INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 34 - Application Interface Product Specification

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

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

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

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, 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

DATE
25 July 91

FOR THE COMMANDER:

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

DATE
25 July 91

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.
This specification establishes the detailed design of the Application Interface computer program.

**BLOCK 11:**

**INTEGRATED INFORMATION SUPPORT SYSTEM**

**Vol VIII - User Interface Subsystem**

**Part 34 - Application Interface Product Specification**
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>Description</th>
<th>Pages</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>3.2</td>
<td>Functional Flow</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3</td>
<td>Interfaces</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Application Programs</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Network Transaction Manager</td>
<td>3-2</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Form Processor</td>
<td>3-3</td>
</tr>
<tr>
<td>3.4</td>
<td>Program Interrupts</td>
<td>3-3</td>
</tr>
<tr>
<td>3.5</td>
<td>Timing and Sequencing Description</td>
<td>3-3</td>
</tr>
<tr>
<td>3.6</td>
<td>Special Control Features</td>
<td>3-3</td>
</tr>
<tr>
<td>3.7</td>
<td>Storage Allocation</td>
<td>3-3</td>
</tr>
<tr>
<td>3.8</td>
<td>Object Code Creation</td>
<td>3-3</td>
</tr>
<tr>
<td>3.9</td>
<td>Adaptation Data</td>
<td>3-3</td>
</tr>
<tr>
<td>3.10</td>
<td>Detailed Design Description</td>
<td>3-3</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Main Program List</td>
<td>3-3</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Module List</td>
<td>3-7</td>
</tr>
<tr>
<td>3.10.3</td>
<td>External Routines List</td>
<td>3-10</td>
</tr>
<tr>
<td>3.10.4</td>
<td>Include File List</td>
<td>3-12</td>
</tr>
<tr>
<td>3.10.5</td>
<td>Where Include File Used List</td>
<td>3-14</td>
</tr>
<tr>
<td>3.10.6</td>
<td>Where External Routine Used List</td>
<td>3-19</td>
</tr>
<tr>
<td>3.10.7</td>
<td>Main Program Parts List</td>
<td>3-22</td>
</tr>
<tr>
<td>3.10.8</td>
<td>Module Documentation</td>
<td>3-51</td>
</tr>
<tr>
<td>3.10.9</td>
<td>Include File Description</td>
<td>3-87</td>
</tr>
<tr>
<td>3.10.10</td>
<td>Hierarchy Chart</td>
<td>3-96</td>
</tr>
<tr>
<td>3.11</td>
<td>Program Listings Comments</td>
<td>3-107</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>A</td>
<td>APPENDIX A</td>
<td>A-1</td>
</tr>
</tbody>
</table>

APPENDIX A  FP/AI MESSAGE FORMATS  A-1
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1</td>
<td>AI Interfaces</td>
<td>3-1</td>
</tr>
<tr>
<td>3-2</td>
<td>AI Data Flow</td>
<td>3-2</td>
</tr>
</tbody>
</table>
SECTION 1
SCAPE

1.1 Identification

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

1.2 Functional Summary

The AI is a collection of procedures that may be linked with an application to enable it to use the Form Processor (FP) and run in the distributed IISS environment. The AI does this by sending/receiving FP requests through the NTM (Network Transaction Manager) to/from the User Interface Monitor (UIM) of the Form Processor.
SECTION 2
DOCUMENTS

2.1 Reference Documents


2.2 Terms and Abbreviations

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

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

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.

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.

Buffer Name: the default file in which the buffer will be saved if no file is given on a save command.

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.

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

Current Cursor Position: the position of the cursor before an edit command or function is issued in the text editor.

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

Cut and Paste Buffer: where deleted lines go and the paste and fill edit commands get their data.
**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.

**Display Start Line**: the first line in the buffer to be displayed.

**Display Size**: the number of lines used in the edit area.

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

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

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

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

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

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

**Item**: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.
Logical Device: a conceptual device which to an application is indistinguishable from a physical device and is then mapped to part or all of a physical device.

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

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

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

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

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

Physical Device: a hardware terminal.

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

Previous Cursor Position: the position of the cursor when the previous edit command was issued.

Previous Edit Command: the function key pressed before the current one.

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

Report Definition Language: 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.

Select Line: one terminus of the select range.
Select Mode: when on, certain commands will be executed over the lines in the selected range. The commands are <DELETE LINE> and replace.

Subform: a form that is used within another form.

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

Top of file: the first line of the buffer.

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

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

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

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

Virtual Terminal Interface Field Map: defines the complete terminal screen by breaking it into pieces of the various forms and items that are displayed. Each area of the terminal screen must be defined as belonging to a particular field in the display list.

Window Manager: a facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.
SECTION 3
REQUIREMENTS

3.1 Structural Description

Applications use the AI just as though they were using the User Interface Form Processor. The Application Interface routines have the same calling sequence as the FP routines, but instead of processing the commands the AI creates messages which are sent to the Form Processor by way of the Network Transaction Manager. The Form Processor then processes the command which is contained in the message. This structure allows the application program to run on a machine other than the host of the User Interface. The detailed structure of the Application Interface is illustrated in section 3.10. Figure 3-1 illustrates the relationship between the AI, NTM, FP and an application.

```
+-------------------+
| application       |
+-------------------+
     |               |
     v               |
+-------------------+
| AI                |
+-------------------+
     |               |
     v               |
+-------------------+
| messages          |
+-------------------+
     |               |
     v               |
+-------------------+
| NTM               |
+-------------------+
     |               |
     v               |
+-------------------+
| messages          |
+-------------------+
     |               |
     v               |
+-------------------+
| UIM/FP            |
+-------------------+

Figure 3-1 AI Interfaces
3.2 Functional Flow

Figure 3-2 is a data flow diagram of the Application Interface.

![Data Flow Diagram of Application Interface](image)

**Figure 3-2 Application Interface Data Flow**

3.3 Interfaces

3.3.1 Application Programs

The interface to the application is identical to that of the Form Processor procedures and is documented in the FP User Manual.

3.3.2 Network Transaction Manager

The AI sends and receives messages to and from the UIM via the NTM. Each FP procedure is identified by a unique number in this message. The remainder of the message consists of FP input parameters for an AI send and FP output parameters for an AI receive. These message formats are documented in Appendix A.
3.3.3 Form Processor

The UIM of the Form Processor receives messages from the NTM, calls the appropriate FP procedure and sends the results back to the AI via the NTM.

3.4 Program Interrupts

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

3.5 Timing and Sequencing Description

The Application Interface control logic is simple. First, the AI receives input parameters from an application and sends the parameters to the FP via the NTM. Then the AI receives output parameters from the FP via the NTM and returns them to the application. This is illustrated in the flow diagram in section 3.2.

3.6 Special Control Features

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

3.7 Storage Allocation

This section does not apply to the AI.

3.8 Object Code Creation

The AI routines were compiled with an ANSI COBOL compiler under VAX/VMS. The source is portable to other compilers on machines such as the IBM.

3.9 Adaptation Data

The AI source is portable to other ANSI COBOL compilers.

3.10 Detailed Design Description

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it
is truly a main program entry point. If not, then it is merely called by other programs not being documented here.
<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDELM</td>
<td>ADD ELEMENT</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>ADD FORM</td>
</tr>
<tr>
<td>CHGLDV</td>
<td>CHANGE LOGICAL DEVICE</td>
</tr>
<tr>
<td>CLSFRM</td>
<td>CLOSE FORM</td>
</tr>
<tr>
<td>CLSLDV</td>
<td>CLOSE LOGICAL DEVICE</td>
</tr>
<tr>
<td>GDATA</td>
<td>GET DATA</td>
</tr>
<tr>
<td>GETATT</td>
<td>GET ATTRIBUTES</td>
</tr>
<tr>
<td>GETBAK</td>
<td>GET BACKGROUND</td>
</tr>
<tr>
<td>GETCUR</td>
<td>GET CURSOR</td>
</tr>
<tr>
<td>GPAGE</td>
<td>GET PAGE</td>
</tr>
<tr>
<td>GWINDO</td>
<td>GET WINDOW</td>
</tr>
<tr>
<td>INITFP</td>
<td>INITIALize Form Processor</td>
</tr>
<tr>
<td>INITVT</td>
<td>INITIAL VTI</td>
</tr>
<tr>
<td>INQLDV</td>
<td>INQUIRE LOGICAL DEVICE</td>
</tr>
<tr>
<td>OISCR</td>
<td>OUTPUT / INPUT SCREEN</td>
</tr>
<tr>
<td>OPNFRM</td>
<td>OPEN FORM</td>
</tr>
<tr>
<td>OPNLVDV</td>
<td>OPEN LOGICAL DEVICE</td>
</tr>
<tr>
<td>OUTSCR</td>
<td>OUTPUT SCREEN</td>
</tr>
<tr>
<td>PARFQN</td>
<td>PARSE FULLY QUALIFIED NAME</td>
</tr>
<tr>
<td>PDATA</td>
<td>PUT DATA</td>
</tr>
<tr>
<td>PMSGLC</td>
<td>PUT MESSAGE LINE CODE</td>
</tr>
<tr>
<td>Module Name</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>PMSGLS</td>
<td>PUT MESSAGE LINE STRING</td>
</tr>
<tr>
<td>PUTATT</td>
<td>PUT ATTRIBUTES</td>
</tr>
<tr>
<td>PUTBAK</td>
<td>PUT BACKGROUND</td>
</tr>
<tr>
<td>PUTCUR</td>
<td>PUT CURSOR</td>
</tr>
<tr>
<td>PUTLOC</td>
<td>PUT CURSOR LOCATION</td>
</tr>
<tr>
<td>RMVPAG</td>
<td>REMOVE PAGE</td>
</tr>
<tr>
<td>RPLFRM</td>
<td>REPLACE FORM</td>
</tr>
<tr>
<td>TERMFP</td>
<td>EXIT FORM PROCESSOR</td>
</tr>
<tr>
<td>TERMVT</td>
<td>TERMINATE VTI</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.
### APPLICATION INTERFACE Module List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDELM</td>
<td>ADD ELEMENT</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>ADD FORM</td>
</tr>
<tr>
<td>CHGLDV</td>
<td>CHANGE LOGICAL DEVICE</td>
</tr>
<tr>
<td>CLSFRM</td>
<td>CLOSE FORM</td>
</tr>
<tr>
<td>CLSLDV</td>
<td>CLOSE LOGICAL DEVICE</td>
</tr>
<tr>
<td>GDATA</td>
<td>GET DATA</td>
</tr>
<tr>
<td>GDATLN</td>
<td>GET DATA LENGTH</td>
</tr>
<tr>
<td>GETATT</td>
<td>GET ATTRIBUTES</td>
</tr>
<tr>
<td>GETBAK</td>
<td>GET BACKGROUND</td>
</tr>
<tr>
<td>GETCUR</td>
<td>GET CURSOR</td>
</tr>
<tr>
<td>GETUIM</td>
<td>GET USER INTERFACE MONITOR AP</td>
</tr>
<tr>
<td>GPAGE</td>
<td>GET PAGE</td>
</tr>
<tr>
<td>GWINDO</td>
<td>GET WINDOW</td>
</tr>
<tr>
<td>INITFP</td>
<td>INITIALize Form Processor</td>
</tr>
<tr>
<td>INITVT</td>
<td>INITIAL VTI</td>
</tr>
<tr>
<td>INQLDV</td>
<td>INQUIRE LOGICAL DEVICE</td>
</tr>
<tr>
<td>OISCR</td>
<td>OUTPUT / INPUT SCREEN</td>
</tr>
<tr>
<td>OPNFRM</td>
<td>OPEN FORM</td>
</tr>
<tr>
<td>OPNLDV</td>
<td>OPEN LOGICAL DEVICE</td>
</tr>
<tr>
<td>OUTSCR</td>
<td>OUTPUT SCREEN</td>
</tr>
<tr>
<td>PARFQN</td>
<td>PARSE FULLY QUALIFIED NAME</td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE Module List

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDATA</td>
<td>PUT DATA</td>
</tr>
<tr>
<td>PMSGLC</td>
<td>PUT MESSAGE LINE CODE</td>
</tr>
<tr>
<td>PMSGLS</td>
<td>PUT MESSAGE LINE STRING</td>
</tr>
<tr>
<td>PUTATT</td>
<td>PUT ATTRIBUTES</td>
</tr>
<tr>
<td>PUTBAK</td>
<td>PUT BACKGROUND</td>
</tr>
<tr>
<td>PUTCUR</td>
<td>PUT CURSOR</td>
</tr>
<tr>
<td>PUTLOC</td>
<td>PUT CURSOR LOCATION</td>
</tr>
<tr>
<td>RMVPAG</td>
<td>REMOVE PAGE</td>
</tr>
<tr>
<td>RPLFRM</td>
<td>REPLACE FORM</td>
</tr>
<tr>
<td>TERMFP</td>
<td>EXIT FORM PROCESSOR</td>
</tr>
<tr>
<td>TERMVT</td>
<td>TERMINATE VTI</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.
<table>
<thead>
<tr>
<th>Module Name</th>
<th>First User</th>
</tr>
</thead>
<tbody>
<tr>
<td>APACCT</td>
<td>GETUIM</td>
</tr>
<tr>
<td>MEMCPY</td>
<td>GETUIM</td>
</tr>
<tr>
<td>NSEND</td>
<td>PUTLOC</td>
</tr>
<tr>
<td>RCV</td>
<td>TERMVT</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.
## APPLICATION INTERFACE Include File List

<table>
<thead>
<tr>
<th>File Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPCODE</td>
<td>Form Processor return codes</td>
</tr>
<tr>
<td>NAPIEVB</td>
<td>CORRECTED C VERSION OF APIEVB.INC</td>
</tr>
<tr>
<td>NAPINME</td>
<td>CORRECTED C VERSION OF APINME.INC</td>
</tr>
<tr>
<td>NBUFAPI</td>
<td>CORRECTED C VERSION OF BUFAPI.INC</td>
</tr>
<tr>
<td>ROUTID</td>
<td>ROUTine ID definitions</td>
</tr>
<tr>
<td>SRVRET</td>
<td>AS THE RETURN GIVEN A TABLE-FULL ERROR</td>
</tr>
<tr>
<td>STD Typ</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.
APPLICATION INTERFACE Where-include-file-used List

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

**FP CODE**

- ADDELML ADD ELEMENT
- ADDFRM ADD FORM
- CHGLDV CHANGE LOGICAL DEVICE
- CLSFRM CLOSE FORM
- CLSLDV CLOSE LOGICAL DEVICE
- GDAT DATA GET DATA
- GDATLN GET DATA LENGTH
- GETATT GET ATTRIBUTES
- GETBAK GET BACKGROUND
- GETCUR GET CURSOR
- GPAGE GET PAGE
- GWINDO GET WINDOW
- INITVT INITIAL VTI
- INQLDV INQUIRE LOGICAL DEVICE
- OISCR OUTPUT / INPUT SCREEN
- OPNFRM OPEN FORM
- OPNLDV OPEN LOGICAL DEVICE
- OUTSCR OUTPUT SCREEN
- PARFQN PARSE FULLY QUALIFIED NAME
- PDATA PUT DATA
- PMSCLC PUT MESSAGE LINE CODE
- PMSGLS PUT MESSAGE LINE STRING
- PUTATT PUT ATTRIBUTES
- PUTBAK PUT BACKGROUND
- PUTCUR PUT CURSOR
- PUTLOC PUT CURSOR LOCATION
- RMVPAG REMOVE PAGE
- RPLFRM REPLACE FORM
- TERMVT TERMINATE VTI

**NAPIEVB**

- GETUIM GET USER INTERFACE MONITOR AP

---

3-15
APPLICATION INTERFACE 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>NAPINME</td>
<td>GETUIM</td>
<td>GET USER INTERFACE MONITOR AP</td>
</tr>
<tr>
<td>NBUFAPI</td>
<td>GETJIM</td>
<td>GET USER INTERFACE MONITOR AP</td>
</tr>
</tbody>
</table>

ROUTID

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDELM</td>
<td>ADD ELEMENT</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>ADD FORM</td>
</tr>
<tr>
<td>CHGLDV</td>
<td>CHANGE LOGICAL DEVICE</td>
</tr>
<tr>
<td>CLSFRM</td>
<td>CLOSE FORM</td>
</tr>
<tr>
<td>CLSLDV</td>
<td>CLOSE LOGICAL DEVICE</td>
</tr>
<tr>
<td>GDATA</td>
<td>GET DATA</td>
</tr>
<tr>
<td>GDATLN</td>
<td>GET DATA LENGTH</td>
</tr>
<tr>
<td>GETATT</td>
<td>GET ATTRIBUTES</td>
</tr>
<tr>
<td>GETBAK</td>
<td>GET BACKGROUND</td>
</tr>
<tr>
<td>GETCUR</td>
<td>GET CURSOR</td>
</tr>
<tr>
<td>GPAGE</td>
<td>GET PAGE</td>
</tr>
<tr>
<td>GWINDO</td>
<td>GET WINDOW</td>
</tr>
<tr>
<td>INITVT</td>
<td>INITIAL VTI</td>
</tr>
<tr>
<td>INQLDV</td>
<td>INQUIRE LOGICAL DEVICE</td>
</tr>
<tr>
<td>OISCR</td>
<td>OUTPUT / INPUT SCREEN</td>
</tr>
<tr>
<td>OPNFRM</td>
<td>OPEN FORM</td>
</tr>
<tr>
<td>OPNLVD</td>
<td>OPEN LOGICAL DEVICE</td>
</tr>
<tr>
<td>OUTSCR</td>
<td>OUTPUT SCREEN</td>
</tr>
<tr>
<td>PARFQN</td>
<td>PARSE FULLY QUALIFIED NAME</td>
</tr>
<tr>
<td>PDATA</td>
<td>PUT DATA</td>
</tr>
<tr>
<td>PMSGLC</td>
<td>PUT MESSAGE LINE CODE</td>
</tr>
<tr>
<td>PMSGLIS</td>
<td>PUT MESSAGE LINE STRING</td>
</tr>
<tr>
<td>PUTATT</td>
<td>PUT ATTRIBUTES</td>
</tr>
<tr>
<td>PUTBAK</td>
<td>PUT BACKGROUND</td>
</tr>
<tr>
<td>PUTCUR</td>
<td>PUT CURSOR</td>
</tr>
<tr>
<td>PUTLOC</td>
<td>PUT CURSOR LOCATION</td>
</tr>
<tr>
<td>RMVPAG</td>
<td>REMOVE PAGE</td>
</tr>
</tbody>
</table>

3-16
 APPLICATION INTERFACE Where-inlude-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>RPLFRM</td>
<td>REPLACE FORM</td>
</tr>
<tr>
<td></td>
<td>TERMVT</td>
<td>TERMINATE VTI</td>
</tr>
</tbody>
</table>

SRVRET

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDELM</td>
<td>ADD ELEMENT</td>
</tr>
<tr>
<td>ADDFRM</td>
<td>ADD FORM</td>
</tr>
<tr>
<td>CHGLDV</td>
<td>CHANGE LOGICAL DEVICE</td>
</tr>
<tr>
<td>CLSFRM</td>
<td>CLOSE FORM</td>
</tr>
<tr>
<td>CLSLDAP</td>
<td>CLOSE LOGICAL DEVICE</td>
</tr>
<tr>
<td>GDATA</td>
<td>GET DATA</td>
</tr>
<tr>
<td>GDATLN</td>
<td>GET DATA LENGTH</td>
</tr>
<tr>
<td>GETATT</td>
<td>GET ATTRIBUTES</td>
</tr>
<tr>
<td>GETBAAK</td>
<td>GET BACKGROUND</td>
</tr>
<tr>
<td>GETCUR</td>
<td>GET CURSOR</td>
</tr>
<tr>
<td>GPAGE</td>
<td>GET PAGE</td>
</tr>
<tr>
<td>GWINDO</td>
<td>GET WINDOW</td>
</tr>
<tr>
<td>INITVT</td>
<td>INITIAL VTI</td>
</tr>
<tr>
<td>INQLDV</td>
<td>INQUIRE LOGICAL DEVICE</td>
</tr>
<tr>
<td>OISCR</td>
<td>OUTPUT / INPUT SCREEN</td>
</tr>
<tr>
<td>OPNFRM</td>
<td>OPEN FORM</td>
</tr>
<tr>
<td>OPNLDAP</td>
<td>OPEN LOGICAL DEVICE</td>
</tr>
<tr>
<td>OUTSCR</td>
<td>OUTPUT SCREEN</td>
</tr>
<tr>
<td>PARFQN</td>
<td>PARSE FULLY QUALIFIED NAME</td>
</tr>
<tr>
<td>PDATA</td>
<td>PUT DATA</td>
</tr>
<tr>
<td>PMSGTC</td>
<td>PUT MESSAGE LINE CODE</td>
</tr>
<tr>
<td>PMSGTL</td>
<td>PUT MESSAGE LINE STRING</td>
</tr>
<tr>
<td>PUTATT</td>
<td>PUT ATTRIBUTES</td>
</tr>
<tr>
<td>PUTBAAK</td>
<td>PUT BACKGROUND</td>
</tr>
<tr>
<td>PUTCUR</td>
<td>PUT CURSOR</td>
</tr>
<tr>
<td>PUTLOC</td>
<td>PUT CURSOR LOCATION</td>
</tr>
<tr>
<td>RMVPAG</td>
<td>REMOVE PAGE</td>
</tr>
<tr>
<td>RPLFRM</td>
<td>REPLACE FORM</td>
</tr>
<tr>
<td>TERMVT</td>
<td>TERMINATE VTI</td>
</tr>
<tr>
<td>Include</td>
<td>Module</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>STDTP</td>
<td>GETUIM</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.
APPLICATION INTERFACE Where-external-routine-used List

<table>
<thead>
<tr>
<th>System</th>
<th>Module Name</th>
<th>Module Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>APACCT</td>
<td>GETUIM</td>
<td>GET USER INTERFACE MONITOR AP</td>
</tr>
<tr>
<td>MEMCPY</td>
<td>GETUIM</td>
<td>GET USER INTERFACE MONITOR AP</td>
</tr>
<tr>
<td>NSEND</td>
<td>ADDELM</td>
<td>ADD ELEMENT</td>
</tr>
<tr>
<td></td>
<td>ADDFRM</td>
<td>ADD FORM</td>
</tr>
<tr>
<td></td>
<td>CHGLDV</td>
<td>CHANGE LOGICAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>CLSFRM</td>
<td>CLOSE FORM</td>
</tr>
<tr>
<td></td>
<td>CLS LDV</td>
<td>CLOSE LOGICAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>GDATA</td>
<td>GET DATA</td>
</tr>
<tr>
<td></td>
<td>GDATLN</td>
<td>GET DATA LENGTH</td>
</tr>
<tr>
<td></td>
<td>GETATT</td>
<td>GET ATTRIBUTES</td>
</tr>
<tr>
<td></td>
<td>GETBAK</td>
<td>GET BACKGROUND</td>
</tr>
<tr>
<td></td>
<td>GETCUR</td>
<td>GET CURSOR</td>
</tr>
<tr>
<td></td>
<td>GPAGE</td>
<td>GET PAGE</td>
</tr>
<tr>
<td></td>
<td>GWINDO</td>
<td>GET WINDOW</td>
</tr>
<tr>
<td></td>
<td>INITVT</td>
<td>INITIAL VTI</td>
</tr>
<tr>
<td></td>
<td>INQ LDV</td>
<td>INQUIRE LOGICAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>OISCR</td>
<td>OUTPUT / INPUT SCREEN</td>
</tr>
<tr>
<td></td>
<td>OPNFRM</td>
<td>OPEN FORM</td>
</tr>
<tr>
<td></td>
<td>OPN LDV</td>
<td>OPEN LOGICAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>OUTSCR</td>
<td>OUTPUT SCREEN</td>
</tr>
<tr>
<td></td>
<td>PARFQN</td>
<td>PARSE FULLY QUALIFIED NAME</td>
</tr>
<tr>
<td></td>
<td>PDATA</td>
<td>PUT DATA</td>
</tr>
<tr>
<td></td>
<td>PM SG LC</td>
<td>PUT MESSAGE LINE CODE</td>
</tr>
<tr>
<td></td>
<td>PM SG LS</td>
<td>PUT MESSAGE LINE STRING</td>
</tr>
<tr>
<td></td>
<td>PUTATT</td>
<td>PUT ATTRIBUTES</td>
</tr>
<tr>
<td></td>
<td>PUTBAK</td>
<td>PUT BACKGROUND</td>
</tr>
<tr>
<td></td>
<td>PUTCUR</td>
<td>PUT CURSOR</td>
</tr>
<tr>
<td></td>
<td>PUTLOC</td>
<td>PUT CURSOR LOCATION</td>
</tr>
<tr>
<td></td>
<td>RMVPAG</td>
<td>REMOVE PAGE</td>
</tr>
<tr>
<td></td>
<td>RPL FRM</td>
<td>REPLACE FORM</td>
</tr>
<tr>
<td></td>
<td>TERM VT</td>
<td>TERMINATE VTI</td>
</tr>
</tbody>
</table>

3-20
APPLICATION INTERFACE Where-external-routine-used List

<table>
<thead>
<tr>
<th>System Module</th>
<th>Module Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCV</td>
<td>ADDELM</td>
<td>ADD ELEMENT</td>
</tr>
<tr>
<td></td>
<td>ADDFRM</td>
<td>ADD FORM</td>
</tr>
<tr>
<td></td>
<td>CHGLDV</td>
<td>CHANGE LOGICAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>CLSFRM</td>
<td>CLOSE FORM</td>
</tr>
<tr>
<td></td>
<td>CLSLDV</td>
<td>CLOSE LOGICAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>GDATA</td>
<td>GET DATA</td>
</tr>
<tr>
<td></td>
<td>GDATLN</td>
<td>GET DATA LENGTH</td>
</tr>
<tr>
<td></td>
<td>GETATT</td>
<td>GET ATTRIBUTES</td>
</tr>
<tr>
<td></td>
<td>GETBAK</td>
<td>GET BACKGROUND</td>
</tr>
<tr>
<td></td>
<td>GETCUR</td>
<td>GET CURSOR</td>
</tr>
<tr>
<td></td>
<td>GPAGE</td>
<td>GET PAGE</td>
</tr>
<tr>
<td></td>
<td>GWINDO</td>
<td>GET WINDOW</td>
</tr>
<tr>
<td></td>
<td>INITVT</td>
<td>INITIAL VTI</td>
</tr>
<tr>
<td></td>
<td>INQLDV</td>
<td>INQUIRE LOGICAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>OISCR</td>
<td>OUTPUT / INPUT SCREEN</td>
</tr>
<tr>
<td></td>
<td>OPNFRM</td>
<td>OPEN FORM</td>
</tr>
<tr>
<td></td>
<td>OPNLDV</td>
<td>OPEN LOGICAL DEVICE</td>
</tr>
<tr>
<td></td>
<td>OUTSCR</td>
<td>OUTPUT SCREEN</td>
</tr>
<tr>
<td></td>
<td>PARFQN</td>
<td>PARSE FULLY QUALIFIED NAME</td>
</tr>
<tr>
<td></td>
<td>PDATA</td>
<td>PUT DATA</td>
</tr>
<tr>
<td></td>
<td>PUTATT</td>
<td>PUT ATTRIBUTES</td>
</tr>
<tr>
<td></td>
<td>PUTBAK</td>
<td>PUT BACKGROUND</td>
</tr>
<tr>
<td></td>
<td>PUTCUR</td>
<td>PUT CURSOR</td>
</tr>
<tr>
<td></td>
<td>PUTLOC</td>
<td>PUT CURSOR LOCATION</td>
</tr>
<tr>
<td></td>
<td>RMVPAG</td>
<td>REMOVE PAGE</td>
</tr>
<tr>
<td></td>
<td>RPLFRM</td>
<td>REPLACE FORM</td>
</tr>
<tr>
<td></td>
<td>TERMVT</td>
<td>TERMINATE VTI</td>
</tr>
</tbody>
</table>

3-21
3.10.7 Main Program Parts List

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

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDELN</td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE 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>ADDFRM</td>
<td>Purpose --&gt; ADD FORM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>

3-24
### APPLICATION INTERFACE 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>CHGLDV</td>
<td>Purpose---&gt;CHANGE LOGICAL DEVICE</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>Main Pgm Name</td>
<td>Module Name</td>
<td>Module Type</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>CLSFRM</td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE 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>CLSLDV</td>
<td>Purpose --&gt; CLOSE LOGICAL DEVICE</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>

3-27
### APPLICATION INTERFACE Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDATA</td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
<tr>
<td>Main Pgm Name</td>
<td>Module Name</td>
<td>Module Type</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>GETATT</td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>
APPLICATION INTERFACE 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>GETBAK</td>
<td>Purpose--&gt;GET BACKGROUND</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE Main Program Parts List

<table>
<thead>
<tr>
<th>Module</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NECUR</td>
<td>GETCUR</td>
<td>Purpose --&gt; GET CURSOR</td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</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>GPAGE</td>
<td>Purpose--&gt;GET PAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE 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>GWINDO</td>
<td>Purpose--&gt;GET WINDOW</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
APPLICATION INTERFACE 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>INITVT</td>
<td>Purpose--&gt;INITIAL VTI</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE 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>INQLDV</td>
<td>Purpose--&gt;INQUIRE LOGICAL DEVICE</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE 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>OISCR</td>
<td>Purpose---&gt;OUTPUT / INPUT SCREEN</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
**APPLICATION INTERFACE Main Program Parts List**

<table>
<thead>
<tr>
<th>Main Pgm Name</th>
<th>Module Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPNFRM</td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</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>OPNLVDV</td>
<td>Purpose--&gt;OPEN LOGICAL DEVICE</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
APPLICATION INTERFACE 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>OUTSCR</td>
<td></td>
<td>Purpose --&gt; OUTPUT SCREEN</td>
</tr>
<tr>
<td></td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>
APPLICATION INTERFACE 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>PARFQN</td>
<td>Purpose--&gt;PARSE FULLY QUALIFIED NAME</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
APPLICATION INTERFACE 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>PDATA</td>
<td>Purpose--&gt;PUT DATA</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GDATLN</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
## APPLICATION INTERFACE 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>PMSGLC</td>
<td>Purpose --&gt; PUT MESSAGE LINE CODF</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm</th>
<th>Module Name</th>
<th>Module Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMSGLS</td>
<td>Purpose--&gt;PUT MESSAGE LINE STRING</td>
<td></td>
</tr>
<tr>
<td></td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
</tbody>
</table>
## APPLICATION INTERFACE 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>PUTATT</td>
<td>Purpose--&gt;PUT ATTRIBUTES</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE 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>PUTBAK</td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>
APPLICATION INTERFACE Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm</th>
<th>Module</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUTCUR</td>
<td></td>
<td>Purpose--&gt;PUT CURSOR</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td></td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td></td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td></td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td></td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td></td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>Main Pgm Name</td>
<td>Module Name</td>
<td>Module Type</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>------------------------------</td>
<td></td>
</tr>
<tr>
<td>PUTLOC</td>
<td>Purpose--&gt;PUT CURSOR LOCATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>
### APPLICATION INTERFACE 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>RMVPAG</td>
<td>APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>
# APPLICATION INTERFACE Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm</th>
<th>Module</th>
<th>Module Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPLFRM</td>
<td></td>
<td>Purpose</td>
<td>REPLACE FORM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3-49 APACCT</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GETUIM</td>
<td>Well-defined module</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MEMCPY</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NSEND</td>
<td>External routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RCV</td>
<td>External routine</td>
</tr>
</tbody>
</table>
APPLICATION INTERFACE Main Program Parts List

<table>
<thead>
<tr>
<th>Main Pgm</th>
<th>Module</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>Type</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>TERMVT</td>
<td>Purpose--&gt;TERMINATE VTI</td>
<td></td>
</tr>
<tr>
<td>APACCT</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>GETUIM</td>
<td>Well-defined module</td>
<td></td>
</tr>
<tr>
<td>MEMCPY</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>NSEND</td>
<td>External routine</td>
<td></td>
</tr>
<tr>
<td>RCV</td>
<td>External routine</td>
<td></td>
</tr>
</tbody>
</table>

3-50
3.10.8 Module Documentation

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

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

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

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

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

The Module Documentation is arranged alphabetically according to Module Name.
APPLICATION INTERFACE Module Documentation

NAME:  ADDELM
PURPOSE:  ADD ELEMENT
LANGUAGE:  VAX-11 COBOL
MODULE TYPE:  PROGRAM
SOURCE FILE:  ADDELM

DESCRIPTION:

INCLUDE FILES:
------------------
FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:
-----------------
GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: ADDFRM
PURPOSE: ADD FORM
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: ADDFRM

DESCRIPTION:

----------

INCLUDE FILES:

----------

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

----------

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: CHGLDV
PURPOSE: CHANGE LOGICAL DEVICE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: CHGLDV

DESCRIPTION:

INCLUDE FILES:

---------------
FPCODE  - Form Processor return codes
ROUTID   - ROUTine ID definitions
SRVRET   - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

---------------
GETUIM   - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: CLSFRM
PURPOSE: CLOSE FORM
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: CLSFRM

DESCRIPTION:

INCLUDE FILES:

FP CODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: CLSLDV
PURPOSE: CLOSE LOGICAL DEVICE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: CLSLDV

DESCRIPTION:

INCLUDE FILES:

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: GDATA
PURPOSE: GET DATA
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: GDATA

DESCRIPTION:

INCLUDE FILES:

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: GDATLN
PURPOSE: GET DATA LENGTH
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: GDATLN

DESCRIPTION:

INCLUDE FILES:

FP CODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
SEND
RCV

CALLED DIRECTLY BY:

PDATA - PUT DATA

USED IN MAIN PROGRAM(S):

PDATA - PUT DATA
APPLICATION INTERFACE Module Documentation

NAME: GETATT
PURPOSE: GET ATTRIBUTES
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: GETATT

DESCRIPTION:

INCLUDE FILES:

- FPCODE - Form Processor return codes
- ROUTID - ROUTine ID definitions
- SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

- GETUIM - GET USER INTERFACE MONITOR AP
- NSEND
- RCV
APPLICATION INTERFACE Module Documentation

NAME: GETBAK
PURPOSE: GET BACKGROUND
LANGUAGE: VAX-I COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: GETBAK

DESCRIPTION:

INCLUDE FILES:

FP CODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: GETCUR
PURPOSE: GET CURSOR
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: GETCUR

DESCRIPTION:

INCLUDE FILES:

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: GETUIM
PURPOSE: GET USER INTERFACE MONITOR AP
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: GETUIM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FPAI
DOCUMENTATION GROUP: FPAI

DESCRIPTION:

SYNOPSIS

FORTRAN VOID GETUIM(UAPNM, UICHAN, RCODE)
    CHAR UAPNM[APNM_LEN], UICHAN[LCHAN_LEN],
        RCODE[RCODE_LEN];

OUTPUTS:

UAPNM - AP NAME OF UIM
UICHAN - LOGICAL CHANNEL FOR UIM
RCODE - RETURN CODE

DESCRIPTION

RETURNS THE AP NAME AND LOGICAL CHANNEL OF THE USER INTERFACE MONITOR.

ARGUMENTS:

----------

UIAPNM = CHAR [APNM_LEN]
UICHAN = CHAR [LCHAN_LEN]
RCODE = CHAR [RCODE_LEN]

IN-LUDE FILES:

----------

.IDTYP - STANDARD TYPE DEFINITIONS
.APINME - CORRECTED C VERSION OF APINME.INC
.BUFAPI - CORRECTED C VERSION OF BUFAPI.INC
.APIEVB - CORRECTED C VERSION OF APIEVB.INC

ROUTINES CALLED:

----------

APACCT
memcpy

CALLED DIRECTLY BY:

----------

ADDELM - ADD ELEMENT
ADDFRM - ADD FORM
CHGLDV - CHANGE LOGICAL DEVICE
CLSF RM - CLOSE FORM
CLSLDV - CLOSE LOGICAL DEVICE
GDATA - GET DATA
GDATLN - GET DATA LENGTH
GETATT - GET ATTRIBUTES
GETBAK - GET BACKGROUND
GETCUR - GET CURSOR
GPAGE - GET PAGE
GWINDO - GET WINDOW
INITVT - INITIAL VTI
INQLDV - INQUIRE LOGICAL DEVICE
OISCR - OUTPUT / INPUT SCREEN
OPNFRM - OPEN FORM
OPNL DV - OPEN LOGICAL DEVICE
OUTSCR - OUTPUT SCREEN
P QRFQ N - PARSE FULLY QUALIFIED NAME
PDATA - PUT DATA
PMSGLC - PUT MESSAGE LINE CODE
PMSGLS - PUT MESSAGE LINE STRING
PUTATT - PUT ATTRIBUTES
PUTBAK - PUT BACKGROUND
PUTCUR - PUT CURSOR
PUTLOC - PUT CURSOR LOCATION
RMVPAG - REMOVE PAGE
RPLFRM - REPLACE FORM
TERMVT - TERMINATE VTI

USED IN MAIN PROGRAM(S):
------------------------------
ADDELM - ADD ELEMENT
ADDFRM - ADD FORM
CHGL DV - CHANGE LOGICAL DEVICE
CLSF RM - CLOSE FORM
CLSLDV - CLOSE LOGICAL DEVICE
GDATA - GET DATA
GETATT - GET ATTRIBUTES
GETBAK - GET BACKGROUND
GETCUR - GET CURSOR
GPAGE - GET PAGE
GWINDO - GET WINDOW
INITVT - INITIAL VTI
INQLDV - INQUIRE LOGICAL DEVICE
OISCR - OUTPUT / INPUT SCREEN
OPNFRM - OPEN FORM
OPNL DV - OPEN LOGICAL DEVICE
OUTSCR - OUTPUT SCREEN
PARFQ N - PARSE FULLY QUALIFIED NAME
PDATA - PUT DATA
PMSGLC - PUT MESSAGE LINE CODE
PMSGLS - PUT MESSAGE LINE STRING
PUTATT - PUT ATTRIBUTES
PUTBAK - PUT BACKGROUND
PUTCUR - PUT CURSOR
PUTLOC - PUT CURSOR LOCATION
APPLICATION INTERFACE Module Documentation

NAME: GPAGE
PURPOSE: GET PAGE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: GPAGE

DESCRIPTION:

INCLUDE FILES:

- FPCODE - Form Processor return codes
- ROUTID - ROUTine ID definitions
- SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

- GETUIM - GET USER INTERFACE MONITOR AP
- NSEND
- RCV
APPLICATION INTERFACE Module Documentation

NAME: GWINDO
PURPOSE: GET WINDOW
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: GWINDO

DESCRIPTION:

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

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

FP CODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

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

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: INITFP
PURPOSE: INITialize Form Processor
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: INITFP

DESCRIPTION:  
--------------
APPLICATION INTERFACE Module Documentation

NAME: INITVT
PURPOSE: INITIAL VTI
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: INITVT

DESCRIPTION:

INCLUDE FILES:

----
FP CODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

----
GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: INQLDV
PURPOSE: INQUIRE LOGICAL DEVICE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: INQLDV

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

INCLUDE FILES:
---------------
FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:
-----------------
GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: OISCR
PURPOSE: OUTPUT / INPUT SCREEN
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: OISCR

DESCRIPTION:

INCLUDE FILES:

FP CODE - Form Processor return codes
ROUT ID - ROUTine ID definitions
SRYRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
USEND
RCV
APPLICATION INTERFACE Module Documentation

NAME:  OPNFRM
PURPOSE: OPEN FORM
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: OPNFRM

DESCRIPTION:

INCLUDE FILES:

FP CODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: OPNLDV
PURPOSE: OPEN LOGICAL DEVICE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: OPNLDV

DESCRIPTION:

INCLUDE FILES:

- FPCODE - Form Processor return codes
- ROUTID - ROUTine ID definitions
- SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

- GETUIM - GET USER INTERFACE MONITOR AP
- USEND
- RCV
APPLICATION INTERFACE Module Documentation

NAME: OUTSCR
PURPOSE: OUTPUT SCREEN
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: OUTSCR

DESCRIPTION:

INCLUDE FILES:

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: PARFQN
PURPOSE: PARSE FULLY QUALIFIED NAME
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: PARFQN

DESCRIPTION:

INCLUDE FILES:

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
USEND
RCV
**APPLICATION INTERFACE Module Documentation**

**NAME:** PDATA  
**PURPOSE:** PUT DATA  
**LANGUAGE:** VAX-11 COBOL  
**MODULE TYPE:** PROGRAM  
**SOURCE FILE:** PDATA

**DESCRIPTION:**

**INCLUDE FILES:**

- **FP CODE** - Form Processor return codes
- **ROUTID** - ROUTine ID definitions
- **SRVRET** - AS THE RETURN GIVEN A TABLE-FULL ERROR

**ROUTINES CALLED:**

- **GDATLN** - GET DATA LENGTH
- **GETUIM** - GET USER INTERFACE MCNITOR AP
- **NSEND**
- **RCV**
APPLICATION INTERFACE Module Documentation

NAME: PMSGLC
PURPOSE: PUT MESSAGE LINE CODE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: PMSGLC

DESCRIPTION:

INCLUDE FILES:

FP CODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
USEND
APPLICATION INTERFACE Module Documentation

NAME:        PMSGLS
PURPOSE:     PUT MESSAGE LINE STRING
LANGUAGE:    VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: PMSGLS

DESCRIPTION:

INCLUDE FILES:

FPCODE       - Form Processor return codes
ROUTID       - ROUTine ID definitions
SRVRET       - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM       - GET USER INTERFACE MONITOR AP
!SEND
APPLICATION INTERFACE Module Documentation

NAME: PUTATT
PURPOSE: PUT ATTRIBUTES
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: PUTATT

DESCRIPTION:

---------

INCLUDE FILES:

---------

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

---------

GETUIM - GET USER INTERFACE MONITOR AP
HSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: PUTBAK
PURPOSE: PUT BACKGROUND
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: PUTBAK

DESCRIPTION:

INCLUDE FILES:

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: PUTCUR
PURPOSE: PUT CURSOR
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: PUTCUR

DESCRIPTION:

INCLUDE FILES:

- FPCODE - Form Processor return codes
- ROUTID - ROUTine ID definitions
- SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

- GETUIM - GET USER INTERFACE MONITOR AP
- NSEND
- RCV
APPLICATION INTERFACE Module Documentation

NAME: PUTLOC
PURPOSE: PUT CURSOR LOCATION
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: PUTLOC

DESCRIPTION:

INCLUDE FILES:

---
FP CODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

---
GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME: RMVPAG
PURPOSE: REMOVE PAGE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: RMVPAG

DESCRIPTION:

INCLUDE FILES:

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
MSEND
RCV
APPLICATION INTERFACE Module Documentation

NAME:                    RPLFRM
PURPOSE:                 REPLACE FORM
LANGUAGE:                VAX-11 COBOL
MODULE TYPE:             PROGRAM
SOURCE FILE:             RPLFRM

DESCRIPTION:

 include files:
 similarities:

 FPCODE       - Form Processor return codes
 ROUTID       - ROUTine ID definitions
 SRVRET       - AS THE RETURN GIVEN A TABLE-FULL ERROR

 routines called:

 GETUIM       - GET USER INTERFACE MONITOR AP
 NSEND
 RCV
APPLICATION INTERFACE Module Documentation

NAME: TERMFP
PURPOSE: EXIT FORM PROCESSOR
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: TERMFP

DESCRIPTION: ---------------
APPLICATION INTERFACE Module Documentation

NAME: TERMVT
PURPOSE: TERMINATE VTI
LANGUAGE: VAX-11 COBOL
MODULE TYPE: PROGRAM
SOURCE FILE: TERMVT

DESCRIPTION:

INCLUDE FILES:

FPCODE - Form Processor return codes
ROUTID - ROUTine ID definitions
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP
NSEND
RCV

3-86
3.10.9 Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code. The language it is written in is also given.
APPLICATION INTERFACE Include File Description

FILE NAME: FPCODE
PURPOSE: Form Processor return codes
LANGUAGE: VAX-11 COBOL

DESCRIPTION:
----------
APPLICATION INTERFACE Include File Description

FILE NAME: NAPIEVB
PURPOSE: CORRECTED C VERSION OF APIEVB.INC
LANGUAGE: C

DESCRIPTION:
----------
APPLICATION INTERFACE Include File Description

FILE NAME: NAPINME
PURPOSE: CORRECTED C VERSION OF APINME.INC
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
APIS GLOBAL NAME DATA.
APPLICATION INTERFACE Include File Description

FILE NAME: NBUFAPI
PURPOSE: CORRECTED C VERSION OF BUFAPI.INC
LANGUAGE: C

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

3-91
APPLICATION INTERFACE Include File Description

FILE NAME: ROUTID
PURPOSE: ROUTine ID definitions
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

DESCRIPTION: THIS INCLUDE MEMBER CONTAINS THE ROUTINE ID DEFINITIONS FOR MESSAGING BETWEEN THE FPAI AND THE FP.

INFORMATION:
TYPE: (C-COBOL, IC-COBOL COPY) IC
SUBSYSTEM: UI
CONFIGURATION ITEM ID:

DESIGNED BY: A. J. WEHRMAN
START DATE: 8/8/85
FINISH DATE: 8/8/85

PROGRAMMED BY: A. J. WEHRMAN
START DATE: 8/8/85
FINISH DATE: 8/8/85
APPLICATION INTERFACE Include File Description

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

DESCRIPTION:

----
MODIFIED 11/2/83 TO INCLUDE RET-CODE-5
MODIFIED 1/9/84 TO INCREASE ALL ERROR CODES TO PIC X(5)
    AND TO ELIMINATE ALPHA'S
MODIFIED 1/26/84 TO ADD RET-CODE FOR GETUSR-NOT-SUCC
    SRV-SUCCESSFUL ADDED FOR GENERIC RETURN
MODIFIED 2/7/84 TO ADD ERROR CODES FOR ENTRY-NOT-FOUND
MODIFIED 2/8/84 TO ADD 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
APPLICATION INTERFACE Include File Description

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

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 USHOR'T(), TINY(), AND UTINY() SHOULD BE USED WHENEVER REFERENCING THEM.

IN ADDITION, THE FOLLOWING UTILITY MACROS ARE DEFINED:

- **LURSHIFT(N, B)** - UNSIGNED LONG RIGHT SHIFT
- **MAX(A, B)** - MAXIMUM OF A AND B
- **MIN(A, B)** - MINIMUM OF A AND B
APPLICATION INTERFACE Include File Description

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

THE FOLLOWING SYMBOLS SHOULD BE DEFINED BASED ON THE COMPILER BEING USED:
USHORT - COMPILER SUPPORTS UNSIGNED SHORT
TINY - COMPILER TREATS CHAR AS SIGNED
UTINY - CHAR IS SIGNED AND COMPILER SUPPORTS UNSIGNED CHAR
VOID - COMPILER SUPPORTS VOID
FORTRAN - COMPILER SUPPORTS FORTRAN
STRASN - DEFINE APPROPRIATE MACRO
SUCCESS - DEFINE APPROPRIATE VALUE IF NOT 0
FAILURE - DEFINE APPROPRIATE VALUE IF NOT 1
3.10.10 Hierarchy Chart

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

There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.
ADDELM...2
ADDFRM...3
APACCT
CHGLDV...4
CLSFRM...5
CLSVDV...1
GDATA...7
GDATLN...20
GETATT...8
GETBAK...9
GETCUR...6
GETTIM...2
GPAGE...11
GWINDO...12
INITVT...13
INQLDV...10
MEMCPY
NSEND
O1SCR...15
OPNFRM...16
OPNLDV...17
OUTSCR...14
PARFQN...19
PDATA...20
PMSGUC...21
PMSGLS...18
PUTATT...23
PUTBAK...24
PUTCUR...25
PUTLOC...22
RCV
RMVPAG...27
RPLFRM...26
TERMVT...26
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."
APPENDIX A

FP/AI MESSAGE FORMATS

The following is a COBOL declaration of the message formats used between the AI and the FP.

FP to AI message formats, FP output parameters.

NOTE: All 01 line records correspond to the FP routines being called. For example, ADDELM-FP-RECORD is used in the routine ADDELM.

01 ADDELM-FP-RECORD.
   05 ADDELM-ELEMENT-NUMBER PIC 9(4).
   05 ADDELM-RCODE PIC X(5).
01 ADDFRM-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 ADDFRM-PAGE-NUMBER PIC 9(4).
   05 ADDFRM-RCODE PIC X(5).
01 CHGLDV-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 CHGLDV-RCODE PIC X(5).
01 CLSFRM-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 CLSFRM-RCODE PIC X(5).
01 CLS LDV-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 CLS LDV-RCODE PIC X(5).
01 GDATA-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 GDATA-BUFFER-LENGTH PIC 9(4).
   05 GDATA-RCODE PIC X(5).
   05 GDATA-BUFFER PIC X(4096).
01 GDATLN-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 GDATLN-BUFFER-LENGTH PIC 9(4).
   03 GDATLN-RCODE PIC X(5).
01 GETATT-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 GETATT-ATTRIBUTE PIC X(10).
   05 GETATT-RCODE PIC X(5).
01 GETBAK-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 GETBAK-ATTRIBUTE PIC X(10).
   05 GETBAK-RCODE PIC X(5).
01 GETCUR-FP-RECORD REDEFINES ADDELM-FP-RECORD.
   05 GETCUR-FIELD-NAME PIC X(120).
   05 GETCUR-FIELD-TYPE PIC X.
   05 GETCUR-ROW PIC 9(4).
   05 GETCUR-COL PIC 9(4).
   05 GETCUR-RCODE PIC X(5).
01 GPAGE-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 GPAGE-FORM-NAME PIC X(10).
  05 GPAGE-RCODE PIC X(5).
01 GWINDO-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 GWINDO-TOTAL-PAGES PIC 9(4).
  05 GWINDO-RCODE PIC X(5).
01 INQLDV-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 INQLDV-LOG-DEV-ID PIC 9(5).
  05 INQLDV-RCODE PIC X(5).
01 OISCR-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 OISCR-FUNCTION PIC 9(4).
  05 OISCR-RCODE PIC X(5).
01 OPNFRM-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 OPNFRM-RCODE PIC X(5).
  01 OPNLDV-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05 OPNLDV-LOG-DEV-ID PIC 9(5).
    05 OPNLDV-RCODE PIC X(5).
  01 OUTSCR-RECORD REDEFINES ADDELM-FP-RECORD.
  05 OUTSCR-RCODE PIC X(5).
  01 PARFQN-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05 PARFQN-PAR-NAME PIC X(120).
    05 PARFQN-PAR-TYPE PIC X.
    05 PARFQN-RCODE PIC X(5).
  01 PDATA-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 PDATA-RCODE PIC X(5).
  01 PUTATT-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 PUTATT-RCODE PIC X(5).
  01 PUTBAK-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 PUTBAK-RCODE PIC X(5).
  01 PUTCUR-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 PUTCUR-RCODE PIC X(5).
  01 PUTLOC-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 PUTLOC-RCODE PIC X(5).
  01 RMVPAG-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 RMVPAG-RCODE PIC X(5).
  01 RPLFRM-FP-RECORD REDEFINES ADDELM-FP-RECORD.
  05 RPLFRM-RCODE PIC X(5).

AI to FP message format, input parameters.

01 INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
01 ADDELM-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 ADDELM-ELEMENT-NAME PIC X(120).
01 ADDFRM-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 ADDFRM-WINDOW-NAME PIC X(120).
  05 ADDFRM-FORM-NAME PIC X(10).
01 CHGLDV-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 CHGLDV-LOG-DEV-ID PIC 9(5).
01 CLSFRM-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 CLSFRM-FORM-NAME PIC X(10).
01 CLSLDV-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 CLSLDV-LOG-DEV-ID PIC 9(5).
01 GDATA-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 GDATA-INSTANCE-ID PIC 9(4).
  05 GDATA-FIELD-NAME PIC X(120).
01 GDATLN-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 GDATLN-FIELD-NAME PIC X(120).
01 GETATT-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 GETATT-FIELD-NAME PIC X(120).
  05 GETATT-DURATION PIC 9(4).
01 GETBAK-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 GETBAK-FIELD-NAME PIC X(120).
  05 GETBAK-DURATION PIC 9(4).
01 GETCUR-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
01 GPAGE-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 GPAGE-WINDOW-NAME PIC X(120).
  05 GPAGE-PAGE-NUMBER PIC 9(4).
01 GWINDO-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 GWINDO-WINDOW-NAME PIC X(120).
01 INQLDV-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
01 OISCR-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 OISCR-WINDOW-NAME PIC X(120).
01 OPNFRM-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 OPNFRM-FORM-NAME PIC X(10).
01 OPNLDV-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
01 OUTSCR-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 OUTSCR-WINDOW-NAME PIC X(120).

01 PARFQN-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 PARFQN-FIELD-NAME PIC X(120).
  05 PARFQN-FIELD-TYPE PIC X.
  05 PARFQN-LEVEL PIC 9(4).

01 PDATA-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 PDATA-FIELD-NAME PIC X(120).
  05 PDATA-BUFFER PIC X(4096).

01 PMSGLC-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 PMSGLC-MSG-CODE PIC X(5).

01 PMSGLS-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 PMSGLS-MSG-STRING PIC X(60).

01 PUTATT-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 PUTATT-FIELD-NAME PIC X(120).
  05 PUTATT-DURATION PIC 9(4).
  05 PUTATT-ATTRIBUTE PIC X(10).

01 PUTBAK-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 PUTBAK-FIELD-NAME PIC X(120).
  05 PUTBAK-DURATION PIC 9(4).
  05 PUTBAK-ATTRIBUTE PIC X(10).

01 PUTCUR-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 PUTCUR-FIELD-NAME PIC X(120).

01 PUTLOC-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 PUTLOC-FIELD-NAME PIC X(120).
  05 PUTLOC-ROW PIC 9(4).
  05 PUTLOC-COL PIC 9(4).

01 RMVPAG-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 RMVPAG-WINDOW-NAME PIC X(120).
  05 RMVPAG-PAGE-NUMBER PIC 9(4).

01 RPLFRM-FPAI-RECORD REDEFINES INPUT-RECORD.
  05 ROUTINE-ID PIC 99.
  05 RPLFRM-WINDOW-NAME PIC X(120).
  05 RPLFRM-PAGE-NUMBER PIC 9(4).
  05 RPLFRM-FORM-NAME PIC X(10).