INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 24 - Report Writer Product Specification

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

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited
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

FOR THE COMMANDER:

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

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

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.
<table>
<thead>
<tr>
<th>REPORT DOCUMENTATION PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. REPORT SECURITY CLASSIFICATION</td>
</tr>
<tr>
<td>2a. SECURITY CLASSIFICATION AUTHORITY</td>
</tr>
<tr>
<td>2b. DECLASSIFICATION/DOWNGRADING SCHEDULE</td>
</tr>
<tr>
<td>3. DISTRIBUTION/AVAILABILITY OF REPORT</td>
</tr>
<tr>
<td>4. PERFORMING ORGANIZATION REPORT NUMBER(S)</td>
</tr>
<tr>
<td>5. MONITORING ORGANIZATION REPORT NUMBER(S)</td>
</tr>
<tr>
<td>6a. NAME OF PERFORMING ORGANIZATION</td>
</tr>
<tr>
<td>7a. NAME OF MONITORING ORGANIZATION</td>
</tr>
<tr>
<td>6b. OFFICE SYMBOL</td>
</tr>
<tr>
<td>7b. ADDRESS (City, State, and ZIP Code)</td>
</tr>
<tr>
<td>8a. NAME OF FUNDING/SPONSORING ORGANIZATION</td>
</tr>
<tr>
<td>9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM.</td>
</tr>
<tr>
<td>8b. OFFICE SYMBOL</td>
</tr>
<tr>
<td>10. SOURCE OF FUNDING NUMS.</td>
</tr>
<tr>
<td>PROGRAM NO.</td>
</tr>
<tr>
<td>PROJECT NO.</td>
</tr>
<tr>
<td>TASK NO.</td>
</tr>
<tr>
<td>WORK UNIT NO.</td>
</tr>
<tr>
<td>11. TITLE</td>
</tr>
<tr>
<td>12. PERSONAL AUTHOR(S)</td>
</tr>
<tr>
<td>13a. TYPE OF REPORT</td>
</tr>
<tr>
<td>13b. TIME COVERED</td>
</tr>
<tr>
<td>14. DATE OF REPORT</td>
</tr>
<tr>
<td>15. PAGE COUNT</td>
</tr>
<tr>
<td>16. SUPPLEMENTARY NOTICES</td>
</tr>
<tr>
<td>17. COSATI CODES</td>
</tr>
<tr>
<td>FIELD</td>
</tr>
<tr>
<td>GROUP</td>
</tr>
<tr>
<td>18. SUBJECT TERMS</td>
</tr>
<tr>
<td>19. ABSTRACT</td>
</tr>
<tr>
<td>This specification establishes the detailed design of the Report Writer computer program.</td>
</tr>
<tr>
<td>BLOCK 11:</td>
</tr>
<tr>
<td>INTEGRATED INFORMATION SUPPORT SYSTEM</td>
</tr>
<tr>
<td>Vol VIII - User Interface Subsystem</td>
</tr>
<tr>
<td>Part 24 - Report Writer Product Specification</td>
</tr>
<tr>
<td>20. DISTRIBUTION/AVAILABILITY OF ABSTRACT</td>
</tr>
<tr>
<td>21. ABSTRACT SECURITY CLASSIFICATION</td>
</tr>
<tr>
<td>22a. NAME OF RESPONSIBLE INDIVIDUAL</td>
</tr>
<tr>
<td>22b. TELEPHONE NO. (Include Area Code)</td>
</tr>
<tr>
<td>22c. OFFICE SYMBOL</td>
</tr>
</tbody>
</table>

**UD FORM 1473, 83 APR**

**EDITION OF 1 JAN 73 IS OBSOLETE**

**Unclassified**

**SECURITY CLASSIFICATION OF THIS PAGE**
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:

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

<table>
<thead>
<tr>
<th>SECTION</th>
<th>CONTENTS</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>SCOPE</td>
<td>1-1</td>
</tr>
<tr>
<td>1.1</td>
<td>Identification</td>
<td>1-1</td>
</tr>
<tr>
<td>1.2</td>
<td>Functional Summary</td>
<td>1-1</td>
</tr>
<tr>
<td>2.0</td>
<td>DOCUMENTS</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1</td>
<td>Reference Documents</td>
<td>2-1</td>
</tr>
<tr>
<td>2.2</td>
<td>Terms and Abbreviations</td>
<td>2-2</td>
</tr>
<tr>
<td>3.0</td>
<td>REQUIREMENTS</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>Structural Description</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1.1</td>
<td>RWG</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1.2</td>
<td>HRW</td>
<td>3-2</td>
</tr>
<tr>
<td>3.2</td>
<td>Functional Flow</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3</td>
<td>Interfaces</td>
<td>3-4</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Forms Language Compiler</td>
<td>3-4</td>
</tr>
<tr>
<td>3.3.2</td>
<td>CDM Data Dictionary</td>
<td>3-4</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Generated Report Interfaces</td>
<td>3-4</td>
</tr>
<tr>
<td>3.4</td>
<td>Program Interrupts</td>
<td>3-4</td>
</tr>
<tr>
<td>3.5</td>
<td>Timing and Sequencing Description</td>
<td>3-4</td>
</tr>
<tr>
<td>3.6</td>
<td>Special Control Features</td>
<td>3-4</td>
</tr>
<tr>
<td>3.7</td>
<td>Storage Allocation</td>
<td>3-5</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Data Base Definition</td>
<td>3-5</td>
</tr>
<tr>
<td>3.7.1.1</td>
<td>File Descriptions</td>
<td>3-5</td>
</tr>
<tr>
<td>3.7.1.2</td>
<td>Table Description</td>
<td>3-7</td>
</tr>
<tr>
<td>3.8</td>
<td>Object Code Creation</td>
<td>3-8</td>
</tr>
<tr>
<td>3.9</td>
<td>Adaptation Data</td>
<td>3-8</td>
</tr>
<tr>
<td>3.10</td>
<td>Detailed Design Description</td>
<td>3-8</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Main Program List</td>
<td>3-8</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Module List</td>
<td>3-10</td>
</tr>
<tr>
<td>3.10.3</td>
<td>External Routines List</td>
<td>3-20</td>
</tr>
<tr>
<td>3.10.4</td>
<td>Include File List</td>
<td>3-23</td>
</tr>
<tr>
<td>3.10.5</td>
<td>Where Include File Used List</td>
<td>3-25</td>
</tr>
<tr>
<td>3.10.6</td>
<td>Where External Routine Used List</td>
<td>3-52</td>
</tr>
<tr>
<td>3.10.7</td>
<td>Main Program Parts List</td>
<td>3-69</td>
</tr>
<tr>
<td>3.10.8</td>
<td>Module Documentation</td>
<td>3-78</td>
</tr>
<tr>
<td>3.10.9</td>
<td>Include File Description</td>
<td>3-368</td>
</tr>
<tr>
<td>3.10.10</td>
<td>Hierarchy Chart</td>
<td>3-384</td>
</tr>
<tr>
<td>3.11</td>
<td>Program Listings Comments</td>
<td>3-437</td>
</tr>
<tr>
<td>4.0</td>
<td>QUALITY ASSURANCE PROVISIONS</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction and Definitions</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2</td>
<td>Computer Programming and Test Evaluation</td>
<td>4-1</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3-1</td>
<td>RWG Structure</td>
<td>3-2</td>
</tr>
<tr>
<td>3-2</td>
<td>Report Writer Environment Data Flow</td>
<td>3-3</td>
</tr>
<tr>
<td>3-3</td>
<td>Hierarchical Report Writer Data Flow</td>
<td>3-4</td>
</tr>
</tbody>
</table>
1.1 Identification

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

1.2 Functional Summary

The RW is used to report selected information stored in the database accessible through the Common Data Model (CDM).

The major functions of the RW are:

1. The placement and formatting of fixed textual information and database information, i.e., CDM data.

2. The summarization of simple statistical attributes of the reported information such as counts, sums, and averages.

3. The retrieval of data from the CDM.
SECTION 2
DOCUMENTS

2.1 Reference Documents


2.2 Terms and Abbreviations

**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 that 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.

**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.

**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.

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

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

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

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

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

IISS Function Screen: the first screen that is displayed after logon. It allows the user to specify the function he wants to access and the device type and device name on which he is working.
Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

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

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

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

**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.
Report Definition Language: (RDL), an extension of the Forms Definition Language that includes retrieval and calculation of database information and is used to define reports.

Report Writer: (RW), part of the Application Generator that generates source code for report programs based on a language input.

Report Writer Generator: (RWG), used to translate report definitions defined using the RDL into programs that access data bases via the CDM.

Subform: a form that is used within another form.

User Interface: (UI), I ISS 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 I ISS User Interface subsystems that are used by applications programmers as they develop I ISS applications. The UIDS includes the Form Editor and the Application Generator.

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


3.1.1 RWG

The RWG is used to translate report definitions defined using the Report Definition Language (RDL) into programs that access databases via the CDM and report the extracted data in a formatted way. Conceptually, the RWG is a compiler that takes RDL as input and generates:

- Binary form definition files that determine the layout of the report pages by parsing the RDL using the modules YTAB.C, FLANSP.C, and WRTFRM.C.

- A database query program that maps the CDM external schema to the presentation schema (forms defined by the FD files). The module NDMLGEN.C calls the COBOL module CDMESQY.PRC to get meta data about the report query 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 query and must be precompiled using the NDML precompiler.

- A control flow program based on the specified conditions. This is the main module of the generated report that uses the Application Interface to put data from the CDM into the report forms and arrange the printed output.
3.1.2 HRW

The HRW is a post processor which takes a report generated by the Report Writer Generator and rearranges it into an appropriate tree structure. The data to be displayed can be either a true hierarchy where each box appears only once or a network where a box may appear more than once.

3.2 Functional Flow

Figure 3-2 is a data flow diagram of the Report Writer environment.
Figure 3-2. Report Writer Environment Data Flow

Figure 3-3 is a data flow diagram of the Hierarchical Report Writer.
3.3 Interfaces

3.3.1 Forms Language Compiler

The RWG uses the modules YTAB.C, FLANSP.C, and WRTFRM.C of the Forms Language Compiler (FLAN) to generate binary form definition files from its RDL 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 query program extracts meta data about the report query from the CDM Data Dictionary to check for legal External Schema to Presentation Schema mappings.

3.3.3 Generated Report Interfaces

The generated report 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 Report Writer.

3.5 Timing and Sequencing Description

This section does not apply to the detailed design of the Report Writer.

3.6 Special Control Features

This section does not apply to the detailed design of the Report Writer.
3.7 Storage Allocation

The Report Writer Generator executable size is 492 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 */
}
typedef struct /* text record */
{
    short row; /* starting row */
    short col; /* starting col */
    short len; /* total length */
    char linefeed;
} TXTREC;

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

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

PURPOSE: This file is a temporary file that pertains to the current report query. It is input to the generated control flow program to produce the printed report and becomes obsolete after the report is generated. If the report
program 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 report name as given on the CREATE REPORT statement of the RDL file - generated C code.

PURPOSE: This is the control flow program generated by the RWG that uses the Application Interface to put data from the CDM into the report forms and arrange the printed output.

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

4. FILE NAME: *X.PRC where * is the report name as given on the CREATE REPORT statement of the RDL file - generated COBOL code that contains:

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

PURPOSE: This code contains the CDM query procedures to do the report query specified by the NDML SELECT and maps the External Schema to the Presentation Schema.

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

5. FILE NAME: *.WRN where * is the report name as given on the CREATE REPORT statement of the RDL 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 RWG are under the control of the CDM and are predefined.
3.8 Object Code Creation

The RWG 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 program can be compiled using any ANSI COBOL compiler.

3.9 Adaptation Data

The C source modules for the RWG 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.
### REPORT WRITER Main Program List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
</tbody>
</table>
3.10.2 **Module List**

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

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTRSV</td>
<td>ACTION RESOLVE</td>
</tr>
<tr>
<td>ADDCHK</td>
<td>ADD POSITION TO CHECK LIST</td>
</tr>
<tr>
<td>ARRANGE</td>
<td>ARRANGE CHART AND ASSIGNS PAGE NUMBERS</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>ASSIGN FILE SECTION</td>
</tr>
<tr>
<td>BLDMOD</td>
<td>BUILD MODULE</td>
</tr>
<tr>
<td>BLDNODE</td>
<td>BUILD NODE</td>
</tr>
<tr>
<td>BLDSUB</td>
<td>BUILD SUBROUTINES</td>
</tr>
<tr>
<td>BSCODE</td>
<td>BUILD SUBROUTINE CODE</td>
</tr>
<tr>
<td>CALCSTAT</td>
<td>CALCULATE STATISTIC</td>
</tr>
<tr>
<td>CCONV</td>
<td>C CONVERSIONS</td>
</tr>
<tr>
<td>CDMESQY</td>
<td>PROGRAM NAME CDMESQY</td>
</tr>
<tr>
<td>CES</td>
<td>C ES</td>
</tr>
<tr>
<td>CESPS</td>
<td>C ES TO PS</td>
</tr>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td>CHKGRP</td>
<td>CHECK FOR GROUP SEPERATORS OR END OF FILE</td>
</tr>
<tr>
<td>CHKSIZE</td>
<td>CHECK SIZE OF ITEMS DOING CONVERSIONS ON</td>
</tr>
<tr>
<td>CLOSEGAP</td>
<td>CLOSE GAP IN TREE</td>
</tr>
<tr>
<td>CLRNDP</td>
<td>CLEAR NODUPLICATE FIELDS</td>
</tr>
<tr>
<td>CLSFLIL</td>
<td>CLOSE FILES</td>
</tr>
</tbody>
</table>
### REPORT WRITER Module List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBCONV</td>
<td>COBOL CONVERSIONS</td>
</tr>
<tr>
<td>COBES</td>
<td>COBOL ES RECORD</td>
</tr>
<tr>
<td>COBESPS</td>
<td>COBOL ES TO PS</td>
</tr>
<tr>
<td>COBPE</td>
<td>COBOL PE</td>
</tr>
<tr>
<td>COPYNODE</td>
<td>COPY A NODE IN TREE</td>
</tr>
<tr>
<td>CPE</td>
<td>C PE</td>
</tr>
<tr>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td>CTLRSV</td>
<td>CONTROL RESOLVE</td>
</tr>
<tr>
<td>DASH</td>
<td>WRITE DASH ' - '</td>
</tr>
<tr>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
</tr>
<tr>
<td>DBFREAD</td>
<td>GENERATE DATA BASE FREAD</td>
</tr>
<tr>
<td>DCLINDX</td>
<td>DECLARE INDEX VARIABLES</td>
</tr>
<tr>
<td>DELNODE</td>
<td>DELETE A SPECIFIED NODE IN TREE</td>
</tr>
<tr>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td>ENDGEN</td>
<td>END GERNERATE</td>
</tr>
<tr>
<td>ERROR</td>
<td>ISSUE ERROR MESSAGE</td>
</tr>
<tr>
<td>ESPSMAP</td>
<td>THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING</td>
</tr>
<tr>
<td>ESPSMAP/INDENT</td>
<td>INDENT</td>
</tr>
<tr>
<td>FATAL</td>
<td>ISSUE FATAL ERROR MESSAGE</td>
</tr>
</tbody>
</table>
# REPORT WRITER Module List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
</tr>
<tr>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
</tr>
<tr>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
</tr>
<tr>
<td>FLDRSV</td>
<td>FIELD RESOLVE</td>
</tr>
<tr>
<td>FLDTYP</td>
<td>FIELD TYPE</td>
</tr>
<tr>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
</tr>
<tr>
<td>FNDFRM</td>
<td>FIND FORM</td>
</tr>
<tr>
<td>FRMPDAT</td>
<td>FORM PDATA</td>
</tr>
<tr>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td>GEN</td>
<td>GENERATE A LINE OF CODE</td>
</tr>
<tr>
<td>GENAA</td>
<td>GENERATE PROCEDURE &quot;ADDACT&quot; ADD AN ACTION</td>
</tr>
<tr>
<td>GENAAL</td>
<td>GENERATE PROCEDURE &quot;ADDAL&quot; ADD ACTION LIST</td>
</tr>
<tr>
<td>GENACT</td>
<td>GENERATE ACTIONS</td>
</tr>
<tr>
<td>GENAE</td>
<td>GENERATE ACTION EXIT</td>
</tr>
<tr>
<td>GENAH</td>
<td>GENERATE ACTION HELP</td>
</tr>
<tr>
<td>GENAI</td>
<td>GENERATE ACTION INSERT</td>
</tr>
<tr>
<td>GENAL</td>
<td>GENERATE ACTION LIST</td>
</tr>
<tr>
<td>GENAP</td>
<td>GENERATE ACTION PAGE</td>
</tr>
<tr>
<td>GENAQ</td>
<td>GENERATE ACTION QUERY (SELECT)</td>
</tr>
<tr>
<td>GENAR</td>
<td>GENERATE ACTION PRESENT</td>
</tr>
<tr>
<td>GENAS</td>
<td>GENERATE ACTION SET</td>
</tr>
<tr>
<td>Module Name</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>GENAT</td>
<td>GENERATE ACTION SIGNAL</td>
</tr>
<tr>
<td>GENBEG</td>
<td>GENERATE BEGINNING OF APPLICATION OR REPORT</td>
</tr>
<tr>
<td>GENCCHG</td>
<td>GENERATE CHANGE DECLARATIONS</td>
</tr>
<tr>
<td>GENDB</td>
<td>GENERATE DATA BASE RECORDS AND FILE DECLARATIONS</td>
</tr>
<tr>
<td>GENDOA</td>
<td>GENERATE PROCEDURE &quot;DOACT&quot; DO ACTION</td>
</tr>
<tr>
<td>GENDS</td>
<td>GENERATE DATA DATA STRUCTURES</td>
</tr>
<tr>
<td>GENFP</td>
<td>GENERATE FORM PATH</td>
</tr>
<tr>
<td>GENFS</td>
<td>GENERATE FORM DATA STRUCTURES</td>
</tr>
<tr>
<td>GENFSD</td>
<td>GENERATE FORM STRUCTURE DATA INITIALIZATION</td>
</tr>
<tr>
<td>GENINS</td>
<td>GENERATE INSERT DECLARATIONS</td>
</tr>
<tr>
<td>GENMAIN</td>
<td>GENERATE MAIN PROGRAM</td>
</tr>
<tr>
<td>GENNDP</td>
<td>GENERATE NODUPLICATE DECLARATIONS</td>
</tr>
<tr>
<td>GENPAG</td>
<td>GENERATE NEWPAG PROCEDURE</td>
</tr>
<tr>
<td>GETCOL</td>
<td>GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING</td>
</tr>
<tr>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM THE USER</td>
</tr>
<tr>
<td>GETFIT</td>
<td>GET SUBTREE THAT FITS ON PAGE</td>
</tr>
<tr>
<td>GETLOWLEF</td>
<td>GET LOWER LEFT CHILD NODE</td>
</tr>
<tr>
<td>GETLOWRIT</td>
<td>GET LOWER RIGHT CHILD NODE</td>
</tr>
</tbody>
</table>
### REPORT WRITER Module List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETPAR</td>
<td>GET PARENT NODE</td>
</tr>
<tr>
<td>GETPTH</td>
<td>GET PATH</td>
</tr>
<tr>
<td>GETSIZE</td>
<td>GET SUBTREE SIZE</td>
</tr>
<tr>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
</tr>
<tr>
<td>GETTOP</td>
<td>GET TOP OF TREE</td>
</tr>
<tr>
<td>GETUPLFT</td>
<td>GET UPPER LEFTMOST NODE</td>
</tr>
<tr>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
</tr>
<tr>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td>HASDATA</td>
<td>DETERMINE IF THERE ARE ANY SELECT STATEMENTS</td>
</tr>
<tr>
<td>HASITEM</td>
<td>THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN</td>
</tr>
<tr>
<td>HASLOWER</td>
<td>HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?</td>
</tr>
<tr>
<td>HBALANC</td>
<td>HORIZONTAL TREE BALANCE</td>
</tr>
<tr>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>INDENT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
</tr>
<tr>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
</tr>
<tr>
<td>INSRSRV</td>
<td>INSERT RESOLVE</td>
</tr>
<tr>
<td>INSWS</td>
<td>INSERT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td>ISOPNE</td>
<td>DETERMINE IF THIS FIELD IS OPEN ENDED</td>
</tr>
<tr>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
</tr>
</tbody>
</table>

3-15
<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKES</td>
<td>MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE</td>
</tr>
<tr>
<td>MAKES/CNUMPIC</td>
<td>C NUMBERS</td>
</tr>
<tr>
<td>MAKES/INDENT</td>
<td>INDENT</td>
</tr>
<tr>
<td>MAKES/NUMPIC</td>
<td>NUMBER PICTURE CLAUSE</td>
</tr>
<tr>
<td>MAKINS</td>
<td>MAKE INSERT</td>
</tr>
<tr>
<td>MAKINT</td>
<td>MAKE EXPRESSION INTO AN INTEGER</td>
</tr>
<tr>
<td>MAKPS</td>
<td>MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE</td>
</tr>
<tr>
<td>MAKQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
</tr>
<tr>
<td>MAKSTR</td>
<td>MAKE EXPRESSION INTO A STRING</td>
</tr>
<tr>
<td>MAKWH</td>
<td>MAKE WHERE</td>
</tr>
<tr>
<td>MAKWHES</td>
<td>MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES</td>
</tr>
<tr>
<td>MAKWHES/COBWHES</td>
<td>COBOL WHERE ES</td>
</tr>
<tr>
<td>MAKWHES/CWHES</td>
<td>C WHERE ES</td>
</tr>
<tr>
<td>MAKWHES/NUMPIC</td>
<td>NUMBER PICTURE CLAUSE</td>
</tr>
<tr>
<td>MAPDB</td>
<td>MAP DATABASE</td>
</tr>
<tr>
<td>MKINC</td>
<td>MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)</td>
</tr>
<tr>
<td>MKPOS</td>
<td>MAKE POSITION NODE</td>
</tr>
<tr>
<td>MLPFRM</td>
<td>MAKE A LIST OF PRESENTED FORMS</td>
</tr>
<tr>
<td>MODPAGE</td>
<td>MODIFY PAGES</td>
</tr>
</tbody>
</table>
## REPORT WRITER Module List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVCLD</td>
<td>MOVE CHILDREN</td>
</tr>
<tr>
<td>MOVECLD</td>
<td>MOVE CHILD'S POSITION</td>
</tr>
<tr>
<td>MYALLOC</td>
<td>MY MALLOC</td>
</tr>
<tr>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td>NDMLLAB</td>
<td>GENERATE LABELS</td>
</tr>
<tr>
<td>NDMLLNK</td>
<td>LINKAGE SECTION</td>
</tr>
<tr>
<td>NEXTLEV</td>
<td>ADVANCE POINTERS TO NEXT LEVEL OF SUBTREE</td>
</tr>
<tr>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
</tr>
<tr>
<td>OPNFIL</td>
<td>GENERATE OPEN FILE SECTION</td>
</tr>
<tr>
<td>PAGNODE</td>
<td>PAGE NODES</td>
</tr>
<tr>
<td>PAGTREE</td>
<td>PAGE TREE</td>
</tr>
<tr>
<td>PEMAP</td>
<td>THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING</td>
</tr>
<tr>
<td>PRNT</td>
<td>PRINT MODULE NAMES HIERARCHICALLY</td>
</tr>
<tr>
<td>PRNTREE</td>
<td>PRINT TREE</td>
</tr>
<tr>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
</tr>
<tr>
<td>PSSTRC/COBSUB</td>
<td>COBOL SUBSTITUTE</td>
</tr>
<tr>
<td>PSSTRC/CSUB</td>
<td>C SUBSTITUTE</td>
</tr>
<tr>
<td>PSSTRC/INDENT</td>
<td>INDENT</td>
</tr>
<tr>
<td>PUTLIN</td>
<td>PRINT LEVEL OF TREE</td>
</tr>
<tr>
<td>READDB</td>
<td>READ DATA BASE</td>
</tr>
</tbody>
</table>
# REPORT WRITER Module List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>README</td>
<td>READ DUMPTREE FILE</td>
</tr>
<tr>
<td>REPOS</td>
<td>REPOSITION MODULE EXPANSIONS</td>
</tr>
<tr>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
</tr>
<tr>
<td>RSETSTAT</td>
<td>RESET STATISTIC</td>
</tr>
<tr>
<td>RWEXPD</td>
<td>REPORT WRITER EXPAND ARRAYS</td>
</tr>
<tr>
<td>RWOPN</td>
<td>REPORT WRITER OPEN FORMS</td>
</tr>
<tr>
<td>RWSP/FIXPRM</td>
<td>FIX UP A FORM</td>
</tr>
<tr>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
</tr>
<tr>
<td>SELECT</td>
<td>GENERATE SELECT CODE</td>
</tr>
<tr>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
</tr>
<tr>
<td>SELLLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
</tr>
<tr>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
</tr>
<tr>
<td>SELOPN</td>
<td>SELECT OPEN</td>
</tr>
<tr>
<td>SELRSV</td>
<td>SELECT RESOLVE</td>
</tr>
<tr>
<td>SELWHR</td>
<td>SELECT WHERE</td>
</tr>
<tr>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
</tr>
<tr>
<td>SORT</td>
<td>SORT MODULE NAMES</td>
</tr>
<tr>
<td>SPLICE</td>
<td>SPLICE TREE INTO ANOTHER TREE</td>
</tr>
<tr>
<td>SPLITNODE</td>
<td>SPLIT A NODE FOR PAGE BREAKS</td>
</tr>
</tbody>
</table>
REPORT WRITER Module List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATRSV</td>
<td>STATISTIC RESOLVE</td>
</tr>
<tr>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
</tr>
<tr>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
<tr>
<td>TRGRSV</td>
<td>TRIGGER RESOLVE</td>
</tr>
<tr>
<td>UQFOR</td>
<td>UNIVERSAL QUALIFIER FOR LOOP</td>
</tr>
<tr>
<td>UQPTH</td>
<td>UNIVERSAL QUALIFIER PATH</td>
</tr>
<tr>
<td>USING</td>
<td>GENERATE USING SECTION</td>
</tr>
<tr>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
</tr>
<tr>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
</tr>
<tr>
<td>WINRSV</td>
<td>WINDOW RESOLVE</td>
</tr>
<tr>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
<tr>
<td>WRTFRM/DBFCLOS</td>
<td>DEFAULT BUFFER CLOSE</td>
</tr>
<tr>
<td>WRTFRM/FORMAT</td>
<td>INSERT FORMAT CODES</td>
</tr>
<tr>
<td>WRTFRM/TBFCLOS</td>
<td>TEXT BUFFER CLOSE</td>
</tr>
<tr>
<td>WRTFRM/WRTDBF</td>
<td>WRITE DEFAULT BUFFER</td>
</tr>
<tr>
<td>WRTFRM/WRTFLD</td>
<td>WRITE FIELD</td>
</tr>
<tr>
<td>WRTFRM/WRTTF</td>
<td>WRITE TEXT BUFFER</td>
</tr>
<tr>
<td>WRTFRM/WRTTXT</td>
<td>WRITE TEXT</td>
</tr>
<tr>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>YYYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>
3.10.3 External Routines List

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

<table>
<thead>
<tr>
<th>Module Name</th>
<th>First User</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>RWEXPD</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>FRNTND</td>
</tr>
<tr>
<td>ATOF</td>
<td>YYLEX</td>
</tr>
<tr>
<td>ATOI</td>
<td>YYLEX</td>
</tr>
<tr>
<td>BLEN</td>
<td>CHKFLD</td>
</tr>
<tr>
<td>CALLOC</td>
<td>GRP/MAIN</td>
</tr>
<tr>
<td>COPFLD</td>
<td>WINRSV</td>
</tr>
<tr>
<td>DELFLD</td>
<td>FLANCI</td>
</tr>
<tr>
<td>ERRPRO</td>
<td>CDMESQY</td>
</tr>
<tr>
<td>ESCPY</td>
<td>COBCONV</td>
</tr>
<tr>
<td>FCLOSE</td>
<td>WRTFRM</td>
</tr>
<tr>
<td>FGETS</td>
<td>DRALEV</td>
</tr>
<tr>
<td>FOPEN</td>
<td>WRTFRM</td>
</tr>
<tr>
<td>FPRINTF</td>
<td>PSSTRC/COBSUB</td>
</tr>
<tr>
<td>FREE</td>
<td>DOINDEX</td>
</tr>
<tr>
<td>FSEEK</td>
<td>CHKFLD</td>
</tr>
<tr>
<td>FTELL</td>
<td>STRIPLEV</td>
</tr>
<tr>
<td>FWRITE</td>
<td>READTREE</td>
</tr>
<tr>
<td>GDATA</td>
<td>HRW/MAIN</td>
</tr>
<tr>
<td>GETC</td>
<td>READTREE</td>
</tr>
<tr>
<td>INITIAL</td>
<td>FRNTND</td>
</tr>
<tr>
<td>INITFP</td>
<td>FRNTND</td>
</tr>
<tr>
<td>INSMAP</td>
<td>PROCGEN</td>
</tr>
<tr>
<td>ISALNUM</td>
<td>YYLEX</td>
</tr>
<tr>
<td>ISALPHA</td>
<td>YYLEX</td>
</tr>
<tr>
<td>ISDIGIT</td>
<td>YYLEX</td>
</tr>
<tr>
<td>ISSPACE</td>
<td>YYLEX</td>
</tr>
<tr>
<td>MAKFLD</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>MALLOC</td>
<td>WINRSV</td>
</tr>
<tr>
<td>MAP</td>
<td>PROCGEN</td>
</tr>
<tr>
<td>MAX</td>
<td>GETSIZE</td>
</tr>
<tr>
<td>MEMCMP</td>
<td>HRW/MAIN</td>
</tr>
<tr>
<td>MEMCPY</td>
<td>STRIPLEV</td>
</tr>
<tr>
<td>MEMSET</td>
<td>DRALEV</td>
</tr>
<tr>
<td>MIN</td>
<td>GETSIZE</td>
</tr>
<tr>
<td>OISCR</td>
<td>FRNTND</td>
</tr>
<tr>
<td>OUTSCR</td>
<td>HRW/MAIN</td>
</tr>
<tr>
<td>PMSGLC</td>
<td>GRP/MAIN</td>
</tr>
<tr>
<td>PMSGMLS</td>
<td>BLDMOD</td>
</tr>
<tr>
<td>PRINTF</td>
<td>PRNT</td>
</tr>
<tr>
<td>Module Name</td>
<td>First User</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>PSESMAIN</td>
<td>PROCGEN</td>
</tr>
<tr>
<td>PTHPTR</td>
<td>UQPTH</td>
</tr>
<tr>
<td>PUTATT</td>
<td>HRW/MAIN</td>
</tr>
<tr>
<td>PUTC</td>
<td>PUTLIN</td>
</tr>
<tr>
<td>PUTCUR</td>
<td>HRW/MAIN</td>
</tr>
<tr>
<td>SPRINTF</td>
<td>GETFILE</td>
</tr>
<tr>
<td>STRASN</td>
<td>CHKARY</td>
</tr>
<tr>
<td>STRCAT</td>
<td>YYPARSE</td>
</tr>
<tr>
<td>STRCHR</td>
<td>PUTLIN</td>
</tr>
<tr>
<td>STRCMP</td>
<td>RWSP/FIXFRM</td>
</tr>
<tr>
<td>STRCPY</td>
<td>GETPTH</td>
</tr>
<tr>
<td>STRLEN</td>
<td>READTREE</td>
</tr>
<tr>
<td>STRNCPY</td>
<td>SAVEES</td>
</tr>
<tr>
<td>STRNCMP</td>
<td>WRTFRM/WRTFLD</td>
</tr>
<tr>
<td>STRSPN</td>
<td>GENAS</td>
</tr>
<tr>
<td>STRUPC</td>
<td>SORT</td>
</tr>
<tr>
<td>SYSMSG</td>
<td>WRTFRM</td>
</tr>
<tr>
<td>TERMFP</td>
<td>GRP/MAIN</td>
</tr>
<tr>
<td>TOUPPER</td>
<td>YYLEX</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>HRW/MAIN</td>
</tr>
<tr>
<td>TRMNATML</td>
<td>GRP/MAIN</td>
</tr>
<tr>
<td>UNGETC</td>
<td>YYLEX</td>
</tr>
<tr>
<td>YYERROR</td>
<td>YYPARSE</td>
</tr>
</tbody>
</table>
3.10.4 Include File List

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

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

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

<table>
<thead>
<tr>
<th>File Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHART</td>
<td>CHART INCLUDE FILE</td>
</tr>
<tr>
<td>CTLCHR</td>
<td>CONTROL CHARACTERS</td>
</tr>
<tr>
<td>CTYPE</td>
<td>***** PURPOSE NOT FOUND BY STRIPPER *****</td>
</tr>
<tr>
<td>ERRPRO</td>
<td>PROCESS ERROR INCLUDE FILE</td>
</tr>
<tr>
<td>FFFV2</td>
<td>FORM FILE FORMAT - VERSION 2</td>
</tr>
<tr>
<td>FLAN.y&quot;</td>
<td>***** PURPOSE NOT FOUND BY STRIPPER *****</td>
</tr>
<tr>
<td>FPCODE</td>
<td>FORM PROCESSOR RETURN CODES</td>
</tr>
<tr>
<td>FPD</td>
<td>FORM PROCESSOR DATA</td>
</tr>
<tr>
<td>FPDINI</td>
<td>FPD INITIALIZATION</td>
</tr>
<tr>
<td>FPPARM</td>
<td>FORM PROCESSOR PARAMETERS</td>
</tr>
<tr>
<td>HRWFRM</td>
<td>HRW FORM DEFINITION</td>
</tr>
<tr>
<td>MATH</td>
<td>***** PURPOSE NOT FOUND BY STRIPPER *****</td>
</tr>
<tr>
<td>NTM</td>
<td>NTM INTERFACE INCLUDE FILE</td>
</tr>
<tr>
<td>RW</td>
<td>REPORT WRITER DEFINITIONS</td>
</tr>
<tr>
<td>SRVRET</td>
<td>AS THE RETURN GIVEN A TABLE-FULL ERROR</td>
</tr>
<tr>
<td>STDIO</td>
<td>***** PURPOSE NOT FOUND BY STRIPPER *****</td>
</tr>
<tr>
<td>STDTYP</td>
<td>STANDARD TYPE DEFINITIONS</td>
</tr>
</tbody>
</table>
3.10.5 Where Include File Used List

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

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>CHART</td>
<td>ARRANGE</td>
<td>ARRANGE CHART AND ASSIGNS PAGE NUMBERS</td>
</tr>
<tr>
<td></td>
<td>BLDMOD</td>
<td>BUILD MODULE</td>
</tr>
<tr>
<td></td>
<td>BLDNODE</td>
<td>BUILD NODE</td>
</tr>
<tr>
<td></td>
<td>CLOSEGAP</td>
<td>CLOSE GAP IN TREE</td>
</tr>
<tr>
<td></td>
<td>COPYNODE</td>
<td>COPY A NODE IN TREE</td>
</tr>
<tr>
<td></td>
<td>DELNODE</td>
<td>DELETE A SPECIFIED NODE IN TREE</td>
</tr>
<tr>
<td></td>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td></td>
<td>DRAWEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td></td>
<td>GETFIT</td>
<td>GET SUBTREE THAT FITS ON PAGE</td>
</tr>
<tr>
<td></td>
<td>GETLOWLEF</td>
<td>GET LOWER LEFT CHILD NODE</td>
</tr>
<tr>
<td></td>
<td>GETLOWRIT</td>
<td>GET LOWER RIGHT CHILD NODE</td>
</tr>
<tr>
<td></td>
<td>GETPAR</td>
<td>GET PARENT NODE</td>
</tr>
<tr>
<td></td>
<td>GETSIZE</td>
<td>GET SUBTREE SIZE</td>
</tr>
<tr>
<td></td>
<td>GETTOP</td>
<td>GET TOP OF TREE</td>
</tr>
<tr>
<td></td>
<td>GETUPLFT</td>
<td>GET UPPER LEFTMOST NODE</td>
</tr>
<tr>
<td></td>
<td>HBALANC</td>
<td>HORIZONTAL TREE BALANCE</td>
</tr>
<tr>
<td></td>
<td>HRW/Main</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td></td>
<td>MODPAGE</td>
<td>MODIFY PAGES</td>
</tr>
<tr>
<td></td>
<td>MOVCILD</td>
<td>MOVE CHILDREN</td>
</tr>
<tr>
<td></td>
<td>MOVECLD</td>
<td>MOVE CHILD'S POSITION</td>
</tr>
<tr>
<td></td>
<td>NEXTLEV</td>
<td>ADVANCE POINTERS TO NEXT LEVEL OF SUBTREE</td>
</tr>
<tr>
<td></td>
<td>PAGNODE</td>
<td>PAGE NODES</td>
</tr>
<tr>
<td></td>
<td>PAGTREE</td>
<td>PAGE TREE</td>
</tr>
<tr>
<td></td>
<td>PRNT</td>
<td>PRINT MODULE NAMES HIERARCHICALLY</td>
</tr>
<tr>
<td></td>
<td>PRNTREE</td>
<td>PRINT TREE</td>
</tr>
<tr>
<td></td>
<td>PUTLIN</td>
<td>PRINT LEVEL OF TREE</td>
</tr>
<tr>
<td></td>
<td>READTREE</td>
<td>READ DUMPTREE FILE</td>
</tr>
<tr>
<td></td>
<td>REPOS</td>
<td>REPOSITION MODULE EXPANSIONS</td>
</tr>
<tr>
<td></td>
<td>SORT</td>
<td>SORT MODULE NAMES</td>
</tr>
<tr>
<td></td>
<td>SPLICE</td>
<td>SPLICE TREE INTO ANOTHER TREE</td>
</tr>
<tr>
<td></td>
<td>SPLITNODE</td>
<td>SPLIT A NODE FOR PAGE BREAKS</td>
</tr>
<tr>
<td></td>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
</tbody>
</table>

CTLCHR

<table>
<thead>
<tr>
<th>ASSIGN</th>
<th>ASSIGN FILE SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLSFIL</td>
<td>CLOSE FILES</td>
</tr>
</tbody>
</table>
### REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
</tr>
<tr>
<td></td>
<td>ENDCGEN</td>
<td>END GENERATE</td>
</tr>
<tr>
<td></td>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
</tr>
<tr>
<td></td>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
</tr>
<tr>
<td></td>
<td>INDENT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
</tr>
<tr>
<td></td>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
</tr>
<tr>
<td></td>
<td>INSWS</td>
<td>INSERT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td></td>
<td>NDMGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td></td>
<td>NDMLLAB</td>
<td>GENERATE LABELS</td>
</tr>
<tr>
<td></td>
<td>NDMLLNK</td>
<td>LINKAGE SECTION</td>
</tr>
<tr>
<td></td>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
</tr>
<tr>
<td></td>
<td>OPNFIL</td>
<td>GENERATE OPEN FILE SECTION</td>
</tr>
<tr>
<td></td>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
</tr>
<tr>
<td></td>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
</tr>
<tr>
<td></td>
<td>SELECT</td>
<td>GENERATE SELECT CODE</td>
</tr>
<tr>
<td></td>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
</tr>
<tr>
<td></td>
<td>SELLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
</tr>
<tr>
<td></td>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
</tr>
<tr>
<td></td>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td></td>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
</tr>
<tr>
<td></td>
<td>USING</td>
<td>GENERATE USING SECTION</td>
</tr>
</tbody>
</table>

### CTYPES

- **MAKACT** MAKE ACTION LIST ELEMENT
- **YYLEX** LEXICAL ANALYZER FOR FLAN
- **YYPARSE** FLAN PARSER

### ERRPRO

- **CDMESQY** PROGRAM NAME CDMESQY

### FFFV2
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/DB</td>
<td>DEFAULT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/FO</td>
<td>INSERT FORMAT CODES</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/TB</td>
<td>TEXT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE DEFAULT BUFFER</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE FIELD</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE TEXT BUFFER</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE TEXT</td>
<td></td>
</tr>
</tbody>
</table>

FLAN.Y"

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

FPCODE

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTRSV</td>
<td>ACTION RESOLVE</td>
<td></td>
</tr>
<tr>
<td>ADDCHK</td>
<td>ADD POSITION TO CHECK LIST</td>
<td></td>
</tr>
<tr>
<td>ASSIGN</td>
<td>ASSIGN FILE SECTION</td>
<td></td>
</tr>
<tr>
<td>CALCSTAT</td>
<td>CALCULATE STATISTIC</td>
<td></td>
</tr>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
<td></td>
</tr>
<tr>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
<td></td>
</tr>
<tr>
<td>CHKFRM</td>
<td>CHECK FORM</td>
<td></td>
</tr>
<tr>
<td>CLSFIL</td>
<td>CLOSE FILES</td>
<td></td>
</tr>
<tr>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
<td></td>
</tr>
<tr>
<td>CTLSERV</td>
<td>CONTROL RESOLVE</td>
<td></td>
</tr>
<tr>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
<td></td>
</tr>
<tr>
<td>ENDGEN</td>
<td>END GERNERATE</td>
<td></td>
</tr>
<tr>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
<td></td>
</tr>
<tr>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
<td></td>
</tr>
<tr>
<td>FLDRSV</td>
<td>FIELD RESOLVE</td>
<td></td>
</tr>
<tr>
<td>FLDTYP</td>
<td>FIELD TYPE</td>
<td></td>
</tr>
<tr>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
<td></td>
</tr>
<tr>
<td>FNDFRM</td>
<td>FIND FORM</td>
<td></td>
</tr>
</tbody>
</table>
# REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETCOL</td>
<td>GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING</td>
<td></td>
</tr>
<tr>
<td>GETPTH</td>
<td>GET PATH</td>
<td></td>
</tr>
<tr>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
<td></td>
</tr>
<tr>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
<td></td>
</tr>
<tr>
<td>HRW/MMAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
<td></td>
</tr>
<tr>
<td>INDENT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
<td></td>
</tr>
<tr>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>INSRSV</td>
<td>INSERT RESOLVE</td>
<td></td>
</tr>
<tr>
<td>INSWS</td>
<td>INSERT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>MAKINS</td>
<td>MAKE INSERT</td>
<td></td>
</tr>
<tr>
<td>MAKINT</td>
<td>MAKE EXPRESSION INTO AN INTEGER</td>
<td></td>
</tr>
<tr>
<td>MAKPS</td>
<td>MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE</td>
<td></td>
</tr>
<tr>
<td>MAKSTR</td>
<td>MAKE EXPRESSION INTO A STRING</td>
<td></td>
</tr>
<tr>
<td>MAKWH</td>
<td>MAKE WHERE</td>
<td></td>
</tr>
<tr>
<td>MAKWHES</td>
<td>MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES</td>
<td></td>
</tr>
<tr>
<td>MAKWHES/C</td>
<td>COBOL WHERE ES</td>
<td></td>
</tr>
<tr>
<td>MAKWHES/C</td>
<td>C WHERE ES</td>
<td></td>
</tr>
<tr>
<td>MAKWHES/N</td>
<td>NUMBER PICTURE CLAUSE</td>
<td></td>
</tr>
<tr>
<td>MKPOS</td>
<td>MAKE POSITION NODE</td>
<td></td>
</tr>
<tr>
<td>MLPFRM</td>
<td>MAKE A LIST OF PRESENTED FORMS</td>
<td></td>
</tr>
<tr>
<td>MYALLOC</td>
<td>MY MALLOC</td>
<td></td>
</tr>
<tr>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
<td></td>
</tr>
<tr>
<td>NDMLLAB</td>
<td>GENERATE LABELS</td>
<td></td>
</tr>
<tr>
<td>NDMLLNK</td>
<td>LINKAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
<td></td>
</tr>
<tr>
<td>OPNFLI</td>
<td>GENERATE OPEN FILE SECTION</td>
<td></td>
</tr>
<tr>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/CO</td>
<td>COBOL SUBSTITUTE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/CS</td>
<td>C SUBSTITUTE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/IN</td>
<td>INDENT</td>
<td></td>
</tr>
<tr>
<td>RSETSTAT</td>
<td>RESFT STATISTIC</td>
<td></td>
</tr>
<tr>
<td>RWEFPD</td>
<td>REPORT WRITER EXPAND ARRAYS</td>
<td></td>
</tr>
<tr>
<td>RWOPN</td>
<td>REPORT WRITER OPEN FORMS</td>
<td></td>
</tr>
<tr>
<td>RWS/PFIXF</td>
<td>FIX UP A FORM</td>
<td></td>
</tr>
<tr>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td>GENERATE SELECT CODE</td>
<td></td>
</tr>
<tr>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
<td></td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>FILE</td>
<td>Module</td>
<td>Purpose</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>SELLN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
<td></td>
</tr>
<tr>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
<td></td>
</tr>
<tr>
<td>SELRSV</td>
<td>SELECT RESOLVE</td>
<td></td>
</tr>
<tr>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>STATRSV</td>
<td>STATISTIC RESOLVE</td>
<td></td>
</tr>
<tr>
<td>STDCode</td>
<td>STANDARD COBOL CODE</td>
<td></td>
</tr>
<tr>
<td>TRGRSV</td>
<td>TRIGGER RESOLVE</td>
<td></td>
</tr>
<tr>
<td>UQPTH</td>
<td>UNIVERSAL QUALIFIER PATH</td>
<td></td>
</tr>
<tr>
<td>USING</td>
<td>GENERATE USING SECTION</td>
<td></td>
</tr>
<tr>
<td>WINRSV</td>
<td>WINDOW RESOLVE</td>
<td></td>
</tr>
<tr>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
<td></td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/DB</td>
<td>DEFAULT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/FO</td>
<td>INSERT FORMAT CODES</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/TB</td>
<td>TEXT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE DEFAULT BUFFER</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE FIELD</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE TEXT BUFFER</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE TEXT</td>
<td></td>
</tr>
</tbody>
</table>

FPD

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTRSV</td>
<td>ACTION RESOLVE</td>
</tr>
<tr>
<td>ADDCHK</td>
<td>ADD POSITION TO CHECK LIST</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>ASSIGN FILE SECTION</td>
</tr>
<tr>
<td>BLDSUB</td>
<td>BUILD SUBROUTINES</td>
</tr>
<tr>
<td>BSCODE</td>
<td>BUILD SUBROUTINE CODE</td>
</tr>
<tr>
<td>CALCFAT</td>
<td>CALCULATE STATISTIC</td>
</tr>
<tr>
<td>CCONV</td>
<td>C CONVERSIONS</td>
</tr>
<tr>
<td>CES</td>
<td>C ES</td>
</tr>
<tr>
<td>CESPS</td>
<td>C ES TO PS</td>
</tr>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td>CHKFMR</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td>CHKGRP</td>
<td>CHECK FOR GROUP SEPERATORS OR END OF FILE</td>
</tr>
<tr>
<td>CHXSIZE</td>
<td>CHECK SIZE OF ITEMS DOING CONVERSIONS ON</td>
</tr>
<tr>
<td>CLRNDP</td>
<td>CLEAR NODUPLICATE FIELDS</td>
</tr>
<tr>
<td>CLSFIL</td>
<td>CLOSE FILES</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COBCCONV</td>
<td>COBOL CONVERSIONS</td>
</tr>
<tr>
<td></td>
<td>COBES</td>
<td>COBOL ES RECORD</td>
</tr>
<tr>
<td></td>
<td>COBESPS</td>
<td>COBOL ES TO PS</td>
</tr>
<tr>
<td></td>
<td>COBPE</td>
<td>COBOL PE</td>
</tr>
<tr>
<td></td>
<td>CPE</td>
<td>C PE</td>
</tr>
<tr>
<td></td>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td></td>
<td>CTLRSV</td>
<td>CONTROL RESOLVE</td>
</tr>
<tr>
<td></td>
<td>DASH</td>
<td>WRITE DASH '-'</td>
</tr>
<tr>
<td></td>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
</tr>
<tr>
<td></td>
<td>DBFREAD</td>
<td>GENERATE DATA BASE FREAD</td>
</tr>
<tr>
<td></td>
<td>DCLINDEX</td>
<td>DECLARE INDEX VARIABLES</td>
</tr>
<tr>
<td></td>
<td>ENDFGEN</td>
<td>END GENERATE</td>
</tr>
<tr>
<td></td>
<td>ESPSMAP</td>
<td>THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING</td>
</tr>
<tr>
<td></td>
<td>ESPSMAP/I</td>
<td>INDENT</td>
</tr>
<tr>
<td></td>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
</tr>
<tr>
<td></td>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
</tr>
<tr>
<td></td>
<td>FLANCIC</td>
<td>FLAN CALLABLE INTERFACE</td>
</tr>
<tr>
<td></td>
<td>FLDRESV</td>
<td>FIELD RESOLVE</td>
</tr>
<tr>
<td></td>
<td>FLDTYPE</td>
<td>FIELD TYPE</td>
</tr>
<tr>
<td></td>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>FNDFRM</td>
<td>FIND FORM</td>
</tr>
<tr>
<td></td>
<td>FRMPDAT</td>
<td>FORM PDATA</td>
</tr>
<tr>
<td></td>
<td>GEN</td>
<td>GENERATE A LINE OF CODE</td>
</tr>
<tr>
<td></td>
<td>GENTAD</td>
<td>GENERATE PROCEDURE &quot;ADDACT&quot; ADD AN ACTION</td>
</tr>
<tr>
<td></td>
<td>GENAAL</td>
<td>GENERATE PROCEDURE &quot;ADDAL&quot; ADD ACTION LIST</td>
</tr>
<tr>
<td></td>
<td>GENACT</td>
<td>GENERATE ACTIONS</td>
</tr>
<tr>
<td></td>
<td>GENAE</td>
<td>GENERATE ACTION EXIT</td>
</tr>
<tr>
<td></td>
<td>GENAH</td>
<td>GENERATE ACTION HELP</td>
</tr>
<tr>
<td></td>
<td>GENAI</td>
<td>GENERATE ACTION INSERT</td>
</tr>
<tr>
<td></td>
<td>GENAL</td>
<td>GENERATE ACTION LIST</td>
</tr>
<tr>
<td></td>
<td>GENAP</td>
<td>GENERATE ACTION PAGE</td>
</tr>
<tr>
<td></td>
<td>GENAQ</td>
<td>GENERATE ACTION QUERY (SELECT)</td>
</tr>
<tr>
<td></td>
<td>GENAR</td>
<td>GENERATE ACTION PRESENT</td>
</tr>
<tr>
<td></td>
<td>GENAS</td>
<td>GENERATE ACTION SET</td>
</tr>
<tr>
<td></td>
<td>GENAT</td>
<td>GENERATE ACTION SIGNAL</td>
</tr>
<tr>
<td></td>
<td>GENBEG</td>
<td>GENERATE BEGINNING OF APPLICATION OR REPORT</td>
</tr>
<tr>
<td></td>
<td>GENCHG</td>
<td>GENERATE CHANGE DECLARATIONS</td>
</tr>
<tr>
<td>Include File</td>
<td>Module Name</td>
<td>Module Purpose</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>GENLB</td>
<td>GENERATE DATA BASE RECORDS AND FILE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENDOA</td>
<td>GENERATE PROCEDURE &quot;DOACT&quot; DO ACTION</td>
<td></td>
</tr>
<tr>
<td>GENDS</td>
<td>GENERATE DATA DATA STRUCTURES</td>
<td></td>
</tr>
<tr>
<td>GENFP</td>
<td>GENERATE FORM PATH</td>
<td></td>
</tr>
<tr>
<td>GENFS</td>
<td>GENERATE FORM DATA STRUCTURES</td>
<td></td>
</tr>
<tr>
<td>GENFSD</td>
<td>GENERATE FORM STRUCTURE DATA INITIALIZATION</td>
<td></td>
</tr>
<tr>
<td>GENINS</td>
<td>GENERATE INSERT DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENMAIN</td>
<td>GENERATE MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td>GENNDP</td>
<td>GENERATE NODUPLICATE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENPAG</td>
<td>GENERATE NEWPAG PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>GETCOL</td>
<td>GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING</td>
<td></td>
</tr>
<tr>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM THE USER</td>
<td></td>
</tr>
<tr>
<td>GETPTH</td>
<td>GET PATH</td>
<td></td>
</tr>
<tr>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
<td></td>
</tr>
<tr>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
<td></td>
</tr>
<tr>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
<td></td>
</tr>
<tr>
<td>HASDATA</td>
<td>DETERMINE IF THERE ARE ANY SELECT STATEMENTS</td>
<td></td>
</tr>
<tr>
<td>HASITEM</td>
<td>THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN</td>
<td></td>
</tr>
<tr>
<td>HASLOWER</td>
<td>HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?</td>
<td></td>
</tr>
<tr>
<td>INDENT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
<td></td>
</tr>
<tr>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>INSRSV</td>
<td>INSERT RESOLVE</td>
<td></td>
</tr>
<tr>
<td>INSWS</td>
<td>INSERT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>ISOPNE</td>
<td>DETERMINE IF THIS FIELD IS OPEN ENDED</td>
<td></td>
</tr>
<tr>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
<td></td>
</tr>
<tr>
<td>MAKES</td>
<td>MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE</td>
<td></td>
</tr>
<tr>
<td>MAKES/CNU</td>
<td>C NUMBERS</td>
<td></td>
</tr>
<tr>
<td>MAKES/IND</td>
<td>INDENT</td>
<td></td>
</tr>
<tr>
<td>MAKES/NUM</td>
<td>NUMBER PICTURE CLAUSE</td>
<td></td>
</tr>
<tr>
<td>MAKINS</td>
<td>MAKE INSERT</td>
<td></td>
</tr>
<tr>
<td>MAKINT</td>
<td>MAKE EXPRESSION INTO AN INTEGER</td>
<td></td>
</tr>
<tr>
<td>MAKPS</td>
<td>MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE</td>
<td></td>
</tr>
</tbody>
</table>
### REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
</tr>
<tr>
<td></td>
<td>MAKSTR</td>
<td>MAKE EXPRESSION INTO A STRING</td>
</tr>
<tr>
<td></td>
<td>MAKWH</td>
<td>MAKE WHERE</td>
</tr>
<tr>
<td></td>
<td>MAKWHES</td>
<td>MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES</td>
</tr>
<tr>
<td></td>
<td>MAKWHES/C</td>
<td>COBOL WHERE ES</td>
</tr>
<tr>
<td></td>
<td>MAKWHES/C</td>
<td>C WHERE ES</td>
</tr>
<tr>
<td></td>
<td>MAKWHES/N</td>
<td>NUMBER PICTURE CLAUSE</td>
</tr>
<tr>
<td></td>
<td>MAPDB</td>
<td>MAP DATABASE</td>
</tr>
<tr>
<td></td>
<td>MKINC</td>
<td>MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)</td>
</tr>
<tr>
<td></td>
<td>MKPOS</td>
<td>MAKE POSITION NODE</td>
</tr>
<tr>
<td></td>
<td>MLPFRM</td>
<td>MAKE A LIST OF PRESENTED FORMS</td>
</tr>
<tr>
<td></td>
<td>MYALLOC</td>
<td>MY MALLOC</td>
</tr>
<tr>
<td></td>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td></td>
<td>NDMLLAB</td>
<td>GENERATE LABELS</td>
</tr>
<tr>
<td></td>
<td>NDMLLNK</td>
<td>LINKAGE SECTION</td>
</tr>
<tr>
<td></td>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
</tr>
<tr>
<td></td>
<td>OPNFIL</td>
<td>GENERATE OPEN FILE SECTION</td>
</tr>
<tr>
<td></td>
<td>PEMAP</td>
<td>THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING</td>
</tr>
<tr>
<td></td>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
</tr>
<tr>
<td></td>
<td>PSSTRC/CO</td>
<td>COBOL SUBSTITUTE</td>
</tr>
<tr>
<td></td>
<td>PSSTRC/CS</td>
<td>C SUBSTITUTE</td>
</tr>
<tr>
<td></td>
<td>PSSTRC/IN</td>
<td>INDENT</td>
</tr>
<tr>
<td></td>
<td>READDB</td>
<td>READ DATA BASE</td>
</tr>
<tr>
<td></td>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
</tr>
<tr>
<td></td>
<td>RSETSTAT</td>
<td>RESET STATISTIC</td>
</tr>
<tr>
<td></td>
<td>RWEXPD</td>
<td>REPORT WRITER EXPAND ARRAYS</td>
</tr>
<tr>
<td></td>
<td>RWOPN</td>
<td>REPORT WRITER OPEN FORMS</td>
</tr>
<tr>
<td></td>
<td>RWSP/FIXF</td>
<td>FIX UP A FORM</td>
</tr>
<tr>
<td></td>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
</tr>
<tr>
<td></td>
<td>SELECT</td>
<td>GENERATE SELECT CODE</td>
</tr>
<tr>
<td></td>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
</tr>
<tr>
<td></td>
<td>SELLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
</tr>
<tr>
<td></td>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
</tr>
<tr>
<td></td>
<td>SELOPN</td>
<td>SELECT OPEN</td>
</tr>
<tr>
<td></td>
<td>SELRSV</td>
<td>SELECT RESOLVE</td>
</tr>
<tr>
<td></td>
<td>SELWHR</td>
<td>SELECT WHERE</td>
</tr>
</tbody>
</table>

3-33
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include</th>
<th>Module</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>File</td>
<td>Name</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>SELWS</td>
<td></td>
<td>SELECT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td>SETNDP</td>
<td></td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
</tr>
<tr>
<td>STATRSV</td>
<td></td>
<td>STATISTIC RESOLVE</td>
</tr>
<tr>
<td>STDCODE</td>
<td></td>
<td>STANDARD COBOL CODE</td>
</tr>
<tr>
<td>TRGRSV</td>
<td></td>
<td>TRIGGER RESOLVE</td>
</tr>
<tr>
<td>UQFOR</td>
<td></td>
<td>UNIVERSAL QUALIFIER FOR LOOP</td>
</tr>
<tr>
<td>UQPTH</td>
<td></td>
<td>UNIVERSAL QUALIFIER PATH</td>
</tr>
<tr>
<td>USING</td>
<td></td>
<td>GENERATE USING SECTION</td>
</tr>
<tr>
<td>VISITA</td>
<td></td>
<td>VISIT ARRAYS ON THIS FORM</td>
</tr>
<tr>
<td>WINRSV</td>
<td></td>
<td>WINDOW RESOLVE</td>
</tr>
<tr>
<td>WRTEXP</td>
<td></td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td>WRTFRM</td>
<td></td>
<td>WRITE FORM</td>
</tr>
<tr>
<td>WRTFRM/DB</td>
<td></td>
<td>DEFAULT BUFFER CLOSE</td>
</tr>
<tr>
<td>WRTFRM/FO</td>
<td></td>
<td>INSERT FORMAT CODES</td>
</tr>
<tr>
<td>WRTFRM/TB</td>
<td></td>
<td>TEXT BUFFER CLOSE</td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td></td>
<td>WRITE DEFAULT BUFFER</td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td></td>
<td>WRITE FIELD</td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td></td>
<td>WRITE TEXT BUFFER</td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td></td>
<td>WRITE TEXT</td>
</tr>
<tr>
<td>YYLEX</td>
<td></td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>YYPARSE</td>
<td></td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>

FPDINI

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDSUB</td>
<td>BUILD SUBROUTINES</td>
<td></td>
</tr>
<tr>
<td>BSCODE</td>
<td>BUILD SUBROUTINE CODE</td>
<td></td>
</tr>
<tr>
<td>CHKGRP</td>
<td>CHECK FOR GROUP SEPERATORS OR END OF FILE</td>
<td></td>
</tr>
<tr>
<td>CLRNDF</td>
<td>CLEAR NODUPLICATE FIELDS</td>
<td></td>
</tr>
<tr>
<td>DBFREAD</td>
<td>GENERATE DATA BASE FREAD</td>
<td></td>
</tr>
<tr>
<td>GEN</td>
<td>GENERATE A LINE OF CODE</td>
<td></td>
</tr>
<tr>
<td>GENPAG</td>
<td>GENERATE NEWPAG PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>THE USER</td>
<td></td>
</tr>
<tr>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
<td></td>
</tr>
<tr>
<td>HASDATA</td>
<td>DETERMINE IF THERE ARE ANY SELECT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STATEMENTS</td>
<td></td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>HASITEM</td>
<td>THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN</td>
<td></td>
</tr>
<tr>
<td>HASLOWER</td>
<td>HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?</td>
<td></td>
</tr>
<tr>
<td>ISOPNE</td>
<td>DETERMINE IF THIS FIELD IS OPEN ENDED</td>
<td></td>
</tr>
<tr>
<td>MAQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
<td></td>
</tr>
<tr>
<td>MAPDB</td>
<td>MAP DATABASE</td>
<td></td>
</tr>
<tr>
<td>READDB</td>
<td>READ DATA BASE</td>
<td></td>
</tr>
<tr>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
<td></td>
</tr>
<tr>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
<td></td>
</tr>
<tr>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
<td></td>
</tr>
</tbody>
</table>

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 FRONT END FOR FORMS
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
HASITEM THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Incl. File</th>
<th>Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HASLOWER</td>
<td>HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INSERT</td>
<td>INSERT A LINE OF GENERATED CODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INSWS</td>
<td>INSERT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISOPNE</td>
<td>DETERMINE IF THIS FIELD IS OPEN ENDED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKINS</td>
<td>MAKE INSERT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKPS</td>
<td>MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKWH</td>
<td>MAKE WHERE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKWHES</td>
<td>MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKWHES/C</td>
<td>COBOL WHERE ES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKWHES/C</td>
<td>C WHERE ES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKWHES/N</td>
<td>NUMBER PICTURE CLAUSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAPDB</td>
<td>MAP DATABASE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NDMIGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NDMILAB</td>
<td>GENERATE LABELS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NDMILNK</td>
<td>LINKAGE SECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPNFIL</td>
<td>GENERATE OPEN FILE SECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSSTRC/CO</td>
<td>COBOL SUBSTITUTE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSSTRC/CS</td>
<td>C SUBSTITUTE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSSTRC/IN</td>
<td>INDENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>READDB</td>
<td>READ DATA BASE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELECT</td>
<td>GENERATE SELECT CODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>USING</td>
<td>GENERATE USING SECTION</td>
</tr>
<tr>
<td></td>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
</tr>
<tr>
<td></td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td></td>
<td>YYParse</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>

HRWFRM
HRW/Main MAIN MODULE FOR HIERARCHICAL REPORT WRITER

MATH

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

NTM

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>ASSIGN FILE SECTION</td>
</tr>
<tr>
<td>BLDSUB</td>
<td>BUILD SUBROUTINES</td>
</tr>
<tr>
<td>BSCODE</td>
<td>BUILD SUBROUTINE CODE</td>
</tr>
<tr>
<td>CHKGRT</td>
<td>CHECK FOR GROUP SEPARATORS OR END OF FILE</td>
</tr>
<tr>
<td>CLRNDDP</td>
<td>CLEAR NODUPLICATE FIELDS</td>
</tr>
<tr>
<td>CLSFIL</td>
<td>CLOSE FILES</td>
</tr>
<tr>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
</tr>
<tr>
<td>DBFREAD</td>
<td>GENERATE DATA BASE FREAD</td>
</tr>
<tr>
<td>ENDGEN</td>
<td>END GENERATE</td>
</tr>
<tr>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
</tr>
<tr>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
</tr>
<tr>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td>GEN</td>
<td>GENERATE A LINE OF CODE</td>
</tr>
<tr>
<td>GENPAG</td>
<td>GENERATE NEWPAG PROCEDURE</td>
</tr>
<tr>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM THE USER</td>
</tr>
<tr>
<td>GRP/Main</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>HASDATA</td>
<td>DETERMINE IF THERE ARE ANY SELECT STATEMENTS</td>
<td></td>
</tr>
<tr>
<td>HASITEM</td>
<td>THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN</td>
<td></td>
</tr>
<tr>
<td>HASLOWER</td>
<td>HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?</td>
<td></td>
</tr>
<tr>
<td>INDENT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
<td></td>
</tr>
<tr>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>INSWS</td>
<td>INSERT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>ISOPNE</td>
<td>DETERMINE IF THIS FIELD IS OPEN ENDED</td>
<td></td>
</tr>
<tr>
<td>MAQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
<td></td>
</tr>
<tr>
<td>MAPDB</td>
<td>MAP DATABASE</td>
<td></td>
</tr>
<tr>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
<td></td>
</tr>
<tr>
<td>NDMLLAB</td>
<td>GENERATE LABELS</td>
<td></td>
</tr>
<tr>
<td>NDMLLNK</td>
<td>LINKAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
<td></td>
</tr>
<tr>
<td>OPNFLF</td>
<td>GENERATE OPEN FILE SECTION</td>
<td></td>
</tr>
<tr>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
<td></td>
</tr>
<tr>
<td>READDDB</td>
<td>READ DATA BASE</td>
<td></td>
</tr>
<tr>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
<td></td>
</tr>
<tr>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td>GENERATE SELECT CODE</td>
<td></td>
</tr>
<tr>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
<td></td>
</tr>
<tr>
<td>SELLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
<td></td>
</tr>
<tr>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
<td></td>
</tr>
<tr>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
<td></td>
</tr>
<tr>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
<td></td>
</tr>
<tr>
<td>USING</td>
<td>GENERATE USING SECTION</td>
<td></td>
</tr>
<tr>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
<td></td>
</tr>
</tbody>
</table>

RW

<table>
<thead>
<tr>
<th>Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTRSV</td>
<td>ACTION RESOLVE</td>
</tr>
<tr>
<td>ADDCHK</td>
<td>ADD POSITION TO CHECK LIST</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>ASSIGN FILE SECTION</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLDSUB</td>
<td>BUILD SUBROUTINES</td>
</tr>
<tr>
<td></td>
<td>BSCODE</td>
<td>BUILD SUBROUTINE CODE</td>
</tr>
<tr>
<td></td>
<td>CALCSTAT</td>
<td>CALCULATE STATISTIC</td>
</tr>
<tr>
<td></td>
<td>CCONV</td>
<td>C CONVERSIONS</td>
</tr>
<tr>
<td></td>
<td>CES</td>
<td>C ES</td>
</tr>
<tr>
<td></td>
<td>CESPS</td>
<td>C ES TO PS</td>
</tr>
<tr>
<td></td>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td></td>
<td>CHKFD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td></td>
<td>CHKGRP</td>
<td>CHECK FOR GROUP SEPERATORS OR END OF FILE</td>
</tr>
<tr>
<td></td>
<td>CHKSIZE</td>
<td>CHECK SIZE OF ITEMS DOING CONVERSIONS ON</td>
</tr>
<tr>
<td></td>
<td>CLRNDF</td>
<td>CLEAR NODUPLICATE FIELDS</td>
</tr>
<tr>
<td></td>
<td>CLSFIL</td>
<td>CLOSE FILES</td>
</tr>
<tr>
<td></td>
<td>COBCONV</td>
<td>COBOL CONVERSIONS</td>
</tr>
<tr>
<td></td>
<td>COBES</td>
<td>COBOL ES RECORD</td>
</tr>
<tr>
<td></td>
<td>COBESPS</td>
<td>COBOL ES TO PS</td>
</tr>
<tr>
<td></td>
<td>COBPE</td>
<td>COBOL PE</td>
</tr>
<tr>
<td></td>
<td>CPE</td>
<td>C PE</td>
</tr>
<tr>
<td></td>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td></td>
<td>CTLRSV</td>
<td>CONTROL RESOLVE</td>
</tr>
<tr>
<td></td>
<td>DASH</td>
<td>WRITE DASH '-'</td>
</tr>
<tr>
<td></td>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
</tr>
<tr>
<td></td>
<td>DBFREAD</td>
<td>GENERATE DATA BASE FREAD</td>
</tr>
<tr>
<td></td>
<td>DCLINDEX</td>
<td>DECLARE INDEX VARIABLES</td>
</tr>
<tr>
<td></td>
<td>ENDGEN</td>
<td>END GENERATE</td>
</tr>
<tr>
<td></td>
<td>ESPSMAP</td>
<td>THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING</td>
</tr>
<tr>
<td></td>
<td>ESPSMAP/I</td>
<td>INDENT</td>
</tr>
<tr>
<td></td>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
</tr>
<tr>
<td></td>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
</tr>
<tr>
<td></td>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
</tr>
<tr>
<td></td>
<td>FLDRESV</td>
<td>FIELD RESOLVE</td>
</tr>
<tr>
<td></td>
<td>FLDTYPE</td>
<td>FIELD TYPE</td>
</tr>
<tr>
<td></td>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>FNDATM</td>
<td>FIND FORM</td>
</tr>
<tr>
<td></td>
<td>FRMPDAT</td>
<td>FORM PDATA</td>
</tr>
<tr>
<td></td>
<td>GEN</td>
<td>GENERATE A LINE OF CODE</td>
</tr>
<tr>
<td></td>
<td>GENAA</td>
<td>GENERATE PROCEDURE &quot;ADDACT&quot; ADD AN ACTION</td>
</tr>
<tr>
<td></td>
<td>GENAAL</td>
<td>GENERATE PROCEDURE &quot;ADDAL&quot; ADD ACTION LIST</td>
</tr>
<tr>
<td></td>
<td>GENACT</td>
<td>GENERATE ACTIONS</td>
</tr>
</tbody>
</table>
# REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENAE</td>
<td>GENERATE ACTION EXIT</td>
<td></td>
</tr>
<tr>
<td>GENAH</td>
<td>GENERATE ACTION HELP</td>
<td></td>
</tr>
<tr>
<td>GENAI</td>
<td>GENERATE ACTION INSERT</td>
<td></td>
</tr>
<tr>
<td>GENAL</td>
<td>GENERATE ACTION LIST</td>
<td></td>
</tr>
<tr>
<td>GENAP</td>
<td>GENERATE ACTION PAGE</td>
<td></td>
</tr>
<tr>
<td>GENAQ</td>
<td>GENERATE ACTION QUERY (SELECT)</td>
<td></td>
</tr>
<tr>
<td>GENAR</td>
<td>GENERATE ACTION PRESENT</td>
<td></td>
</tr>
<tr>
<td>GENAS</td>
<td>GENERATE ACTION SET</td>
<td></td>
</tr>
<tr>
<td>GENAT</td>
<td>GENERATE ACTION SIGNAL</td>
<td></td>
</tr>
<tr>
<td>GENBEG</td>
<td>GENERATE BEGINNING OF APPLICATION OR REPORT</td>
<td></td>
</tr>
<tr>
<td>GENCHG</td>
<td>GENERATE CHANGE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENDB</td>
<td>GENERATE DATA BASE RECORDS AND FILE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENDOA</td>
<td>GENERATE PROCEDURE &quot;DOACT&quot; DO ACTION</td>
<td></td>
</tr>
<tr>
<td>GENDS</td>
<td>GENERATE DATA DATA STRUCTURES</td>
<td></td>
</tr>
<tr>
<td>GENFP</td>
<td>GENERATE FORM PATH</td>
<td></td>
</tr>
<tr>
<td>GENFS</td>
<td>GENERATE FORM DATA STRUCTURES</td>
<td></td>
</tr>
<tr>
<td>GENFSD</td>
<td>GENERATE FORM STRUCTURE DATA INITIALIZATION</td>
<td></td>
</tr>
<tr>
<td>GENINS</td>
<td>GENERATE INSERT DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENMAIN</td>
<td>GENERATE MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td>GENNDP</td>
<td>GENERATE NODUPLICATE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENPAG</td>
<td>GENERATE NEWPAG PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>GETCOL</td>
<td>GET THE COLUMN NAME OF A TABLE.COLUM or COLUMN STRING</td>
<td></td>
</tr>
<tr>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM THE USER</td>
<td></td>
</tr>
<tr>
<td>GETPATH</td>
<td>GET PATH</td>
<td></td>
</tr>
<tr>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
<td></td>
</tr>
<tr>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
<td></td>
</tr>
<tr>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
<td></td>
</tr>
<tr>
<td>HASDATA</td>
<td>DETERMINE IF THERE ARE ANY SELECT STATEMENTS</td>
<td></td>
</tr>
<tr>
<td>HASITEM</td>
<td>THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN</td>
<td></td>
</tr>
<tr>
<td>HASLOWER</td>
<td>HAS A LOWER FORM WHICH READS THE SAME DATA RECORD</td>
<td></td>
</tr>
<tr>
<td>INDENT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
<td></td>
</tr>
<tr>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
<td></td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

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

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSETSTAT</td>
<td>RESET STATISTIC</td>
<td></td>
</tr>
<tr>
<td>RWEXPD</td>
<td>REPORT WRITER EXPAND ARRAYS</td>
<td></td>
</tr>
<tr>
<td>RWOPN</td>
<td>REPORT WRITER OPEN FORMS</td>
<td></td>
</tr>
<tr>
<td>RWSP/FIXF</td>
<td>FIX UP A FORM</td>
<td></td>
</tr>
<tr>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td>GENERATE SELECT CODE</td>
<td></td>
</tr>
<tr>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
<td></td>
</tr>
<tr>
<td>SELLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
<td></td>
</tr>
<tr>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
<td></td>
</tr>
<tr>
<td>SELOPN</td>
<td>SELECT OPEN</td>
<td></td>
</tr>
<tr>
<td>SELRESV</td>
<td>SELECT RESOLVE</td>
<td></td>
</tr>
<tr>
<td>SELWHERE</td>
<td>SELECT WHERE</td>
<td></td>
</tr>
<tr>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARE DUPLICATED</td>
<td></td>
</tr>
<tr>
<td>STATRSV</td>
<td>STATISTIC RESOLVE</td>
<td></td>
</tr>
<tr>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
<td></td>
</tr>
<tr>
<td>TRGRSV</td>
<td>TRIGGER RESOLVE</td>
<td></td>
</tr>
<tr>
<td>UQFOR</td>
<td>UNIVERSAL QUALIFIER FOR LOOP</td>
<td></td>
</tr>
<tr>
<td>UQPTH</td>
<td>UNIVERSAL QUALIFIER PATH</td>
<td></td>
</tr>
<tr>
<td>USING</td>
<td>GENERATE USING SECTION</td>
<td></td>
</tr>
<tr>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
<td></td>
</tr>
<tr>
<td>WINRSV</td>
<td>WINDOW RESOLVE</td>
<td></td>
</tr>
<tr>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
<td></td>
</tr>
<tr>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
</tbody>
</table>

SRVRET

CDMESQY PROGRAM NAME CDMESQY

STdio

<table>
<thead>
<tr>
<th>ADDCHK</th>
<th>ADD POSITION TO CHECK LIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARRANGE</td>
<td>ARRANGE CHART AND ASSIGNS PAGE NUMBERS</td>
</tr>
<tr>
<td>ASSGN</td>
<td>ASSIGN FILE SECTION</td>
</tr>
</tbody>
</table>
REPORT WRITER Where include file used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDSUB</td>
<td>BUILD SUBROUTINES</td>
<td></td>
</tr>
<tr>
<td>BSCODE</td>
<td>BUILD SUBROUTINE CODE</td>
<td></td>
</tr>
<tr>
<td>CCONV</td>
<td>C CONVERSIONS</td>
<td></td>
</tr>
<tr>
<td>CES</td>
<td>C ES</td>
<td></td>
</tr>
<tr>
<td>CESPS</td>
<td>C ES TO PS</td>
<td></td>
</tr>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
<td></td>
</tr>
<tr>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
<td></td>
</tr>
<tr>
<td>CHKFRTM</td>
<td>CHECK FORM</td>
<td></td>
</tr>
<tr>
<td>CHKGRP</td>
<td>CHECK FOR GROUP SEPERATORS OR END OF FILE</td>
<td></td>
</tr>
<tr>
<td>CHKSZIE</td>
<td>CHECK SIZE OF ITEMS DOING CONVERSIONS ON</td>
<td></td>
</tr>
<tr>
<td>CLRNDP</td>
<td>CLEAR NODUPLICATE FIELDS</td>
<td></td>
</tr>
<tr>
<td>CLSFIL</td>
<td>CLOSE FILES</td>
<td></td>
</tr>
<tr>
<td>COBCONV</td>
<td>COBOL CONVERSIONS</td>
<td></td>
</tr>
<tr>
<td>COBES</td>
<td>COBOL ES RECORD</td>
<td></td>
</tr>
<tr>
<td>COBESPS</td>
<td>COBOL ES TO PS</td>
<td></td>
</tr>
<tr>
<td>COBPE</td>
<td>COBOL PE</td>
<td></td>
</tr>
<tr>
<td>CPE</td>
<td>C PE</td>
<td></td>
</tr>
<tr>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
<td></td>
</tr>
<tr>
<td>DASH</td>
<td>WRITE DASH '-',</td>
<td></td>
</tr>
<tr>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
<td></td>
</tr>
<tr>
<td>DBFREAD</td>
<td>GENERATE DATA BASE FREAD</td>
<td></td>
</tr>
<tr>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
<td></td>
</tr>
<tr>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
<td></td>
</tr>
<tr>
<td>ENDEX</td>
<td>END GENERATE</td>
<td></td>
</tr>
<tr>
<td>ESPSMAPI</td>
<td>THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING</td>
<td></td>
</tr>
<tr>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
<td></td>
</tr>
<tr>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
<td></td>
</tr>
<tr>
<td>FLDTYP</td>
<td>FIELD TYPE</td>
<td></td>
</tr>
<tr>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
<td></td>
</tr>
<tr>
<td>GEN</td>
<td>GENERATE A LINE OF CODE</td>
<td></td>
</tr>
<tr>
<td>GENBEZG</td>
<td>GENERATE BEGINNING OF APPLICATION OR REPORT</td>
<td></td>
</tr>
<tr>
<td>GENCCHG</td>
<td>GENERATE CHANGE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENDB</td>
<td>GENERATE DATA BASE RECORDS AND FILE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENDS</td>
<td>GENERATE DATA DATA STRUCTURES</td>
<td></td>
</tr>
<tr>
<td>GENFP</td>
<td>GENERATE FORM PATH</td>
<td></td>
</tr>
</tbody>
</table>

3-43
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>File</th>
<th>Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Include</td>
<td>Module Name</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
<td>------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>GENFS</td>
<td>GENERATE FORM DATA STRUCTURES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GENFSD</td>
<td>GENERATE FORM STRUCTURE DATA INITIALIZATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GENINS</td>
<td>GENERATE INSERT DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GENMAIN</td>
<td>GENERATE MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GENNDP</td>
<td>GENERATE NODUPLICATE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GENPAG</td>
<td>GENERATE NEWPAG PROCEDURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM THE USER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GRP/MAP</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HASDATA</td>
<td>DETERMINE IF THERE ARE ANY SELECT STATEMENTS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HASITEM</td>
<td>THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HASLOWER</td>
<td>HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HRW/MAP</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INSERT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INSWs</td>
<td>INSERT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ISOPNE</td>
<td>DETERMINE IF THIS FIELD IS OPEN ENDED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKES/CNU</td>
<td>MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKES/IND</td>
<td>INDENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKES/NUM</td>
<td>NUMBER PICTURE CLAUSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKINS</td>
<td>MAKE INSERT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAINT</td>
<td>MAKE EXPRESSION INTO AN INTEGER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKPS</td>
<td>MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKSTR</td>
<td>MAKE EXPRESSION INTO A STRING</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKWH</td>
<td>MAKE WHERE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKWHES</td>
<td>MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKWHES/C</td>
<td>COBOL WHERE ES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKWHES/C</td>
<td>C WHERE ES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAKWHES/N</td>
<td>NUMBER PICTURE CLAUSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAPFDB</td>
<td>MAP DATABASE</td>
<td></td>
</tr>
</tbody>
</table>
# REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKINC</td>
<td>MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)</td>
<td></td>
</tr>
<tr>
<td>MKPOS</td>
<td>MAKE POSITION NODE</td>
<td></td>
</tr>
<tr>
<td>MYALLOC</td>
<td>MY MALLOC</td>
<td></td>
</tr>
<tr>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
<td></td>
</tr>
<tr>
<td>NDMLLAB</td>
<td>GENERATE LABELS</td>
<td></td>
</tr>
<tr>
<td>NDMLLINK</td>
<td>LINKAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
<td></td>
</tr>
<tr>
<td>OPENFIL</td>
<td>GENERATE OPEN FILE SECTION</td>
<td></td>
</tr>
<tr>
<td>PEMAP</td>
<td>THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING</td>
<td></td>
</tr>
<tr>
<td>PRNTREE</td>
<td>PRINT TREE</td>
<td></td>
</tr>
<tr>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/CO</td>
<td>COBOL SUBSTITUTE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/CS</td>
<td>C SUBSTITUTE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/IN</td>
<td>INDENT</td>
<td></td>
</tr>
<tr>
<td>PUTLIN</td>
<td>PRINT LEVEL OF TREE</td>
<td></td>
</tr>
<tr>
<td>READDB</td>
<td>READ DATA BASE</td>
<td></td>
</tr>
<tr>
<td>READTREE</td>
<td>READ DUMPTREE FILE</td>
<td></td>
</tr>
<tr>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
<td></td>
</tr>
<tr>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td>GENERATE SELECT CODE</td>
<td></td>
</tr>
<tr>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
<td></td>
</tr>
<tr>
<td>SELLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
<td></td>
</tr>
<tr>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
<td></td>
</tr>
<tr>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
<td></td>
</tr>
<tr>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
<td></td>
</tr>
<tr>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
<td></td>
</tr>
<tr>
<td>USING</td>
<td>GENERATE USING SECTION</td>
<td></td>
</tr>
<tr>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
<td></td>
</tr>
<tr>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
<td></td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/DB</td>
<td>DEFAULT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/FO</td>
<td>INSERT FORMAT CODES</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/TB</td>
<td>TEXT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE DEFAULT BUFFER</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE FIELD</td>
<td></td>
</tr>
</tbody>
</table>

3-45
### REPORT WRITER Where-include-file-used List

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

### STDTYP

<table>
<thead>
<tr>
<th>STDTYP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTRSV</td>
<td>ACTION RESOLVE</td>
</tr>
<tr>
<td>ADDCHK</td>
<td>ADD POSITION TO CHECK LIST</td>
</tr>
<tr>
<td>ARRANGE</td>
<td>ARRANGE CHART AND ASSIGNS PAGE NUMBERS</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>ASSIGN FILE SECTION</td>
</tr>
<tr>
<td>BLDMOD</td>
<td>BUILD MODULE</td>
</tr>
<tr>
<td>BLNODE</td>
<td>BUILD NODE</td>
</tr>
<tr>
<td>BLDSUB</td>
<td>BUILD SUBROUTINES</td>
</tr>
<tr>
<td>BSCODE</td>
<td>BUILD SUBROUTINE CODE</td>
</tr>
<tr>
<td>CALCSTAT</td>
<td>CALCULATE STATISTIC</td>
</tr>
<tr>
<td>CCONV</td>
<td>C CONVERSIONS</td>
</tr>
<tr>
<td>CES</td>
<td>C ES</td>
</tr>
<tr>
<td>CESPS</td>
<td>C ES TO PS</td>
</tr>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td>CHKGRP</td>
<td>CHECK FOR GROUP SEPERATORS OR END OF FILE</td>
</tr>
<tr>
<td>CHKSIZE</td>
<td>CHECK SIZE OF ITEMS DOING CONVERSIONS ON</td>
</tr>
<tr>
<td>CLOSEGAP</td>
<td>CLOSE GAP IN TREE</td>
</tr>
<tr>
<td>CLRNDP</td>
<td>CLEAR NODUPLICATE FIELDS</td>
</tr>
<tr>
<td>CLSFIL</td>
<td>CLOSE FILES</td>
</tr>
<tr>
<td>COBConv</td>
<td>COBOL CONVERSIONS</td>
</tr>
<tr>
<td>COBES</td>
<td>COBOL ES RECORD</td>
</tr>
<tr>
<td>COBESPS</td>
<td>COBOL ES TO PS</td>
</tr>
<tr>
<td>COBPE</td>
<td>COBOL PE</td>
</tr>
<tr>
<td>COPYNODE</td>
<td>COPY A NODE IN TREE</td>
</tr>
<tr>
<td>CPE</td>
<td>C PE</td>
</tr>
<tr>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td>CTLRSV</td>
<td>CONTROL RESOLVE</td>
</tr>
<tr>
<td>DASH</td>
<td>WRITE DASH ' - '</td>
</tr>
<tr>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
</tr>
<tr>
<td>DBFREAD</td>
<td>GENERATE DATA BASE FREAD</td>
</tr>
</tbody>
</table>

3-46
<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCLINDX</td>
<td>DECLARE INDEX VARIABLES</td>
<td></td>
</tr>
<tr>
<td>DELNODE</td>
<td>DELETE A SPECIFIED NODE IN TREE</td>
<td></td>
</tr>
<tr>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
<td></td>
</tr>
<tr>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
<td></td>
</tr>
<tr>
<td>ENGEN</td>
<td>END GENERATE</td>
<td></td>
</tr>
<tr>
<td>ERROR</td>
<td>ISSUE ERROR MESSAGE</td>
<td></td>
</tr>
<tr>
<td>ESPSMAP</td>
<td>THE EXTERNAL SCHEMA AND PRESENTATION SCHEMA MAPPING</td>
<td></td>
</tr>
<tr>
<td>ESPSMAP/I</td>
<td>INDENT</td>
<td></td>
</tr>
<tr>
<td>FATAL</td>
<td>ISSUE FATAL ERROR MESSAGE</td>
<td></td>
</tr>
<tr>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
<td></td>
</tr>
<tr>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
<td></td>
</tr>
<tr>
<td>FLDRESV</td>
<td>FIELD RESOLVE</td>
<td></td>
</tr>
<tr>
<td>FLDTYP</td>
<td>FIELD TYPE</td>
<td></td>
</tr>
<tr>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
<td></td>
</tr>
<tr>
<td>FNDfrm</td>
<td>FIND FORM</td>
<td></td>
</tr>
<tr>
<td>FRMPDAT</td>
<td>FORM PDATA</td>
<td></td>
</tr>
<tr>
<td>FRNND</td>
<td>FRONT END FOR FORMS</td>
<td></td>
</tr>
<tr>
<td>GEN</td>
<td>GENERATE A LINE OF CODE</td>
<td></td>
</tr>
<tr>
<td>GENAA</td>
<td>GENERATE PROCEDURE &quot;ADDACT&quot; ADD AN ACTION</td>
<td></td>
</tr>
<tr>
<td>GENAAL</td>
<td>GENERATE PROCEDURE &quot;ADDAL&quot; ADD ACTION LIST</td>
<td></td>
</tr>
<tr>
<td>GENACT</td>
<td>GENERATE ACTIONS</td>
<td></td>
</tr>
<tr>
<td>GENAE</td>
<td>GENERATE ACTION EXIT</td>
<td></td>
</tr>
<tr>
<td>GENAH</td>
<td>GENERATE ACTION HELP</td>
<td></td>
</tr>
<tr>
<td>GENAI</td>
<td>GENERATE ACTION INSERT</td>
<td></td>
</tr>
<tr>
<td>GENAL</td>
<td>GENERATE ACTION LIST</td>
<td></td>
</tr>
<tr>
<td>GENAP</td>
<td>GENERATE ACTION PAGE</td>
<td></td>
</tr>
<tr>
<td>GENAQ</td>
<td>GENERATE ACTION QUERY (SELECT)</td>
<td></td>
</tr>
<tr>
<td>GENAR</td>
<td>GENERATE ACTION PRESENT</td>
<td></td>
</tr>
<tr>
<td>GENAS</td>
<td>GENERATE ACTION SET</td>
<td></td>
</tr>
<tr>
<td>GENAT</td>
<td>GENERATE ACTION SIGNAL</td>
<td></td>
</tr>
<tr>
<td>GENBEG</td>
<td>GENERATE BEGINNING OF APPLICATION OR REPORT</td>
<td></td>
</tr>
<tr>
<td>GENCCHG</td>
<td>GENERATE CHANGE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENDB</td>
<td>GENERATE DATA BASE RECORDS AND FILE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENDOA</td>
<td>GENERATE PROCEDURE &quot;DOACT&quot; DO ACTION</td>
<td></td>
</tr>
<tr>
<td>GENDS</td>
<td>GENERATE DATA DATA STRUCTURES</td>
<td></td>
</tr>
<tr>
<td>GENFP</td>
<td>GENERATE FORM PATH</td>
<td></td>
</tr>
</tbody>
</table>
**REPORT WRITER Where-include-file-used List**

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENFS</td>
<td>GENERATE FORM DATA STRUCTURES</td>
<td></td>
</tr>
<tr>
<td>GENFSD</td>
<td>GENERATE FORM STRUCTURE DATA INITIALIZATION</td>
<td></td>
</tr>
<tr>
<td>GENINS</td>
<td>GENERATE INSERT DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENMAIN</td>
<td>GENERATE MAIN PROGRAM</td>
<td></td>
</tr>
<tr>
<td>GENNDP</td>
<td>GENERATE NODUPLICATE DECLARATIONS</td>
<td></td>
</tr>
<tr>
<td>GENPAG</td>
<td>GENERATE NEWPAG PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>GETCOL</td>
<td>GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING</td>
<td></td>
</tr>
<tr>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM THE USER</td>
<td></td>
</tr>
<tr>
<td>GETFIT</td>
<td>GET SUBTREE THAT FITS ON PAGE</td>
<td></td>
</tr>
<tr>
<td>GETLOWLEF</td>
<td>GET LOWER LEFT CHILD NODE</td>
<td></td>
</tr>
<tr>
<td>GETLOWRIT</td>
<td>GET LOWER RIGHT CHILD NODE</td>
<td></td>
</tr>
<tr>
<td>GETPAR</td>
<td>GET PARENT NODE</td>
<td></td>
</tr>
<tr>
<td>GETPTH</td>
<td>GET PATH</td>
<td></td>
</tr>
<tr>
<td>GETSIZE</td>
<td>GET SUBTREE SIZE</td>
<td></td>
</tr>
<tr>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
<td></td>
</tr>
<tr>
<td>GETTOP</td>
<td>GET TOP OF TREE</td>
<td></td>
</tr>
<tr>
<td>GETUPLFT</td>
<td>GET UPPER LEFTMOST NODE</td>
<td></td>
</tr>
<tr>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
<td></td>
</tr>
<tr>
<td>GRP/MEM</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
<td></td>
</tr>
<tr>
<td>HASDATA</td>
<td>DETERMINE IF THERE ARE ANY SELECT STATEMENTS</td>
<td></td>
</tr>
<tr>
<td>HASITEM</td>
<td>THIS ROUTINE DETERMINES IF THERE IS AN ITEM WITHIN</td>
<td></td>
</tr>
<tr>
<td>HASLOWER</td>
<td>HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?</td>
<td></td>
</tr>
<tr>
<td>HBALANC</td>
<td>HORIZONTAL TREE BALANCE</td>
<td></td>
</tr>
<tr>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
<td></td>
</tr>
<tr>
<td>INDENT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
<td></td>
</tr>
<tr>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
<td></td>
</tr>
<tr>
<td>INSRSV</td>
<td>INSERT RESOLVE</td>
<td></td>
</tr>
<tr>
<td>INSWS</td>
<td>INSERT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>ISOPNE</td>
<td>DETERMINE IF THIS FIELD IS OPEN ENDED</td>
<td></td>
</tr>
<tr>
<td>MAKACT</td>
<td>MAKE ACTION LIST ELEMENT</td>
<td></td>
</tr>
<tr>
<td>MAKES/CNU</td>
<td>MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE</td>
<td></td>
</tr>
<tr>
<td>MAKES/IND</td>
<td>INDENT</td>
<td></td>
</tr>
<tr>
<td>MAKES/NUM</td>
<td>NUMBER PICTURE CLAUSE</td>
<td></td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKINS</td>
<td>MAKE INSERT</td>
<td></td>
</tr>
<tr>
<td>MAKINT</td>
<td>MAKE EXPRESSION INTO AN INTEGER</td>
<td></td>
</tr>
<tr>
<td>MAKPS</td>
<td>MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE</td>
<td></td>
</tr>
<tr>
<td>MAKQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
<td></td>
</tr>
<tr>
<td>MAKSTR</td>
<td>MAKE EXPRESSION INTO A STRING</td>
<td></td>
</tr>
<tr>
<td>MAKWH</td>
<td>MAKE WHERE</td>
<td></td>
</tr>
<tr>
<td>MAKWHES</td>
<td>MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES</td>
<td></td>
</tr>
<tr>
<td>MAKWHES/C</td>
<td>COBOL WHERE ES</td>
<td></td>
</tr>
<tr>
<td>MAKWHES/C</td>
<td>C WHERE ES</td>
<td></td>
</tr>
<tr>
<td>MAKWHES/N</td>
<td>NUMBER PICTURE CLAUSE</td>
<td></td>
</tr>
<tr>
<td>MAPDB</td>
<td>MAP DATABASE</td>
<td></td>
</tr>
<tr>
<td>MKINC</td>
<td>MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)</td>
<td></td>
</tr>
<tr>
<td>MKPOS</td>
<td>MAKE POSITION NODE</td>
<td></td>
</tr>
<tr>
<td>MLPFRM</td>
<td>MAKE A LIST OF PRESENTED FORMS</td>
<td></td>
</tr>
<tr>
<td>MODPAGE</td>
<td>MODIFY PAGES</td>
<td></td>
</tr>
<tr>
<td>MOVCLD</td>
<td>MOVE CHILDREN</td>
<td></td>
</tr>
<tr>
<td>MOVECLD</td>
<td>MOVE CHILD'S POSITION</td>
<td></td>
</tr>
<tr>
<td>MYALLOC</td>
<td>MY MALLOC</td>
<td></td>
</tr>
<tr>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
<td></td>
</tr>
<tr>
<td>NDMLLAB</td>
<td>GENERATE LABELS</td>
<td></td>
</tr>
<tr>
<td>NDMLLINK</td>
<td>LINKAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>NEXTLEV</td>
<td>ADVANCE POINTERS TO NEXT LEVEL OF SUBTREE</td>
<td></td>
</tr>
<tr>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
<td></td>
</tr>
<tr>
<td>OPNFIL</td>
<td>GENERATE OPEN FILE SECTION</td>
<td></td>
</tr>
<tr>
<td>PAGNODE</td>
<td>PAGE NODES</td>
<td></td>
</tr>
<tr>
<td>PAGTREE</td>
<td>PAGE TREE</td>
<td></td>
</tr>
<tr>
<td>PEMAP</td>
<td>THE PRESENTATION SCHEMA AND THE EXTERNAL SCHEMA AND MAPPING</td>
<td></td>
</tr>
<tr>
<td>PRNT</td>
<td>PRINT MODULE NAMES HIERARCHICALLY</td>
<td></td>
</tr>
<tr>
<td>PRNTREE</td>
<td>PRINT TREE</td>
<td></td>
</tr>
<tr>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/CO</td>
<td>COBOL SUBSTITUTE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/CS</td>
<td>C SUBSTITUTE</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/IN</td>
<td>INDENT</td>
<td></td>
</tr>
<tr>
<td>PUTLIN</td>
<td>PRINT LEVEL OF TREE</td>
<td></td>
</tr>
<tr>
<td>READDB</td>
<td>READ DATA BASE</td>
<td></td>
</tr>
<tr>
<td>READTREE</td>
<td>READ DUMPTREE FILE</td>
<td></td>
</tr>
<tr>
<td>Include File</td>
<td>Module Name</td>
<td>Module Purpose</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>REPOS</td>
<td>REPOSITION MODULE EXPANSIONS</td>
<td></td>
</tr>
<tr>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
<td></td>
</tr>
<tr>
<td>RSETSTAT</td>
<td>RESET STATISTIC</td>
<td></td>
</tr>
<tr>
<td>RWEXPD</td>
<td>REPORT WRITER EXPAND ARRAYS</td>
<td></td>
</tr>
<tr>
<td>RWOPN</td>
<td>REPORT WRITER OPEN FORMS</td>
<td></td>
</tr>
<tr>
<td>RWSP/FIXF</td>
<td>FIX UP A FORM</td>
<td></td>
</tr>
<tr>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td>SELECT GENERATE SELECT CODE</td>
<td></td>
</tr>
<tr>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
<td></td>
</tr>
<tr>
<td>SELLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
<td></td>
</tr>
<tr>
<td>SELMAP</td>
<td>MAP SELECTED DATA TO OUTPUT RECORD</td>
<td></td>
</tr>
<tr>
<td>SELOPEN</td>
<td>SELECT OPEN</td>
<td></td>
</tr>
<tr>
<td>SELRSV</td>
<td>SELECT RESOLVE</td>
<td></td>
</tr>
<tr>
<td>SELWHR</td>
<td>SELECT WHERE</td>
<td></td>
</tr>
<tr>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
<td></td>
</tr>
<tr>
<td>SORT</td>
<td>SORT MODULE NAMES</td>
<td></td>
</tr>
<tr>
<td>SPLICE</td>
<td>SPLICE TREE INTO ANOTHER TREE</td>
<td></td>
</tr>
<tr>
<td>SPLITNODE</td>
<td>SPLIT A NODE FOR PAGE BREAKS</td>
<td></td>
</tr>
<tr>
<td>STATRSV</td>
<td>STATISTIC RESOLVE</td>
<td></td>
</tr>
<tr>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
<td></td>
</tr>
<tr>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
<td></td>
</tr>
<tr>
<td>TRGRSV</td>
<td>TRIGGER RESOLVE</td>
<td></td>
</tr>
<tr>
<td>UQFOR</td>
<td>UNIVERSAL QUALIFIER FOR LOOP</td>
<td></td>
</tr>
<tr>
<td>UQPTH</td>
<td>UNIVERSAL QUALIFIER PATH</td>
<td></td>
</tr>
<tr>
<td>USING</td>
<td>GENERATE USING SECTION</td>
<td></td>
</tr>
<tr>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
<td></td>
</tr>
<tr>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
<td></td>
</tr>
<tr>
<td>WINRSV</td>
<td>WINDOW RESOLVE</td>
<td></td>
</tr>
<tr>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
<td></td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/DB</td>
<td>DEFAULT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/FO</td>
<td>INSERT FORMAT CODES</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/TB</td>
<td>TEXT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE DEFAULT BUFFER</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE FIELD</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE TEXT BUFFER</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WR</td>
<td>WRITE TEXT</td>
<td></td>
</tr>
</tbody>
</table>
REPORT WRITER Where-include-file-used List

<table>
<thead>
<tr>
<th>Include File</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
<td></td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
<td></td>
</tr>
</tbody>
</table>
3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.
**REPORT WRITER Where-external-routine-used List**

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td></td>
<td>RWEXPD</td>
<td>REPORT WRITER EXPAND ARRAYS</td>
</tr>
<tr>
<td>ADFRM</td>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>ATOF</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>ATOI</td>
<td>CCONV</td>
<td>C CONVERSIONS</td>
</tr>
<tr>
<td></td>
<td>CES</td>
<td>C ES</td>
</tr>
<tr>
<td></td>
<td>COBCONV</td>
<td>COBOL CONVERSIONS</td>
</tr>
<tr>
<td></td>
<td>COBES</td>
<td>COBOL ES RECORD</td>
</tr>
<tr>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td></td>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
</tr>
<tr>
<td></td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>BLEN</td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td>CHKSIZE</td>
<td>CHECK SIZE OF ITEMS DOING CONVERSIONS ON</td>
</tr>
<tr>
<td></td>
<td>MKINC</td>
<td>MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)</td>
</tr>
<tr>
<td></td>
<td>PSSTRC/COBCOBOL</td>
<td>SUBSTITUTE</td>
</tr>
<tr>
<td></td>
<td>PSSTRC/CSUC</td>
<td>SUBSTITUTE</td>
</tr>
<tr>
<td></td>
<td>SELLEN</td>
<td>COMPUTE LENGTH OF SELECT PS RECORD</td>
</tr>
</tbody>
</table>

3-53
REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALLOC</td>
<td>CALLOC</td>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td>COPFLD</td>
<td>COPFLD</td>
<td>RWEXP</td>
<td>REPORT WRITER EXPAND ARRAYS</td>
</tr>
<tr>
<td>COPFLD</td>
<td>COPFLD</td>
<td>RWSP/FIXFR</td>
<td>FIX UP A FORM</td>
</tr>
<tr>
<td>COPFLD</td>
<td>COPFLD</td>
<td>WINRSV</td>
<td>WINDOW RESOLVE</td>
</tr>
<tr>
<td>DELFLD</td>
<td>DELFLD</td>
<td>FLANCI</td>
<td>FLAN CALLABLE INTERFACE</td>
</tr>
<tr>
<td>ERRPRO</td>
<td>ERRPRO</td>
<td>CDMESQY</td>
<td>PROGRAM NAME CDMESQY</td>
</tr>
<tr>
<td>ESCPY</td>
<td>ESCPY</td>
<td>CCONV</td>
<td>C CONVERSIONS</td>
</tr>
<tr>
<td>ESCPY</td>
<td>ESCPY</td>
<td>CES</td>
<td>C ES</td>
</tr>
<tr>
<td>ESCPY</td>
<td>ESCPY</td>
<td>COBCONV</td>
<td>COBOL CONVERSIONS</td>
</tr>
<tr>
<td>ESCPY</td>
<td>ESCPY</td>
<td>COBES</td>
<td>COBOL ES RECORD</td>
</tr>
<tr>
<td>ESCPY</td>
<td>ESCPY</td>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
</tr>
<tr>
<td>ESCPY</td>
<td>ESCPY</td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>ESCPY</td>
<td>ESCPY</td>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
</tr>
<tr>
<td>FCLOSE</td>
<td>FCLOSE</td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>FCLOSE</td>
<td>FCLOSE</td>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td>FCLOSE</td>
<td>FCLOSE</td>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
</tbody>
</table>
**REPORT WRITER Where-external-routine-used List**

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

FGETS

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td>READTREE</td>
<td>READ DUMPTREE FILE</td>
</tr>
<tr>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
</tbody>
</table>

FOPEN

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM THE USER</td>
</tr>
<tr>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
</tbody>
</table>

PRINTF

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>ASSIGN FILE SECTION</td>
</tr>
<tr>
<td>CCONV</td>
<td>C CONVERSIONS</td>
</tr>
<tr>
<td>CES</td>
<td>C ES</td>
</tr>
<tr>
<td>CHKSIZE</td>
<td>CHECK SIZE OF ITEMS DOING CONVERSIONS ON</td>
</tr>
<tr>
<td>CLSFIL</td>
<td>CLOSE FILES</td>
</tr>
<tr>
<td>COBCONV</td>
<td>COBOL CONVERSIONS</td>
</tr>
<tr>
<td>COBES</td>
<td>COBOL ES RECORD</td>
</tr>
<tr>
<td>COBESPS</td>
<td>COBOL ES TO PS</td>
</tr>
<tr>
<td>COBPE</td>
<td>COBOL PE</td>
</tr>
<tr>
<td>DATAGEN</td>
<td>DATA DIVISION GENERATE</td>
</tr>
<tr>
<td>ENDDGEN</td>
<td>END GENERATE</td>
</tr>
<tr>
<td>FD</td>
<td>FD SECTION DECLARATIONS</td>
</tr>
<tr>
<td>FILELNK</td>
<td>FILE LINKAGE SECTION GENERATE</td>
</tr>
<tr>
<td>GEN</td>
<td>GENERATE A LINE OF CODE</td>
</tr>
<tr>
<td>INSERT</td>
<td>INSERT PROCEDURE</td>
</tr>
<tr>
<td>INSWS</td>
<td>INSERT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td>MAKES/CNUMC</td>
<td>NUMBERS</td>
</tr>
<tr>
<td>MAKES/NUMPNUMBER</td>
<td>PICTURE CLAUSE</td>
</tr>
<tr>
<td>MAKINS</td>
<td>MAKE INSERT</td>
</tr>
</tbody>
</table>
# REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MAKWH</td>
<td>MAKE WHERE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKWHES</td>
<td>MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKWHES/COCOBOL</td>
<td>WHERE ES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAKWHES/NUNUMBER</td>
<td>PICTURE CLAUSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NDMLLAB</td>
<td>GENERATE LABELS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPNFLIL</td>
<td>GENERATE OPEN FILE SECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PRNTREE</td>
<td>PRINT TREE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSSTRC/COBCOBOL</td>
<td>SUBSTITUTE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSSTRC/CSUC</td>
<td>SUBSTITUTE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USING</td>
<td>GENERATE USING SECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FPUTS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PRNTREE</td>
<td>PRINT TREE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FREE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELNODE</td>
<td>DELETE A SPECIFIED NODE IN TREE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WINRSV</td>
<td>WINDOW RESOLVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FSEEK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
</tbody>
</table>

3-56
**REPORT WRITER Where-external-routine-used List**

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTELL</td>
<td>READTREE</td>
<td>READ DUMPTREE FILE</td>
</tr>
<tr>
<td>FWRITE</td>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
<tr>
<td></td>
<td>WRTFRM/DBFDEFAULT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM/TBFTEXT BUFFER CLOSE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WRTWRITE DEFAULT BUFFER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WRTWRITE FIELD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WRTWRITE TEXT BUFFER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRTFRM/WRTWRITE TEXT</td>
<td></td>
</tr>
<tr>
<td>GDATA</td>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>GETC</td>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td></td>
<td>READTREE</td>
<td>READ DUMPTREE FILE</td>
</tr>
<tr>
<td></td>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
<tr>
<td></td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>INITIAL</td>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
</tbody>
</table>

INITFP
### REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
<td></td>
</tr>
<tr>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
<td></td>
</tr>
<tr>
<td>INSMAP</td>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
</tr>
<tr>
<td>ISALNUM</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>ISALPHA</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>ISDIGIT</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>ISSPACE</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>MAKFLD</td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
<tr>
<td>MALLOC</td>
<td>BLDMOD</td>
<td>BUILD MODULE</td>
</tr>
<tr>
<td></td>
<td>BLDNODE</td>
<td>BUILD NODE</td>
</tr>
<tr>
<td></td>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td></td>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MLPFRM</td>
<td>MAKE A LIST OF PRESENTED FORMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MYALLOC</td>
<td>MY MALLOC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SORT</td>
<td>SORT MODULE NAMES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UQPTH</td>
<td>UNIVERSAL QUALIFIER PATH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WINRSV</td>
<td>WINDOW RESOLVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAP</td>
<td>PROCGEN PROCEDURE DIVISION GENERATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DCLINDX</td>
<td>DECLARE INDEX VARIABLES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GETSIZE</td>
<td>GET SUBTREE SIZE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HBALANC</td>
<td>HORIZONTAL TREE BALANCE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MEMCMP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MEMCPY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WRTFRM/WRTWRITE</td>
<td>WRITE FIELD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>
## REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>------</td>
<td>------</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>MEMSET</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DRAWLEV</td>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>GETSIZE</td>
<td>GET SUBTREE SIZE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
<tr>
<td></td>
<td>OISCR</td>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td></td>
<td>OUTSCR</td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td></td>
<td>PMSGLC</td>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td></td>
<td>PMSGLS</td>
<td>BLDMOD</td>
<td>BUILD MODULE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ERROR</td>
<td>ISSUE ERROR MESSAGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FATAL</td>
<td>ISSUE FATAL ERROR MESSAGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
</tr>
</tbody>
</table>

3-60
REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINTF</td>
<td>PRNT</td>
<td>PRINT MODULE NAMES HIERARCHICALLY</td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
<tr>
<td>PSESMAP</td>
<td>PROCGEN</td>
<td>PROCEDURE DIVISION GENERATE</td>
</tr>
<tr>
<td>PTHPTR</td>
<td>GETPTH</td>
<td>GET PATH</td>
</tr>
<tr>
<td></td>
<td>UQPTH</td>
<td>UNIVERSAL QUALIFIER PATH</td>
</tr>
<tr>
<td>PUTATT</td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>PUTC</td>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td></td>
<td>ESPSMAP/ININDENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INDEMT</td>
<td>INDENT A LINE OF GENERATED CODE</td>
</tr>
<tr>
<td></td>
<td>MAKES/INDEMT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRNTREE</td>
<td>PRINT TREE</td>
</tr>
<tr>
<td></td>
<td>PSSTRC/ININDENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PUTLIN</td>
<td>PRINT LEVEL OF TREE</td>
</tr>
<tr>
<td>PUTCUR</td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>------</td>
<td>-------</td>
<td>------</td>
</tr>
</tbody>
</table>

SPRINTF

- BLDMOD BUILD MODULE
- BSCODE BUILD SUBROUTINE CODE
- CALCSTAT CALCULATE STATISTIC
- CHKGRT CHECK FOR GROUP SEPARATORS OR END OF FILE
- CLRNDP CLEAR NODUPLICATE FIELDS
- DBFREAD GENERATE DATA BASE FREAD
- DCLINDEX DECLARE INDEX VARIABLES
- DINDEX DO CHART INDEX
- DRAWLEV DRAW A LEVEL OF THE CHART
- ERROR ISSUE ERROR MESSAGE
- FATAL ISSUE FATAL ERROR MESSAGE
- FRMPDAT FORM PDATA
- FRNTND FRONT END FOR FORMS
- 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
## REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GETFILE</td>
<td>RETURN A FILE POINTER BASED ON INPUT FROM THE USER</td>
</tr>
<tr>
<td></td>
<td>MAKQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
</tr>
<tr>
<td></td>
<td>MAPDB</td>
<td>MAP DATABASE</td>
</tr>
<tr>
<td></td>
<td>MKINC</td>
<td>MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)</td>
</tr>
<tr>
<td></td>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td></td>
<td>PRNTREE</td>
<td>PRINT TREE</td>
</tr>
<tr>
<td></td>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
</tr>
<tr>
<td></td>
<td>RSETSTAT</td>
<td>RESET STATISTIC</td>
</tr>
<tr>
<td></td>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
</tr>
<tr>
<td></td>
<td>SELOPN</td>
<td>SELECT OPEN</td>
</tr>
<tr>
<td></td>
<td>SELWHR</td>
<td>SELECT WHERE</td>
</tr>
<tr>
<td></td>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
</tr>
<tr>
<td></td>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
</tr>
<tr>
<td></td>
<td>UQFOR</td>
<td>UNIVERSAL QUALIFIER FOR LOOP</td>
</tr>
<tr>
<td></td>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
</tr>
<tr>
<td></td>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
</tr>
<tr>
<td></td>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td></td>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>

### STRASN

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CHKARY</td>
<td>CHECK ARRAY</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>CHECK FORM</td>
</tr>
<tr>
<td>RWEXPD</td>
<td>REPORT WRITER EXPAND ARRAYS</td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
</tbody>
</table>

### STRCAT

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DCLINDX</td>
<td>DECLARE INDEX VARIABLES</td>
</tr>
<tr>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td>MAKES/CNUMC</td>
<td>NUMBERS</td>
</tr>
<tr>
<td>MAKQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRCHR</td>
<td>DASH</td>
<td>WRITE DASH '-'</td>
</tr>
<tr>
<td></td>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td></td>
<td>GENDOA</td>
<td>GENERATE PROCEDURE &quot;DOACT&quot; DO ACTION</td>
</tr>
<tr>
<td></td>
<td>GETCOL</td>
<td>GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING</td>
</tr>
<tr>
<td></td>
<td>GETPPTH</td>
<td>GET PATH</td>
</tr>
<tr>
<td></td>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
</tr>
<tr>
<td></td>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td></td>
<td>MAPDB</td>
<td>MAP DATABASE</td>
</tr>
<tr>
<td></td>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
</tr>
<tr>
<td></td>
<td>PUTLIN</td>
<td>PRINT LEVEL OF TREE</td>
</tr>
<tr>
<td></td>
<td>UQPTH</td>
<td>UNIVERSAL QUALIFIER PATH</td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
<tr>
<td>STRCMP</td>
<td>BLDMOD</td>
<td>BUILD MODULE</td>
</tr>
<tr>
<td></td>
<td>DOINDEX</td>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td></td>
<td>FNDATT</td>
<td>FIND ATTRIBUTE</td>
</tr>
<tr>
<td></td>
<td>FNDFRM</td>
<td>FIND FORM</td>
</tr>
<tr>
<td></td>
<td>GENAR</td>
<td>GENERATE ACTION PRESENT</td>
</tr>
<tr>
<td></td>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
</tr>
<tr>
<td></td>
<td>GFLDPT</td>
<td>GET FIELD POINTER</td>
</tr>
<tr>
<td></td>
<td>RWSP/FIXFR</td>
<td>FIX UP A FORM</td>
</tr>
<tr>
<td></td>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
</tr>
<tr>
<td></td>
<td>SORT</td>
<td>SORT MODULE NAMES</td>
</tr>
<tr>
<td></td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td></td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
<tr>
<td>STRCPY</td>
<td>BLDMOD</td>
<td>BUILD MODULE</td>
</tr>
<tr>
<td></td>
<td>CSTASH</td>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td></td>
<td>DCLINDX</td>
<td>DECLARE INDEX VARIABLES</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENAS</td>
<td>GENERATE ACTION SET</td>
<td></td>
</tr>
<tr>
<td>GETCOL</td>
<td>GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING</td>
<td></td>
</tr>
<tr>
<td>GETPTH</td>
<td>GET PATH</td>
<td></td>
</tr>
<tr>
<td>GETTBL</td>
<td>GET A TABLE NAME</td>
<td></td>
</tr>
<tr>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
<td></td>
</tr>
<tr>
<td>INSW</td>
<td>INSERT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>MAKES</td>
<td>MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE</td>
<td></td>
</tr>
<tr>
<td>MAKQR</td>
<td>MAKE QUALIFIED REFERENCE</td>
<td></td>
</tr>
<tr>
<td>NULBLK</td>
<td>BLANK FILL A STRING</td>
<td></td>
</tr>
<tr>
<td>SELGEN</td>
<td>SELECT GENERATE</td>
<td></td>
</tr>
<tr>
<td>SELWS</td>
<td>SELECT WORKING STORAGE SECTION</td>
<td></td>
</tr>
<tr>
<td>SORT</td>
<td>SORT MODULE NAMES</td>
<td></td>
</tr>
<tr>
<td>UQPPTH</td>
<td>UNIVERSAL QUALIFIER PATH</td>
<td></td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WRTWRITE FIELD</td>
<td>WRITE FORM</td>
<td></td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STRLEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK FIELD</td>
</tr>
<tr>
<td>CHECK FORM</td>
</tr>
<tr>
<td>CHARACTER STASH</td>
</tr>
<tr>
<td>DECLARE INDEX VARIABLES</td>
</tr>
<tr>
<td>DO CHART INDEX</td>
</tr>
<tr>
<td>DRAW A LEVEL OF THE CHART</td>
</tr>
<tr>
<td>ISSUE ERROR MESSAGE</td>
</tr>
<tr>
<td>ISSUE FATAL ERROR MESSAGE</td>
</tr>
<tr>
<td>GENERATE ACTION SET</td>
</tr>
<tr>
<td>GENERATE FORM STRUCTURE DATA</td>
</tr>
<tr>
<td>INITIALIZE</td>
</tr>
<tr>
<td>MAKES</td>
</tr>
<tr>
<td>MAKE THE EXTERNAL SCHEMA RECORD STRUCTURE</td>
</tr>
<tr>
<td>MAP DATABASE</td>
</tr>
<tr>
<td>MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)</td>
</tr>
<tr>
<td>PRINT TREE</td>
</tr>
<tr>
<td>PRINT LEVEL OF TREE</td>
</tr>
<tr>
<td>READ DUMPTREE FILE</td>
</tr>
</tbody>
</table>
# REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>REPORT WRITER</td>
<td>Where-external-routine-used List</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSETNDP</td>
<td>RESET NODUPLICATE FIELDS TO VALUE OF NODUP%D</td>
</tr>
<tr>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
</tr>
<tr>
<td>SETNDP</td>
<td>SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED</td>
</tr>
<tr>
<td>STRIPLEV</td>
<td>DRAW STRIP CHART LEVEL</td>
</tr>
<tr>
<td>VISITA</td>
<td>VISIT ARRAYS ON THIS FORM</td>
</tr>
<tr>
<td>WARNING</td>
<td>ISSUE WARNING MESSAGE</td>
</tr>
<tr>
<td>WRTEXP</td>
<td>WRITE EXPRESSION</td>
</tr>
<tr>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
<tr>
<td>WRTFRM/WRTWRITE</td>
<td>WRITE TEXT</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRNCMP</td>
<td>C CONVERSIONS</td>
</tr>
<tr>
<td>SAVEES</td>
<td>SAVE ES INFORMATION</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRNCPY</td>
<td>MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE</td>
</tr>
<tr>
<td>NDMGEN</td>
<td>NDL COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td>WRTFRM/WRTWRITE</td>
<td>WRITE FIELD</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRSPN</td>
<td>GENERATE ACTION SET</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRUPC</td>
<td>GET PATH</td>
</tr>
<tr>
<td>SORT</td>
<td>SORT MODULE NAMES</td>
</tr>
<tr>
<td>STDCODE</td>
<td>STANDARD COBOL CODE</td>
</tr>
<tr>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>
# REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSMSG</td>
<td>CHKFLD</td>
<td>CHECK FIELD</td>
</tr>
<tr>
<td></td>
<td>NDMLGEN</td>
<td>NDML COBOL APPLICATION GENERATOR</td>
</tr>
<tr>
<td></td>
<td>WRTFRM</td>
<td>WRITE FORM</td>
</tr>
<tr>
<td>TERMFP</td>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>TOUPPER</td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>FRNTND</td>
<td>FRONT END FOR FORMS</td>
</tr>
<tr>
<td></td>
<td>HRW/MAIN</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>TRMNDML</td>
<td>GRP/MAIN</td>
<td>GENERATE APPLICATION/REPORT PROGRAM</td>
</tr>
<tr>
<td>UNGETC</td>
<td>READTREE</td>
<td>READ DUMPTREE FILE</td>
</tr>
<tr>
<td></td>
<td>YYLEX</td>
<td>LEXICAL ANALYZER FOR FLAN</td>
</tr>
</tbody>
</table>
REPORT WRITER Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYERROR</td>
<td>YYPARSE</td>
<td>FLAN PARSER</td>
</tr>
</tbody>
</table>
3.10.7 Main Program Parts List

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

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP/MAIN</td>
<td>Purpose--&gt;GENERATE APPLICATION/REPORT PROGRAM</td>
<td></td>
</tr>
<tr>
<td>ABS</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ACTRSV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ADDCHK</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ADDFRM</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ASSIGN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ATOF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ATOI</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>BLDSUB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>BLEN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>BSCODE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CALCSTAT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CALLOC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>CCONV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CDMESQY</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CES</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CESPS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CHKARY</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CHKFLD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CHKFRM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CHKGRP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CHKSIZE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CLNDP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CLSFIL</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>COBCONV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>COBES</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>COBESPS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>COBPE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>COPFLD</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>CPE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CSTASH</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CTRLRSV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DASH</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DATAGEN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DBREAD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DCLINDX</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DELFLD</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ENDGEN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ERROR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>Main Pgm Name</td>
<td>Module Name</td>
<td>Type</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ERRPRO</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ESCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ESPSMAP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ESPSMAP/INDENT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FATAL</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FCLOSE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FILELNK</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FL'NCI</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FLDRSV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FLDTYP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FNDATT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FNDFRM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FOPEN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FPRINTF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FREE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FRMPDAT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FRN'TND</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>FWRITE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GDATA</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GEN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAA</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAAL</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENACT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAH</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAI</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAL</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAQ</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENAT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENBEG</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENCHG</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENDB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENDOA</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENDS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENFP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENFS</td>
<td>Well-defined module</td>
<td></td>
</tr>
</tbody>
</table>
# REPORT WRITER Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENFSD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENINS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENMAIN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENNDP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GENPAG</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETCOL</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETFILE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETPTH</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETTBL</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GFLDPT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>HASDATA</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>HASITEM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>HASLOWER</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>INDENT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>INITIAL</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>INITFP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>INSERT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>INSMAP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>INSRSV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>INSWS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ISALNUM</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ISALPHA</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ISDIGIT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ISOPNE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ISSPACE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAKACT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKES</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKES/CNUMPIC</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKES/INDENT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKES/NUMPIC</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKFLD</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAKINS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKINT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKPS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKQR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKSTR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKWH</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKWHES</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKWHES/COBWHES</td>
<td>Well-defined module</td>
<td></td>
</tr>
</tbody>
</table>
### REPORT WRITER Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAKWHES/CWHESS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAKWHES/NUMPIC</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MALLOC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAPDB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MEMCMP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MEMSET</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MKINC</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MKPOS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MLPFRM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MYALLOC</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>NDMLGEN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>NDMLLAB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>NDMLLNK</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>NULBLK</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>OISCR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>OPNFIL</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PEMAP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PMSGLC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PMSGLS</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PRINTF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PROCGEN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PSESMAP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/COBSUB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/CSUB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PSSTRC/INDENT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PTHPTR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PUTC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>READDB</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>RSETNDP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>RSETSTAT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>RWEXPD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>RWOPN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>RWSP/FIXFRM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SAVEES</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SELECT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SELGEN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SELLEN</td>
<td>Well-defined module</td>
<td></td>
</tr>
</tbody>
</table>
REPORT WRITER Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELMAP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SELOPN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SELRSV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SELWHR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SELWS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SETNDP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SPRINTF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STATRSV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>STDCODE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>STRASN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRCAT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STCHR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRCMP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRLEN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRNCMP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRNCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRSPN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRUPC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>SYSMSG</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>TERMFP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>TOUPPER</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>TRGRSV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>TRMNAT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>TRMNDML</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>UNGETC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>UQFOR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>UQPTH</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>USING</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>VISITA</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WARNING</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WINRSV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WRTEXP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WRTFRM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/DBFLOS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/FORMAT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/TBFCLOS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WRTDBF</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WRTFLD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>WRTFRM/WRTTBF</td>
<td>Well-defined module</td>
<td></td>
</tr>
</tbody>
</table>
REPORT WRITER Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRTFRM/WRTTXT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>YYERROR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>YYLEX</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>YYPARSE</td>
<td>Well-defined module</td>
<td></td>
</tr>
</tbody>
</table>

3-75
**REPORT WRITER Main Program Parts List**

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRW/MAIN</td>
<td>Purpose</td>
<td>MAIN MODULE FOR HIERARCHICAL REPORT WRITER</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>ARRANGE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ATOI</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>BLDMOD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>BLDNODE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>CLOSEGAP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>COPYNODE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DELNODE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DOI INDEX</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>DRAWLEV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>ESCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FCLOSE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FGETS</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FOPEN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FPRINTF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FPUTS</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FREE</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FSEEK</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>FTELL</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GDATA</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETFIT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETLOWLEF</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETLOWRIT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETPAR</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETSIZE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETTOP</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETUPLFT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>HBALANC</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>INITIAL</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>INITFP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MALLOC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MEMCMP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MEMCOPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MEMSET</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>MODPAGE</td>
<td>Well-defined module</td>
<td></td>
</tr>
</tbody>
</table>
# REPORT WRITER Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVCLD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MOVECLD</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>NEXTLEV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>OISCR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>OUTSCR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PAGNODE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PAGTREE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PMSGLS</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PRNTREE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>PUTATT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PUTC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PUTCUR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>PUTLIN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>READTREE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>REPOS</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SORT</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SPLICE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SPLITNODE</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>SPRINTF</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRCHR</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRCMP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRIPLEV</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>STRENN</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>STRUPC</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>TERMFP</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>TRMNAT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>UNGETC</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
3.10.8 Module Documentation

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

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

NAME: Name of program Module.

PURPOSE: Purpose of Module as detailed in the source code.

LANGUAGE: Programming language source code is written in.
The choices are:
- VAX-11 FORTRAN
- C (I/S-1 Workbench 'C')
- VAX-11 COBOL

MODULE TYPE: Whether a Program, Subroutine, or Function.

SOURCE FILE: Name of Source File from file specification.

SOURCE FILE TYPE: Source File Extension from file specification.

HOST: Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.

SUBSYSTEM: IISS sub-system this file resides in.

SUBDIRECTORY: Sub-directory of that subsystem in which this file resides.

DOCUMENTATION GROUP: Name of documentation group of which this source file is a member.

DESCRIPTION: A description of the module as obtained 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.
REPORT WRITER 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:

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

ROUTINES CALLED:

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

3-80
CALLED DIRECTLY BY:
---------------------
TRGRSv     - TRIGGER RESOLVE

USED IN MAIN PROGRAM(S):
------------------------
GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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
ADS THE SPECIFIED POSITION TO THE OVERLAP CHECK LIST

ARGUMENTS:

POSPTR = POS *

INCLUDE FILES:

STDTYPE - 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
REPORT WRITER Module Documentation

NAME: ARRANGE
PURPOSE: ARRANGE CHART AND ASSIGNS PAGE NUMBERS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID
SOURCE FILE: ARRANGE
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
ARRANGE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE ASSIGNS PAGE NUMBERS.

INCLUDE FILES:

STDTYP  - STANDARD TYPE DEFINITIONS
STDIO   - **** PURPOSE NOT FOUND BY STRIPPER ****
CHART   - CHART INCLUDE FILE

FUNCTIONS CALLED:

GETLOWLEF  - GET LOWER LEFT CHILD NODE
GETTOP    - GET TOP OF TREE

CALLED DIRECTLY BY:

HRW/MAIN  - MAIN MODULE FOR HIERARCHICAL REPORT WRITER

USED IN MAIN PROGRAM(S):

HRW/MAIN  - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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: 
----------------

PRINTF
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
REPORT WRITER Module Documentation

NAME: BLDMOD
PURPOSE: BUILD MODULE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: MODULE * ()
SOURCE FILE: BLDMOD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

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

SYNOPSIS

BLDMOD()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

RETURNS A POINTER TO THE NAMED MODULE. THE MODULE IS
ALLOCATED IF IT
DOES NOT ALREADY EXIST

ARGUMENTS:
----------

MODULE_NAME = CHAR []
FILEPOS = LONG
WIDTH = INT
DEPTH = INT
TOP_POS = INT
BOT_POS = INT
L_MARGIN = INT

INCLUDE FILES:
---------------

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

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

STRCMP

3-85
SPRINTF
PMSGLS
MALLOC
STRCPY

CALLED DIRECTLY BY:
-------------------
COPYNODE   - COPY A NODE IN TREE
GETPAR     - GET PARENT NODE
READTREE   - READ DUMPTREE FILE

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN    - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: BLDNODE
PURPOSE: BUILD NODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: NODE *
SOURCE FILE: BLDNODE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS

BLDNODE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

THIS ROUTINE BUILDS A LINKED LIST FOR TREE RELATIONSHIPS, IT
SETS UP PARENT - CHILD RELATIONSHIPS AS WELL AS LEFT - RIGHT.

ARGUMENTS:

PARENT_PTR = NODE *
LEFT_PTR = NODE *
RIGHT_PTR = NODE *
MODULE_PTR = MODULE *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

MALLOC

3-87
CALLED DIRECTLY BY:

COPYNODE - COPY A NODE IN TREE
GETPAR - GET PARENT NODE
MOVCLD - MOVE CHILDREN
READTREE - READ DUMPTREE FILE
SPLITNODE - SPLIT A NODE FOR PAGE BREAKS

USED IN MAIN PROGRAM(S):

HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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
FPFPARM    - 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
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 OF 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:
-----------------
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:
---------------------
SPRINTF - GENERATE A LINE OF CODE
GEN     - READ DATABASE
READB   - READ DATABASE
RSETNDP - 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:
----------------------
BLDSUB  - BUILD SUBROUTINES
USED IN MAIN PROGRAM(S):
---------------------
GRP/MAIN   - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
---------------
   STDTYP  -  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   -  GENERATE A LINE OF CODE

CALLED DIRECTLY BY:
---------------------
   FRMPDAT    -  FORM PDATA
   CALCSTAT   -  CALCULATE STATISTIC

USED IN MAIN PROGRAM(S):
------------------------
   GRP/MAIN    -  GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-------------

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

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

ATOI
ESCPY
STRNCMP
PRINTF
MAKES/INDENT - INDENT

CALLED DIRECTLY BY:
-------------------

MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE

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

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

SRVRRET - 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
REPORT WRITER 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:

STDTYP - STANDARD TYPE DEFINITIONS
STDOUT - **** 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
REPORT WRITER 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
STDO - **** 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
REPORT WRITER 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:

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:

MYALLOC - MY MALLOC
ABS
STRASN

CALLED DIRECTLY BY:

CHKFRM - CHECK FORM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

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

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

DESCRIPTION
CHECKS THE CURRENT FIELD FOR COMPLETENESS AND CONSISTENCY

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

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

CALLED DIRECTLY BY:
-------------------
YYPARSE  - FLAN PARSER

USED IN MAIN PROGRAM(S):
-----------------------
GRP/MAIN  - GENERATE APPLICATION/REPORT PROGRAM

3-100
REPORT WRITER Module Documentation

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

DESCRIPTION:

SYNOPSIS

VOID CHKFRM()

DESCRIPTION

CHECKS THE CURRENT FORM FOR COMPLETENESS AND CONSISTENCY

INCLUDE FILES:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

-----------------------------------------------------------------
YYPARSE - FLAN PARSER

3-101
USED IN MAIN PROGRAM(S):

GRP/MAIN   - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: CHKGRP
PURPOSE: CHECK FOR GROUP SEPARATORS 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 SEPARATOR OR END OF FILE. IF SO IT CLEARS THE
NODUP%D FIELDS
AND READS ANOTHER RECORD.

ARGUMENTS:

Fx = FIELD *

INCLUDE FILES:

---
STDTYP - STANDARD TYPE DEFINITIONS
STDTO - **** 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
---

3-103
ROUTINES CALLED:
------------------
HASLOWER - HAS A LOWER FORM WHICH READS THE SAME DATA RECORD?
SPRINTF
GEN - GENERATE A LINE OF CODE
CLRNDP - CLEAR NONDUPLICATE 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
REPORT WRITER 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:
---------------
STDTYP - STANDARD TYPE DEFINITIONS
STDOUT - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:
-----------------
BLEN
PRINTF

CALLED DIRECTLY BY:
---------------------
COBESPS - COBOL ES TO PS
COBPE - COBOL PE
USED IN MAIN PROGRAM(S):
--------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: CLOSEGAP
PURPOSE: CLOSE GAP IN TREE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID()
SOURCE FILE: CLSGAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
CLOSEGAP()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE CUTS A SECTION OUT OF THE TREE. THE SECTION FROM FIRST_PTR TO LAST_PTR AND ALL OF THE RELATED CHILDREN ARE UNLINKED AND THE RESULTING HOLE IS SPLICED.

ARGUMENTS:

FIRST_PTR = NODE *
LAST_PTR = NODE *

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

NEXTLEV - ADVANCE POINTERS TO NEXT LEVEL OF SUBTREE

CALLED DIRECTLY BY:

COPYNODE - COPY A NODE IN TREE

3-107
DFLNODE - DELETE A SPECIFIED NODE IN TREE
MVCLD - MOVE CHILDREN
REPOS - REPOSITION MODULE EXPANSIONS
SPLITNODE - SPLIT A NODE FOR PAGE BREAKS

USED IN MAIN PROGRAM(S):
-------------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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: 
----------------- 
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:

----------
SPRF
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
REPORT WRITER 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
STDSIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPARM - 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
FPINTF
INDENT - INDENT A LINE OF GENERATED CODE

CALLED DIRECTLY BY:

CLSFIL - CLOSE FILES
PROCGEN - PROCEDURE DIVISION GENERATE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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
RECCNT = INT

INCLUDE FILES:

STDTYP - 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
PRINTF
MAKES/INDENT - INDENT

CALLED DIRECTLY BY:

MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE

USED IN MAIN PROGRAM(S):

GRP/Main - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

STDTYP - STANDARD TYPE DEFINITIONS
STDOI - **** PURPOSE NOT FOUND BY STRIPPER ****
FPDF - 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
REPORT WRITER 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: 

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

ROUTINES CALLED: 

DASH - WRITE DASH '-'
GETBL - GET A TABLE NAME
GETCOL - GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
CHKSIZE - CHECK SIZE OF ITEMS DOING CONVERSIONS ON
PRINTF
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
REPORT WRITER 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:

STDTYP - 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
REPORT WRITER Module Documentation

NAME: COPYNODE
PURPOSE: COPY A NODE IN TREE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID()
SOURCE FILE: COPNODE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
COPYNODE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
ARGUMENTS:

PAGE_PTR = NODE *
NODE_PTR = NODE *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

BLDMOD - BUILD MODULE
BLDNODE - BUILD NODE
CLOSEGAP - CLOSE GAP IN TREE

CALLED DIRECTLY BY:

PAGNODE - PAGE NODES
USED IN MAIN PROGRAM(S):
----------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: CPE
PURPOSE: CPE
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:

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

ROUTINES CALLED:
-----------------
STRCPY
STRENC
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

3-119
REPORT WRITER Module Documentation

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

DESCRIPTION:

SYNOPSIS

CTLRSV(CTLPTR)

CTLLST *CTL PTR;

INPUTS:

CTL PTR - 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:

STDTYP - 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
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-----------------

STDTYP - STANDARD TYPE DEFINITIONS
STDCIO - **** 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
REPORT WRITER Module Documentation

NAME: DATAGEN
PURPOSE: DATA DIVISION GENERATE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: NDMILGEN
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
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:

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

Called Directly By:

STDCODE - STANDARD COBOL CODE
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
----------------
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:

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
REPORT WRITER 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:

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

ROUTINES CALLED:

---
MAX
STRCPY
PRINTF
STRLN
STRCAT
GEN - GENERATE A LINE OF CODE
CALLED DIRECTLY BY:

GENDOA  - GENERATE PROCEDURE "DOACT" DO ACTION

USED IN MAIN PROGRAM(S):

GRP/MAIN  - GENERATE APPLICATION/REPORT PROGRAM
NAME: DELNODE
PURPOSE: DELETE A SPECIFIED NODE IN TREE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: DELNODE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
DELNODE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
DELETES THE SPECIFIED NODE, FIXES ITS NEIGHBORS' POINTERS, AND DELETES ITS MODULE IF IT WAS THE LAST REFERENCE TO IT.

ARGUMENTS:

NODE_PTR = NODE *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

DELNODE - DELETE A SPECIFIED NODE IN TREE
CLOSEGAP - CLOSE GAP IN TREE
FREE
CALLED DIRECTLY BY:
---------------------
DELNODE     - DELETE A SPECIFIED NODE IN TREE
HRW/MAIN    - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
MOVCLD      - MOVE CHILDREN
REPOS       - REPOSITION MODULE EXPANSIONS

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN     - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: DOINDEX
PURPOSE: DO CHART INDEX
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: DOINDEX
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

-------------

SYNOPSIS

VOID DOINDEX(OUTCHART, PAGE_WIDTH, PAGE_DEPTH)
FILE OUTCHART;
INT PAGE_WIDTH;
INT PAGE_DEPTH;

INPUTS:
OUTCHART - OUTPUT FILE
PAGE_WIDTH - OUTPUT PAGE WIDTH
PAGE_DEPTH - OUTPUT PAGE DEPTH

DESCRIPTION

PRINTS THE INDEX FOR THE CHART. IF A MODULE HAS AN EXPANSION, THE PAGE NUMBER WHERE THE EXPANSION APPEARS FOLLOWS THE MODULE NAME.

ARGUMENTS:

-------------

OUTCHART = FILE *
PAGE_WIDTH = INT
PAGE_DEPTH = INT

INCLUDE FILES:

-------------

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

-------------

SORT - SORT MODULE NAMES
MALLOC
STRCMP
PUTC
FREE

3-131
FPUTS
MEMSET
STRLEN
MEMCPY
PRINTF
GETTOP - GET TOP OF TREE

CALLED DIRECTLY BY:
------------------
PRNTREE - PRINT TREE

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: DRAWLEV
PURPOSE: DRAW A LEVEL OF THE CHART
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: DRAWLEV
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS

DRAWLEV()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

THIS ROUTINE CONTROLS THE DRAWING OF A LEVEL OF THE CHART. A LEVEL CONSISTS OF MAXLINE LINES:

0 | TOP CONNECTING LINE
MAXLINE-BETWEEN-DEPTH | \\ | > DATA FROM INPUT FILE
| NAME | \\
MAXLINE-BETWEEN+----------+ LINE SPANNING CHILDREN
MAXLINE-1 | | CONNECTING LINES FOR CHILDREN

ARGUMENTS:

--------------
TEMPFILE = FILE *
OUTCHART = FILE *
START_PTR = NODE *
CHARSET = INT
PAGE_WIDTH = INT

3-133
INCLUDE FILES:

- STDTYP  - STANDARD TYPE DEFINITIONS
- STDIO   - **** PURPOSE NOT FOUND BY STRIPPER ****
- CHART   - CHART INCLUDE FILE

ROUTINES CALLED:

- MALLOC
- FSEEK
- GETC
- MEMCPY
- FGETS
- STRLEN
- PUTLIN  - PRINT LEVEL OF TREE
- FREE
- MEMSET
- SPRINTF
- GETTOP  - GET TOP OF TREE

CALLED DIRECTLY BY:

- PRNTREE  - PRINT TREE

USED IN MAIN PROGRAM(S):

- HRW/MAIN  - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: ENDGEN
PURPOSE: END 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:
------
GENERATE THE ENDING CODE WHICH CLOSES THE FILES AND DOES THE
NDML ERROR PROCESSING

ARGUMENTS:
-------
LANG = INT

INCLUDE FILES:
--------
STDTYPE - STANDARD TYPE DEFINITIONS
STDIO  - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD    - FORM PROCESSOR DATA
FPARM  - FORM PROCESSOR PARAMETERS
FP CODE- 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:
-------------------
STD CODE  - STANDARD COBOL CODE

USED IN MAIN PROGRAM(S):
------------------------
GRP/MAIN  - GENERATE APPLICATION/REPORT PROGRAM
NAME: ERROR
PURPOSE: ISSUE ERROR MESSAGE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID
SOURCE FILE: FLUIERR
SOURCE FILE TYPE: .C

DESCRIPTION:
SYNOPSIS
VOID ERROR(S, A, B, C, D, E, F)

DESCRIPTION
PRINTS AN ERROR MESSAGE ON STDERR AND INCREASES THE NUMBER OF ERRORS

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

INCLUDE FILES:
STDTYPE - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:
PMSGLS
STRLEN
PRINTF

CALLED DIRECTLY BY:
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
CHKFLD - CHECK FIELD
CHKFRM - CHECK FORM
ADDCHK - ADD POSITION TO CHECK LIST
YYLEX - LEXICAL ANALYZER FOR FLAN
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):
---------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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

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
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
---------------
STDTYP - 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
REPORT WRITER 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)

DESCRIPTION
PRINTS A FATAL MESSAGE ON STDERR AND EXITS

ARGUMENTS:
-----------
S = CHAR *
A = CHAR *
B = CHAR *
C = CHAR *
D = CHAR *
E = CHAR *
F = CHAR *

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

ROUTINES CALLED:
----------------
SPRINTF
STRLEN
PMSGCLS

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
REPORT WRITER Module Documentation

NAME:        FD
PURPOSE:     FD SECTION DECLARATIONS
LANGUAGE:    C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NDMILGEN
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
REPORT WRITER 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:
-------------

STDYP     - 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

3-144
REPORT WRITER 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
REPORT WRITER 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:
--------------
STDTYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:
-----------------
FLDRSV - FIELD RESOLVE
CTRLRSV - CONTROL RESOLVE
STATRSV - STATISTIC RESOLVE

CALLED DIRECTLY BY:
-------------------
RWOPN - REPORT WRITER OPEN FORMS
FLDRSV - FIELD RESOLVE
USED IN MAIN PROGRAM(S):
---------------------
   GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

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

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

DESCRIPTION
RETURNS A STRING OF THE SPECIFIED FIELD TYPE

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

INCLUDE FILES:
-------------------
STDTYP - STANDARD TYPE DEFINITIONS
STDCIO - **** 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
REPORT WRITER Module Documentation

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

DESCRIPTION:
--------------
SYNOPSIS

ATTMAP *FNDATT(S)
CHAR *S;

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

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

INCLUDE FILES:
---------------

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

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

CALLED DIRECTLY BY:
---------------------

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

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

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-----------------------
STDTYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

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

CALLED DIRECTLY BY:
-----------------------
GENAR - GENERATE ACTION PRESENT
MLPFRM - MAKE A LIST OF PRESENTED FORMS
WINRSV - WINDOW RESOLVE
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
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:

STDTYP - 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

3-152
USED IN MAIN PROGRAM(S):

GRP/MAIN  - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

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

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
INITIAL
MEMCMP
TRMNAT
PMSGLC
INITFP
ADDFRM
GDATA
OISCR
PRINTF

CALLED DIRECTLY BY:
-------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):
-------------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-------------
STDTYP - STANDARD TYPE DEFINITIONS
SDIO - **** 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:
-------------
FPRINTF
CALLED DIRECTLY BY:
----------------------

- GENMAIN - GENERATE MAIN PROGRAM
- GENAAL - GENERATE PROCEDURE "ADDAL" ADD ACTION LIST
- GENAA - GENERATE PROCEDURE "ADDACT" ADD AN ACTION
- GENDOA - GENERATE PROCEDURE "DOACT" DO ACTION
- DCLINDEX - 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
- SELOPEN - 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 NODUPPLICATE 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
REPORT WRITER 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:

STDTYP - 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
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:
-------------------
STDTYPEP - 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
REPORT WRITER 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:
-------------------
STDTYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:
-------------------
GENAAL - GENERATE PROCEDURE "ADDAL" ADD ACTION LST
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
REPORT WRITER 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
REPORT WRITER 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:
----------
STDTYP - 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
USED IN MAIN PROGRAM(S):

GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST

3-164
USED IN MAIN PROGRAM(S):
----------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

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

ROUTINES CALLED:

UQFOR - UNIVERSAL QUALIFIER FOR LOOP
GEN - GENERATE A LINE OF CODE
CFNAP - 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:
-----------------
GENDOA - GENERATE PROCEDURE "DOACT" DO ACTION

USED IN MAIN PROGRAM(S):
-----------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
---------------
STDTYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:
-----------------
GEN - GENERATE A LINE OF CODE
FRMPDAT - FORM PDATA
PRINTF

CALLED DIRECTLY BY:
---------------------
GENAL - GENERATE ACTION LIST
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-------------
STDTYP - 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

3-170
USED IN MAIN PROGRAM(S):
---------------------
GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

STDTYP - 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
HASIT"M - THIS ROUTINE DETERMINES IF THERE IS AN ITEM
WITHIN
STRCMP
FRMPDAT - FORM PDATA
SPRINTF

3-172
CALLED DIRECTLY BY:
---------------------
GENAL       - GENERATE ACTION LIST

USED IN MAIN PROGRAM(S):
------------------------
GRP/MAIN      - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

STDTYP - 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
REPORT WRITER 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:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

GENAL - GENERATE ACTION LIST
USED IN MAIN PROGRAM(S):

---------------------

GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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

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:
---------------
STDTYPE - STANDARD TYPE DEFINITIONS
STDOUT - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:
-----------------
GEN - GENERATE A LINE OF CODE
PRINTF
CALLED DIRECTLY BY:

----------------------
GENMAIN       - GENERATE MAIN PROGRAM

USED IN MAIN PROGRAM(S):
----------------------
GRP/MAIN       - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-------------
STDTYP - STANDARD TYPE DEFINITIONS
STDOUT - **** 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
REPORT WRITER 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: 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:

STDYP - STANDARD TYPE DEFINITIONS
STDB - **** PURPOSE NCT 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
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
REPORT WRITER 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;
   ) DBR&D;  CARRIAGE RETURN PAD.
   }D - NUMBER OF SELECT (0 IS FIRST).

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

3-185
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
REPORT WRITER 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:

STDTPY - 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
REPORT WRITER 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
STDDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:
---------------
ISOPENE - DETERMINE IF THIS FIELD IS OPEN ENDED
HASITEM - THIS ROUTINE DETERMINES IF THERE IS AN ITEM

3-189
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
REPORT WRITER 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:

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
STRLN
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
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];
INSERTED.
CHAR HOSTID[3];
} INSERT1;
ONE PRESENTATION SCHEME
CHAR DUMMY;
NO INSERTS.
} INSERTPS;
ONE PRESENTATION SCHEME
DUMMY FIELD IF THERE ARE
NAME OF INSERT STRUCTURE .
INCLUDE FILES:
---------------
STDTYP - STANDARD TYPE DEFINITIONS
STDDO - **** 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
REPORT WRITER Module Documentation

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

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. GENGCS 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
14. THE STARTUP CONDITION'S ACTIONS ARE ADDED TO THE ACTION LIST.
INCLUDE FILES:

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

ROUTINES CALLED:

GENBEG - GENERATE BEGINNING OF APPLICATION OR REPORT
MKINC - MAKE INCLUDE (ACTUALLY STRUCTURE TAGS ALA MAKINC)
GEN - GENERATE A LINE OF CODE
GENFSD - GENERATE FORM STRUCTURE DATA INITIALIZATION
SPRINTF
GENFS - GENERATE FORM DATA STRUCTURES
GENFP - GENERATE FORM PATH
GENNDP - GENERATE NODUPLICATE DECLARATIONS
GENCHG - GENERATE CHANGE DECLARATIONS
GENDS - GENERATE DATA DATA STRUCTURES
GENDB - GENERATE DATA BASE RECORDS AND FILE DECLARATIONS
GENINS - GENERATE INSERT DECLARATIONS

CALLED DIRECTLY BY:

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: GENNDP
PURPOSE: GENERATE NODUPLICATE 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:

STDTP ' - STANDARD TYPE DEFINITIONS
ST Dio - **** 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
REPORT WRITER 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:

-------------
STDTPY - STANDARD TYPE DEFINITIONS
STDIO  - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD   - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPDPARM - 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
REPORT WRITER 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: RWSP
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
FPFCODE - 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
REPORT WRITER 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:

STDTYP - STANDARD TYPE DEFINITIONS
STDLIO - ***** 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
PRINTF

CALLED DIRECTLY BY:

----------
GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):

----------
GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: GETFIT
PURPOSE: GET SUBTREE THAT FITS ON PAGE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: NODE * ()
SOURCE FILE: GETFIT
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
GETFIT()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE RETURNS A POINTER TO THE LARGEST SUBTREE (UP TO THE ENTIRE TREE) THAT WILL FIT ON A PAGE.

ARGUMENTS:

ARGUMENTS:

NODE_PTR = NODE *
PAGE_WIDTH = INT
PAGE_DEPTH = INT

INCLUDE FILES:

INCLUDE FILES:

STDYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

ROUTINES CALLED:

GETSIZE - GET SUBTREE SIZE
GETFIT - GET SUBTREE THAT FITS ON PAGE
CALLED DIRECTLY BY:
---------------------
GETFIT      - GET SUBTREE THAT FITS ON PAGE
PAGTREE     - PAGE TREE

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN      - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
NAME: GETLOWLEF
PURPOSE: GET LOWER LEFT CHILD NODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: NODE * ()
SOURCE FILE: GETLWLF
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:
-------------
SYNOPSIS
GETLOWLEF()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
GIVEN A NODE, GET THE FIRST NODE ON THE NEXT LOWER LEVEL
THAT IS A CHILD OF THIS NODE OR A CHILD OF SOME NODE RIGHT
OF THIS NODE

ARGUMENTS:
-----------
START_PTR = NODE *

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

CALLED DIRECTLY BY:
---------------------
ARRANGE - ARRANGE CHART AND ASSIGNS PAGE NUMBERS
HBALANC - HORIZONTAL TREE BALANCE
MOVECLD - MOVE CHILD'S POSITION
PAGNODE - PAGE NODES
PRNTREE - PRINT TREE
READTREE - READ DUMPTREE FILE
SPLICE - SPLICE TREE INTO ANOTHER TREE

3-206
USED IN MAIN PROGRAM(S):
-------------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: GETLOWRIT
PURPOSE: GET LOWER RIGHT CHILD NODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: NODE * ()
SOURCE FILE: GETLWRT
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

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

SYNOPSIS
GETLOWRIT()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
GIVEN A NODE, GET THE RIGHT-MOST NODE ON THE NEXT LOWER LEVEL THAT IS A CHILD OF THIS NODE OR A CHILD OF SOME NODE TO THE LEFT OF THIS NODE

ARGUMENTS:
---------
START_PTR = NODE *

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

CALLED DIRECTLY BY:
-------------------
READTREE - READ DUMPTREE FILE
SPLICE - SPLICE TREE INTO ANOTHER TREE
USED IN MAIN PROGRAM(S):

-------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: GETPAR
PURPOSE: GET PARENT NODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: NODE *
SOURCE FILE: GETPAR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

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

SYNOPSIS
GETPAR()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE RETURNS THE SPECIFIED PARENT NODE WHICH IS CREATED IF REQUIRED.

ARGUMENTS:
----------

PARENT NAME = CHAR []
FILEPOS = LONG
WIDTH = INT
DEPTH = INT
TOP_POS = INT
BOT_POS = INT
L_MARGIN = INT

INCLUDE FILES:
---------------

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

3-210
ROUTINES CALLED:
-----------------
  BLDMOD   - BUILD MODULE
  BLDNODE   - BUILD NODE

CALLED DIRECTLY BY:
-------------------
  READTREE   - READ DUMPTREE FILE

USED IN MAIN PROGRAM(S):
------------------------
  HRW/MAIN   - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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
NAME: GETSIZE
PURPOSE: GET SUBTREE SIZE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: GETSIZE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
HEADER()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE RETURNS THE WIDTH AND DEPTH OF A SUB-TREE
ARGUMENTS:

FIRST_PTR = NODE *
WIDTH = INT *
DEPTH = INT *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

NEXTLEV - ADVANCE POINTERS TO NEXT LEVEL OF SUBTREE
MIN
MAX
CALLED DIRECTLY BY:
----------------------
GETFIT       - GET SUBTREE THAT FITS ON PAGE
PAGTREE     - PAGE TREE
PRNTREE     - PRINT TREE

USED IN MAIN PROGRAM(S):
----------------------
HRW/MAIN    - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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:

STDSTYP - 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
MAKWHES/COBWHES - COBOL WHERE ES
SELGEN - SELECT GENERATE
SELWS - SELECT WORKING STORAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: GETTOP
PURPOSE: GET TOP OF TREE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: NODE * ()
SOURCE FILE: GETTOP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
GETTOP()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
ARGUMENTS:

START_PTR = NODE *

INCLUDE FILES:

STDTYPE - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

CALLED DIRECTLY BY:

ARRANGE - ARRANGE CHART AND ASSIGN PAGE NUMBERS
DOINDEX - DO CHART INDEX
DRAWLEV - DRAW A LEVEL OF THE CHART

USED IN MAIN PROGRAM(S):

HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: GETUPLFT
PURPOSE: GET UPPER LEFTMOST NODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: NODE * ()
SOURCE FILE: GETUPLF
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

-------------

SYNOPSIS
GETUPLFT()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
GIVEN A NODE, GET THE NODE ON THE NEXT HIGHER LEVEL
FARthest TO THE LEFT

ARGUMENTS:

-------------
START_PTR = NODE *

INCLUDE FILES:

-------------
STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

CALLED DIRECTLY BY:

------------
HBALANC - HORIZONTAL TREE BALANCE
PAGNODE - PAGE NODES

USED IN MAIN PROGRAM(S):

-------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: GFLDPT
PURPOSE: GET FIELD POINTER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD *
SOURCE FILE: FLANSP
SOURCE FILE TYPE: .C
HOST: UI
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
REPORT WRITER Module Documentation

NAME: GRP/MMAIN
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:

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

3-220
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

FRNTND - FRONT END FOR FORMS
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
REPORT WRITER 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:
---------------
STDTYP - STANDARD TYPE DEFINITIONS
STDLIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION

3-222
ROUTINES CALLED:
-------------
HASDATA    - DETERMINE IF THERE ARE ANY SELECT STATEMENTS

CALLED DIRECTLY BY:
------------------
GENAR       - GENERATE ACTION PRESENT
BLD SUB     - 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
REPORT WRITER 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 *

INCLUD; FILES:
--------------
STDTPY - STANDARD TYPE DEFINITIONS
STDDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPDPPARM - 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:

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
REPORT WRITER 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:

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

3-226
CALLED DIRECTLY BY:

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

USED IN MAIN PROGRAM(S):
-------------------
GRP/MON      - GENERATE APPLICATION/REPORT PROGRAM
NAME: HBALANC
PURPOSE: HORIZONTAL TREE BALANCE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID()
SOURCE FILE: HBALANC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS

HBALANC()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

THIS ROUTINE BALANCES THE TREE. STARTING AT THE BOTTOM LEFT OF THE TREE AND MOVING FIRST RIGHT AND THEN UPWARD, EACH NODE IS POSITIONED TO THE RIGHT OF ITS NEIGHBOR. IF THE CENTER OF ITS CHILDREN IS FURTHER RIGHT, THE NODE IS MOVED RIGHT, OTHERWISE THE CHILDREN ARE MOVED RIGHT.

ARGUMENTS:

FIRST_PTR = NODE *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

GETLOWLEFT - GET LOWER LEFT CHILD NODE
MOVECLD  - MOVE CHILD'S POSITION
MAX
GETUPLFT  - GET UPPER LEFTMOST NODE

CALLED DIRECTLY BY:
---------------------
HRW/MAIN  - MAIN MODULE FOR HIERARCHICAL REPORT WRITER

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN  - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: HRW/MAIN
PURPOSE: MAIN MODULE FOR HIERARCHICAL REPORT WRITER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT()
SOURCE FILE: HRW
SOURCE FILE TYPE: .C
HOST: UI
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
MAIN()

DESCRIPTION
MAIN PROGRAM FOR HIERARCHICAL REPORT WRITER.

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDLIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
CHART - CHART INCLUDE FILE
HRWFHRM - HRW FORM DEFINITION

ROUTINES CALLED:

INITIAL
INITFP
ADDFRM
GDATA
ESCPY
FOPEN
ATOI
MEMCMP
PMSGDLS
PUTATT
PUTCUR
FCLOSE
DELENODE - DELETE A SPECIFIED NODE IN TREE
TERMFP
TRMNAT
READTREE - READ DUMPTREE FILE
REPOS - REPOSITION MODULE EXPANSIONS
MODPAGE - MODIFY PAGES
PAGTREE - PAGE TREE

3-230
ARRANGE - ARRANGE CHART AND ASSIGNS PAGE NUMBERS
HBALANC - HORIZONTAL TREE BALANCE
PRNTREE - PRINT TREE
OUTSCR
OISCR

3-231
REPORT WRITER 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:
---------------

STDYP   - 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:
---------------

MAKWHES/COBWHES - COBOL WHERE ES
MAKWHES         - MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
SELGEN          - SELECT GENERATE
ASSIGN          - ASSIGN FILE SECTION
FD               - FD SECTION DECLARATIONS
CLSFLIL         - CLOSE FILES
ENDGEN          - END GENERATE
PROCGEN         - 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
REPORT WRITER 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:

STDYP = 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
PRINTF
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
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.
TRG PTR - 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 *
TRG PTR = TRGLST *
ACTPTR = ACTLST *

INCLUDE FILES:

-----------
STDTYP  - 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

3-235
CALLED DIRECTLY BY:

ACTRSV  - ACTION RESOLVE

USED IN MAIN PROGRAM(S):

GRP/MAIN  - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

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

ROUTINES CALLED:

---------------
NULBLK   - BLANK FILL A STRING
STRCPY   - WRITE DASH '-'
INDENT   - INDENT A LINE OF GENERATED CODE
FPRINTF  - 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

3-237
REPORT WRITER 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:

STDTYPE - STANDARD TYPE DEFINITIONS
STDOI - **** 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

3-238
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
REPORT WRITER 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 ****
STDYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
MATH - **** PURPOSE NOT FOUND BY STRIPPER ****

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

CALLED DIRECTLY BY:
---------------------
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):
--------------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

3-240
REPORT WRITER 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 *RECCNTPTR;

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
RECCNT_PTR =  INT *

INCLUDE FILES:
-------------
STDTYP   - 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
INSWS - INSERT WORKING STORAGE SECTION

USED IN MAIN PROGRAM(S):

---------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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
REPORT WRITER Module Documentation

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

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

ARGUMENTS:
-----------

\[ M = \text{INT} \]
\[ T = \text{INT} \]

INCLUDE FILES:
----------------

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

STDTYP - 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
REPORT WRITER 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:

STDTYP - STANDARD TYPE DEFINITIONS
STDDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPFARM - FORM PROCESSOR PARAMETERS
FPFCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

NDMILNK - LINKAGE SECTION

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

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

DESCRIPTION:

SYNOPSIS
ENODE *MAKINT(EP)
    ENODE *EP;

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

ARGUMENTS:

---------
EP = ENODE *

INCLUDE FILES:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

--------------
YYPARSE - FLAN PARSER

USED IN MAIN PROGRAM(S):

---------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:      UI
SUBSYSTEM:   RW
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:

---------------
STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPARM - 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
REPORT WRITER 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
STDDO   - **** 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:
-----------------
STRCAT
STRCPY
SPRINTF
ISOPNE  - DETERMINE IF THIS FIELD IS OPEN ENDED

CALLED DIRECTLY BY:
---------------------
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
REPORT WRITER 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:
-------------
STDYP - 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
REPORT WRITER 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
FPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

PSSTRC/INDENT - INDENT
FPINTF
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
REPORT WRITER 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:
---------------
STDTYP - STANDARD TYPE DEFINITIONS
STDDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FP CODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:
-----------------
INDENT - INDENT A LINE OF GENERATED CODE
PRINTF
MAKWHES/CWHES - C WHERE ES
MAKWHES/COBWHES - COBOL WHERE ES

3-254
CALLED DIRECTLY BY:
-------------
SELWS       - SELECT WORKING STORAGE SECTION

USED IN MAIN PROGRAM(S):
-------------
GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:        -------------------
STDYP  -  STANDARD TYPE DEFINITIONS
STDDIO -  **** PURPOSE NOT FOUND BY STRIPPER ****
FPD    -  FORM PROCESSOR DATA
FPARM  -  FORM PROCESSOR PARAMETERS
FPCODE -  FORM PROCESSOR RETURN CODES
RW     -  REPORT WRITER DEFINITIONS

ROUTINES CALLED:      -------------------
MAKWHES/NMPIC - NUMBER PICTURE CLAUSE
PRINTF
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
REPORT WRITER 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:

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

CALLED DIRECTLY BY:

MAKWHES - MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

- STDYP - 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
REPORT WRITER 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:

-------------
STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPFARM - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
ROUNMENTS CALLED:
------------------
STRCHR
MAKQR - MAKE QUALIFIED REFERENCE
PRINTF
STRLN
GEN - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:
-------------------
BSCODE - BUILD SUBROUTINE CODE

USED IN MAIN PROGRAM(S):
-----------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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
{ 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
A SUBFORM OF

3-261
FORM 3.
} ; (* SEL45 *)
DEFINE FRM3
ENDIF

ARGUMENTS:
---------
FP = FIELD *

INCLUDE FILES:
---------
STDYP - 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
REPORT WRITER Module Documentation

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

DESCRIPTION:

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

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

ARGUMENTS:

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

INCLUDE FILES:

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

3-263
ROUTINES CALLED:
---------------
  MYALLOC   - MY MALLOC

CALLED DIRECTLY BY:
-------------------
  YYPARSE     - FLAN PARSER

USED IN MAIN PROGRAM(S):
------------------------
  GRP/MAIN   - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module 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:

STDTYP - 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
REPORT WRITER Module Documentation

NAME: MODPAGE
PURPOSE: MODIFY PAGES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MODPAGE
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

----------------------------------
SYNOPSIS
MODPAGE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE MOVES ANY NODE WHICH IS REFERENCED MORE THAN ONCE TO ITS OWN PAGE.

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

ROUTINES CALLED:
----------------------------------
SPLITNODE - SPLIT A NODE FOR PAGE BREAKS

CALLED DIRECTLY BY:
----------------------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER

USED IN MAIN PROGRAM(S):
----------------------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER

3-266
REPORT WRITER Module Documentation

NAME: MOVCLD
PURPOSE: MOVE CHILDREN
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MOVCLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS

MOVCLD()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

MOVE ALL THE CHILDREN FROM FROM_NODE TO TO_NODE, FOLLOWING ANY EXISTING
TO_NODE CHILDREN

ARGUMENTS:

-----------
FROM_NODE = NODE *
TO_NODE = NODE *

INCLUDE FILES:

-------------
STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

--------------
BLDNODE - BUILD NODE
CLOSEGAP - CLOSE GAP IN TREE
SPLICE - SPLICE TREE INTO ANOTHER TREE
DELNODE - DELETE A SPECIFIED NODE IN TREE
CALLED DIRECTLY BY:
---------------------
REPOS       - REPOSITION MODULE EXPANSIONS

USED IN MAIN PROGRAM(S):
--------------------------
HRW/MAIN       - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
NAME: MOVECLD
PURPOSE: MOVE CHILD'S POSITION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MOVECLD
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW
DESCRIPTION:

SYNOPSIS
MOVECLD()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
INCREMENT THIS CHILD'S POSITION AND EVERYTHING TO THE
RIGHT ALONG THIS LEVEL (INCLUDING THEIR CHILDREN)
ON DOWN AND THIS CHILD'S CHILDREN ON DOWN)

ARGUMENTS:

FIRST_PTR = NODE *
OFFSET = INT

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

GETLOWLEF - GET LOWER LEFT CHILD NODE
CALLED DIRECTLY BY:

-------------------------------
HBALANC - HORIZONTAL TREE BALANCE
PAGTREE - PAGE TREE

USED IN MAIN PROGRAM(S):

-------------------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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:
----------------------
STDYP - STANDARD TYPE DEFINITIONS
STDCIO - **** 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
REPORT WRITER 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:
---------------
STDTYP - STANDARD TYPE DEFINITIONS
STDIO  - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD   - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FFCODE - FORM PROCESSOR RETURN CODES
RW    - REPORT WRITER DEFINITIONS
NTM   - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:
----------------
STRNCOPY
SPKNTF
FOPEN
SYMSG
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
REPORT WRITER 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:

--------------------
STDTYP - STANDARD TYPE DEFINITIONS
STDIO - ***** PURPOSE NOT FOUND BY STRIPPER *****
FPD - FORM PROCESSOR DATA
FPARM - 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
PRINTF

CALLED DIRECTLY BY:

-----------------
PROCGEN - PROCEDURE DIVISION GENERATE

USED IN MAIN PROGRAM(S):

---------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

ARGUMENTS:

LANGUAGE=

LANGUAGE=

LANG = INT
TYPE = CHAR

INCLUDE FILES:

INCLUDE FILES:

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

ROUTINES CALLED:

ROUTINES CALLED:

MAKWH - MAKE WHERE
MAKINS - MAKE INSERT

CALLED DIRECTLY BY:

CALLED DIRECTLY BY:

DATAGEN - DATA DIVISION GENERATE

USED IN MAIN PROGRAM(S):

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: NEXTLEV
PURPOSE: ADVANCE POINTERS TO NEXT LEVEL OF SUBTREE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: NEXTLEV
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

-----------

SYNOPSIS

NEXTLEV()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

ADVANCES L_PTR AND R_PTR TO THE NEXT LEVEL OF A SUBTREE

ARGUMENTS:

-----------

L_PTR = NODE **
R_PTR = NODE **

INCLUDE FILES:

-----------

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

CALLED DIRECTLY BY:

-----------

CLOSEGAP - CLOSE GAP IN TREE
GETSIZE - GET SUBTREE SIZE
REPOS - REPOSITION MODULE EXPANSIONS

USED IN MAIN PROGRAM(S):

-----------

HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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:
---------------
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:
----------------
STRCHR
STRCPY

CALLED DIRECTLY BY:
-------------------
MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
SELEGEN - SELECT GENERATE
SAVEES - SAVE ES INFORMATION
SELEWS - 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

3-278
REPORT WRITER 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
PRINTF
OPNFIL - GENERATE OPEN FILE SECTION

CALLED DIRECTLY BY:

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

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: PAGNODE
PURPOSE: PAGE NODES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PAGNODE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS

PAGNODE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

THIS ROUTINE DIVIDES ANY NODES WHICH ARE TOO BIG TO FIT ON A SINGLE PAGE.

ARGUMENTS:

FIRST_PTR = NODE *
PAGE_WIDTH = INT

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

GETLOWLEF - GET LOWER LEFT CHILD NODE
GETUPLFT - GET UPPER LEFTMOST NODE
COPYNODE - COPY A NODE IN TREE
CALLED DIRECTLY BY:
---------------------
PAGTREE       - PAGE TREE

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN       - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: PAGTREE
PURPOSE: PAGE TREE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PAGTREE
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

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

SYNOPSIS

PAGTREE(

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

THIS ROUTINE DIVIDES THE TREE INTO PAGES.

ARGUMENTS:
-----------

PAGE_WIDTH = INT
PAGE_DEPTH = INT

INCLUDE FILES:
----------------

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

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

PAGNODE - PAGE NODES
GETFIT - GET SUBTREE THAT FITS ON PAGE
SPLITNODE - SPLIT A NODE FOR PAGE BREAKS
GETSIZE - GET SUBTREE SIZE
MOVECLD - MOVE CHILD'S POSITION
CALLED DIRECTLY BY:
-------------------
HRW/MAIN   - MAIN MODULE FOR HIERARCHICAL REPORT WRITER

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN   - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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 = CMDTYPE *

INCLUDE FILES:

STDTYPE - STANDARD TYPE DEFINITIONS
STDO - **** 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
REPORT WRITER Module Documentation

NAME: PRNT
PURPOSE: PRINT MODULE NAMES HIERARCHICALLY
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID()
SOURCE FILE: PRNT
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

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

SYNOPSIS
PRNT()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS IS A ROUTINE TO PRINT MODULE NAMES IN A HIERARCHICAL ORDER.
USEFUL FOR DEBUGGING PURPOSES.

ARGUMENTS:
------------
FIRST_PTR = NODE *

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

ROUTINES CALLED:
-----------------
PRINTF
PRN" - PRINT MODULE NAMES HIERARCHICALLY

CALLED DIRECTLY BY:
---------------------
PRNT - PRINT MODULE NAMES HIERARCHICALLY
REPORT WRITER Module Documentation

NAME: PRNTREE
PURPOSE: PRINT TREE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PRNTREE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
PRNTREE(TOP_MODULE_PTR, TOP_NODE_PTR, OUTCHART, CHARSET,
STRIP,
PAGE_WIDTH, PAGE_DEPTH)

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE PRINTS THE TREE.

ARGUMENTS:

TEMPFILE = FILE *
OUTCHART = FILE *
CHARSET = INT
STRIP = BOOL
PAGE_WIDTH = INT
PAGE_DEPTH = INT

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

FPRINTF
GETSIZE   - GET SUBTREE SIZE
PUTC
GETLOWLEF - GET LOWER LEFT CHILD NODE
STRIPLEV - DRAW STRIP CHART LEVEL
DRAWLEV  - DRAW A LEVEL OF THE CHART
DOINDEX  - DO CHART INDEX
FPUTS
SPRINTF
STRLEN

CALLED DIRECTLY BY:
----------------------
HRW/MAIN   - MAIN MODULE FOR HIERARCHICAL REPORT WRITER

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN   - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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:  
----------------

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:  
------------------

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
PRINTF
PSESMAP

CALLED DIRECTLY BY:  
---------------------

STDCODE - STANDARD COBOL CODE
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

   BLEN
   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

3-291
REPORT WRITER 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:

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

ROUTINES CALLED:

BLEN
FPINTT
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
REPORT WRITER 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
STDO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPARM - FORM PROCESSOR PARAMETERS
FPDRC - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS

ROUTINES CALLED:

PUTC

CALLED DIRECTLY BY:

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

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: PUTLIN
PURPOSE: PRINT LEVEL OF TREE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID()
SOURCE FILE: PUTLIN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
PUTLIN()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
PRINT A FORMATTED LEVEL OF THE TREE

ARGUMENTS:

OUTCHART = FILE *
MAXLINE = INT
LINE = CHAR **
CHARSET = INT

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

STRLEN
STRCHR
PUTC
CALLED DIRECTLY BY:
---------------------
DRAWLEV  - DRAW A LEVEL OF THE CHART
STRIPLEV - DRAW STRIP CHART LEVEL

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN  - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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:
---------------------
STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPL - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPFARM - 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
REPORT WRITER Module Documentation

NAME: READTREE
PURPOSE: READ DUMPTREE FILE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RDTREE
SOURCE FILE TYPE: .C
HOST: SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
READTREE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE READS THE FILE CREATED BY DUMPTREE AND
REBUILDS THE TREE.

ARGUMENTS:

TEMPFILE = FILE *
INVERT = BOOL

INCLUDE FILES:

STDTYPE - STANDARD TYPE DEFINITIONS
STDCPO - **** PURPOSE NOT FOUND BY STRIPPER ****
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

BLDNODE - BUILD NODE
FTELL
GETC
UNGETC
FGETS
STRIEN
GETPAR    - GET PARENT NODE
GETLOWRIT  - GET LOWER RIGHT CHILD NODE
GETLOWLEF  - GET LOWER LEFT CHILD NODE
BLDMOD     - BUILD MODULE

CALLED DIRECTLY BY:
---------------------
HRW/MAIN    - MAIN MODULE FOR HIERARCHICAL REPORT WRITER

USED IN MAIN PROGRAM(S):
---------------------
HRW/MAIN    - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
SYNOPSIS
REPOS()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
REPOSITION MODULE EXPANSIONS TO THE FIRST REFERENCE TO THE MODULE EXPANSION.

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

CLOSEGAP - CLOSE GAP IN TREE
SPlice  - SPLICE TREE INTO ANOTHER TREE
NEXTLEV - ADVANCE POINTERS TO NEXT LEVEL OF SUBTREE
MOVCld - MOVE CHILDREN
DELNODE - DELETE A SPECIFIED NODE IN TREE

CALLED DIRECTLY BY:

HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER

3-300
USED IN MAIN PROGRAM(S):

HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: RSETNDP
PURPOSE: RESET NODUPLICATE 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:
--------------
STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPPLARM - 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
REPORT WRITER 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:

STDTYP - 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 - GENERATE A LINE OF CODE

CALLED DIRECTLY BY:

FRMPDAT - FORM PDATA
RSETSTAT - RESET STATISTIC

USED IN MAIN PROGRAM(S):

GRP/M  - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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: 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 BE CREATING A SPECIAL FIELD TO HOLD THE FIELD'S PROMPTS. USELST MUST BE A POINTER TO A POINTER BECAUSE DELFLD IS USED AND THAT'S WHAT IT NEEDS.

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

INCLUDE FILES:
-------------
STDTYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
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
REPORT WRITER 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:

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

ROUTINES CALLED:

------
RWEXPD - 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

3-307
REPORT WRITER 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:

STDTYP - 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
CALLED DIRECTLY BY:
---------------------
RWEXPD        - REPORT WRITER EXPAND ARRAYS

USED IN MAIN PROGRAM(S):
------------------------
GRP/MAIN        - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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
REPORT WRITER Module Documentation

NAME: SELECT
PURPOSE: GENERATE SELECT CODE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID
SOURCE FILE: NDMLGEN
SOURCE FILE TYPE: .C
HOST: UI
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:

SELGEN - SELECT GENERATE

CALLED DIRECTLY BY:

STDCODE - STANDARD COBOL CODE

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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
PFTR = 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
FP CODE - 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
OPNFILE - GENERATE OPEN FILE SECTION
PRINTF
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
REPORT WRITER 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:

STDTYP - STANDARD TYPE DEFINITIONS
STGIO - **** 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
~TLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

BLEN

CALLED DIRECTLY BY:

FD - FD SECTION DECLARATIONS

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
--------------

STDTYPE - STANDARD TYPE DEFINITIONS
STDOI0 - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FP CODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

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

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

CALLED DIRECTLY BY:
---------------------

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

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

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM

3-315
REPORT WRITER 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
SUBDIRECTORY: 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:

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

ROUTINES CALLED:

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

CALLED DIRECTLY BY:

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

3-316
USED IN MAIN PROGRAM(S):

-------------------

GRP/MAIN  - GENERATE APPLICATION/REPORT PROGRAM
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:

   STDTPY - 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
REPORT WRITER Module Documentation

NAME: SELWHR
PURPOSE: SELECT WHERE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT
SOURCE FILE: GENACT
SOURCE FILE TYPE: .C
HOST: UI
SUBSYSTEM: RW
SUBDIRECTORY: RW/WR
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:

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

ROUTINES CALLED:

SPRINTF
MAKQR - 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
REPORT WRITER 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
FP CODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

---
MAKPS - MAKES THE PRESENTATION SCHEMA RECORD STRUCTURE
NULBLK - BLANK FILL A STRING
STRCPY - WRITE DASH '-'
INDENT - INDENT A LINE OF GENERATED CODE
PRINTF -
MAKWHES - MAKE THE WHERE CLAUSE EXTERNAL SCHEMA VARIABLES
SELWS - SELECT WORKING STORAGE SECTION
MAKES - MAKES THE EXTERNAL SCHEMA RECORD STRUCTURE
GETCOL - GET THE COLUMN NAME OF A TABLE.COLUMN OR COLUMN STRING
GETTBL - GET A TABLE NAME
STRCMP -
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
REPORT WRITER 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:
---------------
STDTYPE - STANDARD TYPE DEFINITIONS
STDDIO - *** 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:

SETNDP - SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED
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
SETNDP - SET NODUPLICATE FIELDS TO BLANK IF THEY ARE DUPLICATED

USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME:                     SORT
PURPOSE:                  SORT MODULE NAMES
LANGUAGE:                 C
MODULE TYPE:              SUBROUTINE
FUNCTION TYPE:            VOID ()
SOURCE FILE:              SORT
SOURCE FILE TYPE:         .C
HOST:                     
SUBSYSTEM:                UI
SUBDIRECTORY:             HRW
DOCUMENTATION GROUP:      HRW

DESCRIPTION:

SYNOPSIS

SORT()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

ARGUMENTS:

MAX_LEN = INT

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

MALLOCC
STRCPYC
STRUPPC
STRCMP

CALLED DIRECTLY BY:

DOINDEX - DO CHART INDEX
USED IN MAIN PROGRAM(S):

HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: SPLICE
PURPOSE: SPLICE TREE INTO ANOTHER TREE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: SPLICE
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

------------

SYNOPSIS
SPLICE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

ARGUMENTS:

----------

FIRST_NODE_PTR = NODE *

INCLUDE FILES:

------------

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

------------

GETLOWLEF - GET LOWER LEFT CHILD NODE
GETLOWRIT - GET LOWER RIGHT CHILD NODE

CALLED DIRECTLY BY:

----------

MOVCLD - MOVE CHILDREN
REPOS - REPOSITION MODULE EXPANSIONS
USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER Module Documentation

NAME: SPLITNODE
PURPOSE: SPLIT A NODE FOR PAGE BREAKS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: SPLNODE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS

SPLITNODE()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION

WHEN A PAGE BREAK OCCURS A DUPLICATE NODE IS CREATED IN ORDER TO BEGIN A NEW PAGE. THE CHILDREN OF THE OLD NODE BECOME THE CHILDREN OF THE NEW NODE

ARGUMENTS:

FIRST_PTR = NODE *
NODE_PTR = NODE *

INCLUDE FILES:

STDTYP - STANDARD TYPE DEFINITIONS
CHART - CHART INCLUDE FILE

ROUTINES CALLED:

BLDNODE - BUILD NODE
CLOSEGAP - CLOSE GAP IN TREE

3-330
CALLED DIRECTLY BY:
---------------------
MODPAGE       - MODIFY PAGES
PAGTREE       - PAGE TREE

USED IN MAIN PROGRAM(S):
------------------------
HRW/MAIN       - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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:

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

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-----------------------
STDTYP - STANDARD TYPE DEFINITIONS
STDOI - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
RW - REPORT WRITER DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
CTCHR - CONTROL CHARACTERS

ROUTINES CALLED:
-----------------------
ENDEGEN - END GERNERATE
SELECT - GENERATE SELECT CODE
SPRINTF
PROCGEN - PROCEDURE DIVISION GENERATE
F!RINTF
DATAGEN - DATA DIVISION GENERATE
STRUPC
INSERT - INSERT PROCEDURE
CALLED DIRECTLY BY:

-------------------
NDMLGEN - NDML COBOL APPLICATION GENERATOR

USED IN MAIN PROGRAM(S):

-------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER Module Documentation

NAME: STRIPLEV
PURPOSE: DRAW STRIP CHART LEVEL
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: STRPLEV
SOURCE FILE TYPE: .C
HOST: 
SUBSYSTEM: UI
SUBDIRECTORY: HRW
DOCUMENTATION GROUP: HRW

DESCRIPTION:

SYNOPSIS
STRIPLEV()

INPUTS/OUTPUTS:

INPUTS:

OUTPUTS:

DESCRIPTION
THIS ROUTINE DRAWS A LEVEL OF A STRIPPED CHART. A LEVEL CONSISTS OF MAXLINE LINES:

0 | TOP CONNECTING LINE
MAXLINE-BETWEEN-DEPTH | \\ 
| NAME > DATA FROM INPUT FILE 
| 
MAXLINE-BETWEEN +----------+ LINE SPANNING CHILDREN
MAXLINE-1 | | | CONNECTING LINES FOR CHILDREN

ARGUMENTS:

TEMPFILE = FILE *
OUTCHART = FILE *
START_PTR = NODE *
 CHARSET = INT
START_POS = INT
PAGE_WIDTH = INT
INCLUDE FILES:

-------------------
STDTYP      - STANDARD TYPE DEFINITIONS
STDDIO      - **** PURPOSE NOT FOUND BY STRIPPER ****
CHART       - CHART INCLUDE FILE

ROUTINES CALLED:

-------------------
MALLOC
MIN
PUTLIN      - PRINT LEVEL OF TREE
FREE
FSEEK
GETC
MEMCPY
FGETS
STRLEN
MEMSET

CALLED DIRECTLY BY:

-------------------
PRNTREE      - PRINT TREE

USED IN MAIN PROGRAM(S):

-------------------
HRW/MAIN      - MAIN MODULE FOR HIERARCHICAL REPORT WRITER
REPORT WRITER 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:

---------
STDYP  - STANDARD TYPE DEFINITIONS
FPD    - FORM PROCESSOR DATA
FPASCADE - 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
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-------------
STDTYP - 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

3-340
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
-----------------
STDYP - 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
---

3-343
REPORT WRITER 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:
--------------
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
FPRIINTF

CALLED DIRECTLY BY:
---------------------
PROCGEN - PROCEDURE DIVISION GENERATE

USED IN MAIN PROGRAM(S):
--------------------------
GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
---------

STDTYPE - STANDARD TYPE DEFINITIONS
STDCIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPFPARM - 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
REPORT WRITER Module Documentation

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

DESCRIPTION:

SYNOPSIS
   VOID WARNING(S, A, B, C, D, E, F)

DESCRIPTION
   PRINTS A WARNING MESSAGE ON STDERR

ARGUMENTS:

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

INCLUDE FILES:

   STDTYP - STANDARD TYPE DEFINITIONS

ROUTINES CALLED:

   PMSGLS
   STRLEN
   SPRINTF

CALLED DIRECTLY BY:

   GENAT - GENERATE ACTION SIGNAL
   CHKFRM - CHECK FORM
   YYLEX - LEXICAL ANALYZER FOR FLAN
   YYFARSE - FLAN PARSER
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
---------
STDTYP - 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
REPORT WRITER 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:

    STDTYPEP - 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:

CHECK FIELD
WRITE EXPRESSION

USED IN MAIN PROGRAM(S):

GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:
------------------
STDYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FFFV2 - FORM FILE FORMAT - VERSION 2

ROUTINES CALLED:
------------------
SPRINTF
OPEN
SYSMSG
WRITE
CLOSE
WRTFRM/WRTTXT - WRITE TEXT
WRTFRM/WRTFLD - WRITE FIELD
WRTFRM/WRTTBF - WRITE TEXT BUFFER
WRTFRM/TBFCLOSE - TEXT BUFFER CLOSE
WRTFRM/WRTDBF - WRITE DEFAULT BUFFER
WRTFRM/DBFCLOSE - DEFAULT BUFFER CLOSE

3-352
STRASN
STRCPY
STRLEN

CALLED DIRECTLY BY:
---------------------
GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM

USED IN MAIN PROGRAM(S):
------------------------
GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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
FP CODE - 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

3-354
REPORT WRITER 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:

STDTPY - 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

3-355
REPORT WRITER Module Documentation

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

DESCRIPTION:
---------------
SYNOPSIS
TBFCLOS(FPTR, I, LINE)
FILE *FPTR;
INT I;
CHAR LINE[];

DESCRIPTION
WRITES THE LAST LINE OF THE TEXT LINE BUFFER.

ARGUMENTS:
-----------
FPTR = FILE *
I = INT
LINE = CHAR []

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

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

ROUTINES CALLED:

FWRITE

CALLED DIRECTLY BY:

WRTFRM - WRITE FORM

3-357
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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:

STDTYPE – STANDARD TYPE DEFINITIONS
STDO – **** PURPOSE NOT FOUND BY STRIPPER ****
FPD – FORM PROCESSOR DATA
FP CODE – FORM PROCESSOR RETURN CODES
FFFV2 – FORM FILE FORMAT – VERSION 2

ROUTINES CALLED:

FWRITE
SCPY
WRTFRM/FORMAT – INSERT FORMAT CODES
SCPY
MEMCPY

CALLED DIRECTLY BY:

WRTFRM – WRITE FORM
USED IN MAIN PROGRAM(S):

GRP/MAIN - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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: FDFF/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
STDOI - **** 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
REPORT WRITER 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:

STDTYPE - 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

3-363
REPORT WRITER 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 ****
STDTYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPParm - FORM PROCESSOR PARAMETERS
RW - REPORT WRITER DEFINITIONS
MATH - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

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

3-364
TOUPPER
ATOI
ISSPACE

CALLED DIRECTLY BY:
-------------------
YYPARSE    - FLAN PARSER

USED IN MAIN PROGRAM(S):
------------------------
GRP/MAIN    - GENERATE APPLICATION/REPORT PROGRAM
REPORT WRITER 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 ****
STDTP - STANDARD TYPE DEFINITIONS
STDI0 - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPARM - 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
STRLFN
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

3-366
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.
REPORT WRITER Include File Description

FILE NAME: CHART
PURPOSE: CHART INCLUDE FILE
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
GLOBAL DECLARATIONS FOR CHART.
DESCRIPTION:

DESCRIPTION
DEFINITIONS OF ALL CONTROL CHARACTERS TO AVOID CHARACTER SET DEPENDENCIES.
REPORT WRITER Include File Description

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

DESCRIPTION:
----------
REPORT WRITER Include File Description

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

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

DESCRIPTION
RECORD LAYOUTS FOR THE BINARY FORM DEFINITION FILE
REPORT WRITER Include File Description

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

DESCRIPTION:

DESCRIPTION
AUXILIARY DATA STRUCTURES USED BY FLAN.
FILE NAME: FPCODE
PURPOSE: FORM PROCESSOR RETURN CODES
LANGUAGE: C

DESCRIPTION:
-----------
REPORT WRITER Include File Description

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

DESCRIPTION:

DATA DEFINITIONS FOR ALL FORM PROCESSOR (INCLUDING MONITOR) DATA.
FILE NAME: FPDINI
PURPOSE: FPD INITIALIZATION
LANGUAGE: C

DESCRIPTION:

INITIALIZED VERSION OF UID FOR INCLUSION IN MAIN PROGRAM.
REPORT WRITER Include File Description

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

DESCRIPTION:

DESCRIPTION: THESE DATA DEFINITIONS ARE USED IN THE FORM PROCESSOR ROUTINES.
REPORT WRITER Include File Description

FILE NAME: HRWFRM
PURPOSE:  HRW FORM DEFINITION
LANGUAGE:  C

DESCRIPTION:
-----------
REPORT WRITER Include File Description

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

DESCRIPTION:

DESCRIPTION
INCLUDE FILE FOR NTM INTERFACE
REPORT WRITER Include File Description

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

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

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)

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 WHTHST-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
REPORT WRITER Include File Description

FILE NAME: STDTYP
PURPOSE: STANDARD TYPE DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
THIS FILE ENSURES THAT THE FOLLOWING STANDARD TYPES ARE AVAILABLE:
FLOAT - SINGLE PRECISION FLOAT
DOUBLE - DOUBLE PRECISION FLOAT
LONG - 32 BIT (OR LARGER) SIGNED INTEGER
LBITS - 32 BITS (OR MORE) FOR BIT MANIPULATION
INT - NATURAL SIZE SIGNED INTEGER
UNSIGNED - NATURAL SIZE UNSIGNED INTEGER
BOOL - NATURAL SIZE LOGICAL (ZERO / NON-ZERO ONLY)
SHORT - 16 BIT (OR LARGER) SIGNED INTEGER
USHORT - 16 BIT (OR LARGER) UNSIGNED INTEGER
BITS - 16 BITS (OR MORE) FOR BIT MANIPULATION
CHAR - SINGLE MACHINE CHARACTER (REAL CHARACTERS ALWAYS POSITIVE)
TINY - 8 BIT (OR LARGER) SIGNED INTEGER
UTINY - 8 BIT (OR LARGER) UNSIGNED INTEGER
TBITS - 8 BITS (OR MORE) FOR BIT MANIPULATION
TBOOL - 8 BIT (OR LARGER) LOGICAL (ZERO / NON-ZERO ONLY)
METACHAR - 16 BIT (OR LARGER) AUGMENTED CHARACTER (SIGNED)
VOID - FUNCTION THAT RETURNS NO VALUE
FORTRAN - STORAGE CLASS FOR FOREIGN (NON-C) ROUTINES OR C ROUTINES WHICH ARE CALLABLE FROM FOREIGN ROUTINES

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

IN ADDITION, THE FOLLOWING UTILITY MACROS ARE DEFINED:
LURSHIFT(N, B) - UNSIGNED LONG RIGHT SHIFT
MAX(A, B) - MAXIMUM OF A AND B
MIN(A, B) - MINIMUM OF A AND B
REPORT WRITER 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 preceding section. 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.
PS 620344501
30 September 1990

15

<table>
<thead>
<tr>
<th>YYPARSE</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>(CONT)</th>
<th>STRCMP</th>
<th>STRLEN</th>
<th>WARNING</th>
<th>(CONT)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PMSGLS</th>
<th>STRLEN</th>
<th>SPRINTF</th>
</tr>
</thead>
</table>

16

<table>
<thead>
<tr>
<th>WRTFRM</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>(CONT)</th>
<th>WRTFRM/WRTTXT</th>
<th>WRTFRM/WRTFLD</th>
<th>(CONT)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>WRITE</th>
<th>STRLEN</th>
</tr>
</thead>
</table>

17

<table>
<thead>
<tr>
<th>RWEXPD</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>COPFLD</th>
<th>ABS</th>
<th>FNDATT</th>
<th>STRASN</th>
<th>RWSP/FIXFRM</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>RWEXPD</th>
<th>COPFLD</th>
<th>FNDATT</th>
<th>STRCMP</th>
</tr>
</thead>
</table>

3-390
PS 620344501
30 September 1990

80

\begin{verbatim}
PAGNODE

GETLOWLEF

GETUPLFT

COPYNODE

BLDMOD

BDNODE

CLOSEGAP

PRNTREE

PRINTF

GETSIZE

PUTC

GETLOWLEF

(CONT)

CHKFLD

(CONT)

WRTEXP

BLEN

MEMCPY

SYSMSG

(CONT)
\end{verbatim}
PS 620344501
30 September 1990

125

| MAKES |
+-------+
| (CONT) | COBES | CCNV | COBCONV | STRCPY | (CONT) |
+--------+-------+-----+---------+--------+-------+

| ATOI | ESCPY | STRNCMP | FPRINTF | MAKES/INDENT |
+-------+-------+--------+---------+-------------+

126

| SAVEES |
+-------+
| (CONT) | NULBLK | STRNCMP | STRLEN | ESCPY | ATOI |
+--------+--------+---------+--------+-------+--------+

127

| MAKWHES/COBWHES |
+------------------+
| (CONT) | GETTBL | GETCOL |
+--------+--------+--------+

3-427
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>620</td>
</tr>
<tr>
<td>ACTRSV</td>
<td>30</td>
</tr>
<tr>
<td>ADDCHK</td>
<td>43</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>41</td>
</tr>
<tr>
<td>ARRANGE</td>
<td>94</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>42</td>
</tr>
<tr>
<td>ATOF</td>
<td>106</td>
</tr>
<tr>
<td>ATOI</td>
<td>30</td>
</tr>
<tr>
<td>BLDMOD</td>
<td>20</td>
</tr>
<tr>
<td>BLDNODE</td>
<td>23</td>
</tr>
<tr>
<td>BLDSUB</td>
<td>39</td>
</tr>
<tr>
<td>BLEN</td>
<td>69</td>
</tr>
<tr>
<td>BSCODE</td>
<td>69</td>
</tr>
<tr>
<td>CALCSTAT</td>
<td>69</td>
</tr>
<tr>
<td>CALLOC</td>
<td>125</td>
</tr>
<tr>
<td>CCONV</td>
<td>120</td>
</tr>
<tr>
<td>CDMESQY</td>
<td>120</td>
</tr>
<tr>
<td>CES</td>
<td>120</td>
</tr>
<tr>
<td>CESPS</td>
<td>120</td>
</tr>
<tr>
<td>CHKARY</td>
<td>125</td>
</tr>
<tr>
<td>CHKFLD</td>
<td>120</td>
</tr>
<tr>
<td>CHKFRM</td>
<td>120</td>
</tr>
<tr>
<td>CHKGRP</td>
<td>120</td>
</tr>
<tr>
<td>CHKSIZE</td>
<td>120</td>
</tr>
<tr>
<td>CLOSEGAP</td>
<td>120</td>
</tr>
<tr>
<td>CLRNDP</td>
<td>120</td>
</tr>
<tr>
<td>CLSFIL</td>
<td>120</td>
</tr>
<tr>
<td>COBCONV</td>
<td>120</td>
</tr>
<tr>
<td>COBES</td>
<td>120</td>
</tr>
<tr>
<td>COBESPS</td>
<td>120</td>
</tr>
<tr>
<td>COBPE</td>
<td>120</td>
</tr>
<tr>
<td>COPFLD</td>
<td>120</td>
</tr>
<tr>
<td>COPYNODE</td>
<td>120</td>
</tr>
<tr>
<td>CPE</td>
<td>120</td>
</tr>
<tr>
<td>CSTASH</td>
<td>120</td>
</tr>
<tr>
<td>CTLRSV</td>
<td>120</td>
</tr>
<tr>
<td>DASH</td>
<td>120</td>
</tr>
<tr>
<td>DATAGEN</td>
<td>120</td>
</tr>
<tr>
<td>DBFREAD</td>
<td>120</td>
</tr>
<tr>
<td>DCLINDX</td>
<td>120</td>
</tr>
<tr>
<td>DCLFLD</td>
<td>120</td>
</tr>
<tr>
<td>DELFLD</td>
<td>120</td>
</tr>
<tr>
<td>DELNODE</td>
<td>120</td>
</tr>
<tr>
<td>DOINDEX</td>
<td>120</td>
</tr>
<tr>
<td>DRAWLEV</td>
<td>120</td>
</tr>
<tr>
<td>ENDEX</td>
<td>120</td>
</tr>
<tr>
<td>ERROR</td>
<td>120</td>
</tr>
<tr>
<td>ERRPRO</td>
<td>120</td>
</tr>
<tr>
<td>ESCPY</td>
<td>120</td>
</tr>
<tr>
<td>ESPSMAP</td>
<td>120</td>
</tr>
<tr>
<td>ESPSMAP/INDENT</td>
<td>120</td>
</tr>
<tr>
<td>FATAL</td>
<td>120</td>
</tr>
<tr>
<td>FCLOSE</td>
<td>120</td>
</tr>
<tr>
<td>FD</td>
<td>120</td>
</tr>
<tr>
<td>FGGETS</td>
<td>120</td>
</tr>
<tr>
<td>FILELNK</td>
<td>120</td>
</tr>
<tr>
<td>FLANCI</td>
<td>120</td>
</tr>
<tr>
<td>FLDRSV</td>
<td>120</td>
</tr>
<tr>
<td>FLTYP</td>
<td>120</td>
</tr>
<tr>
<td>FNDATT</td>
<td>120</td>
</tr>
<tr>
<td>FNDFRM</td>
<td>120</td>
</tr>
<tr>
<td>FOPEN</td>
<td>120</td>
</tr>
<tr>
<td>FPRINTF</td>
<td>120</td>
</tr>
<tr>
<td>FPUTS</td>
<td>120</td>
</tr>
<tr>
<td>FREE</td>
<td>120</td>
</tr>
<tr>
<td>FRMPD</td>
<td>120</td>
</tr>
<tr>
<td>FRNNTND</td>
<td>120</td>
</tr>
<tr>
<td>FSEEK</td>
<td>120</td>
</tr>
<tr>
<td>FTELL</td>
<td>120</td>
</tr>
<tr>
<td>FWRITE</td>
<td>120</td>
</tr>
<tr>
<td>GDATA</td>
<td>120</td>
</tr>
<tr>
<td>GEN</td>
<td>120</td>
</tr>
<tr>
<td>GENAA</td>
<td>120</td>
</tr>
<tr>
<td>GENAAL</td>
<td>120</td>
</tr>
<tr>
<td>GENACT</td>
<td>120</td>
</tr>
<tr>
<td>GENAE</td>
<td>120</td>
</tr>
<tr>
<td>GENAH</td>
<td>120</td>
</tr>
<tr>
<td>GENAI</td>
<td>120</td>
</tr>
<tr>
<td>GENAL</td>
<td>120</td>
</tr>
<tr>
<td>GENAP</td>
<td>120</td>
</tr>
<tr>
<td>GENAQ</td>
<td>120</td>
</tr>
<tr>
<td>GENAR</td>
<td>120</td>
</tr>
<tr>
<td>GENAS</td>
<td>120</td>
</tr>
<tr>
<td>GENAT</td>
<td>120</td>
</tr>
<tr>
<td>GENBEG</td>
<td>120</td>
</tr>
<tr>
<td>GENCHG</td>
<td>120</td>
</tr>
<tr>
<td>GENDB</td>
<td>120</td>
</tr>
<tr>
<td>GENDOA</td>
<td>120</td>
</tr>
<tr>
<td>GENDS</td>
<td>120</td>
</tr>
<tr>
<td>GENFP</td>
<td>120</td>
</tr>
<tr>
<td>GENFS</td>
<td>120</td>
</tr>
<tr>
<td>GENFSD</td>
<td>120</td>
</tr>
<tr>
<td>GENINS</td>
<td>120</td>
</tr>
<tr>
<td>GENMAIN</td>
<td>120</td>
</tr>
<tr>
<td>GENNDP</td>
<td>120</td>
</tr>
</tbody>
</table>

3-430
GENPAG ............ 13  MAKWHES/NUMPIC ... 122
GETC ............... 92  MAKWHES/NUMPIC ...
GETCOL ............  5  MAPDB ...............  51
GETFILE ............ 61  MAP
GETFIT ............ 61  MEMCMP
GETLOWLEF ..........  MEMCPY
GETLOWRIT ........  MEMSET
GETPAR ............ 59  MIN
GETPTH ............ 19  MKINC .............. 32
GETSIZE ............ 61  MKPOS ............  27
GETTBL ............ 88  MLPPFRM .......... 18
GETTOP ............ 13  MODPAGE .......... 41
GETUPLFT ..........  5  MAPDB ..............  51
GFLDT ............ 64  MAKELOC ............  61
GRP/MAIN ...........  2  MOVECLD ..........  60
HASDATA ............ 23  MOVECLD ..........  61
HASITEM ............ 32  MYALLOC ..........  10
HASLOWER .......... 62  NDMLGEN ..........  37
HBALANC ............ 62  NDMLLAB ..........  77
HRW/Main ..........  3  NDMLLNK ..........  93
INDENT ............ 75  NEXLEVEL
INITIAL ............ 10  NULBLK ............  76
INITFP ............. 92  OISCR
INSERT ............. 92  OPNFILE ..........  89
INSMAP .............  6  OUTSCR
INSRSV ............. 21  PAGNODE ..........  80
INSWS ............. 105  PAGTREE ..........  61
ISALNUM .......... 105  PEMAP ..........  76
ISALPHA .......... 105  MSGLC
ISDIGIT .......... 105  MSGLS
ISOPNE .......... 103  PRINTF
ISSPACE ........... 103  PRNT ............  1
MAKACT ............ 10  PRNTREE ..........  81
MAKES ............. 120  PROCGEN ..........  77
MAKES/CNUMPIC ... 129  PSSTRC/COSUB ... 103
MAKES/INDENT ... 120  PSSTRC/CSUB ... 102
MAKES/NUMPIC ... 128  PSSTRC/INDENT ... 93
MAKFLD ........... 45  PTHPTR
MAKINS ............ 101  PUTATT
MAKINT ............ 10  PUTC
MAKPS ............ 112  PUTCUR
MAKQR ............ 45  PUTLIN .......... 107
MAKSTR ............ 64  READDB ..........  35
MAKWH ............ 93  READTREE ..........  39
MAKWHES .......... 113  REPOS ..........  40
MAKWHES/COBWHES ..122  RSETNDP ..........  50
MAKWHES/CWHES ... 122  RSETSTAT ..........  54
RWEXPD ............ 17  WRTFRM/TBFCLOS .... 29
RWOPN ............... 12  WRTFRM/WRTDBF .... 44
RWSP/FIXFRM ......... 17  WRTFRM/WRTFLD .... 28
SAVEES .............. 126  WRTFRM/WRTTBF .... 29
SELECT .............. 76  YYERROR
SELENE .............. 106  YYLEX .............. 98
SELMAP ............... 91  YYPARSE ............  6
SELOPN ............... 71  ...
SELRSV ............... 31  ...
SELWHR ............... 56  ...
SELWS .............. 104  ...
SETNPD ............... 52  ...
SORT ................. 117  ...
SPLICE ............... 40  ...
SPLITNODE ........... 41  ...
SPRINTF
STATRSV ............. 20  ...
STDCODE .............. 58  ...
STRAIN
STRCAT
STRCATR
STRCMP
STRCPY
STRIPELV .......... 107  ...
STRLLEN
STRNCMP
STRNCPY
STRUPC
SYSMSG
TERMFP
TOUPPER
TRGRSV .............. 21  ...
TRMNAT
TRANNDML
UNGETC
UQFOR ............... 36  ...
UQPETH ............... 30  ...
USING ............... 77  ...
VISITAF ............. 68  ...
WARNING ............. 15  ...
WINRSV ............... 19  ...
WRTEXP ............... 96  ...
WRTFRM ............... 11  ...
WRTFRM/DBFCLOS .... 44  ...
WRTFRM/FORMAT

3-432
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."