT

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: PSSTRC/INDENT
PURPOSE: INDENT
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PSSTRC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

M = INT

INCLUDE FILES:

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

ROUTINES CALLED:

PUTC

CALLED DIRECTLY BY:

PSSTRC/CSU - C SUBSTITUTE
PSSTRC/COBSUB - COBOL SUBSTITUTE
MAKWH - MAKE WHERE
MAKINS - MAKE INSERT

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: READDB
PURPOSE: READ DATA BASE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

READDB(FP)
FIELD *FP;

INPUTS/OUTPUTS:
NONE

INPUTS:
FP - FIELD POINTER

OUTPUTS:
NONE

DESCRIPTION

TRAVERSES THE LIST OF SELECTS LOOKING FOR ONES THAT TARGET
TO ITEMS
ON THE FORM INDICATED BY THE INPUT PRARMETER. WHEN ONE IS
FOUND IT
CALLS DBFREAD TO READ A DATA RECORD AND CHECK FOR CHANGE
CONDITIONS.

ARGUMENTS:

FP = FIELD *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

HASLOWER - HAS A LOWER FORM WHICH READS THE SAME DATA
 RECORD?
DBFREAD - GENERATE DATA BASE FREAD

CALLED DIRECTLY BY:

BSCODE - BUILD SUBROUTINE CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: RSETNDP
PURPOSE: RESET NODUPPLICATE FIELDS TO VALUE OF
NODUP%D
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

RSETNDP(FP, DP)
FIELD *FP, *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
FP - PARENT FORM OF DP (HELP IN RECURSION).
DP - FIELD THAT MIGHT HAVE NODUP OPTION.

OUTPUTS:
NONE

DESCRIPTION

TRAVERSES THE FORM HIERARCHY LOOKING FOR ITEMS UNDER FP
WHICH HAVE
THE NODUP OPTION. WHEN IT FINDS ONE IT GENERATES CODE TO
COPY THE
NODUP%D VALUE TO THE FORM FIELD.

ARGUMENTS:

FP = FIELD *
DP = FIELD *

INCLUDE FILES:

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

ROUTINES CALLED:

RSETNDP - RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D
MAKQR - MAKE QUALIFIED REFERENCE
SPRINTF
STRLEN
GEN - GENERATE A LINE OF CODE
HASDATA - DETERMINE IF THERE ARE ANY SELECT STATEMENTS

CALLED DIRECTLY BY:

BSCODE - BUILD SUBROUTINE CODE
RSETNDP - RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: RSETSTAT
PURPOSE: RESET STATISTIC
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION: -----

ARGUMENTS: -----

FP = FIELD *
DP = FIELD *

INCLUDE FILES: -----

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED: -----

RSETSTAT - RESET STATISTIC
MAKQR - MAKE QUALIFIED REFERENCE
SPRINTF
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY: -----

FRMPDAT - FORM PDATA
RSETSTAT - RESET STATISTIC

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

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: RWEXPD
PURPOSE: REPORT WRITER EXPAND ARRAYS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
CHAR *RWEXPD(FDP, USELST)
    FIELD *FDP;
    FIELD **USELST;
```

INPUTS:

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

DESCRIPTION

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

ARGUMENTS:

```
-----
FDP =      FIELD *
USELST =    FIELD **
```

INCLUDE FILES:

```
-----
STD TYP      - STANDARD TYPE DEFINITIONS
FPD          - FORM PROCESSOR DATA
FPCODE       - FORM PROCESSOR RETURN CODES
RW           - REPORT WRITER DEFINITIONS
```


ROUTINES CALLED:

COPFLD
ABS
FNDATT - FIND ATTRIBUTE
STRASN
RWSP/FIXFRM - FIX UP A FORM

CALLED DIRECTLY BY:

RWSP/FIXFR - FIX UP A FORM
RWOPN - REPORT WRITER OPEN FORMS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: RWOPN
PURPOSE: REPORT WRITER OPEN FORMS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
VOID RWOPN()

DESCRIPTION

CREATES AN "OPEN LIST" OF FORMS. FROM THE STRUCTURES
CREATED BY FLAN
SUBFORMS ARE COPIED IN PLACE AND ARRAYS ARE EXPANDED TO
THEIR FULL
SIZE.

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

RWEYPD - REPORT WRITER EXPAND ARRAYS
MLPFRM - MAKE A LIST OF PRESENTED FORMS
WINRSV - WINDOW RESOLVE
FLDRSV - FIELD RESOLVE
TRGRSV - TRIGGER RESOLVE

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

CHAR *FIXFRM(DP, USELST)
FIELD *DP;
FIELD **USELST;

INPUTS:

DP - DUMMY FORM FIELD TO BE FIXED UP.
USELST - WHERE TO LOOK FOR THE SUBFORM TO COPY.

DESCRIPTION

FIXES A SUBFORM BY LOCATING IT AND ATTACHING IT IN PLACE
AND
EXPANDING IT IF REQUIRED.

ARGUMENTS:

DP = FIELD *
USELST = FIELD **

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

RWEXPD - REPORT WRITER EXPAND ARRAYS
COPFLD
FNDATT - FIND ATTRIBUTE
STRCMP

PS 620344502
30 September 1990

CALLED DIRECTLY BY:

RWEXPD - REPORT WRITER EXPAND ARRAYS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SAVEES
PURPOSE: SAVE ES INFORMATION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

ITMNAM = CHAR [CDMCOLNAMLEN +1]
DPTR = CDMdtype *
REC_CNT = INT

INCLUDE FILES:

STDtyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

NULBLK - BLANK FILL A STRING
STRNCMP
STRLEN
ESCPY
ATOI

CALLED DIRECTLY BY:

SELWS - SELECT WORKING STORAGE SECTION
INSWS - INSERT WORKING STORAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SELECT
PURPOSE: GENERATE SELECT CODE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

SELGEN - SELECT GENERATE

CALLED DIRECTLY BY:

STDCODE - STANDARD COBOL CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SELGEN
PURPOSE: SELECT GENERATE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT
PPTR = SELECT *
SPTR = SELECT *
TOPSEL = SELECT *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

SELGEN - SELECT GENERATE
NULBLK - BLANK FILL A STRING
STRCPY
PEMAP - THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA
AND MAPPING
SPRINTF
DASH - WRITE DASH '-'
GETTBL - GET A TABLE NAME
GETCOL - GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN
STRING
OPNFIL - GENERATE OPEN FILE SECTION
FPRINTF
INDENT - INDENT A LINE OF GENERATED CODE

CALLED DIRECTLY BY:

SELGEN - SELECT GENERATE
SELECT - GENERATE SELECT CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SELLEN
PURPOSE: COMPUTE LENGTH OF SELECT PS RECORD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SPTR = SELECT *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

BLEN

CALLED DIRECTLY BY:

FD - FD SECTION DECLARATIONS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SELMAP
PURPOSE: MAP SELECTED DATA TO OUTPUT RECORD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

LANG = INT
SPTR = SELECT *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

ESPSMAP - THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA
MAPPING
SELMAP - MAP SELECTED DATA TO OUTPUT RECORD

CALLED DIRECTLY BY:

PROCEN - PROCEDURE DIVISION GENERATE
SELMAP - MAP SELECTED DATA TO OUTPUT RECORD

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SELOPN
PURPOSE: SELECT OPEN
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIPECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

SELOPN(SP)
SELECT *SP;

INPUTS:

SP - POINTER TO SELECT TO HAVE ITS DATA FILE OPENED.

DESCRIPTION

GENERATES CODE TO OPEN THE DATA FILE ASSOCIATED WITH THIS
SELECT
ACTION.

ARGUMENTS:

SP = SELECT *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

SPRINTF
GEN - GENERATE A LINE OF CODE
DBFREAD - GENERATE DATA BASE FREAD
SELOPN - SELECT OPEN

CALLED DIRECTLY BY:

GENAQ - GENERATE ACTION QUERY (SELECT)
SELOPN - SELECT OPEN

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SELRSV
PURPOSE: SELECT RESOLVE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

```
VOID SELRSV(SELPTR, TRGPTR, ACTPTR)
  SELECT *SELPTR;
  TRGPTR *TRGPTR;
  ACTPTR *ACTPTR;
```

INPUTS:

SELPTR - SELECT FROM WHICH TO LOOK FOR PATHS.
TRGPTR - CONDITION THIS SELECT IS ASSOCIATED WITH.
ACTPTR - ACTION THIS SELECT IS ASSOCIATED WITH.

DESCRIPTION

RESOLVES ALL QUALIFIED NAMES INTO FIELD POINTERS FOR ALL
NAMES
WHICH ARE ROOTED IN SELECT (SELECT, VARIABLE LIST, WHERE
LIST).

ARGUMENTS:

SELPTR = SELECT *
TRGPTR = TRGLST *
ACTPTR = ACTLST *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

SELRSV - SELECT RESOLVE
UQPTH - UNIVERSAL QUALIFIER PATH
ERROR - ISSUE ERROR MESSAGE
GETPTH - GET PATH

CALLED DIRECTLY BY:

SELRSV - SELECT RESOLVE
ACTRSV - ACTION RESOLVE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SELWHR
PURPOSE: SELECT WHERE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
SELWHR(SP, TP, AP)
    SELECT *SP;
    TRGLST *TP;
    ACTLST *AP;
```

INPUTS:

SP - POINTER TO SELECT ACTION (NEEDED SINCE SELECTS CAN BE
NESTED).
TP - CONDITION ASSOCIATED WITH THIS ACTION.
AP - THIS ACTION.

DESCRIPTION

GENERATES CODE TO COPY DATA FROM A FORM STRUCTURE TO THE
WHERE STRUCTURE
FOR THOSE SELECTS WHICH HAVE A QUALIFIED NAME IN THE WHERE
CLAUSE.

ARGUMENTS:

```
SP =      SELECT *
TP =      TRGLST *
AP =      ACTLST *
```

INCLUDE FILES:

```
STDTPF    - STANDARD TYPE DEFINITIONS
FPD        - FORM PROCESSOR DATA
RW         - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

```
SPRINTF    -
MAKQF      - MAKE QUALIFIED REFERENCE
GEN        - GENERATE A LINE OF CODE
SELWHR     - SELECT WHERE
```

CALLED DIRECTLY BY:

GENAQ - GENERATE ACTION QUERY (SELECT)
SELWHR - SELECT WHERE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SELWS
PURPOSE: SELECT WORKING STORAGE SECTION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SPTR = SELECT *
LANG = INT

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

MAEPS - MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
NUIBLK - BLANK FILL A STRING
SICOPY
DASH - WRITE DASH '-'
INDENT - INDENT A LINE OF GENERATED CODE
FPRINTF
MAKWHES - MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
SILWS - SELECT WORKING STORAGE SECTION
MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
SICOL - GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN
STRING
SITBI - GET A TABLE NAME
SICOMP
SAVEES - SAVE ES INFORMATION

CALLED DIRECTLY BY:

DATAGEN - DATA DIVISION GENERATE
SELWS - SELECT WORKING STORAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: SETNDP
PURPOSE: SET NODUPLICATE FIELDS TO BLANK IF THEY
ARE DUPLICATED
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

----- SYNOPSIS

SETNDP(FP, DP)
FIELD *FP, *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
FP - PARENT FORM OF DP (HELP IN RECURSION).
DP - FIELD THAT MIGHT HAVE NODUP OPTION.

OUTPUTS:
NONE

DESCRIPTION

TRAVERSES THE FORM HIERARCHY LOOKING FOR ITEMS UNDER FP
WHICH HAVE
THE NODUP OPTION. WHEN IT FINDS ONE IT GENERATES CODE TO
CHECK FOR
DUPLICATE VALUES AND BLANKS THE FORM FORM IF THERE ARE
DUPLICATE
VALUES.

ARGUMENTS:

FP - FIELD *
DP - FIELD *

INCLUDE FILES:

SIDIYF - STANDARD TYPE DEFINITIONS
SIDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPL - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION

FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

SETNDP - SET NODUPLICATE FIELDS TO BLANK IF THEY ARE
 DUPLICATED
MAQQR - MAKE QUALIFIED REFERENCE
SPRINTF
STRLEN
GEN - GENERATE A LINE OF CODE
HASDATA - DETERMINE IF THERE ARE ANY SELECT STATEMENTS

CALLED DIRECTLY BY:

BSCODE - BUILD SUBROUTINE CODE
SETNDP - SET NODUPLICATE FIELDS TO BLANK IF THEY ARE
 DUPLICATED

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: STATRSV
PURPOSE: STATISTIC RESOLVE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
STATRSV(STATPTR)  
  STATLST *STATPTR;
```

INPUTS:

STATPTR - STATISTIC LIST FROM WHICH TO LOOK FOR PATHS.

DESCRIPTION

RESOLVES ALL QUALIFIED NAMES INTO FIELD POINTERS FOR ALL
NAMES
WHICH ARE ROOTED IN STATLST (STATISTIC LIST).

ARGUMENTS:

```
STATPTR =          STATLST *
```

INCLUDE FILES:

```
STDTPY      - STANDARD TYPE DEFINITIONS  
FPD          - FORM PROCESSOR DATA  
FPCODE      - FORM PROCESSOR RETURN CODES  
RW          - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

```
GETPTH      - GET PATH  
ERROR       - ISSUE ERROR MESSAGE
```

CALLED DIRECTLY BY:

```
FLDRSV      - FIELD RESOLVE
```

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: STDCODE
PURPOSE: STANDARD COBOL CODE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

GENERATE THE NECESSARY DIVISIONS FOR COBOL AND THE DATA
STRUCTURES NECESSARY FOR NTM PROCESSING.
FOR A COBOL PROGRAM TO DO JUST NDML AND WRITE DATA TO FILES
MUST CONSTRUCT FILE SECTION CORRECTLY.
FOR A C PROGRAM WOULD BE DECLARING ALL NTM VARIABLES AS
EXTERNAL TO THE C GENERATED PROCEDURE.

ARGUMENTS:

LANG = INT
APNAME = CHAR *
TYPE = CHAR

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NIM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

ENDGEN - END GENERATE
SELECT - GENERATE SELECT CODE
SPRINTF
PROCGEN - PROCEDURE DIVISION GENERATE
FPRINTF
DATAGEN - DATA DIVISION GENERATE
STRUPC
INSERT - INSERT PROCEDURE

PS 620344502
30 September 1990

CALLED DIRECTLY BY:

NDMLGEN - NDML COBOL APPLICATION GENERATOR

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: TRGRSV
PURPOSE: TRIGGER RESOLVE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
TRGRSV(TRGPTR)
  TRGLST *TRGPTR;
```

INPUTS:

TRGPTR - CONDITION LIST FROM WHICH TO LOOK FOR PATHS.

DESCRIPTION

RESOLVES ALL QUALIFIED NAMES INTO FIELD POINTERS FOR ALL
NAMES
WHICH ARE ROOTED IN TRGLST (CONDITION LIST).

ARGUMENTS:

TRGPTR = TRGLST *

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

```
GETPTH - GET PATH
ERROR - ISSUE ERROR MESSAGE
ACTRSV - ACTION RESOLVE
UQPTH - UNIVERSAL QUALIFIER PATH
```

CALLED DIRECTLY BY:

RWOPN - REPORT WRITER OPEN FORMS

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: UQFOR
PURPOSE: UNIVERSAL QUALIFIER FOR LOOP
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
UQFOR(FLDP, TYPE)
    FLDLST *FLDP;
    CHAR TYPE;
```

INPUTS:

FLDP - POINTER TO LIST OF FIELDS WHICH REQUIRE UNIVERSAL
QUALIFICATION.

TYPE - 'T' FOR CONDITIONAL INDEX (TINDX%D) 'A' FOR ACTION
(AINDX%D).

DESCRIPTION

GENERATES THE FOR LOOP FOR UNIVERSAL QUALIFICATION.

ARGUMENTS:

FLDP =	FLDLST *
TYPE =	CHAR

INCLUDE FILES:

STDYTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
RW	- REPORT WRITER DEFINITIONS

ROUTINES CALLED:

SPRINTF	
GEN	- GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

GENTRG	- GENERATE TRIGGERS
GENAL	- GENERATE ACTION LIST

PS 620344502
30 September 1990

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: UQPTH
PURPOSE: UNIVERSAL QUALIFIER PATH
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

```
CHAR *UQPTH(PATH, DP, TFLDPP, AFLDPP)
CHAR  PATH[];
FIELD *DP;
FLDLST **TFLDPP, **AFLDPP;
```

INPUTS/OUTPUTS:

TFLDPP - POINTER TO POINTER OF CONDITION INDEX FIELDS.
AFLDPP - POINTER TO POINTER OF ACTION INDEX FIELDS.

INPUTS:

PATH - PATH WITH UNIVERSAL QUALIFIERS IN IT.
DP - FIRST INSTANCE OF PATH.

DESCRIPTION

MAKES A LIST OF FIELDS WHICH REQUIRE UNIVERSAL
QUALIFICATION FOR
A CONDITION AND ACTION.

ARGUMENTS:

```
PATH = CHAR []
DP = FIELD *
TFLDPP = FLDLST **
AFLDPP = FLDLST **
```

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
```

ROUTINES CALLED:

MALLOC

PTHPTR
STRCHR
STRCPY

CALLED DIRECTLY BY:

INSRSV - INSERT RESOLVE
SELRSV - SELECT RESOLVE
TRGRSV - TRIGGER RESOLVE
ACTRSV - ACTION RESOLVE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: USING
PURPOSE: GENERATE USING SECTION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

ARGUMENTS:

SPTR = SELECT *

INCLUDE FILES:

STDYTP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
FPPARM	- FORM PROCESSOR PARAMETERS
FPCODE	- FORM PROCESSOR RETURN CODES
RW	- REPORT WRITER DEFINITIONS
NTM	- NTM INTERFACE INCLUDE FILE
CTLCHR	- CONTROL CHARACTERS

ROUTINES CALLED:

INDENT	- INDENT A LINE OF GENERATED CODE
FPRINTF	

CALLED DIRECTLY BY:

PROCEN	- PROCEDURE DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN	- GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: VISITA
PURPOSE: VISIT ARRAYS ON THIS FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GRP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS

VISITA(DP)
FIELD *DP;

INPUTS/OUTPUTS:
NONE

INPUTS:
(DP) - FIELD POINTER

OUTPUTS:
NONE

DESCRIPTION

VISIT AN ARRAY BY GENERATING A LOOP TO GO THRU THE
ELEMENTS IN THE ARRAY.
CHECK FOR GROUP SEPERATORS/END OF FILE, OVERFLOW
CONDITIONS AND CALL THE
PROCEDURE WHICH IMPLEMENTS THE SUBFORM.

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
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

VISITA - VISIT ARRAYS ON THIS FORM
HASDATA - DETERMINE IF THERE ARE ANY SELECT STATEMENTS
GEN - GENERATE A LINE OF CODE
MAKQR - MAKE QUALIFIED REFERENCE
HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
 WITHIN
CHKGRP - CHECK FOR GROUP SEPERATORS OR END OF FILE
SPRINTF
STRLEN

CALLED DIRECTLY BY:

BSCODE - BUILD SUBROUTINE CODE
VISITA - VISIT ARRAYS ON THIS FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

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

DESCRIPTION

PRINTS A WARNING MESSAGE ON STDERR

ARGUMENTS:

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

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

```
PMSGLS
STRLEN
SPRINTF
```

CALLED DIRECTLY BY:

```
CHKFRM - CHECK FORM
YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER
GENAT - GENERATE ACTION SIGNAL
```

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: WINRSV
PURPOSE: WINDOW RESOLVE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: RWSP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: RW
DOCUMENTATION GROUP: RW/AP

DESCRIPTION:

SYNOPSIS
WINRSV()

DESCRIPTION

ALL FORMS WHICH ARE PRESENTED IN WINDOWS ARE ADDED TO
THOSE WINDOWS
SO QUALIFIED NAMES MAY BE RESOLVED INTO POINTERS.

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

MALLOC
GETPTH - GET PATH
FNDFRM - FIND FORM
COPFLD
FREE

CALLED DIRECTLY BY:

RWOPN - REPORT WRITER OPEN FORMS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

CHAR *WRTEXP(EP)
ENODE *EP;

INPUTS:

EP - EXPRESSION TO WRITE

OUTPUTS:

RETURNS A POINTER TO THE WRITTEN EXPRESSION OR NULL
FOR ERRORS

DESCRIPTION

RETURNS A POINTER TO THE CHARACTER STRING REPRESENTING
THE GIVEN
EXPRESSION, OR NULL IF AN ERROR IS DETECTED.

ARGUMENTS:

EP = ENODE *

INCLUDE FILES:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

CHKFLD - CHECK FIELD
WRTEXP - WRITE EXPRESSION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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:

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

ROUTINES CALLED:

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

STRASN
STRCPY
STRLEN

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

----- SYNOPSIS

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

DESCRIPTION

WRITES THE LAST LINE OF THE DEFAULT LINE BUFFER.

ARGUMENTS:

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

INCLUDE FILES:

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

ROUTINES CALLED:

FWRITE

CALLED DIRECTLY BY:

WRTFRM - WRITE FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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:

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

ROUTINES CALLED:

FWRITE

CALLED DIRECTLY BY:

WRTFRM - WRITE FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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: FDFE/FLAN

DESCRIPTION:

SYNOPSIS

```
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:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** 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):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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

```
WRTFLD(FPTR, FLDPTR)
FILE *FPTR;
FIELD *FLDPTR;
```

DESCRIPTION

WRITES THE FIELD RECORD FOR THE SPECIFIED FIELD STRUCTURE.

ARGUMENTS:

```
FPTR = FILE *
FLDPTR = FIELD *
```

INCLUDE FILES:

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

ROUTINES CALLED:

```
FWRITE
STRCPY
WRTFRM/FORMAT - INSERT FORMAT CODES
STRNCPY
MEMCPY
```

CALLED DIRECTLY BY:

```
WRTFRM - WRITE FORM
```

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR 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
FFV2   - FORM FILE FORMAT - VERSION 2
```

ROUTINES CALLED:

FWRITE

CALLED DIRECTLY BY:

WRTFRM - WRITE FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

NAME: WRTFRM/WRTTXT
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:

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
STRLEN

CALLED DIRECTLY BY:

WRTFRM - WRITE FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

SYNOPSIS

INT LEX()

OUTPUTS:

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

DESCRIPTION

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

INCLUDE FILES:

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

ROUTINES CALLED:

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

TOUPPER
ATOI
ISSPACE

CALLED DIRECTLY BY:

YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

APPLICATION GENERATOR Module Documentation

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

DESCRIPTION:

DESCRIPTION
DEFINITION LANGUAGE GRAMMAR.

INCLUDE FILES:

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

ROUTINES CALLED:

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

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

CALLED DIRECTLY BY:

FLANCI - FLAN CALLABLE INTERFACE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT 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.

APPLICATION GENERATOR Include File Description

FILE NAME: CTLCHR
PURPOSE: CONTROL CHARACTERS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
DEFINITIONS OF ALL CONTROL CHARACTERS TO AVOID CHARACTER
SET
DEPENDENCIES.

APPLICATION GENERATOR Include File Description

FILE NAME: ERRPRO
PURPOSE: PROCESS ERROR INCLUDE FILE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

APPLICATION GENERATOR Include File Description

FILE NAME: FFFV2
PURPOSE: FORM FILE FORMAT - VERSION 2
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
RECORD LAYOUTS FOR THE BINARY FORM DEFINITION FILE

APPLICATION GENERATOR Include File Description

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

DESCRIPTION:

DESCRIPTION
AUXILIARY DATA STRUCTURES USED BY FLAN.

APPLICATION GENERATOR Include File Description

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

DESCRIPTION:

APPLICATION GENERATOR Include File Description

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

DESCRIPTION:

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

APPLICATION GENERATOR Include File Description

FILE NAME: FPDINI
PURPOSE: FPD INITIALIZATION
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INITIALIZED VERSION OF UID FOR INCLUSION IN MAIN PROGRAM.

APPLICATION GENERATOR Include File Description

FILE NAME: FPPARM
PURPOSE: FORM PROCESSOR PARAMETERS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION: THESE DATA DEFINITIONS ARE USED
IN THE FORM PROCESSOR ROUTINES.

APPLICATION GENERATOR Include File Description

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

DESCRIPTION:

DESCRIPTION
INCLUDE FILE FOR NTM INTERFACE

APPLICATION GENERATOR Include File Description

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

DESCRIPTION:

DESCRIPTION

APPLICATION GENERATOR Include File Description

FILE NAME: SRVRET
PURPOSE: AS THE RETURN GIVEN A TABLE-FULL ERROR
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

MODIFIED 11/2/83 TO INCLUDE RET-CODE-5 *
MODIFIED 1/9/84 TO INCREASE ALL ERROR CODES TO PIC X(5) *
AND TO ELIMINATE ALPHA'S *
MODIFIED 1/26/84 TO ADD RET-CODE FOR GETUSR-NOT-SUCC *
SRV-SUCCESSFUL ADDED FOR GENERIC RETURN *
MODIFIED 2/7/84 TO ADD ERROR CODES FOR ENTRY-NOT-FOUND *
MODIFIED 2/8/84 TO ADD WTHST-NOT-SUCCESSFUL *
MODIFIED 2/20/84 TO ADD TSTMOD NEW CODES. *
MODIFIED 20 AUG 84 INITIALIZE ALL LOCAL VARIABLES TO
SPACES OR 0.
MODIFIED 5/21/85 TO ADD RCL AND FILGEN RETURN CODES

APPLICATION GENERATOR Include File Description

FILE NAME: STDTP
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
TBCOL	- 8 BIT (OR LARGER) LOGICAL (ZERO / NON-ZERO ONLY)
METACHAR	- 16 BIT (OR LARGER) AUGMENTED CHARACTER (SIGNED)
VOID	- FUNCTION THAT RETURNS NO VALUE
FORTTRAN	- STORAGE CLASS FOR FOREIGN (NON-C) ROUTINES OR C ROUTINES WHICH ARE CALLABLE FROM FOREIGN ROUTINES

SINCE NOT ALL COMPILERS SUPPORT USHORT, TINY, AND UTINY,
THE FUNCTIONS
USHORT(), TINY(), AND UTINY() SHOULD BE USED WHENEVER
REFERENCING THEM.

IN ADDITION, THE FOLLOWING UTILITY MACROS ARE DEFINED:

LURSHIFT(N, B)	- UNSIGNED LONG RIGHT SHIFT
MAX(A, B)	- MAXIMUM OF A AND B
MIN(A, B)	- MINIMUM OF A AND B

APPLICATION GENERATOR Include File Description

ABS(A)	- ABSOLUTE VALUE OF A
STRASN(A, B)	- TRANSPORTABLE A = B FOR STRUCTURES
NULL	- NULL POINTER VALUE (0)
TRUE	- 1
FALSE	- 0
SUCCESS	- EXIT(SUCCESS) INDICATES SUCCESSFUL COMPLETION
FAILURE	- EXIT(FAILURE) INDICATES ERRORS

THE FOLLOWING SYMBOLS SHOULD BE DEFINED BASED ON THE
COMPILER BEING USED:

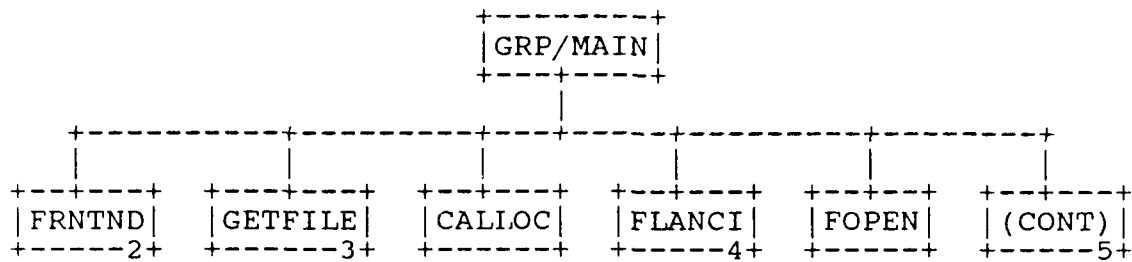
USHORT	- COMPILER SUPPORTS UNSIGNED SHORT
TINY	- COMPILER TREATS CHAR AS SIGNED
UTINY	- CHAR IS SIGNED AND COMPILER SUPPORTS UNSIGNED CHAR
VOID	- COMPILER SUPPORTS VOID
FORTRAN	- COMPILER SUPPORTS FORTRAN
STRASN	- DEFINE APPROPRIATE MACRO
SUCCESS	- DEFINE APPROPRIATE VALUE IF NOT 0
FAILURE	- DEFINE APPROPRIATE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

3.10.10 Hierarchy Chart

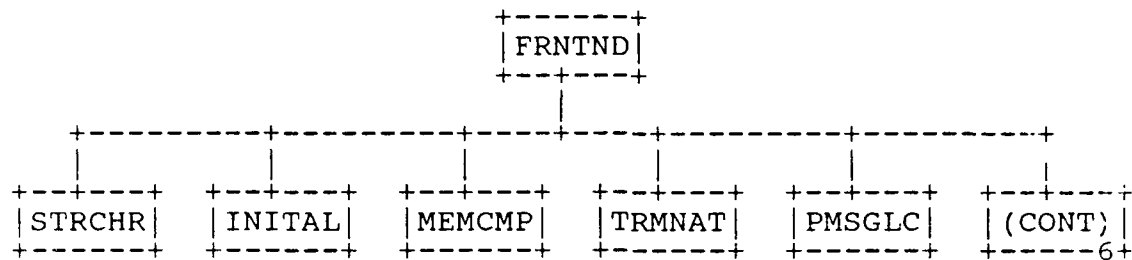
The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the first page are considered Main Programs as described in section 3.10.1 above.

There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.

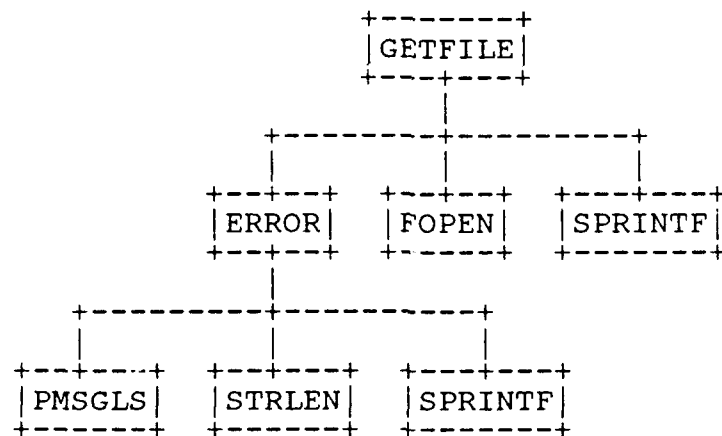
1



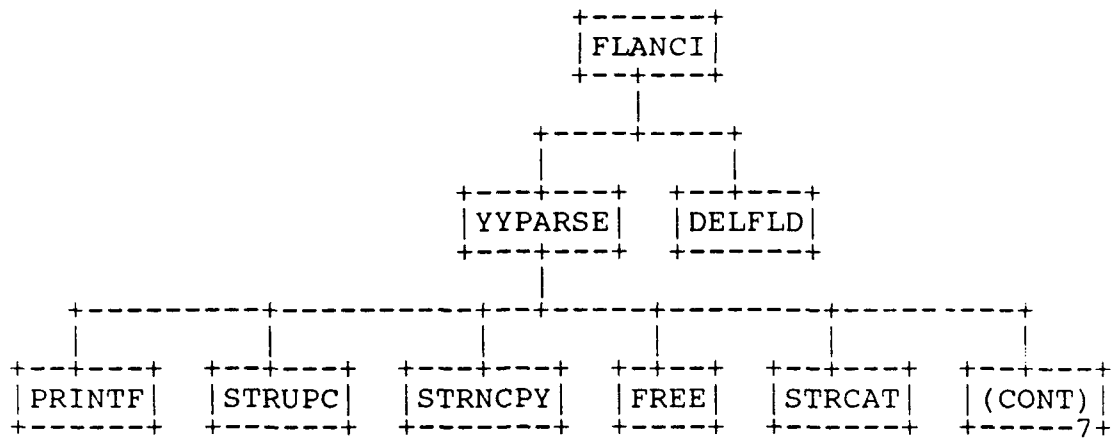
2



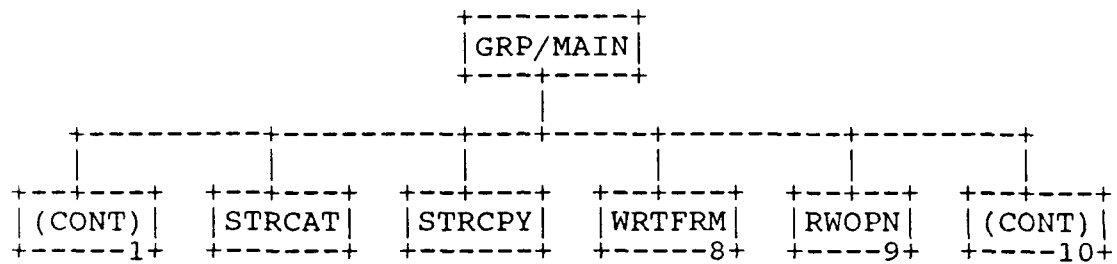
3



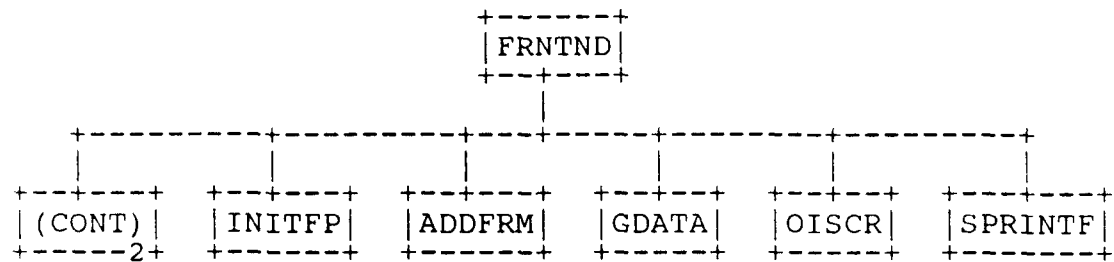
4



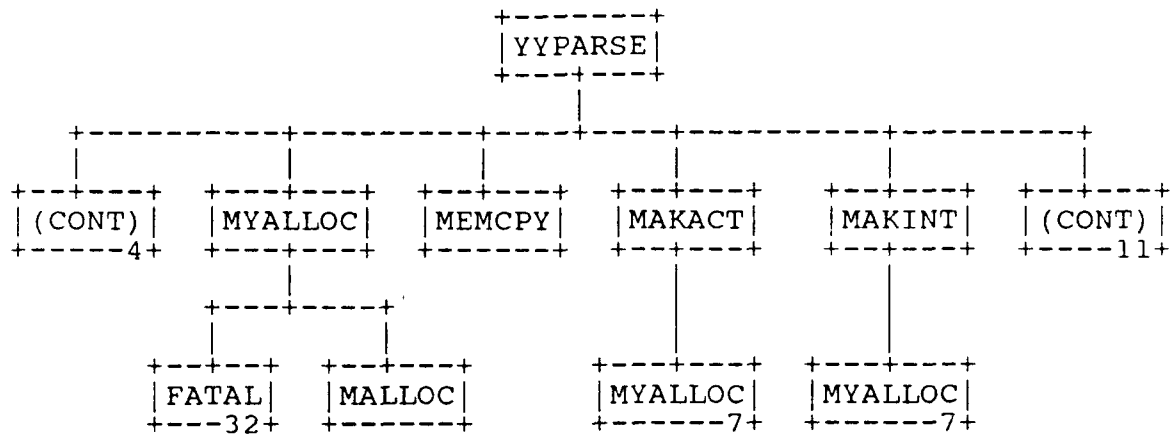
5



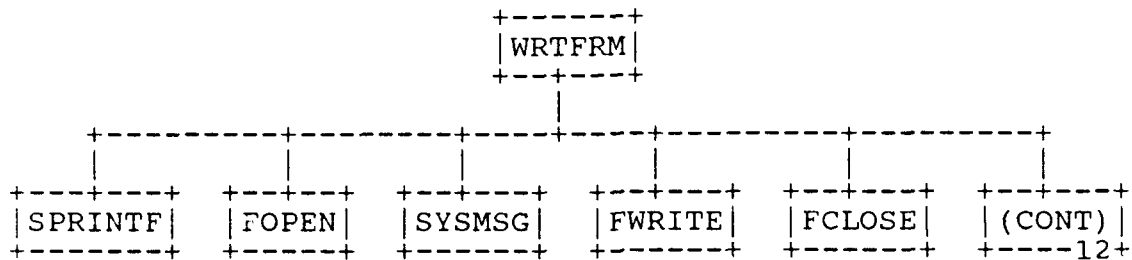
6



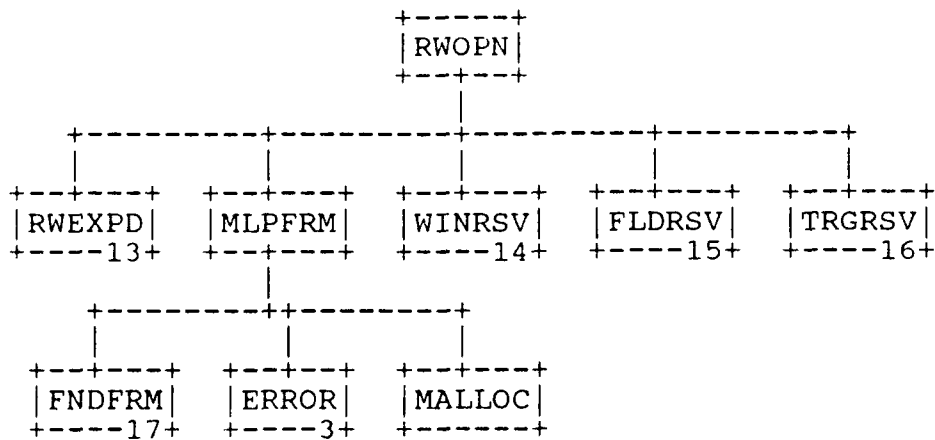
7



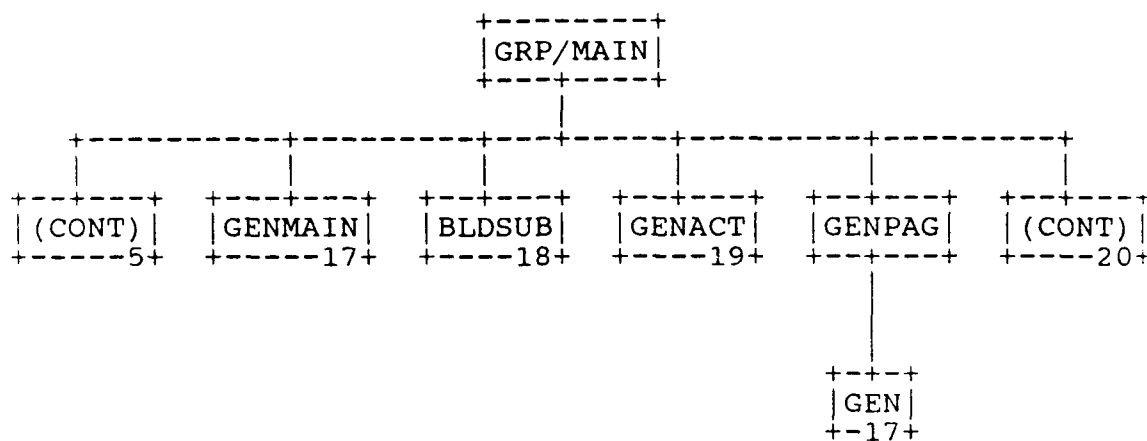
8



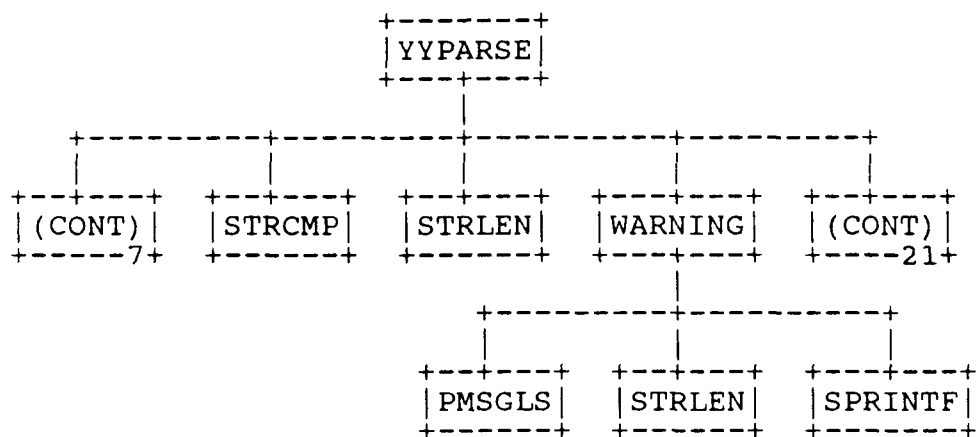
9



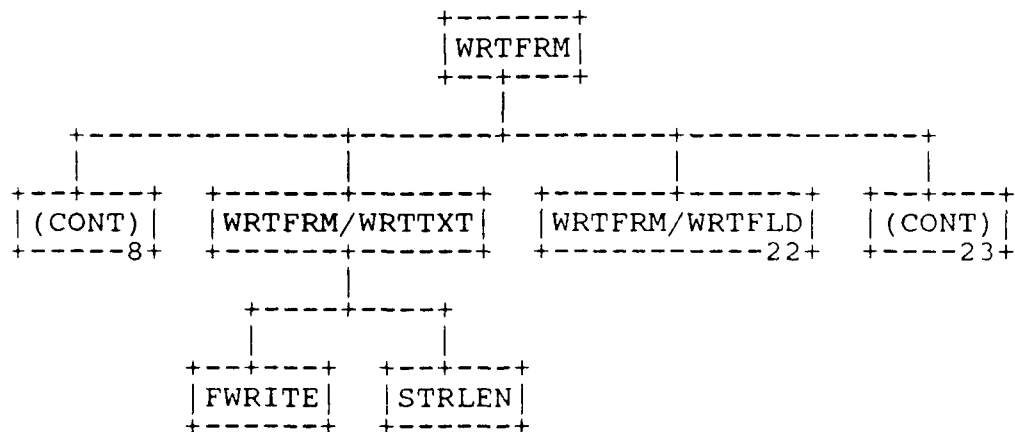
10



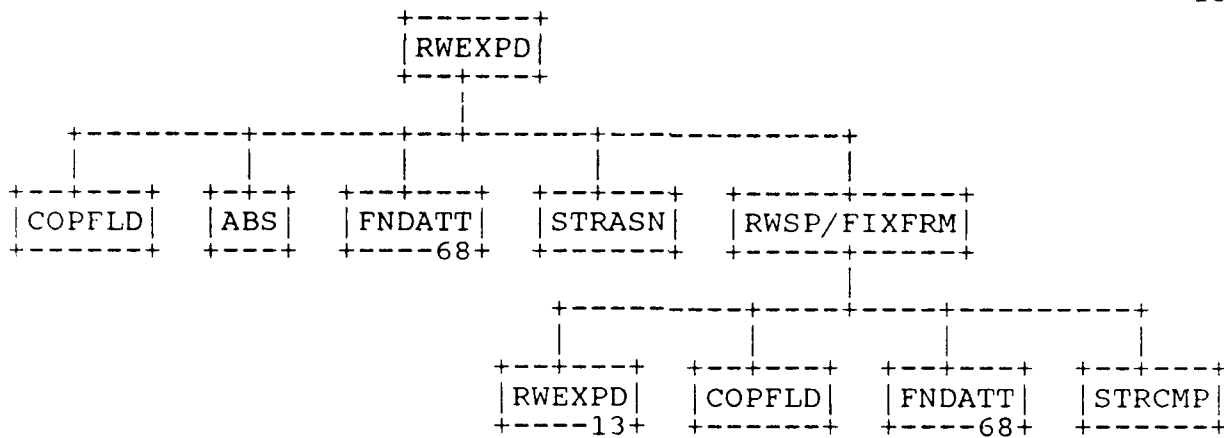
11



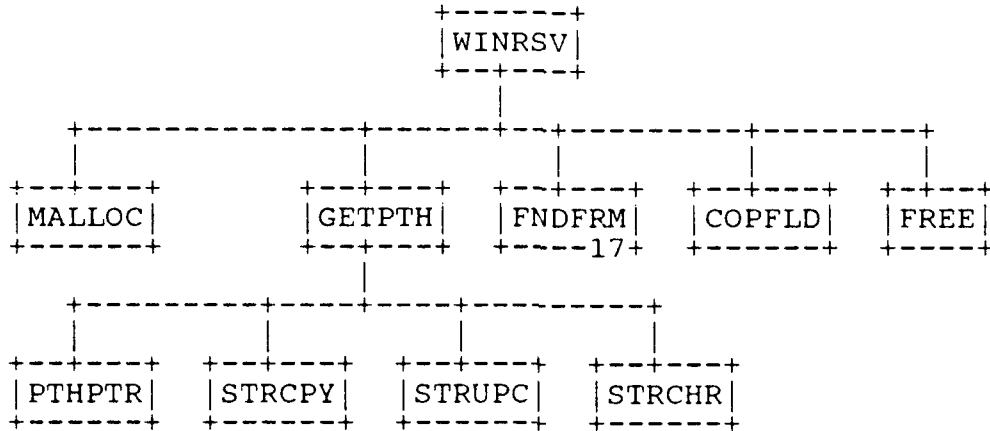
12



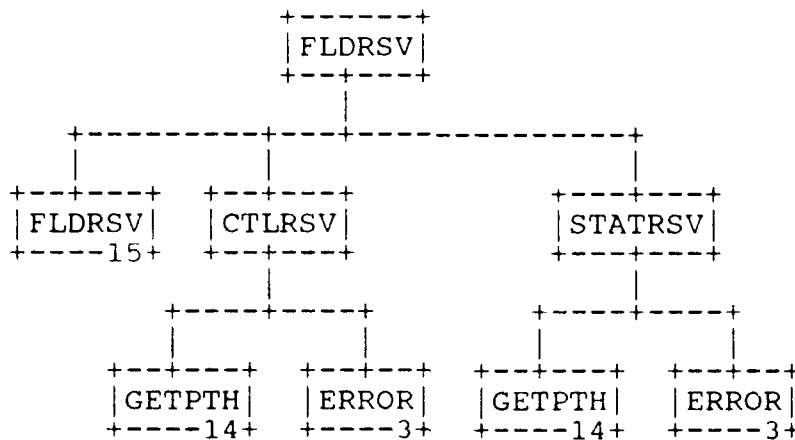
13



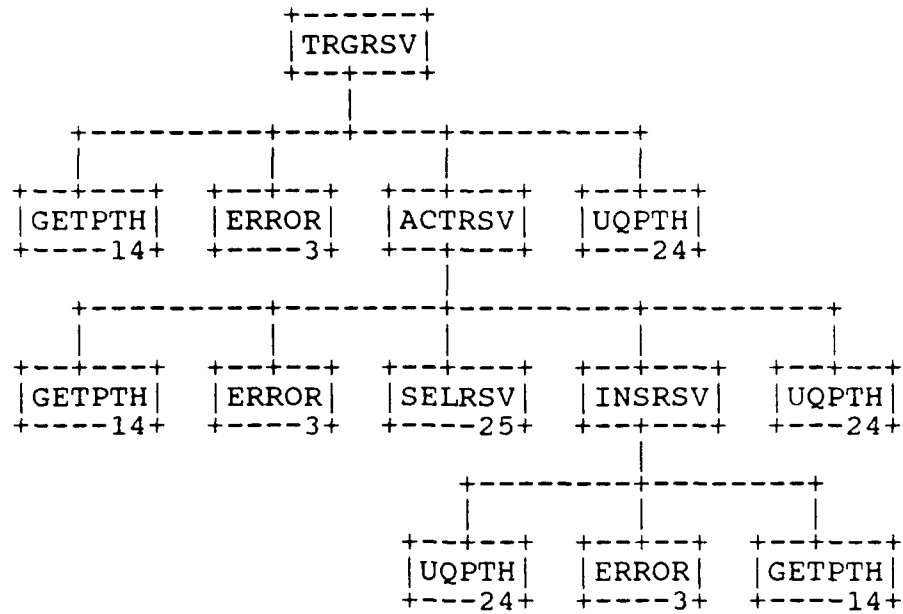
14



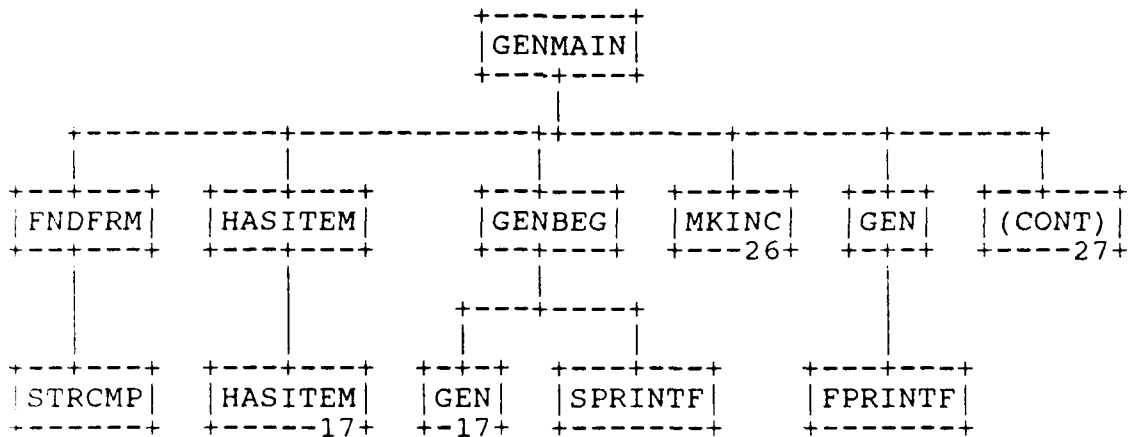
15



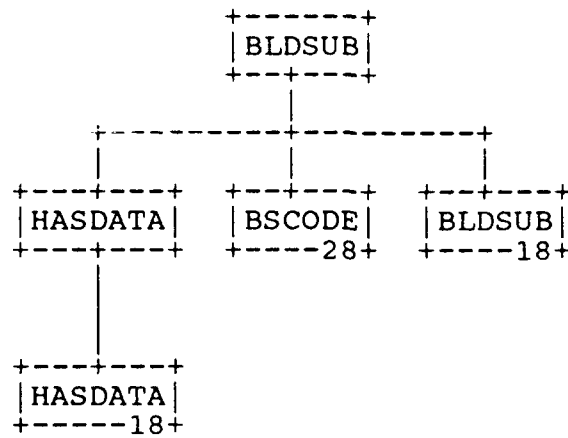
16



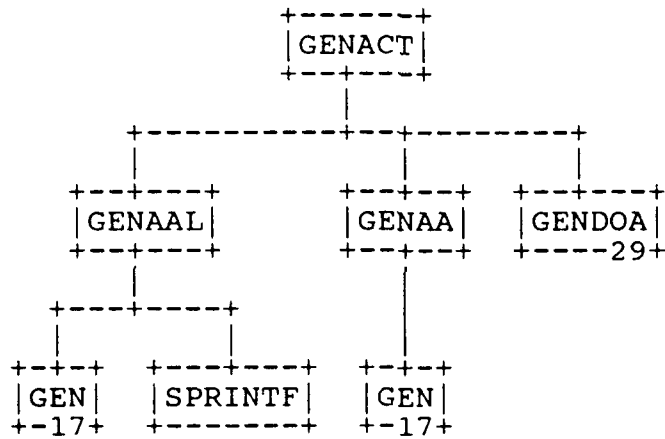
17



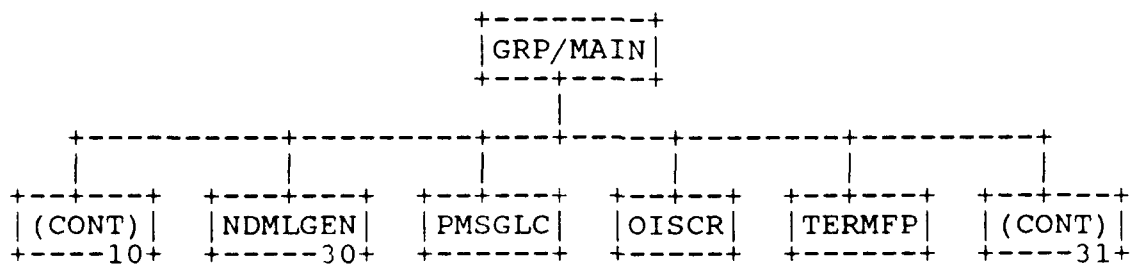
18



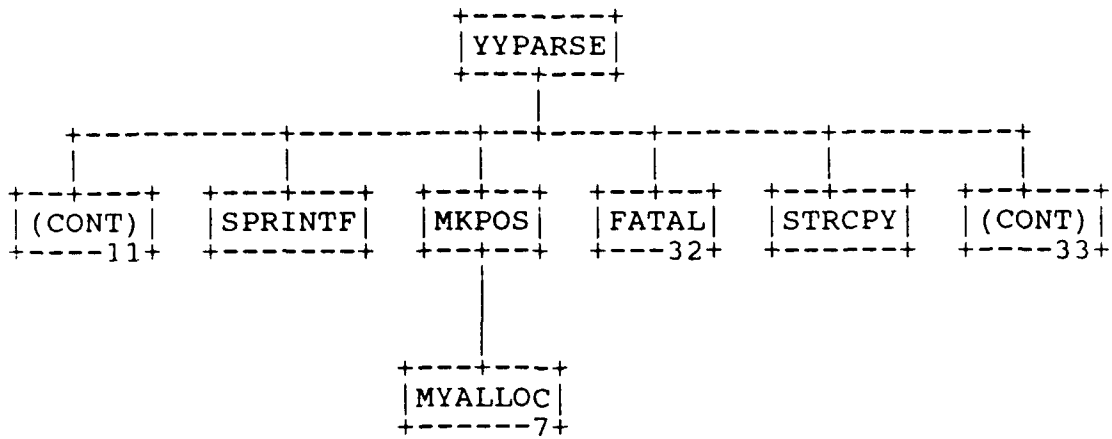
19



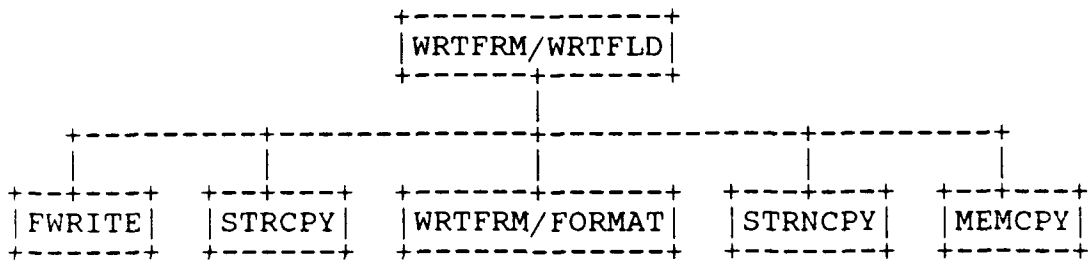
20



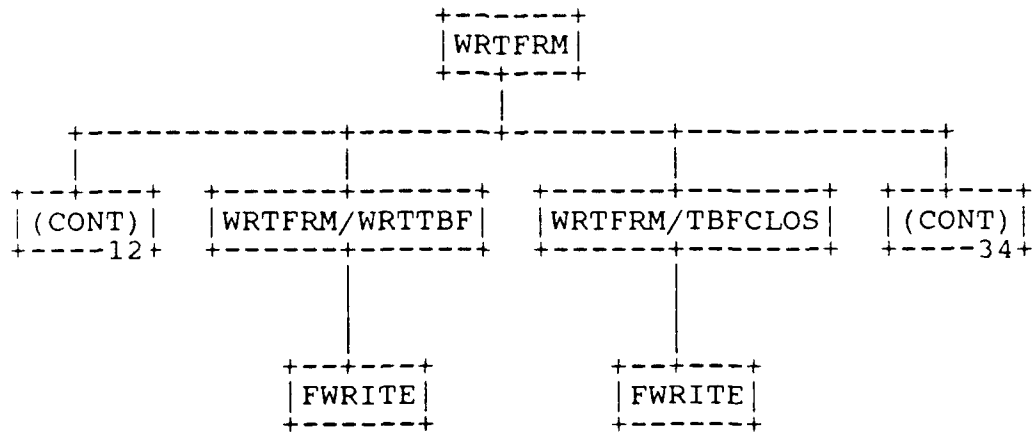
21



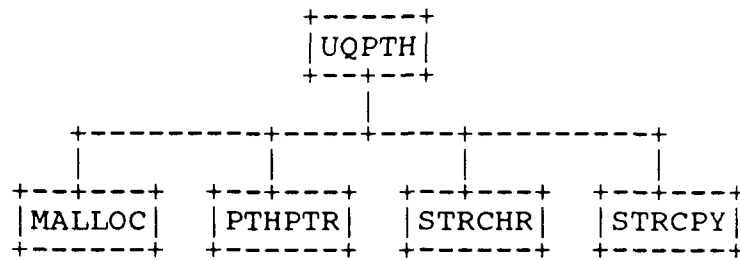
22



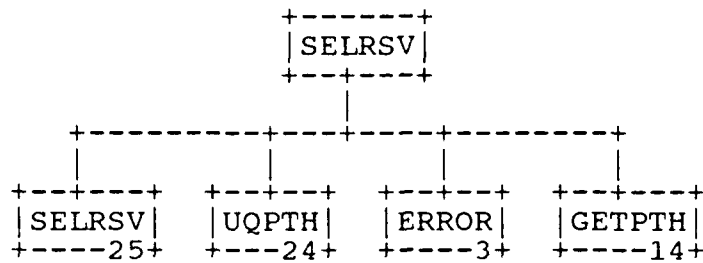
23



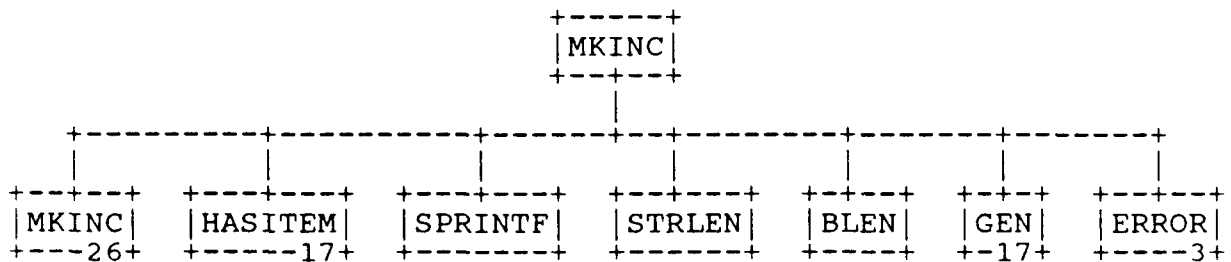
24



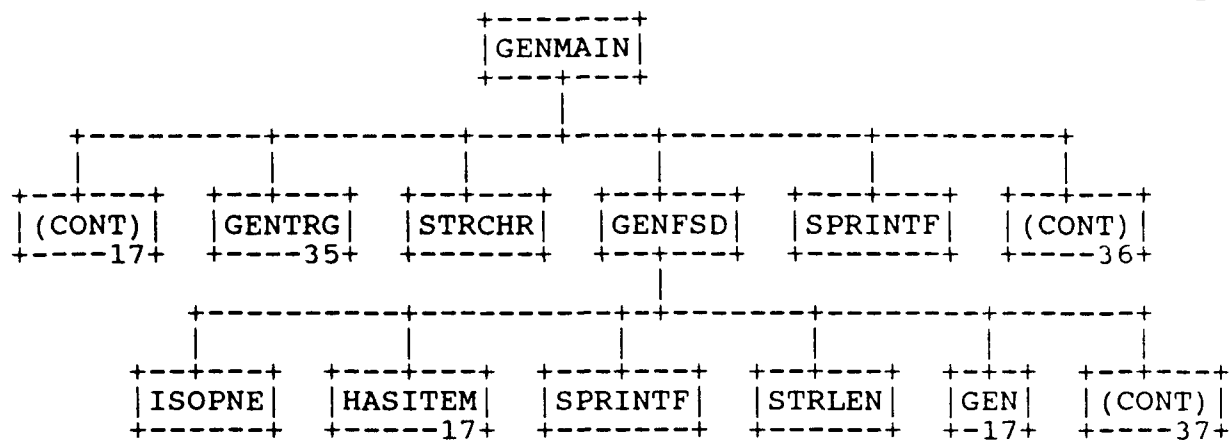
25



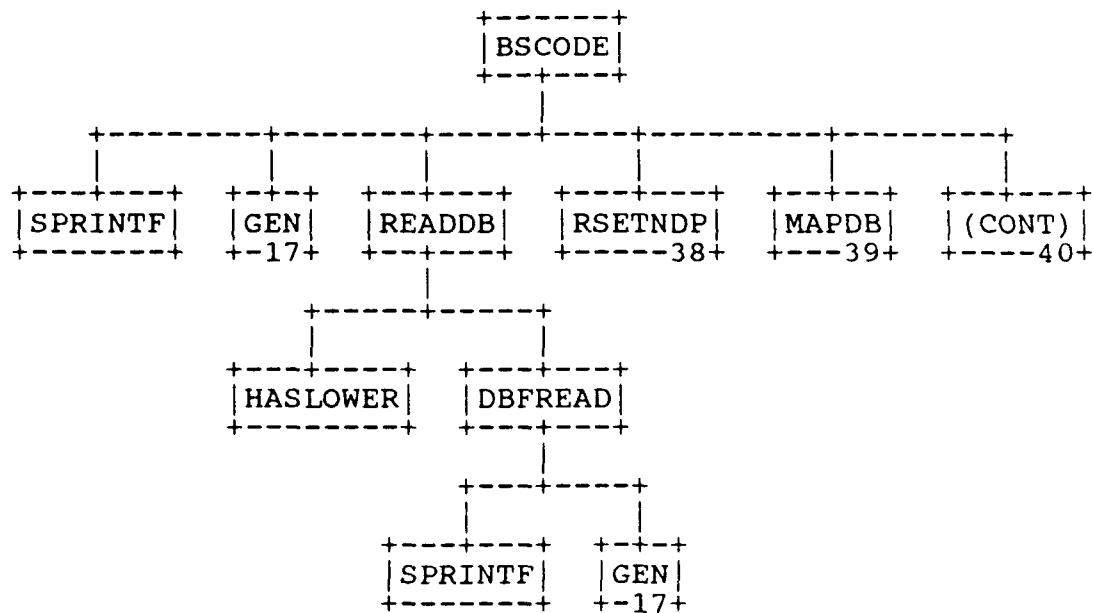
26

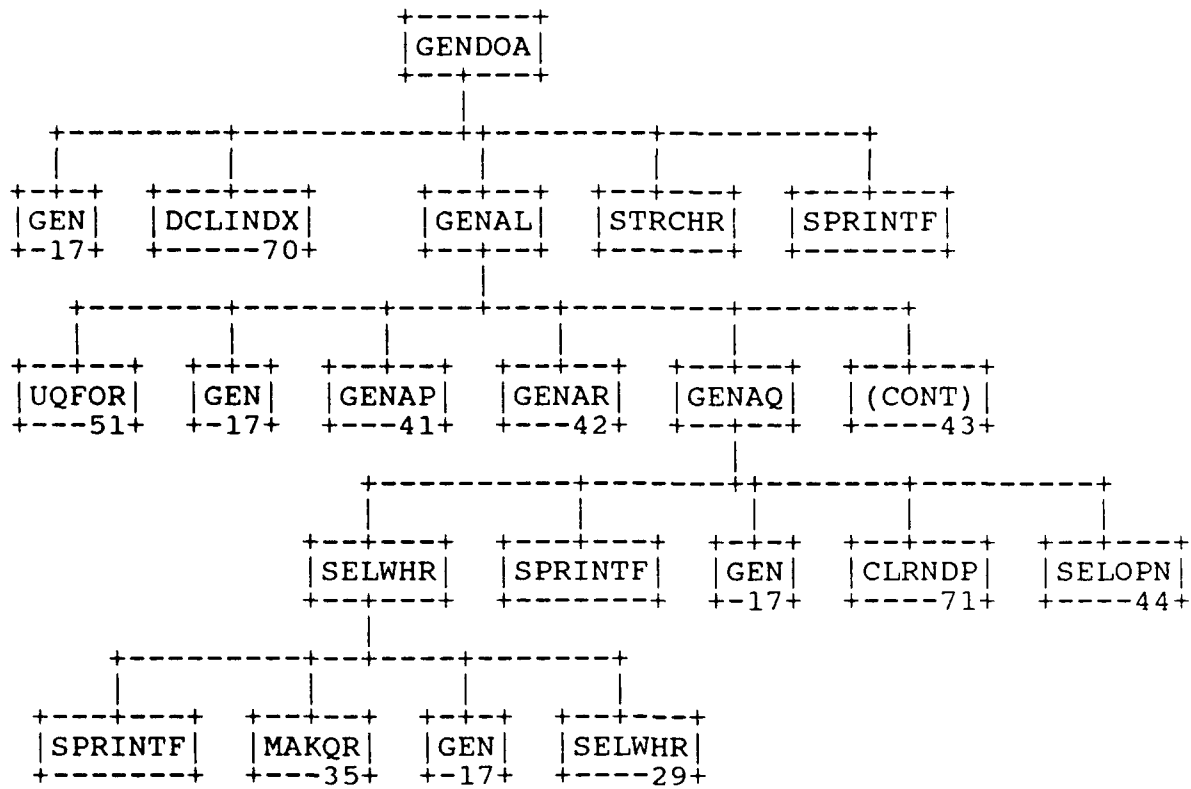


27

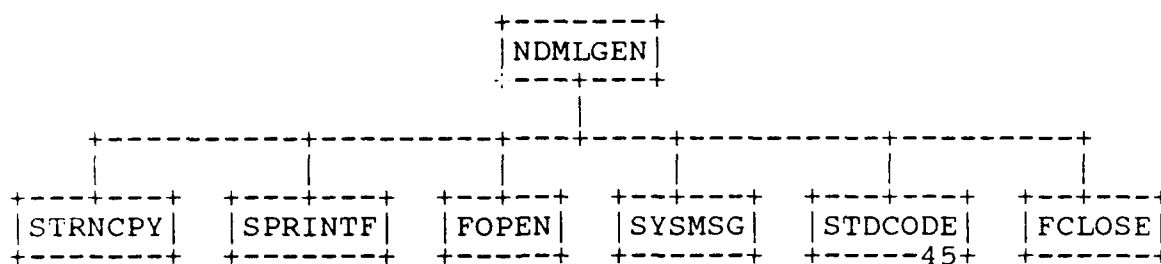


28

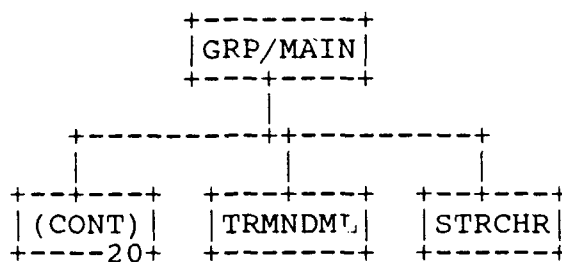




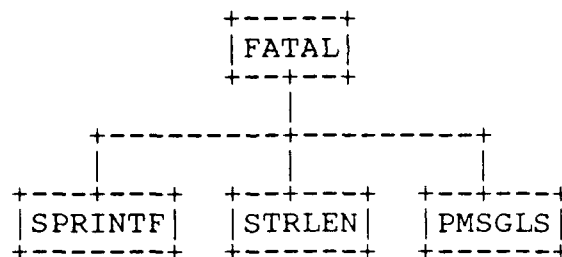
30



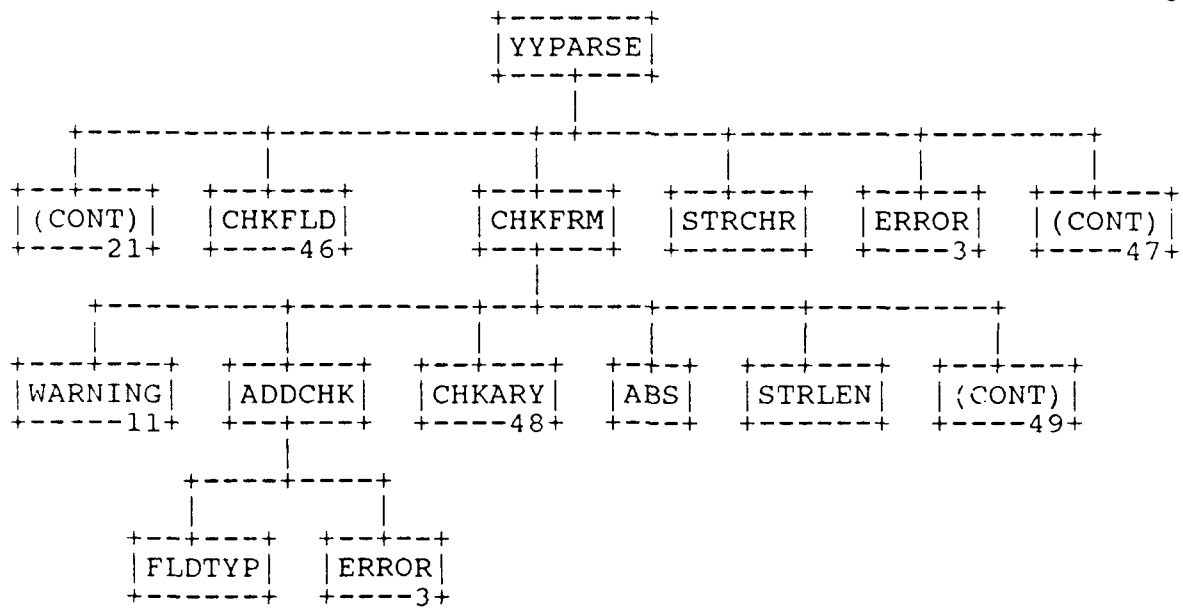
31



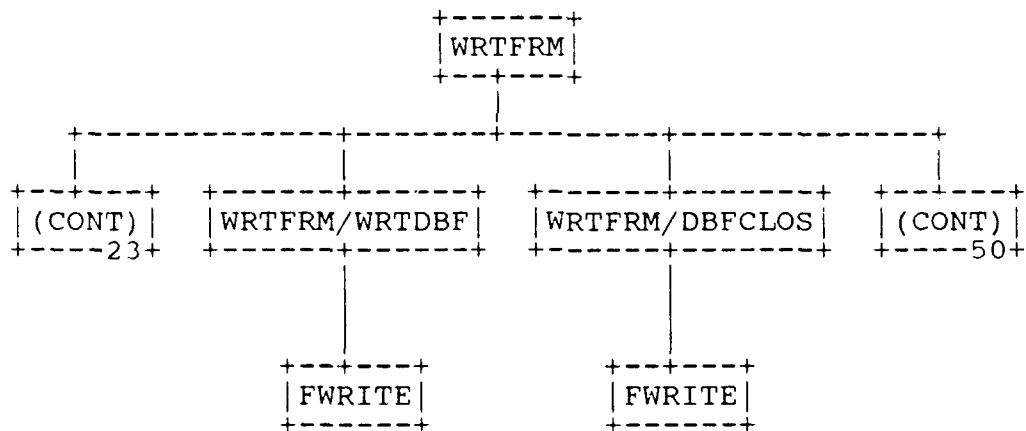
32



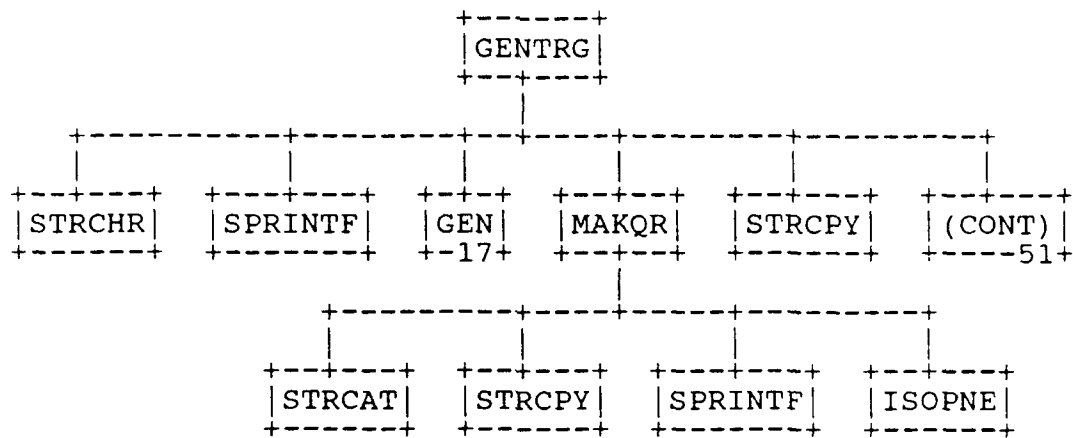
33



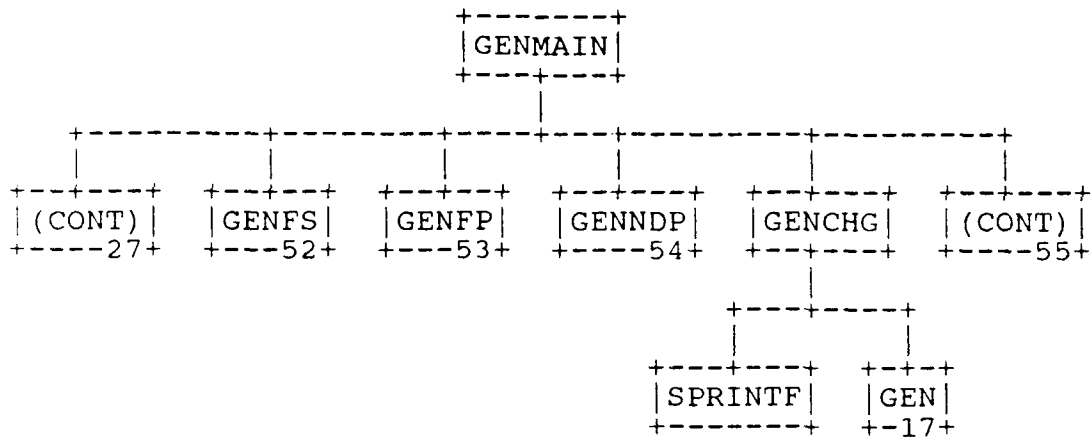
34



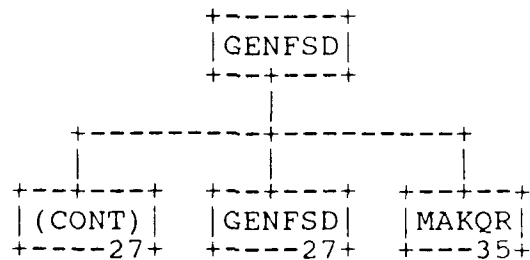
35



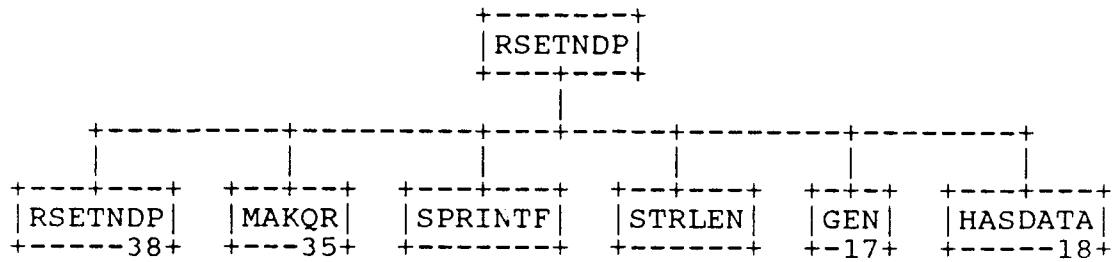
36



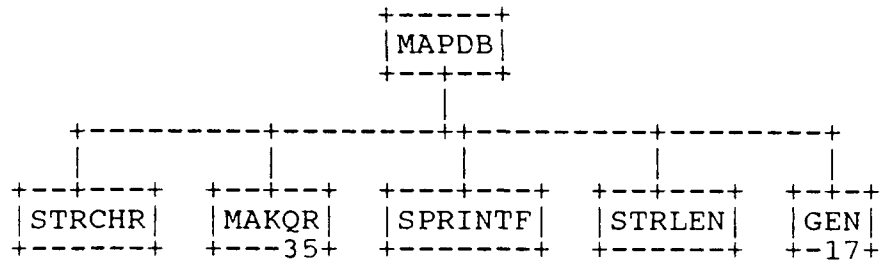
37



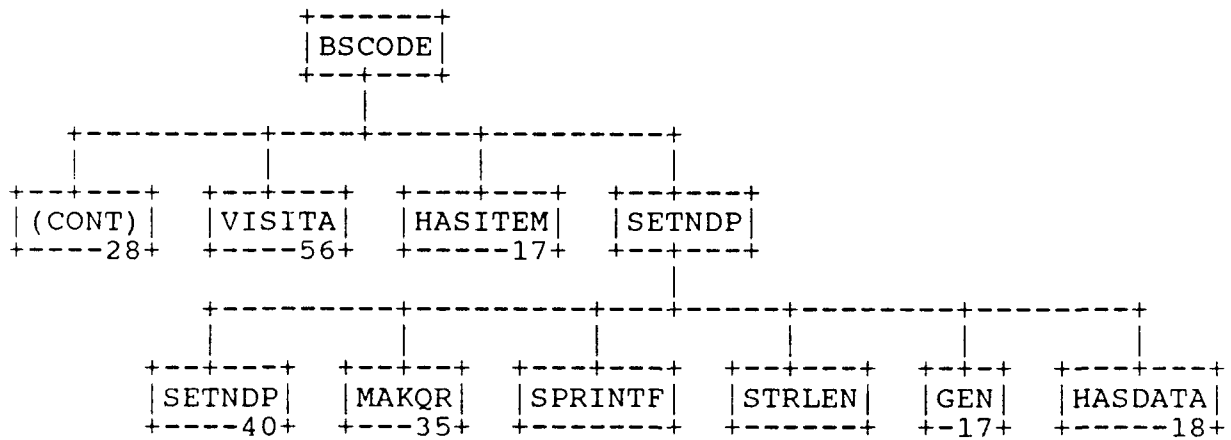
38



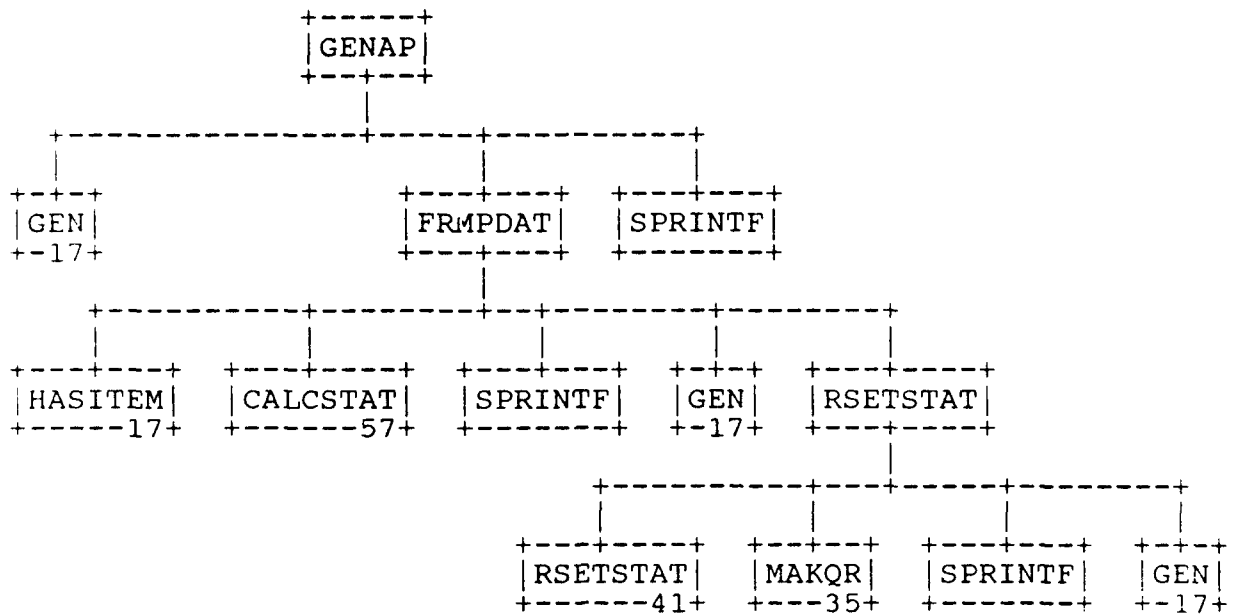
39



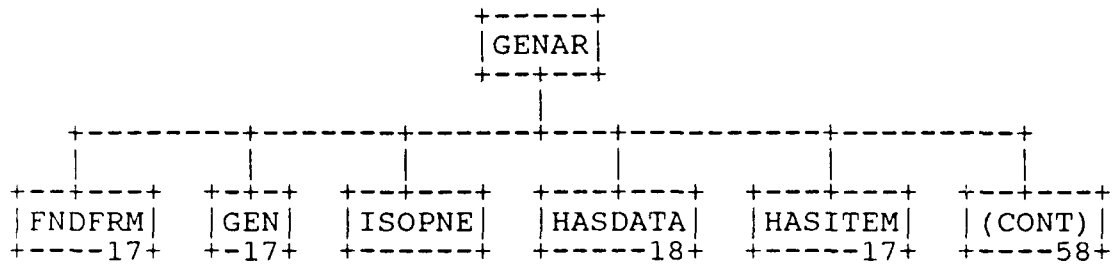
40



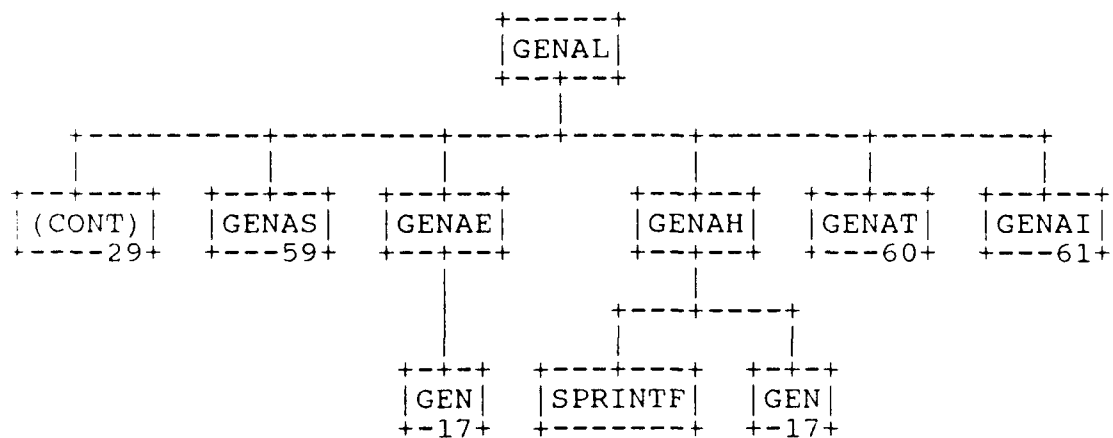
41



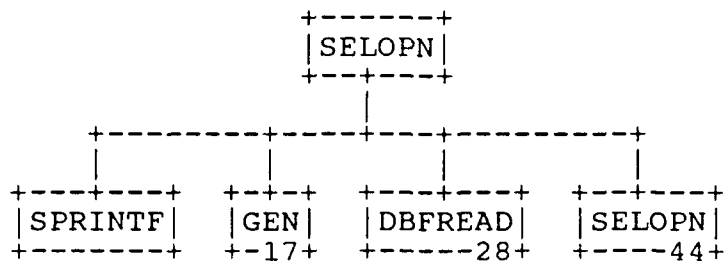
42



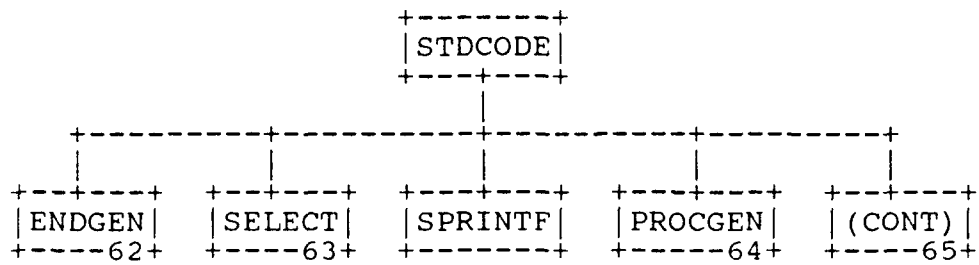
43



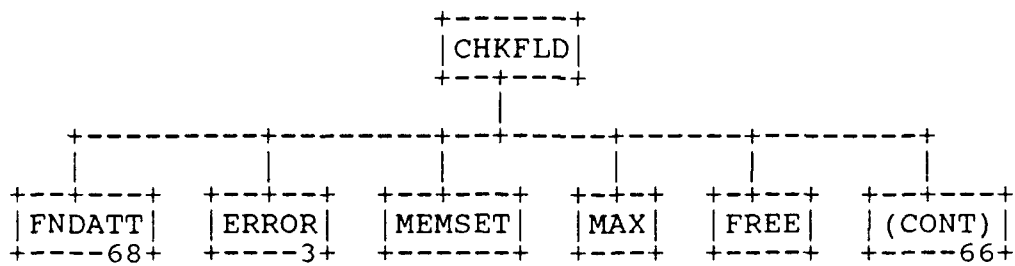
44



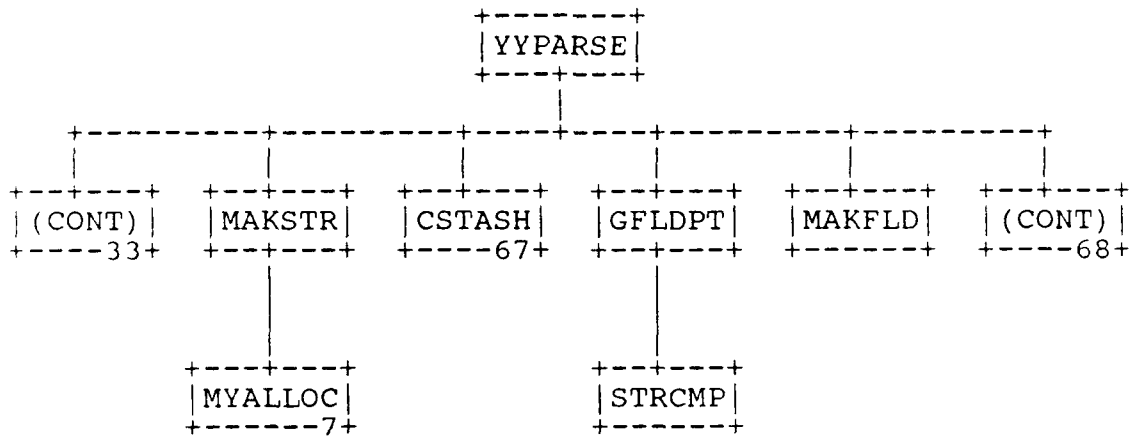
45



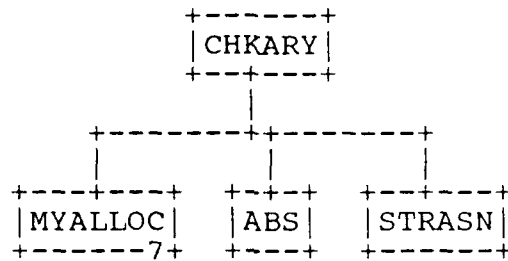
46



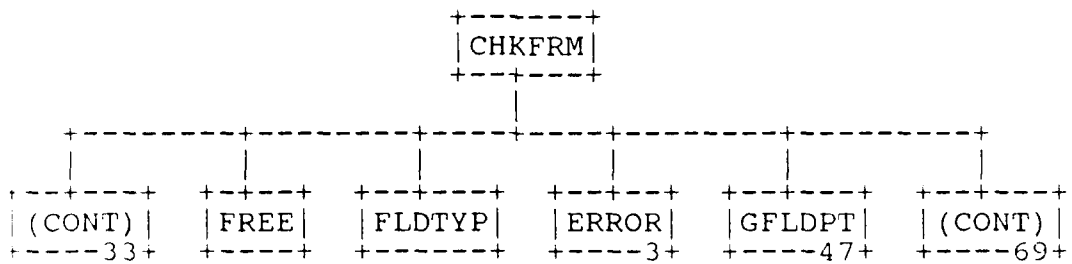
47



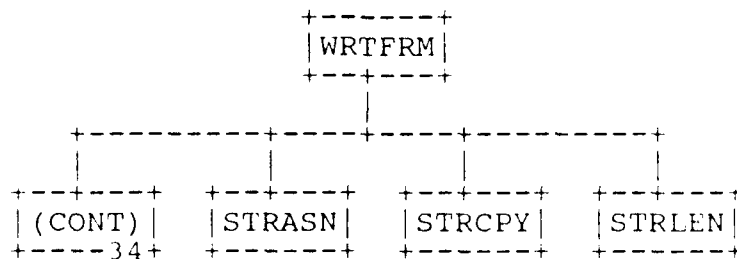
48



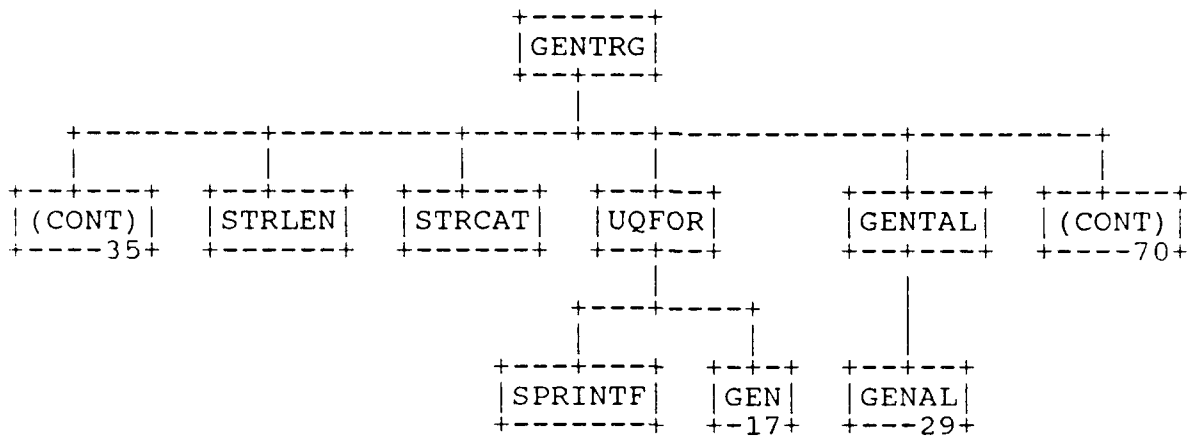
49



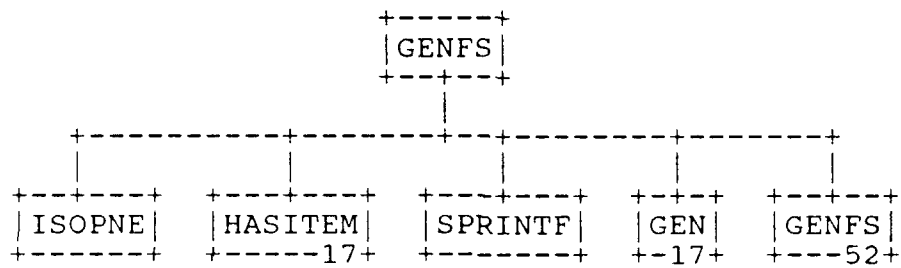
50



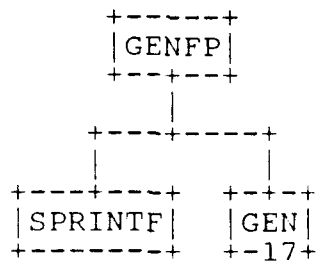
51



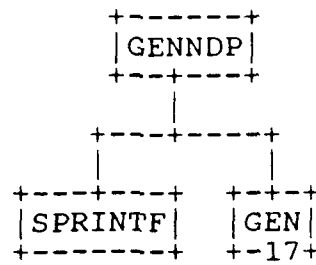
52



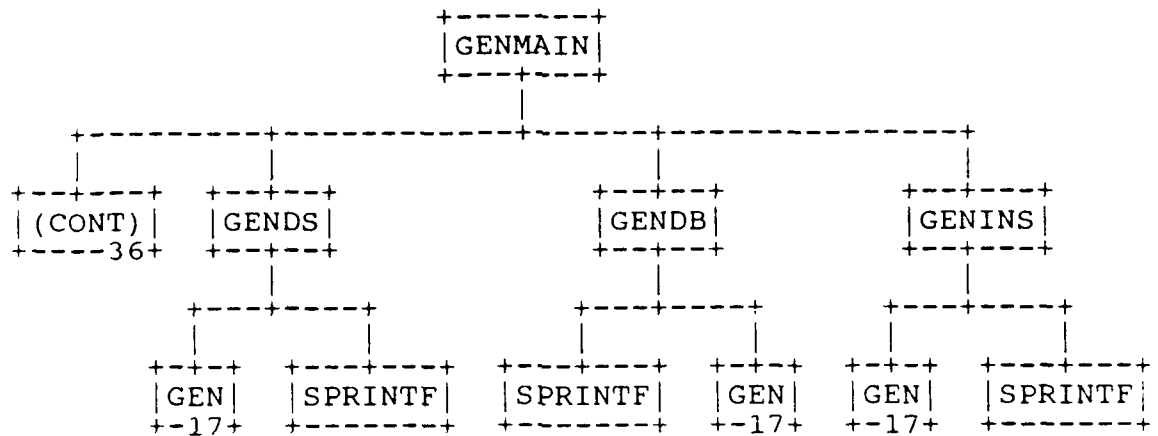
53



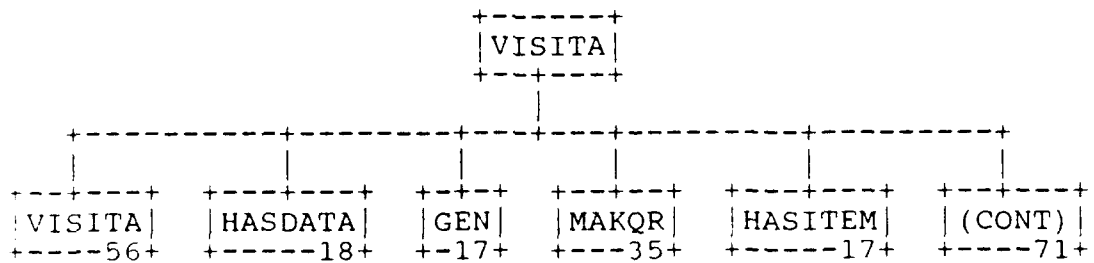
54



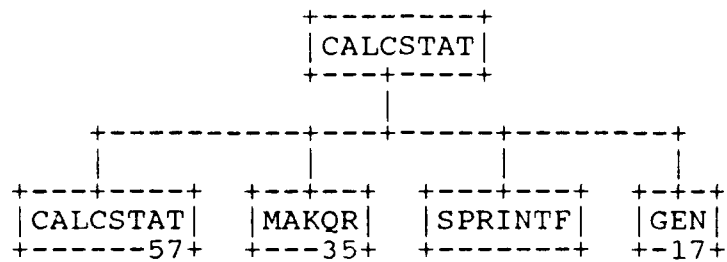
55



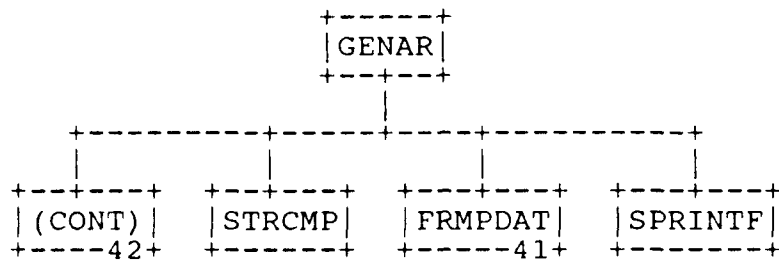
56



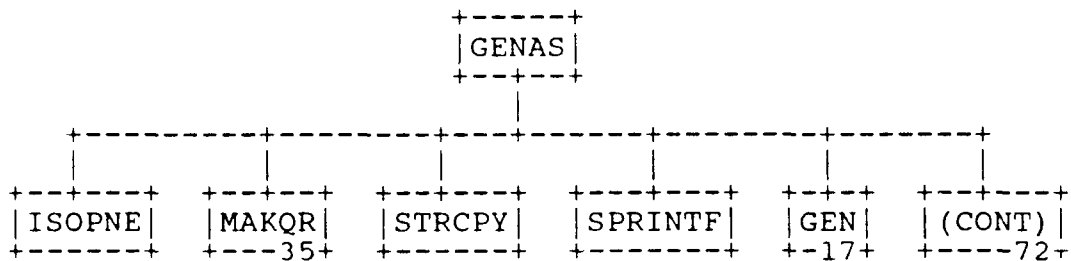
57



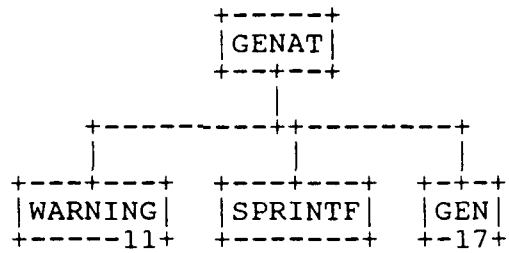
58



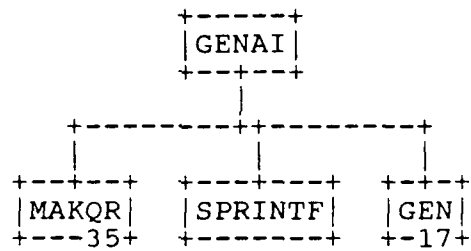
59



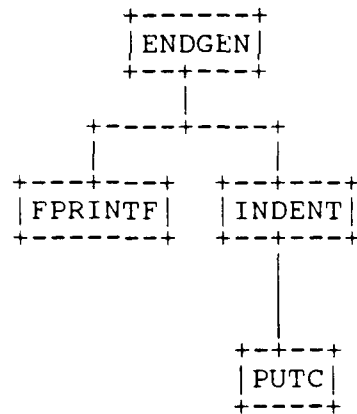
60



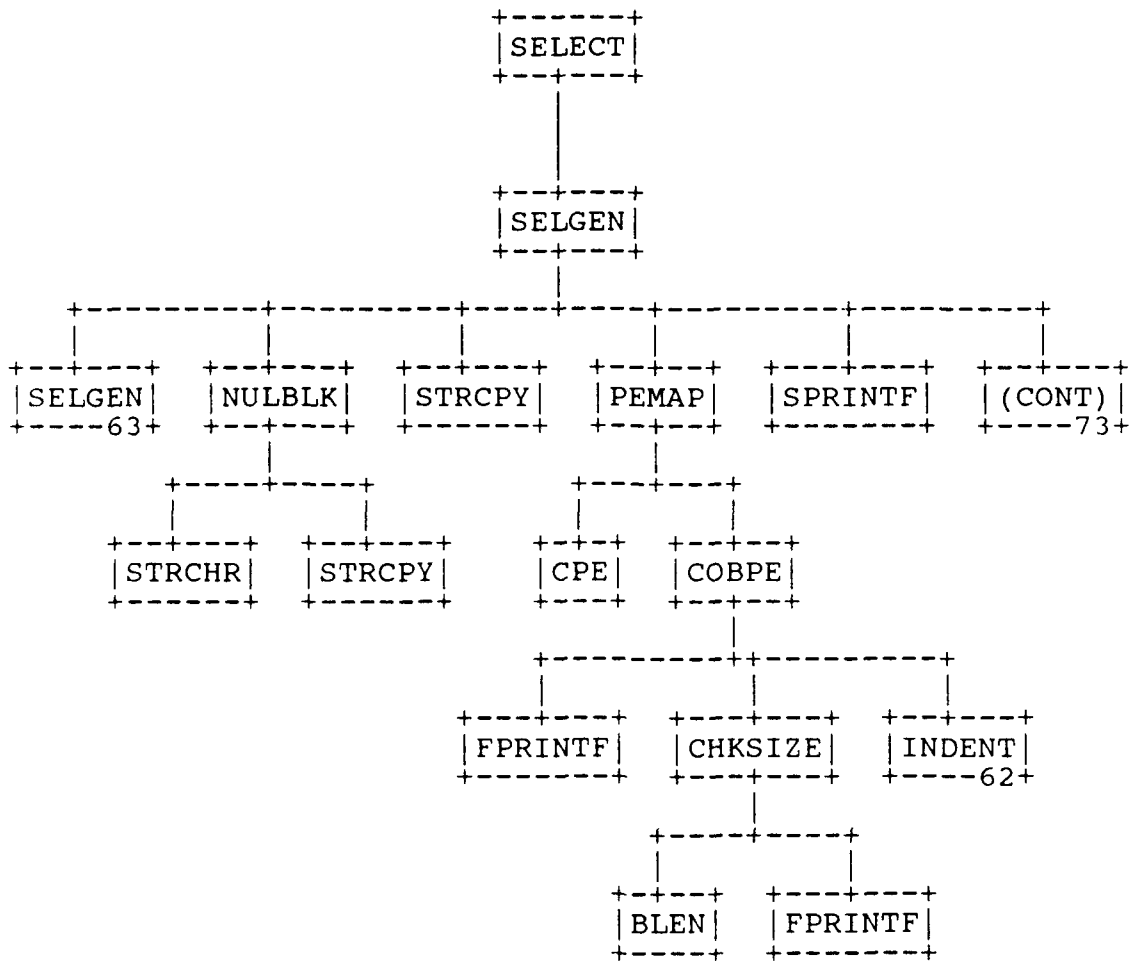
61



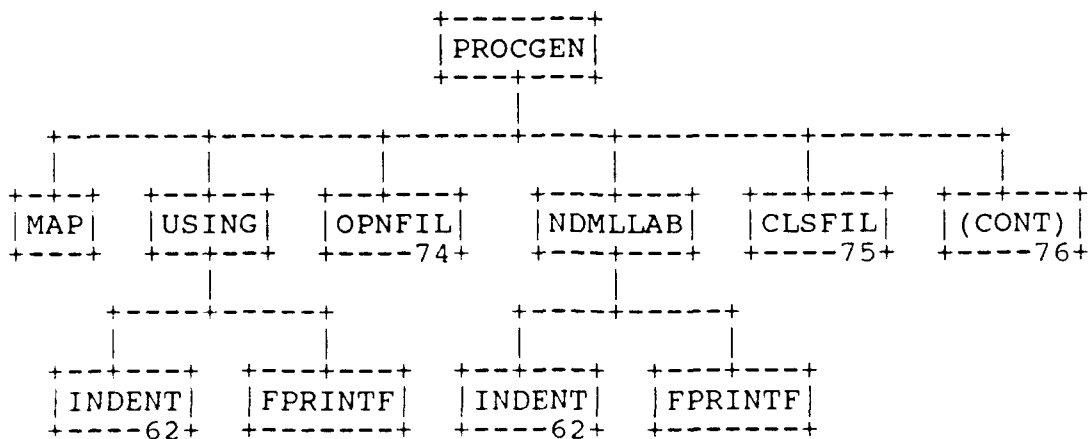
62



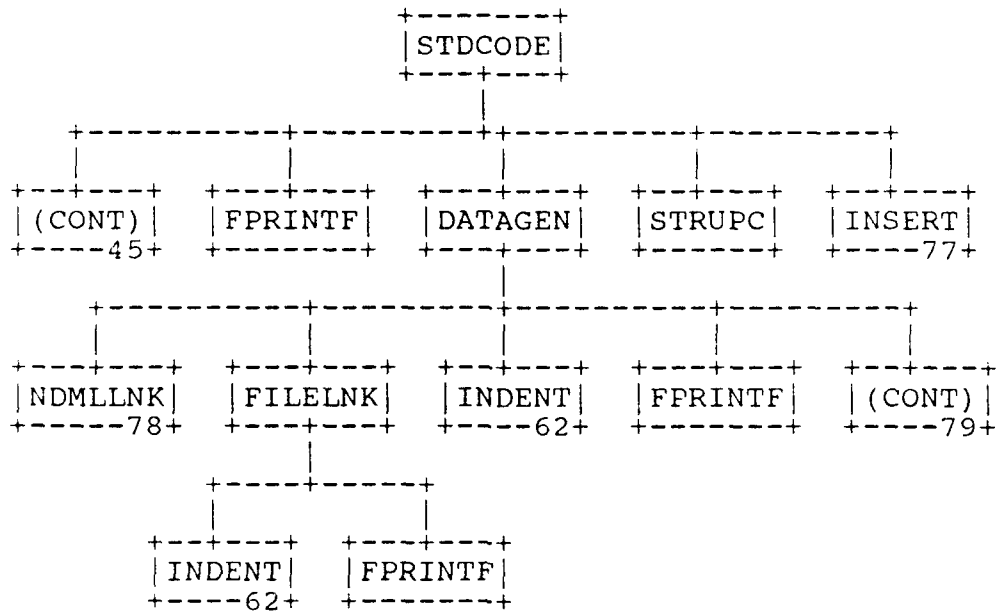
63



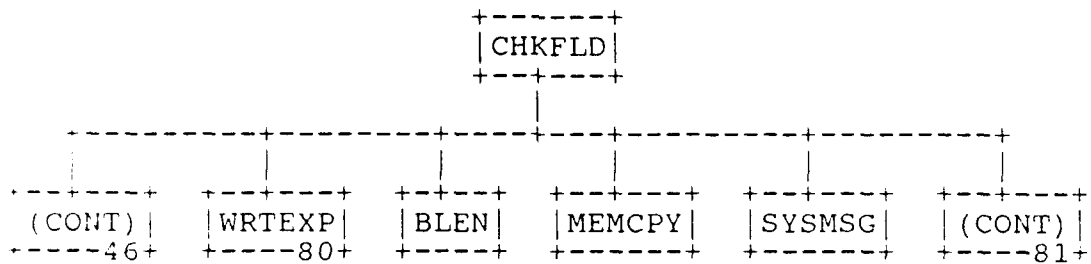
64

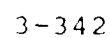


65

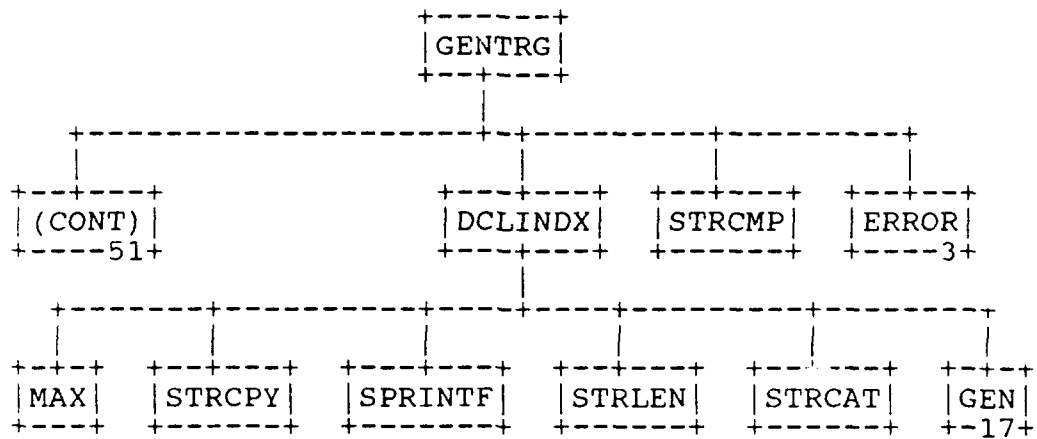


66

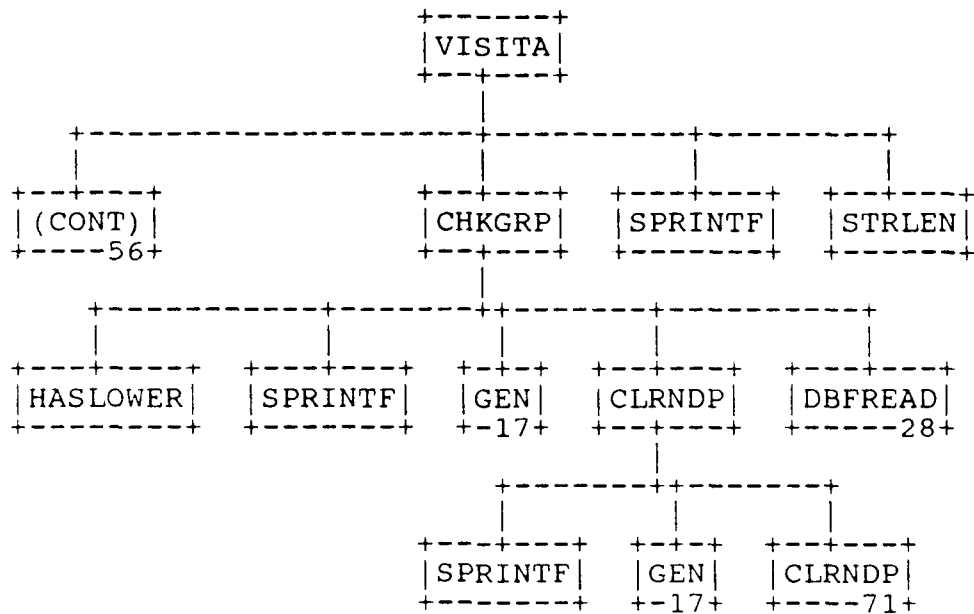




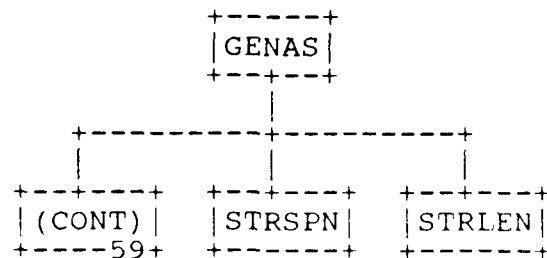
70



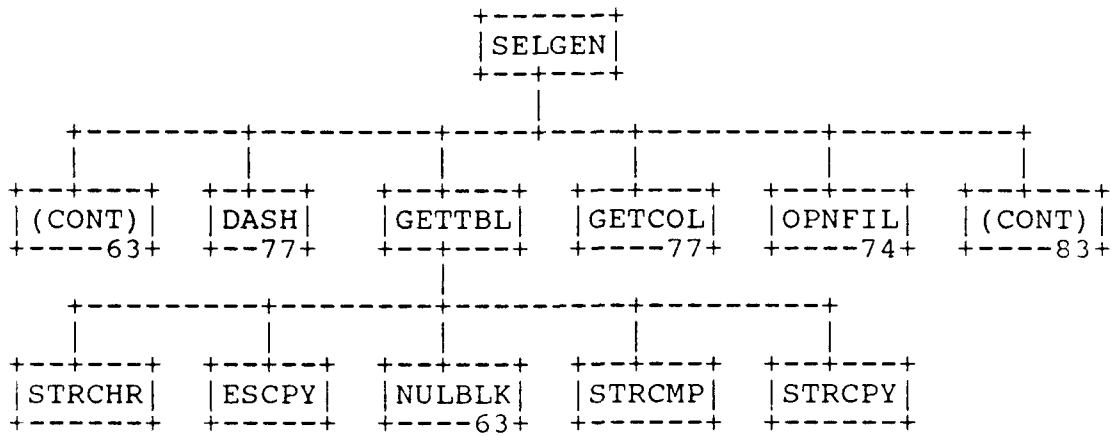
71



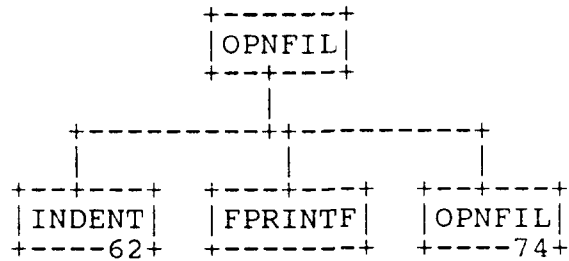
72



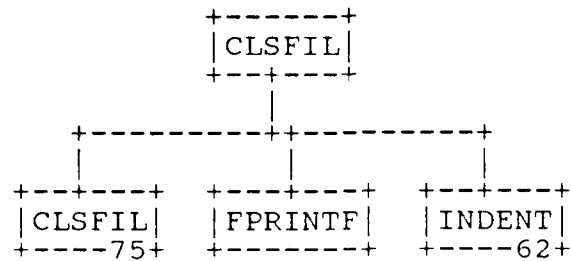
73



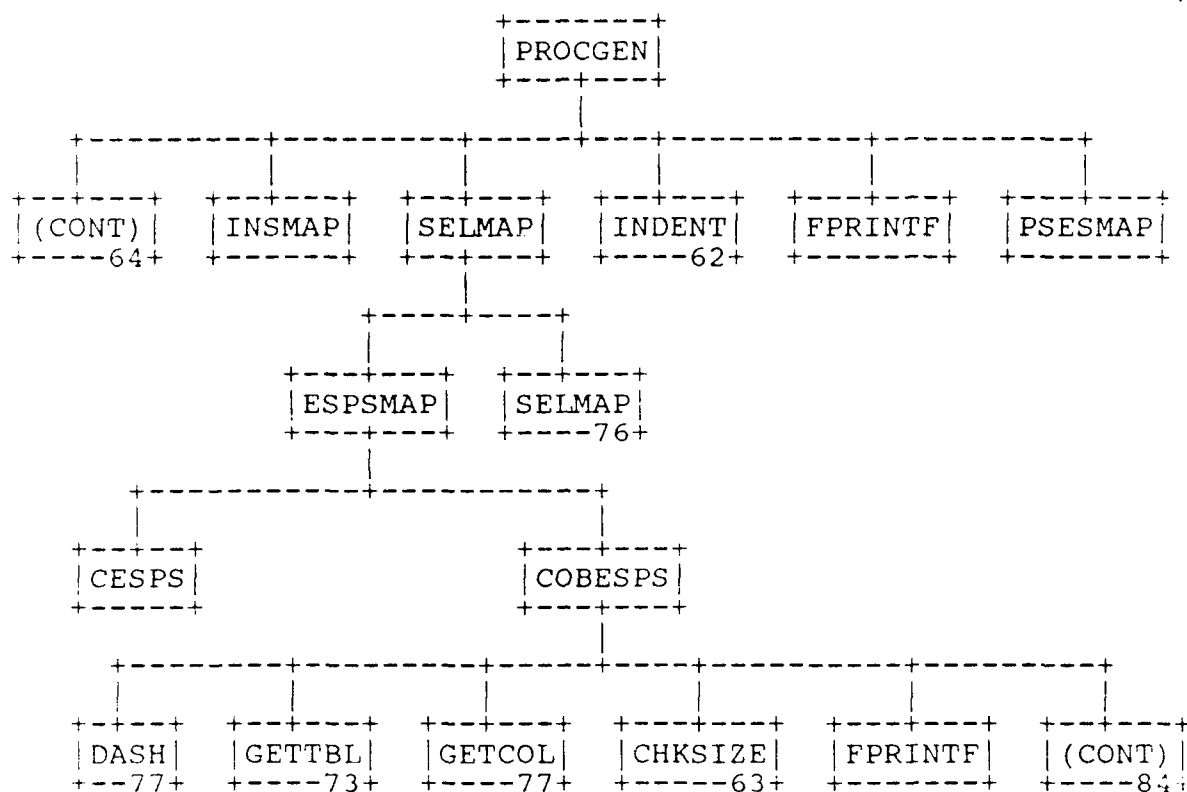
74



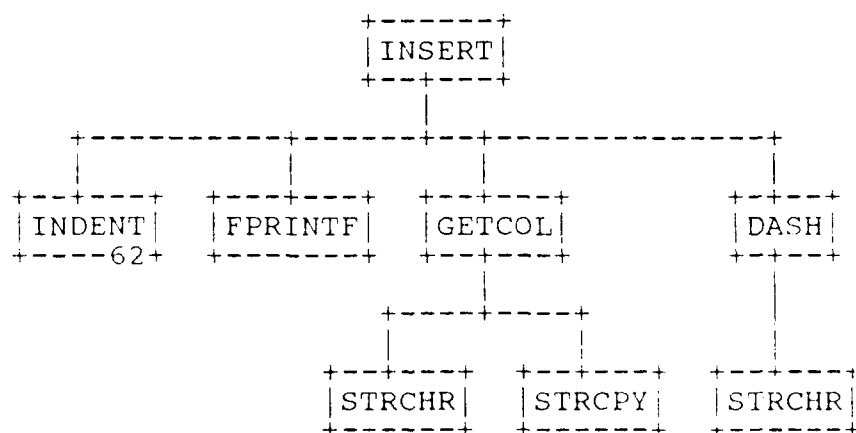
75



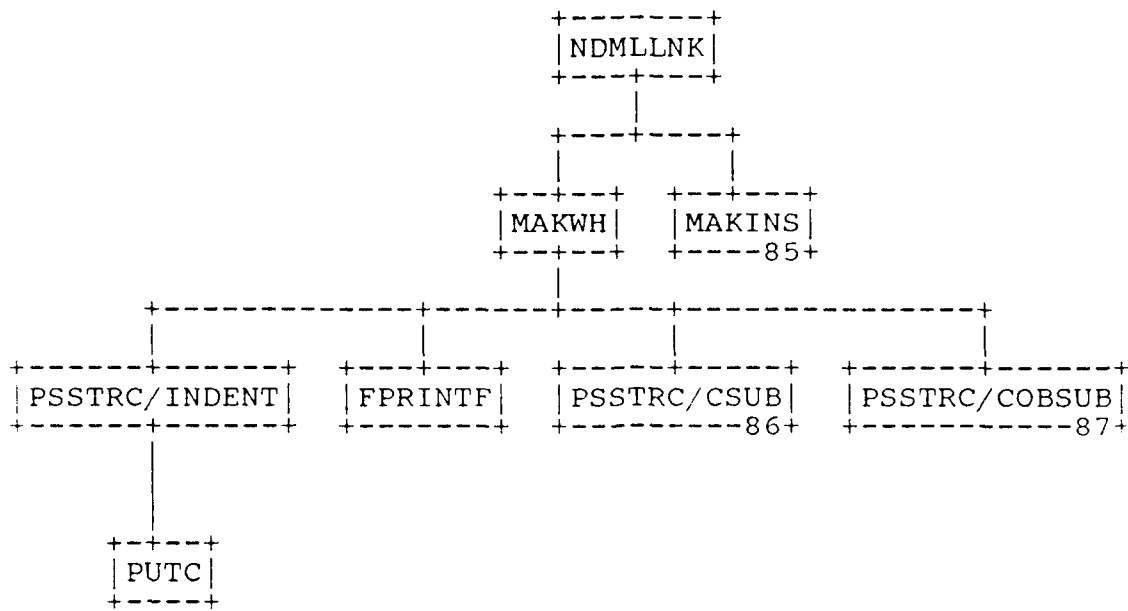
76



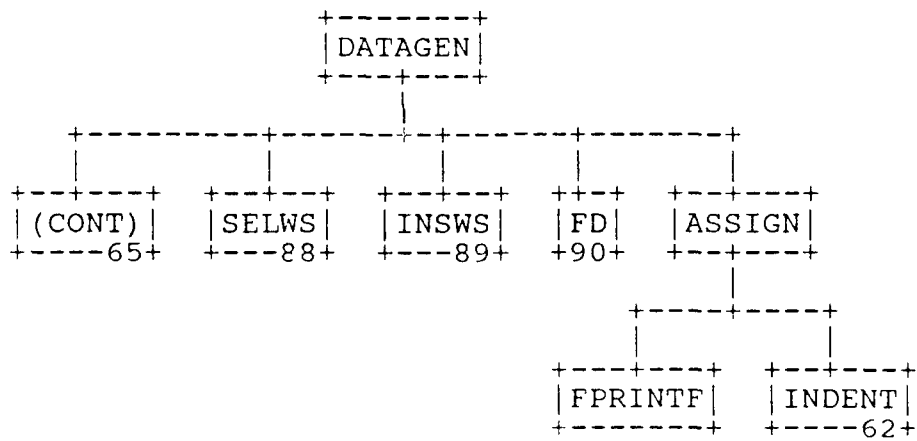
77



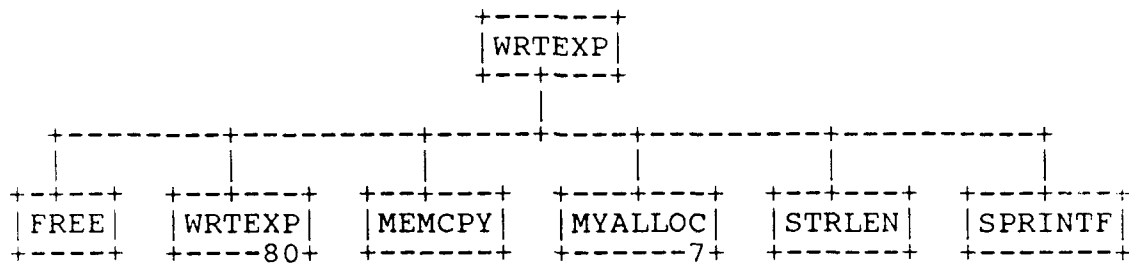
78



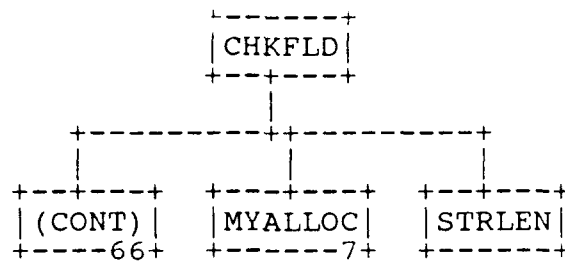
79



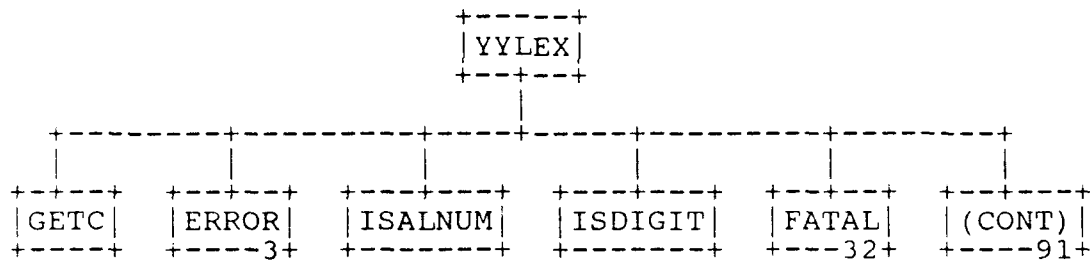
80



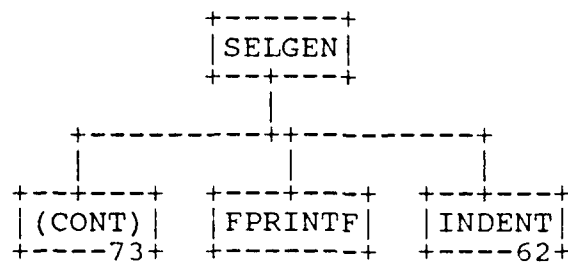
81



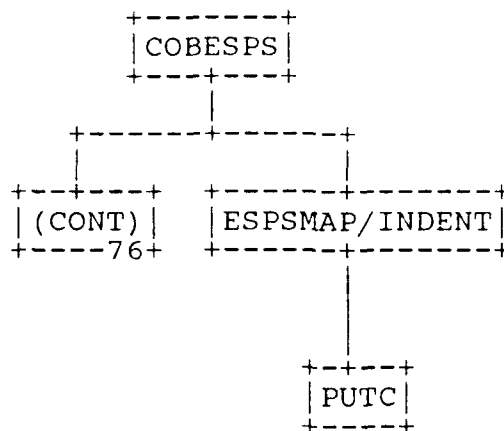
82



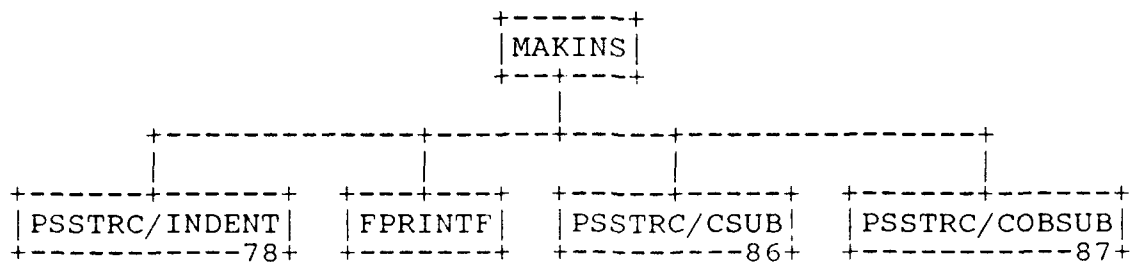
83



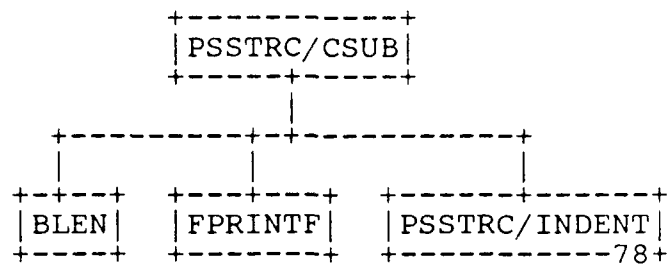
84



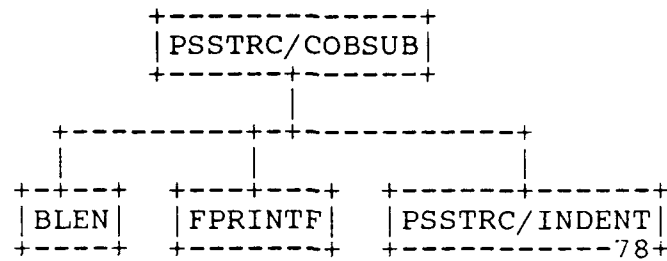
85



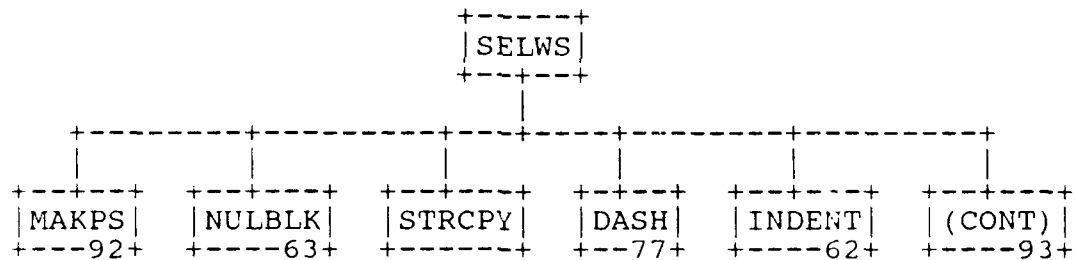
86



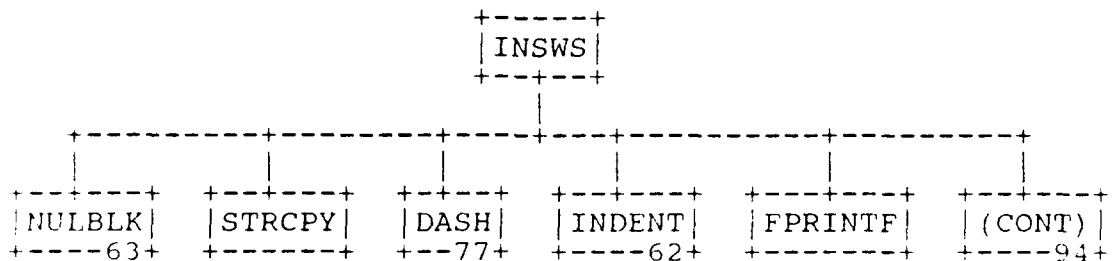
87



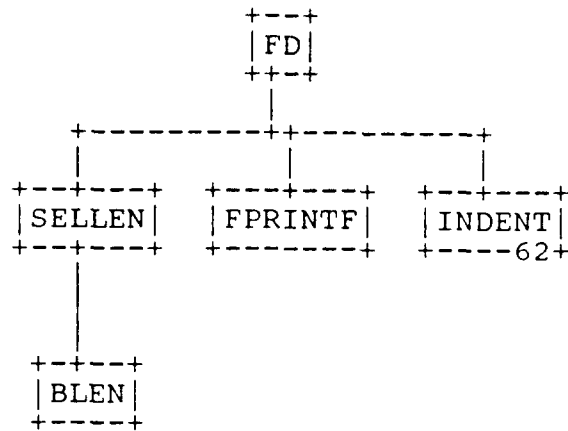
88



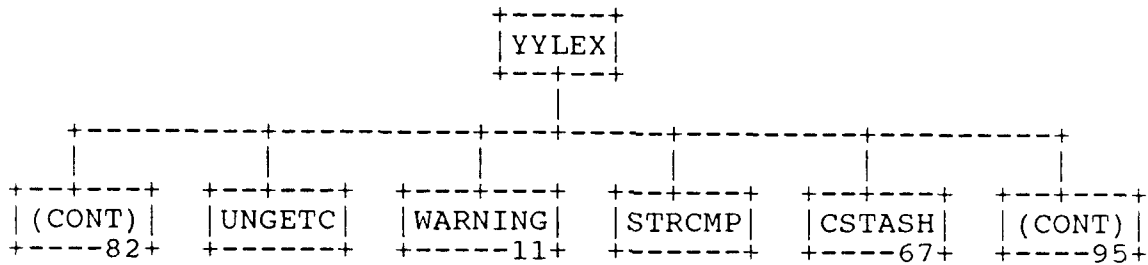
89



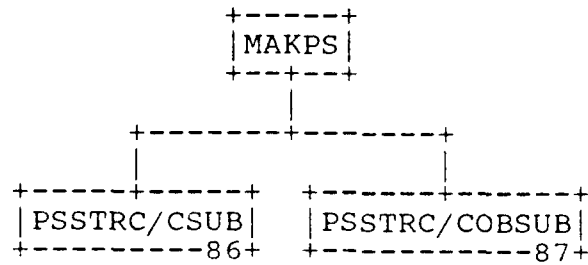
90



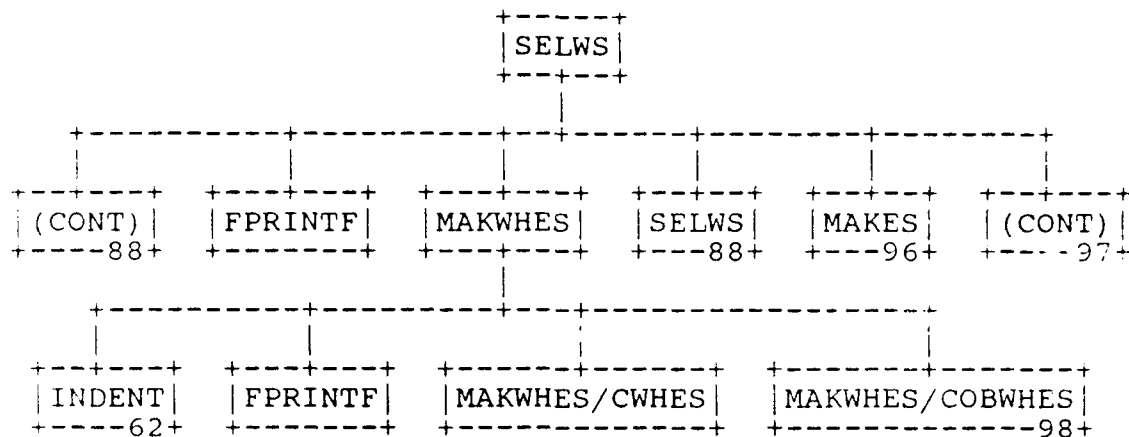
91



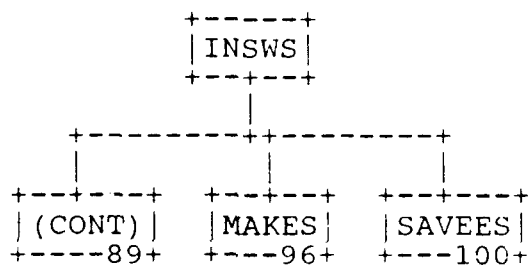
92



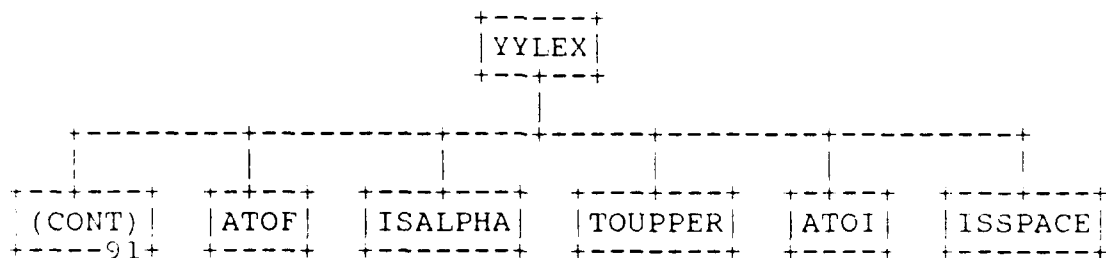
93



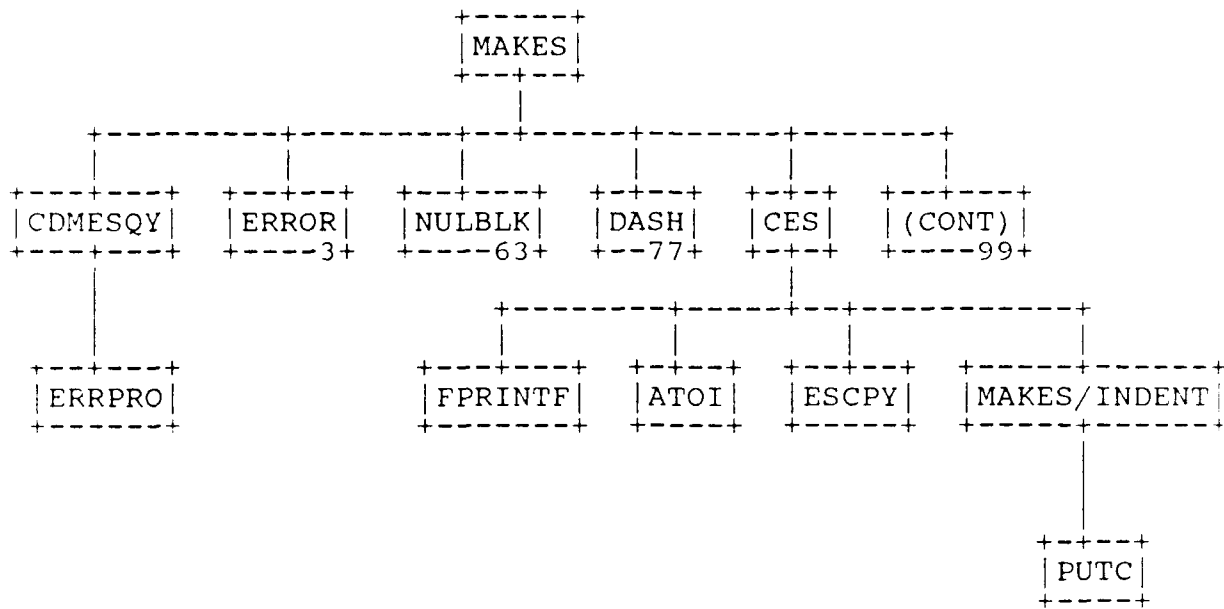
94



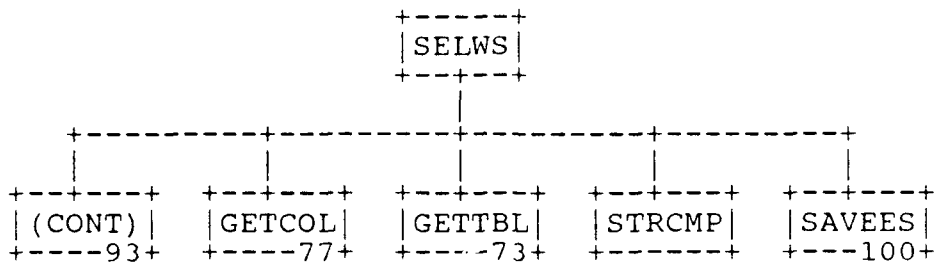
95



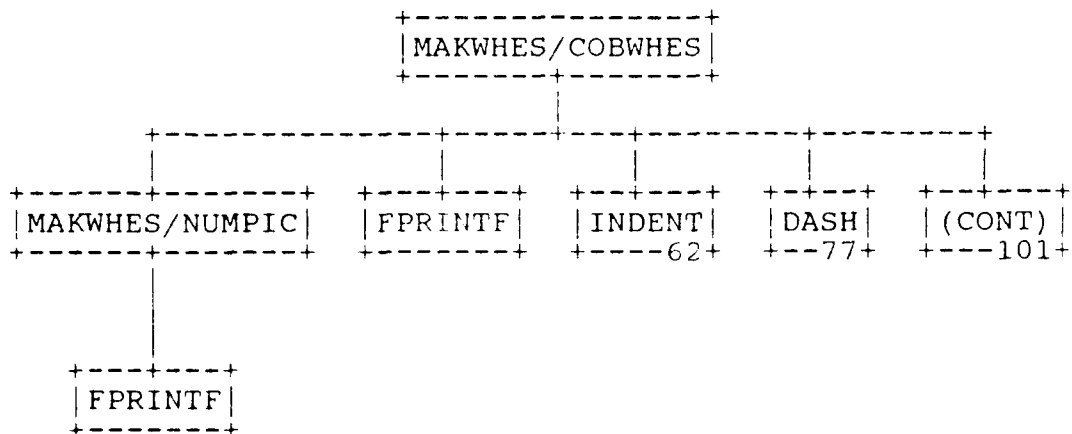
96



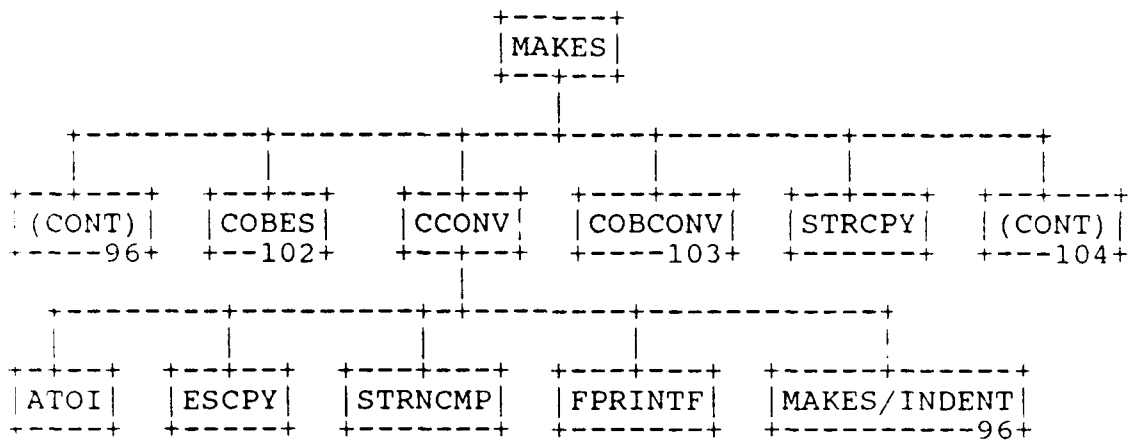
97



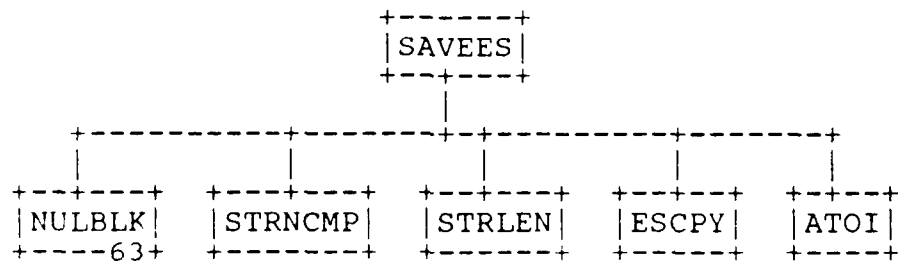
98



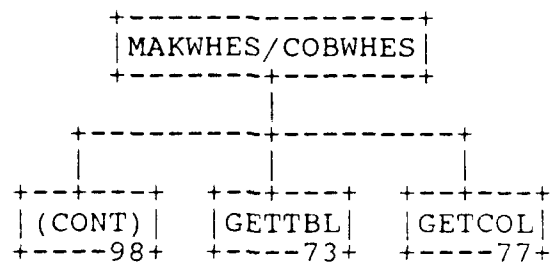
99



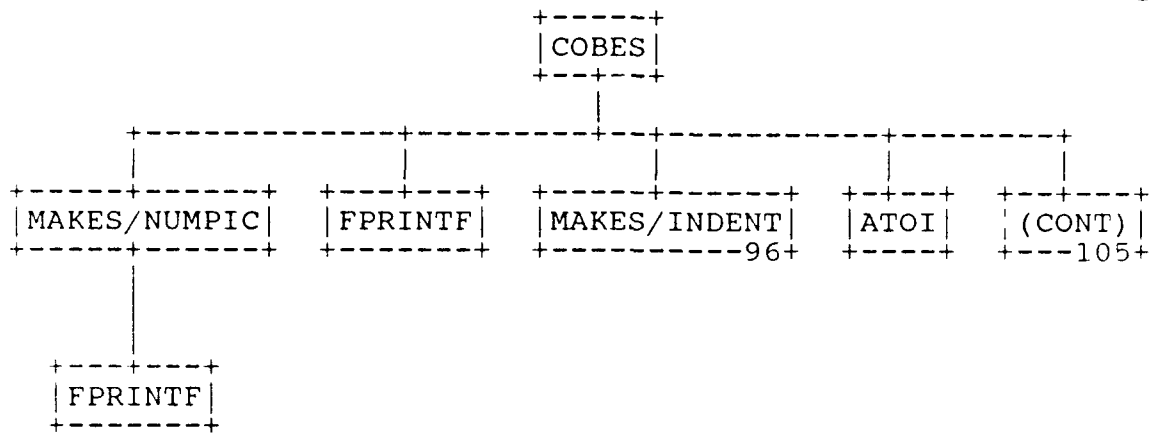
100



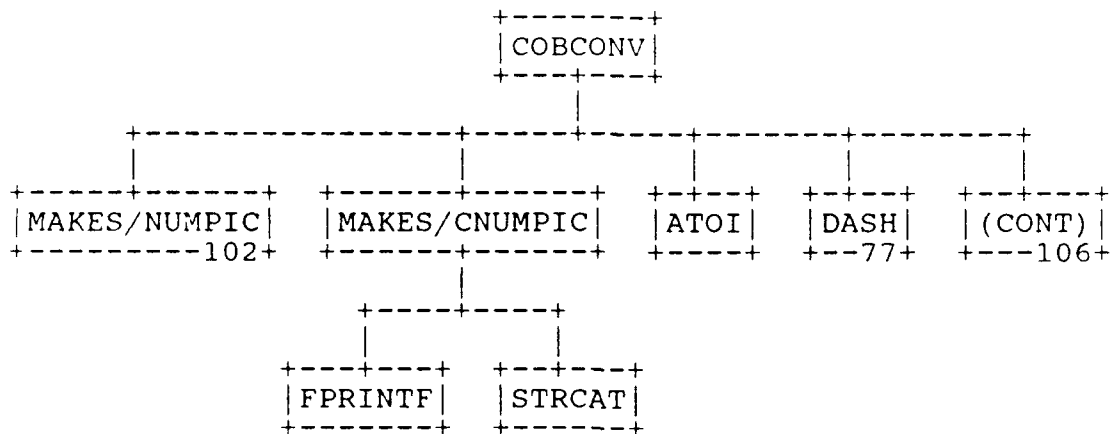
101



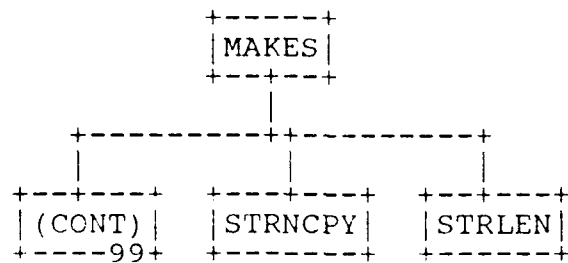
102



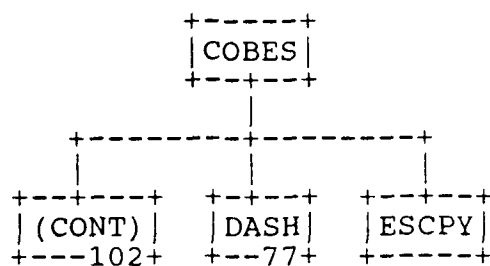
103



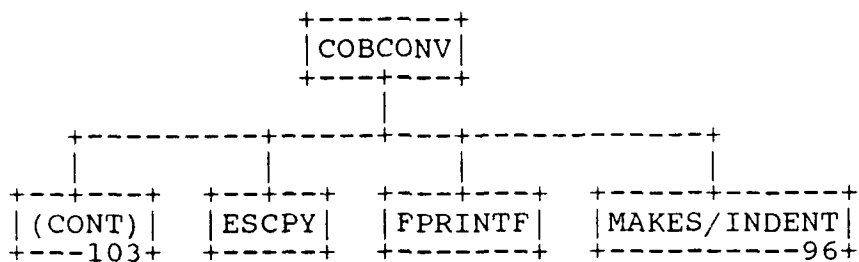
104



105



106



ABS		FLDRSV.....15
ACTRSV.....16		FLDTYP
ADDCHK.....33		FNDATT.....68
ADDFRM		FNDFRM.....17
ASSIGN.....79		FOPEN
ATOF		FPRINTF
ATOI		FREE
BLDSUB.....18		FRMPDAT.....41
BLN		FRNTND.....2
BSCODE.....28		FWRITE
CALCSTAT.....57		GDATA
CALLOC		GEN.....17
CCONV.....99		GENAA.....19
CDMESQY.....96		GENAAL.....19
CES.....96		GENACT.....19
CESPS		GENAE.....43
CHKARY.....48		GENAH.....43
CHKFLD.....46		GENAI.....61
CHKFRM.....33		GENAL.....29
CHKGRP.....71		GENAP.....41
CHKSIZE.....63		GENAQ.....29
CLRNDP.....71		GENAR.....42
CLSFIL.....75		GENAS.....59
COBCONV.....103		GENAT.....60
COBES.....102		GENBEG.....17
COBESPS.....76		GENCHG.....36
COBPE.....63		GENDB.....55
COPFLD		GENDOA.....29
CPE		GENDS.....55
CSTASH.....67		GENFP.....53
CTLRSV.....15		GENFS.....52
DASH.....77		GENFSD.....27
DATAGEN.....65		GENINS.....55
DBFREAD.....28		GENMAIN.....17
DCLINDX.....70		GENNDP.....54
DELFLD		GENPAG.....10
ENDGEN.....62		GENTAL.....51
ERROR.....3		GENTRG.....35
ERRPRO		GETC
ESCPY		GETCOL.....77
ESPSMAP.....76		GETFILE.....3
ESPSMAP/INDENT....84		GETPTH.....14
FATAL.....32		GETTBL.....73
FCLOSE		GFLDPT.....47
FD.....90		GRP/MAIN.....1
FILELNK.....65		HASDATA.....18
FLANCI.....4		HASITEM.....17

HASLOWER		PMSGLC	
INDENT.....62		PMSGLS	
INITAL		PRINTF	
INITFP		PROCGEN.....64	
INSERT.....77		PSESMAP	
INSMAP		PSSTRC/COBSUB.....87	
INRSRV.....16		PSSTRC/CSUB.....86	
INWS.....89		PSSTRC/INDENT.....78	
ISALNUM		PTHPTR	
ISALPHA		PUTC	
ISDIGIT		READDB.....28	
ISOPNE		RSETNDP.....38	
ISSPACE		RSETSTAT.....41	
MAKACT.....7		RWEXPD.....13	
MAKES.....96		RWOPN.....9	
MAKES/CNUMPIC....103		RWSP/FIXFRM.....13	
MAKES/INDENT.....96		SAVEES.....100	
MAKES/NUMPIC....102		SELECT.....63	
MAKFLD		SELGEN.....63	
MAKINS.....85		SELLEN.....90	
MAKINT.....7		SELMAP.....76	
MAKPS.....92		SELOPN.....44	
MAKQR.....35		SELRSV.....25	
MAKSTR.....47		SELWHR.....29	
MAKWH.....78		SELWS.....88	
MAKWHE.....93		SETNDP.....40	
MAKWHE/COBWHE...98		SPRINTF	
MAKWHE/CWHE		STATRSV.....15	
MAKWHE/NUMPIC....98		STDCODE 45	
MALLOC		STRASN	
MAP		STRCAT	
MAPDB.....39		STRCHR	
MAY		STRCMP	
MEMCMP		STRCPY	
MEMCPY		STRLEN	
MEMSET		STRNCMP	
MKINC.....26		STRNCPY	
MKPOS.....21		STRSPN	
MLPFRM.....9		STRUPC	
MYALLOC.....7		SYSMSG	
NDMLGEN.....30		TERMFP	
NDMLLAB.....64		TOUPPER	
NDMLLNK.....78		TRGRSV.....16	
NULBLK.....63		TRMNAT	
OISCR		TRMNDML	
OPNFIL.....74		UNGUTC	
PEMAP.....63		UQFOR.....51	

UQPTH.....24
USING.....64
VISITA.....56
WARNING.....11
WINRSV.....14
WRTEXP.....80
WRTFRM.....8
WRTFRM/DBFCLOS....34
WRTFRM/FORMAT
WRTFRM/TBFCLOS....23
WRTFRM/WRTDBF....34
WRTFRM/WRTFLD....22
WRTFRM/WRTTBF....23
WRTFRM/WRTTXT....12
YYERROR
YYLEX.....82
YYPARSE....4

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."