ASG-Application Definition and Analysis™ User's Guide

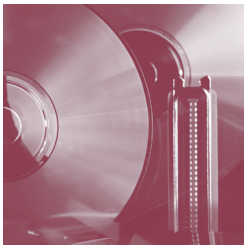
Version: 6.0

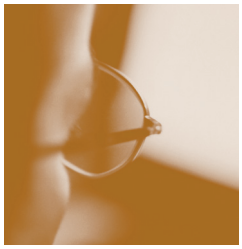
Publication Number: ALL0200-60

Publication Date: February 2002

The information contained herein is the confidential and proprietary information of Allen Systems Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of Allen Systems Group, Inc.

© 1993 - 2002 Allen Systems Group, Inc. All rights reserved.
All names and products contained herein are the trademarks or registered trademarks of their respective holders.





ASG Worldwide Headquarters Naples, Florida USA | asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 941.435.2200 Fax: 941.263.3692 Toll Free: 1.800.932.5536

ASG Documentation/Product Enhancement Fax Form

Please FAX comments regarding ASG products and/or documentation to (941) 263-3692.

Company Name	Telephone Number	Site ID	Contact name

Product Name/Publication	Version #	Publication Date
Product:		
Publication:		
Tape VOLSER:		

Enhancement Request:

© 2002 Allen Systems Group, Inc.
All names and products are trademarks or registered trademarks of their respective holders.

ASG Support Numbers

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours. To expedite response time, please follow these procedures.

Please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely or displayed
- A description of the specific steps that immediately preceded the problem
- The severity code (ASG Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)
- Verify whether you received an ASG Service Pack for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack.

If You Receive a Voice Mail Message:

1. Follow the instructions to report a production-down or critical problem.
2. Leave a detailed message including your name and phone number. A Support representative will be paged and will return your call as soon as possible.
3. Please have the information described above ready for when you are contacted by the Support representative.

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Business Hours Support

Your Location	Phone	Fax	E-mail
United States and Canada	800.354.3578	941.263.2883	support@asg.com
Australia	61.2.9460.0411	61.2.9460.0280	support.au@asg.com
England	44.1727.736305	44.1727.812018	support.uk@asg.com
France	33.141.028590	33.141.028589	support.fr@asg.com
Germany	49.89.45716.222	49.89.45716.400	support.de@asg.com
Singapore	65.332.2922	65.337.7228	support.sg@asg.com
All other countries:	1.941.435.2200		support@asg.com

Non-Business Hours - Emergency Support

Your Location	Phone	Your Location	Phone
United States and Canada	800.354.3578		
Asia	65.332.2922	**Japan/Telecom**	0041.800.9932.5536
Australia	0011.800.9932.5536	**Netherlands**	00.800.3354.3578
Denmark	00.800.9932.5536	**New Zealand**	00.800.9932.5536
France	00.800.3354.3578	**Singapore**	001.800.3354.3578
Germany	00.800.3354.3578	**South Korea**	001.800.9932.5536
Hong Kong	001.800.9932.5536	**Sweden/Telia**	009.800.9932.5536
Ireland	00.800.9932.5536	**Switzerland**	00.800.9932.5536
Israel/Bezeq	014.800.9932.5536	**Thailand**	001.800.9932.5536
Japan/IDC	0061.800.9932.5536	**United Kingdom**	00.800.3354.3578
		All other countries	1.941.435.2200

ASG Web Site

Visit http://www.asg.com, ASG's World Wide Web site.

Submit all product and documentation suggestions to ASG's product management team at http://www.asg.com/asp/emailproductsuggestions.asp.

If you do not have access to the web, FAX your suggestions to product management at (941) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

Contents

Preface vii

About this Publication viii

Related Publications ix

ASG-Existing Systems Workbench (ASG-ESW) x

Invoking ESW Products xiii

ESW Product Integration xiv
Examples xv

Publication Conventions xvii

1 Application Definition and Analysis Concepts 1

Purpose of Application Definition and Analysis 1

Contents of an Application Definition 1

Analyzing and Refining an Application Definition 2

2 Application Analysis Process Overview 3

About the Process 3

Step 1 - Assessing Your Task 4
Defining Your Requirements 5
Collecting Information About Related Application Components 5

Step 2 - Defining the Application 5
Allocating an AKR 6
Defining Application Components 6

Step 3 - Analyzing and Refining 7
Analyzing the Application 8
Resolving Analysis Errors 9
Refining the Application Definition 9

3 Assessing Your Task 11

Defining the Requirements. 11

Gathering Information About Related Application Components 11
Application Background Information Worksheets 13
Program Source Code Description Worksheets 14
Program Source Members Worksheet 15
IDMS Library Description Worksheet 17
IDMS DD and Dataset Names Worksheet 17
Compile/Link-Edit JCL Description Worksheet. 18
Load Library Worksheets 19
Load Library Members Worksheet. 19
JCL/PROC Library Description Worksheets 20
CICS Component Source Library Description Worksheets 21
IMS Component Source Library Description Worksheets 24
Other Language Source Library Description Worksheets 26

4 Defining the Application 27

Overview 27

Creating the Application 29
Creating an Application Definition. 30

Defining Application Components. 31
Defining COBOL Source Libraries 31
Adding COBOL Programs 39
Removing COBOL Library Attributes 45
Adding Load Libraries 46
Adding Load Module Members 47
Adding JCL Members. 51
Removing JCL Library Attributes 54
Adding CICS CSD Files 55
Specifying CSD Group Names 56
Adding CICS Libraries for FCT, PCT, PPT Tables, and BMS Maps 57
Adding CICS FCT, PCT, and PPT Tables, and BMS Map Members 60
Removing CICS Library Attributes 63
Adding IMS Libraries 64
Specifying Maclibs (IMS Libraries) 66
Adding IMS Members. 67
Removing IMS Library Attributes 70

Reviewing the Application Definition 71

Modifying the Application Definition 72
Modifying Library Information. 73
Modifying Member Information 76

Importing the Application Definition **78**

Exporting the Application Definition **78**

5 Analyzing the Application 79

Introduction **79**

Performing Application Analysis **82**
Selecting the Analysis Type 82
Submitting the Analyze Job 83
Restarting the Analyze 86
Scheduling an Application Analyze Through Batch Submission 87

Refining the Definition **95**
Reviewing the Analyze Results 95
Resolving Analysis Errors 95
Researching Dead, Discovered, and Missing Components 96
Identifying Dead Components 97
Identifying Discovered Components 98
Identifying Missing Components 102

6 The Application Maintenance Facility 103

Introduction **103**

Opening the AMF **103**
Alliance and Recap 103
Estimate 104
Line Commands and Menu Options in AMF 105

Editing the Application Definition **109**
The Auto-Analysis Feature 114
Changing the AMF View Include Options 115

Appendix A
Online and Batch AKR Utilities 121

Online Utility **121**

Batch Utility **122**
Job Control Statements 123
Control Cards 123

Command Formats **124**
Command Syntax 124
ANLZSTAT Command 125
COMMENTS Command 126
CONVERT Command 126
COPY Command 128

DELETE Command 129
EXPORT Command 130
HELP Command 131
INIT Command 132
MOVE Command 133
PRINT Command 134
PUNCH Command 135

Batch AKR Reports 137
AKR Utility Directory Report 137
AKR Utility Log 138
AKR Punch Directory File 139

Appendix B
Analyze Options and Parameters 145

Analyze Options 145

Analyze Parameters 151

Appendix C
Import Language Reference 153

Import Language 153
Import Language Syntax 153
Import File Structure 154
Sample Import File 170

Import Language Keywords 172
Application Information Section 172
Default Section 172
Library and Member Definition Section 174

Appendix D
Other Language Support 177

Assembler Language Support 177
ASM Support Limitations 177
Defining Assembler Components 179

PL/I Language Support 182
PL/I Support Limitations 182
Defining PL/I Components 183

Analyzing NATURAL Applications for Alliance and Estimate 183
Overview 183
Step 1 - Extracting the Application from the Database 183

Step 2 - Transferring The Extracted File To MVS (Optional) 187
Step 3 - Splitting Up the Extracted File Into Individual Components 187
Step 4 - Defining the Application Under Alliance or Estimate 188
Step 5 - Submitting the Analysis Job 189

Index. 191

Preface

This *ASG-Application Definition and Analysis User's Guide* describes defining and analyzing an application in ASG-Alliance (herein called Alliance), ASG-Recap (herein called Recap), and ASG-Estimate (herein called Estimate). The ASG-Application Definition and Analysis Facility (herein called Application Definition Facility) is the component residing in Alliance, Recap, and Estimate that provides an environment in which to build, analyze, and refine definitions of application components. Once analyzed, the application definition provides information about how the application components relate to one another.

The emphasis of the Application Definition Facility is to allow the programmer/analyst to quickly define all the components of an application for analysis. This analysis provides accurate information used to assess the resources required for application changes and the impact of proposed system changes and system failures.

Application definitions created using the Application Definition Facility are stored in an Application Knowledge Repository (AKR) that exists on the target MVS platform. By definition, the results of analyzing the items listed in the application definition, (i.e., analyzing the application), are stored within the same application object in the AKR.

Allen Systems Group, Inc. (ASG) provides professional support to resolve any questions or concerns regarding the installation or use of any ASG product. Telephone technical support is available around the world, 24 hours a day, 7 days a week.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on any ASG product.

About this Publication

This publication consists of these chapters:

- Chapter 1, "Application Definition and Analysis Concepts," discusses the concepts behind what an application definition is and how it is used.
- Chapter 2, "Application Analysis Process Overview," explains how to successfully complete application definition and analysis tasks.
- Chapter 3, "Assessing Your Task," describes the process of task assessment and provides worksheets and instructions to use for gathering application inventory information.
- Chapter 4, "Defining the Application," provides detailed instructions for creating an application definition.
- Chapter 5, "Analyzing the Application," provides detailed instructions on analyzing and refining an application definition.
- Chapter 6, "The Application Maintenance Facility," describes in detail the Application Maintenance Facility (AMF) available in Alliance, Estimate, and Recap.

Related Publications

The documentation library for the ASG-Application Definition Facility consists of these publications (where *nn* represents the product version number):

- *ASG-Alliance Installation Guide* (ALX0300-*nn*) contains instructions for the installation and maintenance of ASG-Alliance.
- *ASG-Alliance User's Guide* (ALX0200-*nn*) contains instructions for using ASG-Alliance.
- *ASG-Application Definition and Analysis User's Guide* (ALL0200-*nn*) describes defining and analyzing an application in ASG-Alliance, ASG-Recap, and ASG-Estimate.
- *ASG-Center Installation Guide* (CNX0300-*nn*) contains installation and maintenance information for ASG-Center, the common set of libraries shared by the ASG-Existing Systems Workbench suite of products.
- *ASG-Estimate Installation Guide* (TMX0300-*nn*) contains instructions for the installation and maintenance of ASG-Estimate.
- *ASG-Estimate User's Guide* (TMX0200-*nn*) contains instructions for using ASG-Estimate.
- *ASG-Recap Installation Guide* (RCX0300-*nn*) contains instructions for the installation and maintenance of ASG-Recap.
- *ASG-Recap User's Guide* (RCX0200-*nn*) contains instructions for using ASG-Recap.

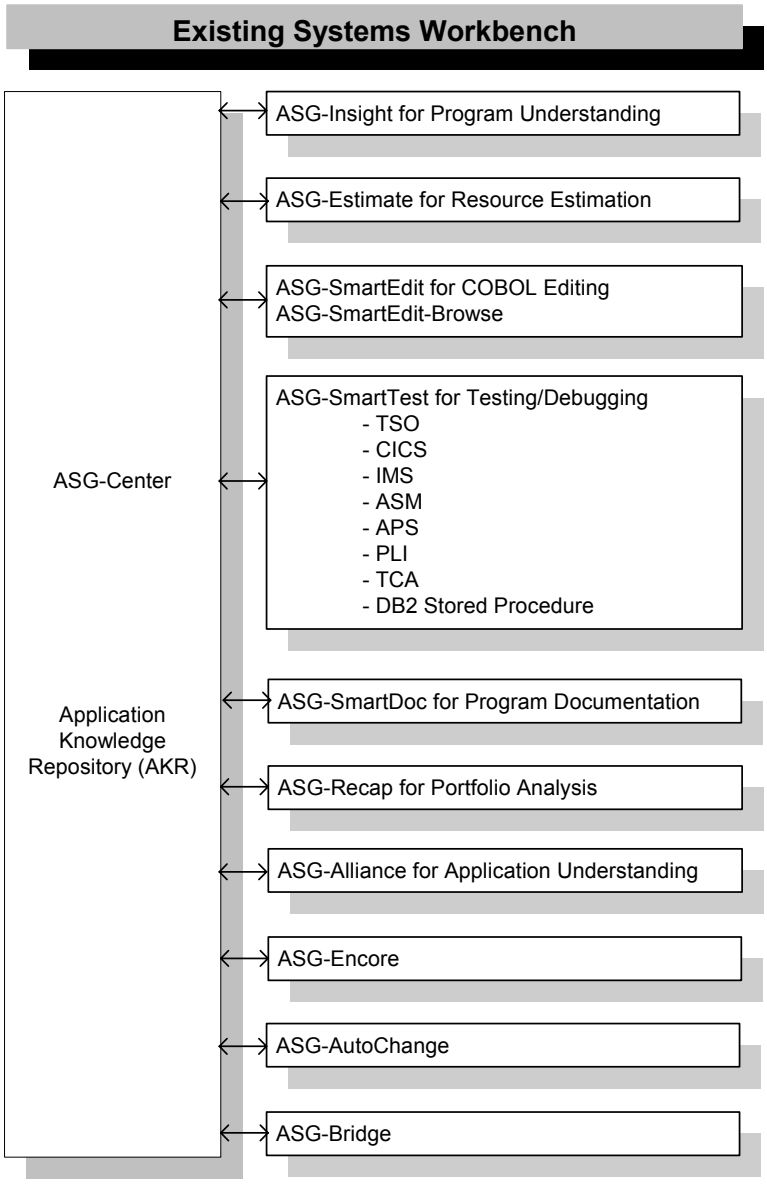
Note:

To obtain a specific version of a publication, contact the ASG Service Desk.

ASG-Existing Systems Workbench (ASG-ESW)

ASG-ESW (herein called ESW) is an integrated suite of components designed to assist organizations in enhancing, redeveloping, or re-engineering their existing systems. ESW products use the Application Knowledge Repository (AKR) to store source program analysis information generated by the Analytical Engine. Figure 1 represents the components of ESW.

Figure 1 • ASG Existing Systems Workbench



This table contains the name and description of each ESW component:

ESW Product	Herein Called	Description
ASG-Alliance	Alliance	The application understanding component that is used by IT professionals to conduct an analysis of every application in their environment. Alliance supports the analysis and assessment of the impact of change requests upon an entire application. Alliance allows the programmer/analyst to accurately perform application analysis tasks in a fraction of the time it would take to perform these tasks without an automated analysis tool. The impact analysis from Alliance provides application management with additional information for use in determining the resources required for application changes.
ASG-AutoChange	AutoChange	The COBOL code change tool that makes conversion teams more productive by enabling quick and safe changes to be made to large quantities of code. AutoChange is an interactive tool that guides the user through the process of making source code changes.
ASG-Bridge	Bridge	The bridging product that enables field expansion for program source code, without being required to simultaneously expand the fields in files or databases. Because programs are converted in smaller groups, or on a one-by-one basis, and do not require file conversion, testing during the conversion process is simpler and more thorough.
ASG-Center	Center	The common platform for all ESW products. Center provides the common Analytical Engine to analyze the source program and store this information in the AKR. This common platform provides a homogeneous environment for all ESW products to work synergistically.

ESW Product	Herein Called	Description
ASG-Encore	Encore	The program re-engineering component for COBOL programs. Encore includes analysis facilities and allows you to extract code based on the most frequently used re-engineering criteria. The code generation facilities allow you to use the results of the extract to generate a standalone program, a callable module, a complement module, and a CICS server. Prior to code generation, you can view and modify the extracted Logic Segment using the COBOL editor.
ASG-Estimate	Estimate	The resource estimation tool that enables the user to define the scope, determine the impact, and estimate the cost of code conversion for COBOL, Assembler, and PL/I programs. Estimate locates selected data items across an application and determines how they are used (moves, arithmetic operations, and compares). Time and cost factors are applied to these counts, generating cost and personnel resource estimates.
ASG-Insight	Insight	The program understanding component for COBOL programs. Insight allows programmers to expose program structure, identify data flow, find program anomalies, and trace logic paths. It also has automated procedures to assist in debugging program abends, changing a computation, and resolving incorrect program output values.
ASG-Recap	Recap	The portfolio analysis component that evaluates COBOL applications. Recap reports provide function point analysis and metrics information, program quality assessments, intra-application and inter-application comparisons and summaries, and historical reporting of function point and metrics information. The portfolio analysis information can also be viewed interactively or exported to a database, spreadsheet, or graphics package.
ASG-SmartDoc	SmartDoc	The program documentation component for COBOL programs. SmartDoc reports contain control and data flow information, an annotated source listing, structure charts, program summary reports, exception reports for program anomalies, and software metrics.

ESW Product	Herein Called	Description
ASG-SmartEdit	SmartEdit	The COBOL editing component that can be activated automatically when the ISPF/PDF Editor is invoked. SmartEdit provides comprehensive searching, inline copybook display, and syntax checking. SmartEdit allows you to include an additional preprocessor (for example, the APS generator) during syntax checking. SmartEdit supports all versions of IBM COBOL, CICS, SQL, and CA-IDMS.
ASG-SmartTest	SmartTest	The testing/debugging component for COBOL, PL/I, Assembler, and APS programs in the TSO, MVS Batch, CICS (including file services), and IMS environments. SmartTest features include program analysis commands, execution control, intelligent breakpoints, test coverage, pseudo code with COBOL source update, batch connect, disassembled object code support, and full screen memory display.

Invoking ESW Products

The method you use to invoke an ESW product depends on your system setup. If you need assistance to activate a product, see your systems administrator. If your site starts a product directly, use the ISPF selection or CLIST as indicated by your systems administrator. If your site uses the ESW screen to start a product, initiate the ESW screen using the ISPF selection or CLIST as indicated by your systems administrator and then typing in the product command on the command line.

The product names can also vary depending on whether you access a product directly or through ESW. See "ESW Product Integration" on page xiv for more information about using ESW.

To initialize ESW products from the main ESW screen, select the appropriate option on the action bar pull-downs or type the product shortcut on the command line.

Product Name	Shortcut	ESW Pull-down Options
Alliance	AL	Understand ▸ Application
AutoChange	CC	Change ▸ Conversion Set
Bridge	BR	Change ▸ ASG-Bridge
Encore (Re-engineer)	EN	Re-engineer ▸ Program
Estimate	ES	Measure ▸ ASG-Estimate
Insight (Understand)	IN	Understand ▸ Program
Recap (Portfolio Analysis)	RC	Measure ▸ Portfolio
SmartDoc (Document)	DC	Document ▸ Program
SmartEdit	SE	Change ▸ Program **Or** Change ▸ Program with Options
SmartTest	ST	Test ▸ Module/Transaction

ESW Product Integration

Because ESW is an integrated suite of products, you are able to access individual ESW products directly or through the main ESW screen. As a result, you might see different fields, values, action bar options, and pull-down options on a screen or pop-up depending on how you accessed the screen or pop-up.

Certain ESW products also contain functionality that interfaces with other ESW products. Using SmartTest as an example, if Alliance is installed, SmartTest provides a dynamic link to Alliance that can be used to display program analysis information. If Insight is installed and specified during the analyze, the Insight program analysis functions are automatically available for viewing logic/data relationships and execution path. For example, the Scratchpad option is available on the Options pull-down if you have Insight installed. Access to these integrated products requires only that they be installed and executed in the same libraries.

Examples

Example 1. Figure 2 shows the Encore Primary screen that displays when you access Encore directly.

The Encore Primary screen contains these eight action bar menu items: File, View, Extract, Generate, Search, List, Options, and Help.

Figure 2 • Encore Primary Screen

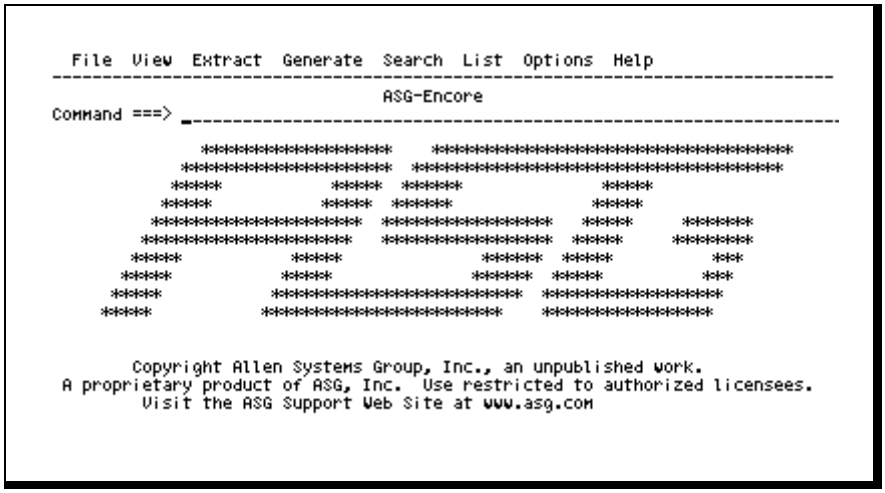
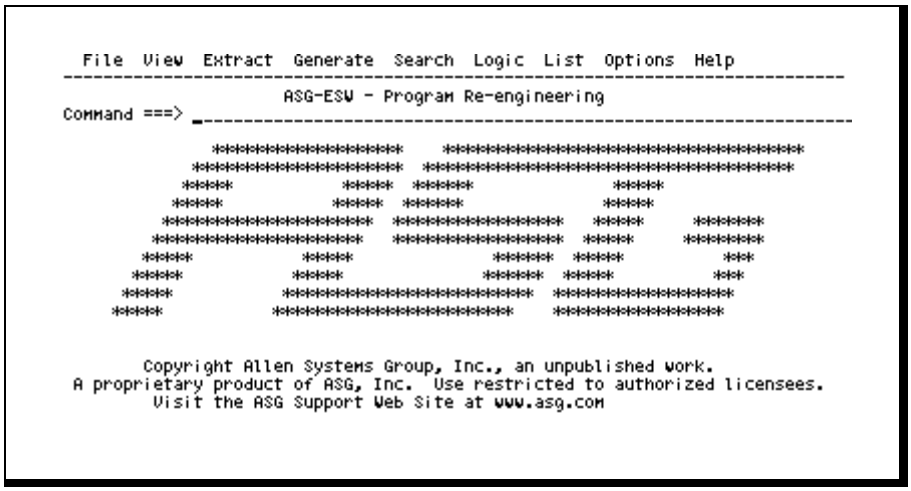


Figure 3 shows the Encore Primary screen that displays when you access Encore through ESW by selecting Re-engineer ▸ Program from the ESW action bar menu. Notice that the Primary screen name changes to ASG-ESW - Program Re-engineering when you enter Encore through ESW. Also, the Logic menu item displays if Insight is installed.

Figure 3 • ESW Encore Primary Screen



Example 2. Figure 4 shows the File - Analyze Submit pop-up that displays when you access SmartTest directly. Figure 5 shows the File - Analyze Submit pop-up that displays when you access SmartTest through ESW.

Notice that the Analyze features field in Figure 5 lists additional ESW products than shown on Figure 4. This field is automatically customized to contain the ESW products you have installed on your system.

The actions shown on these screens also vary. For example, the D action (ASG-SmartDoc Options) is available on the File - Analyze Submit screen if the SmartDoc product is installed on your system. In Figure 4, the ASG-SmartDoc Options action is not available.

Figure 4 • File - Analyze Submit Screen

```
                          File - Analyze Submit
Command ===> ____________________________________________________________

           E - Edit JCL                                S - Submit JCL

Compile and link JCL (PDS or sequential):
   Data set name  'USER12.REL.CNTL(VIAPCOBC)'

Analyze features (Y/N):
   ASG-SmartTest:  Y   Extended Analysis:  N

AKR data set name 'USER12.GENERAL.AKR'
AKR program name                       (if overriding PROGRAM-ID)

Analyze options:
   ______________________________________________________________________
   ______________________________________________________________________
   ______________________________________________________________________

Compile? (Y/N) . . . . . . . . . Y     (Y if needed by features)
Link load module reusable? (Y/N) Y
```

Figure 5 • File - Analyze Submit Screen (Accessed through ESW)

```
                          File - Analyze Submit
Command ===> ____________________________________________________________

           E - Edit JCL     S - Submit JCL     D - ASG-SmartDoc Options

Compile and link JCL (PDS or sequential):
   Data set name  'USER12.REL.CNTL(HTEST)'

Analyze features (Y/N):
   ASG-Insight:  Y   ASG-SmartTest:  Y   Extended Analysis:  N
   ASG-SmartDoc:  N   ASG-Encore:  N
AKR data set name 'USER12.GENERAL.AKR'
AKR program name  __________           (if overriding PROGRAM-ID)

Analyze options:
   ______________________________________________________________________
   ______________________________________________________________________
   ______________________________________________________________________

Compile? (Y/N) . . . . . . . . . Y     (Y if needed by features)
Link load module reusable? (Y/N) Y     (ASG-SmartTest
```

Publication Conventions

ASG uses these conventions in technical publications:

Convention	Represents
ALL CAPITALS	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
`lowercase italic monospace`	Information that you provide according to your particular situation. For example, you would replace *`filename`* with the actual name of the file.
`Monospace`	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.
Vertical Separator Bar (|) with underline	Options available with the default value underlined (e.g., Y|N).

1 Application Definition and Analysis Concepts

This chapter discusses the concepts behind what an application definition is and how it is used, and contains these sections:

Section	Page
Purpose of Application Definition and Analysis	1
Contents of an Application Definition	1
Analyzing and Refining an Application Definition	2

Purpose of Application Definition and Analysis

The purpose of the Application Definition Facility is to create application definitions and to analyze the items stored in the application definitions. An application definition is a list of items that define the components of an application. Analyzing the items listed in an application definition provides an understanding of the components in an application and how they are related to each other.

Contents of an Application Definition

An application is a group of programs and associated components that perform a related function. An application definition includes programs, executable modules, database definitions, screen definitions, execution jobs, and transaction definitions. The definition of these elements exists in source form. The location of the source form is what is added to the application definition. The Application Analytical Engine (AAE) reads, parses, and analyzes the source form provided by the application definition to determine the inter-relationships of application components.

The Application Definition Facility allows you to define applications that exist on an MVS platform. Under the MVS platform, the source form of the items is found in the libraries dataset name (DSN), dataset, and file. Libraries are divided into members (unless it is a sequential library). A member contains a discrete source form of the definition of one or more components. Therefore, the items in an application definition are libraries and members.

Analyzing and Refining an Application Definition

Requesting analysis of an application definition saves the current definition and invokes the Analytical Engine to run against the items in the saved application definition. You can specify these analysis parameters:

- Incremental, Full, or Auto analysis
- Analytical Engine parameters, also known as analyze options
- Analysis storage requirements
- Choice of including Semantic Linking in the analysis
- Choice of analysis type: Alliance, Recap, or Estimate
- Choice of using the analysis recovery mechanism

2 Application Analysis Process Overview

This chapter explains how to successfully complete application definition and analysis tasks, and contains these sections:

Section	Page
About the Process	3
Step 1 - Assessing Your Task	4
Step 2 - Defining the Application	5
Step 3 - Analyzing and Refining	7

About the Process

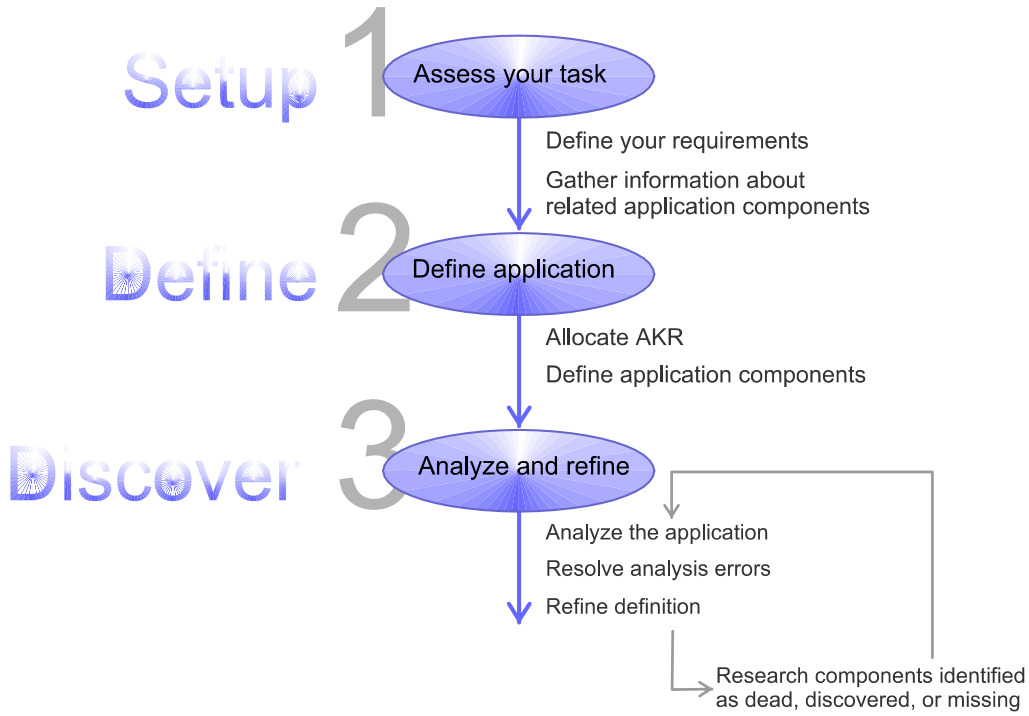
The ESW family of products provide automated, interactive tools you can use to analyze and understand the nature and relationships of application components.

As shown in Figure 1 on page 4, the application definition and analysis process takes place in three steps:

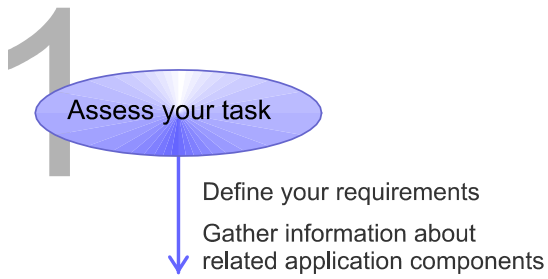
- Assessing your task
- Defining your application
- Analyzing and refining your application

These high-level guidelines are designed to facilitate your understanding of how application definition and analysis work. Detailed instructions on performing the series of actions that constitute each step are provided in subsequent chapters referenced under each heading.

Figure 1 • Application definition and analysis process outline



Step 1 - Assessing Your Task



In this step, define the task you have been assigned and determine the applications and application components affected by that task. Also, gather information about the application in order to create its definition. Whatever the requirement is, careful planning before you start any ESW product saves both time and resources. This stage is an effective step toward the successful implementation of your goals.

Defining Your Requirements

Your initial task is to delineate the scope of the requirement by identifying the change and the applications involved. For example, your assignment may be to determine which components of the Manufacturing application are affected by a larger part number field and provide the information necessary to develop a project plan for converting this application.

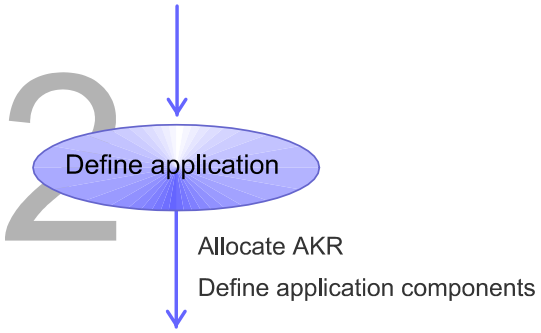
Note:

For step-by-step instructions, see "Defining the Application" on page 27.

Collecting Information About Related Application Components

Collecting data about affected applications is performed manually. The Application Inventory Worksheets section of this guide (see Chapter 3, "Assessing Your Task," on page 11) provides a series of forms you can use to organize the information. Copy and fill out these worksheets, as needed, to begin the application definition process. See "Gathering Information About Related Application Components" on page 11 for more information about Application Inventory Worksheets.

Step 2 - Defining the Application



In this step, you need to prepare the Application Definition Facility to perform the analysis of your application. First, you allocate a repository space for the storage of data, called the AKR. Then, you provide information about the application, the process called application definition. This definition establishes the boundaries of the application slated for maintenance. From this context, the Application Definition Facility identifies the components targeted for assessment and performs the analysis.

Allocating an AKR

The AKR is the repository for all information used by the ESW products. ESW provides both online and batch utilities for managing the AKR. The file organization for an AKR is BDAM or VSAM. You can create a new AKR, or you can expand an existing one. Each AKR can contain multiple applications, or you can define one application per AKR. You may want to allocate one AKR for each major business unit in your company (e.g., manufacturing, accounting, and sales). The size of the AKR depends on the number and size of the application definitions it contains.

When the AKR is allocated, you can start defining your application.

Note:

For step-by-step instructions, see Appendix D – Managing the AKR in the *ASG-Alliance User's Guide*.

Defining Application Components

An application is a group of programs and associated components that perform a related function. An application definition is a list of items that define the components in an application. Using this inventory, the Application Definition Facility invokes the Application Analytical Engine (AAE). The AAE analyzes the items in the list to determine the components of the application and their inter-relationships. You can create a new application definition, open an existing one, or import an application definition from another source.

An often-used technique for organizing applications is to put all the applications of a business unit into a single AKR. For example, the applications for an accounting business may include Accounts Payable, Accounts Receivable, General Ledger, Fixed Assets, Inventory, and Payroll.

To create the application definition within the AKR, specify the definition of the components in their source form. The types of source include program source, load modules, JCL, CICS, and IMS system definition instructions. The source exists in libraries and members, except for sequential files. Therefore, the items in an application definition are libraries and members.

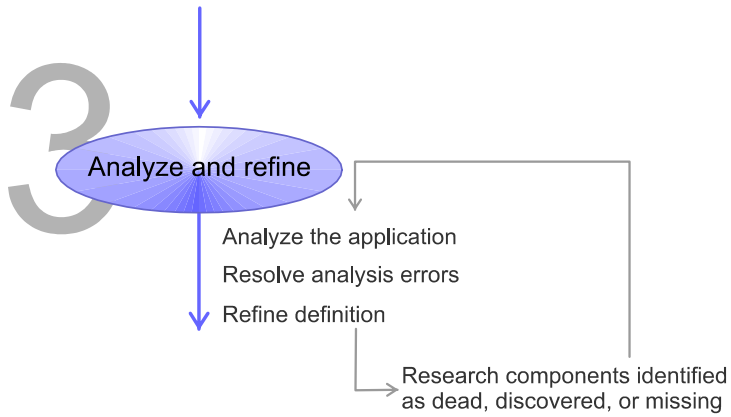
Specify attributes for each library and member in the definition. For example, you can add analysis parameters, copy/include libraries, or preprocessor information. You must define libraries separately for each type of source its members contain (e.g., COBOL and JCL).

Before submitting the application for analysis, you can view the application definition to verify the specified components. After you review the data (libraries, application attributes, and library and member definitions) and make any necessary changes, you can analyze the application and review the results of the application analysis.

Note:

For step-by-step instructions, see "Defining the Application" on page 27.

Step 3 - Analyzing and Refining



In this step you need to analyze and refine the application definition to finalize the analysis results. Defining an application is an iterative process consisting of these steps:

- Run the analysis
- Review the results
- Make modifications
- Rerun the analysis until you receive the expected analysis results

During the analyze job, the product may detect problem areas and stop. The information provided in this chapter enables you to perform analysis recovery and restart procedures. You can also work with resource estimates, analysis options, and batch submission of the analyze job. The analysis summary report provides you with analysis results.

The analysis identifies discovered, missing, or dead application entities inside your application (see "Researching Components Identified as Discovered, Missing, or Dead" on page 10 for more information). With each analyze job, you assess these components to determine their validity. Not only does this verify the accuracy of the collected data, but it enables you to discover unaccounted aspects of your application.

Output from the analyze job is stored in the AKR where you can retrieve it by using the online facilities. This output constitutes the data ESW products use in performing tasks. You can also export the data to a database manager.

Analyzing the Application

The Application Analytical Engine (AAE) analyzes the contents of the application specified in the application definition. You can perform an analysis on an entire application, or incrementally on sections of an application.

- A full analysis analyzes every member in the application and can require a large amount of resources.
- An incremental analysis analyzes only the members you select and is less resource-intensive.
- An auto analysis is a batch analysis job that automatically selects all members that have not been analyzed, have bad analyze return codes or have been modified since the last analyze. See "Analyze Resource Estimates" on page 81 for more information about estimating analyze resources.

The last piece of the analysis is semantic linking, a process that defines the relationships between the components of the application. Semantic linking is a resource-intensive phase of the analyze job that determines how collected entities relate to each other. These relationships are shown as lines connecting components when you perform the detailed analysis. The semantic linking phase is required before you can use the information in the AKR for impact assessment.

If your application is large, you may not have the resources to run a full analysis. In this case, you would only perform incremental analyses on portions of your application, with the semantic linking option disabled. Follow the same process on each section as if analyzing the entire application (submit the section, review the results, modify the application definition, and resubmit). After all the components in the application definition have been analyzed, submit an application analysis batch job with the semantic linking option enabled.

Application analysis produces a summary report named VIAARPT that contains run-time statistics and diagnostic messages for each analyze job. You can use this report to help understand how the job ran, the return code from the individual analysis of each member, and to identify analysis errors. For example, you may have forgotten to define a copy library or missed a syntactic error. This file is temporary and is not stored in the AKR. It is an output product from running the analyze job.

Note:

For step-by-step instructions, see "Analyzing the Application" on page 79.

Resolving Analysis Errors

Occasionally, lack of CPU time, limited DASD space, or other operating system problems can prevent the application analysis from completing. Unless you change the default setting for recovery, you can restart the analysis after resolving the problem. To protect the data previously collected, the Application Definition Facility produces a temporary copy of the AKR with each analyze.

Semantic linking is a common area for abends stemming from excessive CPU use. To prevent this from occurring, you can reduce resource requirements by waiting until you have resolved all other definition problems before running the analysis for semantic linking.

With each analyze job, the product produces a log file of product-related errors named VIALOG. Use this log to find out why an error occurred where it did and to determine what can be done to fix the problem. You can also look up explanations in the online help following the instructions provided in this chapter.

Data files are produced with each analyze job to track the analysis process, and to help identify and resolve analysis errors. VIAARPT is a file summarizing analysis activities and results. Each analyzed item displays in a list with a return code. Return codes not equal to zero identify errors or warnings. VIAPRINT is a file detailing the analysis results that are summarized in VIAARPT. VIAPRINT lists information required to identify and resolve errors for each analyzed item, if applicable.

Note:

For step-by-step instructions, see "Resolving Analysis Errors" on page 95.

Refining the Application Definition

When analysis errors are resolved, refine the application definition. Begin by determining the validity of the components detected by the analysis as potential problems. Then reanalyze the application, resolve errors, and identify any questionable components until the analysis results are as resolved as possible. Repeat this cycle until all questionable items have been addressed. A clean output from the analysis (return codes of 0 through 8) ensures the accuracy of the impact assessment.

Note:

While return codes of 8 are usually considered acceptable, a return code of 8 for an Assembler module is considered an error by the AAE.

For step-by-step instructions, see "Refining the Definition" on page 95.

Researching Components Identified as Discovered, Missing, or Dead

The application analysis produces lists of these three types of questionable components:

- Dead components - items that are part of the application definition, but are not used by any other component of the application.
- Discovered components - items that were not present in the original application definition, but were referenced by a component item.
- Missing components - items that were present at the time of the original definition, but can no longer be found.

For more information, see "Researching Dead, Discovered, and Missing Components" on page 96.

To refine the application definition, review each item identified to determine if any action should be taken. For example, before resubmitting the definition for analysis, you may want to add discovered components to the definition, redefine or remove references to missing components, or decide whether dead components should be removed from the definition.

When you have refined the definition, you are ready to resubmit the application for a new analysis. Continue this cycle until you have accurately resolved all dead, discovered, and missing components.

When application analysis is complete, you can review the data collected about application entities and relationships.

Note:

For step-by-step instructions, see "Researching Dead, Discovered, and Missing Components" on page 96.

3 Assessing Your Task

This chapter describes the process of task assessment, provides worksheets and instructions to use for gathering application inventory information, and contains these sections:

Section	Page
Defining the Requirements	11
Gathering Information About Related Application Components	11

Defining the Requirements

Your initial task is to determine the scope of the software change requirement by identifying the change and the applications involved.

Gathering Information About Related Application Components

This chapter provides a series of forms you can use to organize the information. These are the Application Inventory Worksheets:

Worksheet name	Identifies…	See Page
Application Background Information	• Application name	13
	• Application functionality description	13
	• Application environment description	13
	• Application interfaces description	14

Worksheet name	Identifies…	See Page
Program Source Code Description	• Program source library dataset name	14
	• Program source library - Source manager type	14
	• Program source library default language level	15
	• Program copy/include/maclib libraries	15
IDMS Library Description	Source library/JCL dataset name	17
IDMS DD and Dataset Names	IDMS DD and dataset names	17
Compile/Link-Edit JCL Description	Compile/Link-edit JCL library dataset name	18
Compile/Link-Edit JCL Library Members	JCL member name and description	18
Load Library Description	Load library dataset name	19
Load Library/Module Members	Load module names	19
JCL/PROC Library Description	• JCL library members	20
	• Catalogued procedure libraries	21
CICS Component Source Library Description	• CICS FCT, PCT, PPT tables and BMS map source libraries	22
	• CICS FCT, PCT, PPT tables and BMS map source members	22
	• CSD files	23
	• CICS/MACLIBS/copylibs	23
IMS Component Source Library Description	• IMS source libraries	24
	• IMS source members	24
	• IMS MACLIBS/copylibs	25
	• IMS stage 1 libraries	25
Other Language Component Source Library Description	• Other language source libraries	26
	• Other language source members	26

Application Background Information Worksheets

Use these forms to document application background information.

Application name: (expand/explain acronyms)

Application functionality description (What does it do?)

Application environment description: (Percentage of Batch vs. online, CICS, or IMS, DB2, IDMS, etc.)

Application interfaces description: (Are there external sources and destination of data?)

Program Source Code Description Worksheets

Use these forms to specify information about a program source library. If the application source resides within multiple source libraries, prepare a separate form for each library.

Program source library dataset name:

Program source library - Source manager type: (Circle one)					
PDS	Panvalet	Librarian	Sequential	Endevor	Other
If other, describe here:					

Program source library default language level: (Circle one - make choice based on majority)				
COBOL 68	COBOL 74	COBOL II	COBOL II R3	COBOL/370
COBOL for MVS and VM		PL/I 1.5		PL/I 2.x
ASMF		ASMH		HLASM

Program copy/include/maclib libraries: (List in proper concatenation sequence)

Program Source Members Worksheet

A prefix/suffix pattern match on a member name is acceptable, for example, VIA*.

Program source member name	Language level (default override)		Program source member name	Language level (default override)

Program source member name	Language level (default override)		Program source member name	Language level (default override)

IDMS Library Description Worksheet

Use these forms to specify IDMS information, if any of the COBOL, PL/I, or ASM source modules contain IDMS statements. If the modules reside within multiple source libraries, prepare a separate form for each library.

Source library dataset name	Compile/Link JCL dataset name for members containing IDMS statements

IDMS DD and Dataset Names Worksheet

Specify the IDMS DD and dataset names in proper concatenation sequence.

IDMS DD names	Dataset name of DD

Compile/Link-Edit JCL Description Worksheet

Use these forms to specify information about a compile/link-edit JCL library. If the application compile/link-edit JCL resides within multiple JCL libraries, prepare a separate form for each library. Identify at least one example of each type of compile necessary to support the application.

Compile/link-edit JCL library data name

Compile/Link-Edit JCL Library Members Worksheet Load Library/Modules Description

JCL member name	Description (Pre/post-processors, etc.)

Load Library Worksheets

Use these forms to specify information about a load library. If the application load modules reside within multiple load libraries, prepare a separate form for each library.

Load library dataset name

Load Library Members Worksheet

A prefix/suffix pattern match on a member name is acceptable, for example, VIA*

Load module		Load module		Load module

JCL/PROC Library Description Worksheets

Use these forms to specify information about execution JCL and PROC libraries. If the application JCL resides within multiple JCL libraries, prepare a separate form for each library.

JCL Library Members Worksheet

A prefix/suffix pattern match on a member name is acceptable, for example, VIA*.

JCL member		JCL member		JCL member

Cataloged Procedure Libraries Worksheet

PROC library dataset name: (List in proper concatenation sequence)

CICS Component Source Library Description Worksheets

Use these forms to specify information about CICS FCT, PCT, and PPT tables, and BMS map source libraries. If the application CICS components reside within multiple CICS source libraries, prepare a separate form for each library.

CICS FCT, PCT, PPT Tables and BMS Map Source Libraries Worksheet

CICS source dataset name

CICS FCT, PCT, PPT Tables, and BMS Map Source Members Worksheet

A prefix/suffix pattern match on a member name is acceptable, for example, VIA*.

Member name	Member type (FCT, PCT, PPT tables, BMS maps)		Member name	Member type (FCT, PCT, PPT tables, BMS maps)

CSD Files Worksheet

CSD file name	Group name

CICS/MACLIBS/Copylibs Worksheet

MACLIB/copylib dataset name (Specify in proper concatenation sequence)

IMS Component Source Library Description Worksheets

Use these forms to specify information about IMS PSB, DBD, MFS, and DFSMDA source libraries, and Stage 1 input libraries. If the application IMS components reside within multiple IMS source libraries, prepare a separate form for each library.

IMS Source Libraries Worksheet

IMS source dataset name

IMS Source Worksheet

A prefix/suffix pattern match on a member name is acceptable, for example, VIA*.

Member name	Member type (PSB, DBD, MFS, DFSMDA, Stage 1)		Member name	Member type (PSB, DBD, MFS, DFSMDA, Stage 1)

IMS MACLIBS/Copylibs Worksheet

MACLIB/copylib dataset name (Specify in proper concatenation sequence)

IMS Stage 1 Libraries Worksheet

Stage 1 dataset name

Other Language Source Library Description Worksheets

Use these forms to specify information about PL/1, Assembler, NATURAL, or other language source libraries. If the application components reside within multiple source libraries, prepare a separate form for each library.

Other Language Source Libraries Worksheet Other Language Source Worksheet

Source dataset name

A prefix/suffix pattern match on a member name is acceptable, for example, VIA*.

Member name	Member type		Member name	Member type

4 Defining the Application

This chapter provides detailed instructions about creating an application definition and contains these sections:

Section	Page
Overview	27
Creating the Application	29
Defining Application Components	31
Reviewing the Application Definition	71
Modifying the Application Definition	72
Importing the Application Definition	78
Exporting the Application Definition	78

Overview

The AAE analyzes the application definition to provide information that is used by Estimate, Alliance, and Recap to define the scope of software change requirements.

Figure 2 on page 28, Figure 3 on page 28, and Figure 4 on page 29 show the application definition hierarchy at increasing levels of complexity.

In Figure 2 on page 28, the basic components of the application definition hierarchy are the AKR, applications, libraries, and members.

These are the relationships between these components:

- AKR contains applications
- Applications have libraries
- Libraries have members

Figure 2 • Application Definition Architecture 1

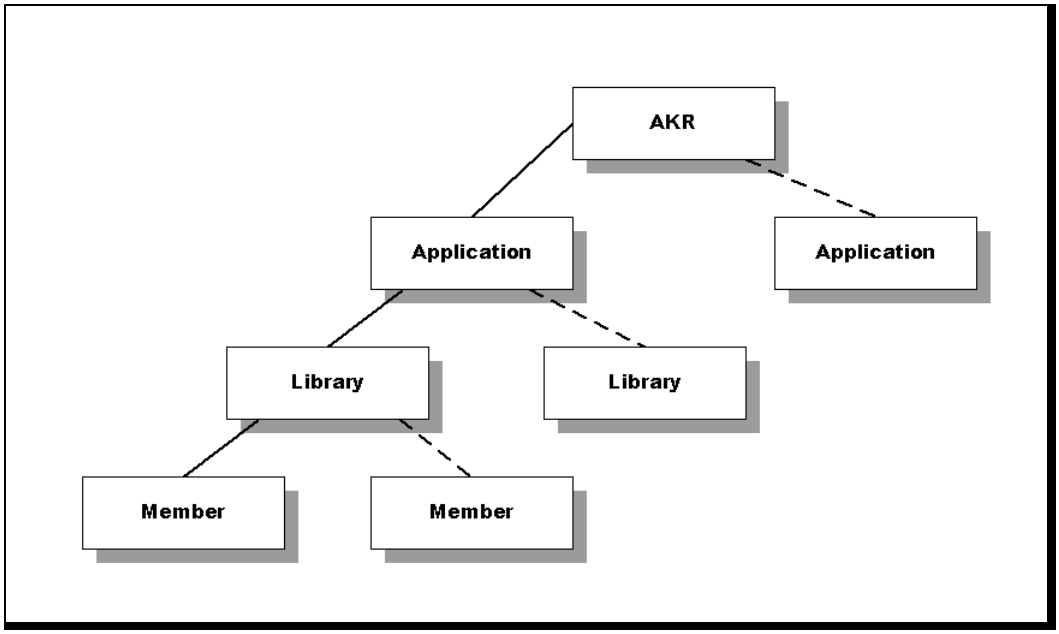


Figure 3 shows this information:

- An application can have more than one type of library.
- The library and its members are categorized by what they contain.

Figure 3 • Application Definition Architecture 2

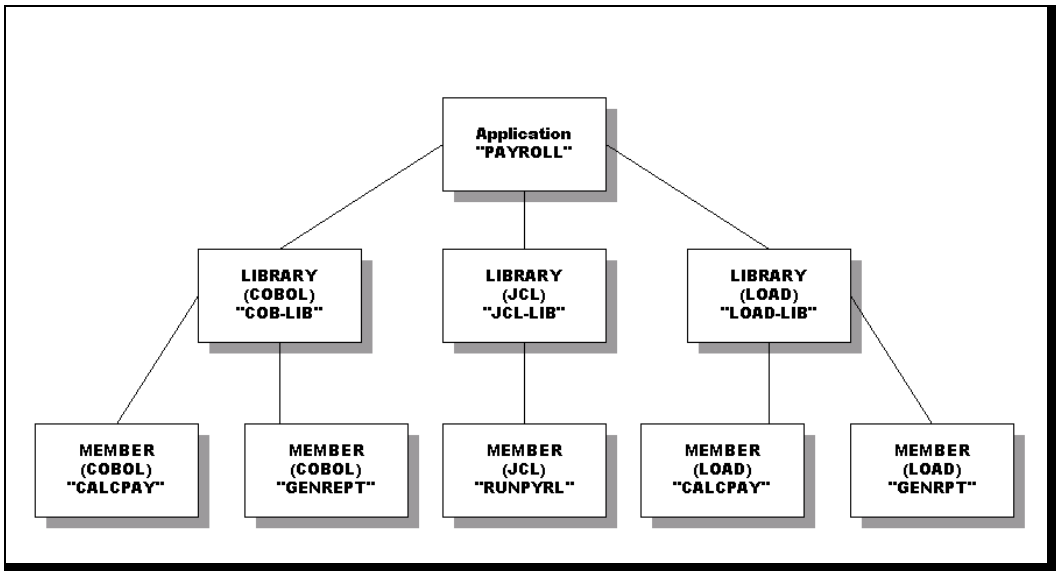
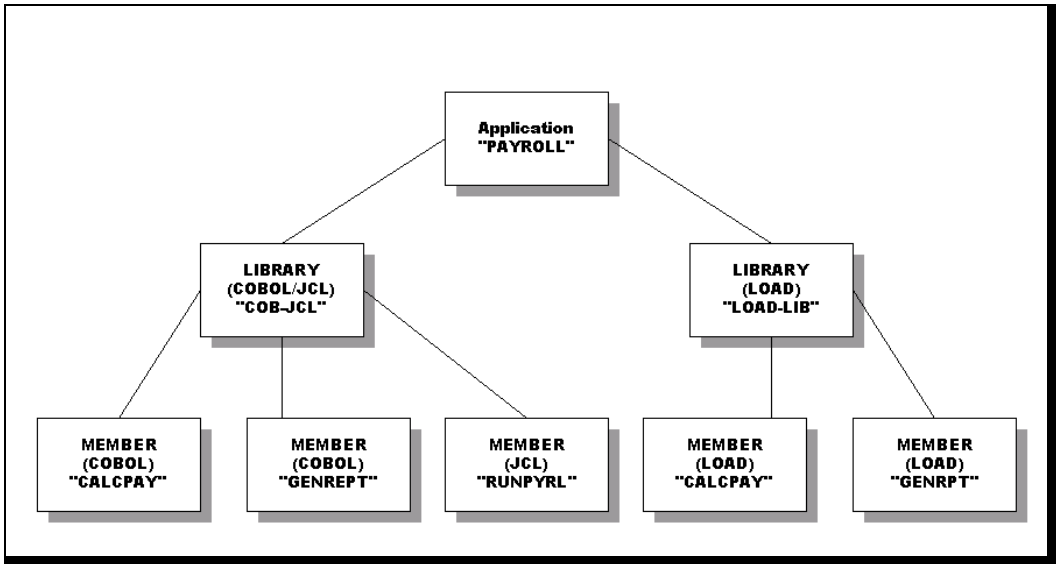


Figure 4 shows this information:

- A library may contain multiple types of members.
- A member may contain only one type of component.
- The member can inherit the options of the library or override them with unique options.

Figure 4 • Application Definition Architecture 3



Creating the Application

After gathering the required information and allocating the AKR, you are ready to create an application definition. As described in Chapter 2, "Application Analysis Process Overview," on page 3, an application is a group of programs and associated components that perform a related function. Multiple applications can be grouped together in an AKR that represents a single business unit, such as accounting. For example, typical applications within an accounting AKR would be Accounts Payable, Accounts Receivable, General Ledger, Fixed Assets, Inventory, and Purchasing.

The process of creating an application definition includes these definitions:

- AKR dataset name where the application information is stored
- Application name and an optional description
- Program source libraries, attributes, and source members
- Load libraries and module members
- JCL libraries and members

- CICS CSD files and group names
- CICS libraries for FCT, PCT, PPT tables and BMS maps, and members of each
- IMS libraries and members

Creating an Application Definition

To create an application

1 Select File ▸ New on the primary product screen to display the File - New Application pop-up.

2 Enter the appropriate information shown in this table:

Field Name	Specifies the...
Data set name field	Name of the AKR where the source resides. If the high-level qualifier of the name is different than your TSO prefix or user ID, enter the name in single quotes ('').
Application name field	Name of application (up to 10 characters). For example, if you are defining a General Ledger application within an accounting AKR, an appropriate application name is GEN-LEDG.
Volser field	Volume serial number, if the AKR is not cataloged.
Password field	Password, if the AKR is password protected.

Note:

When creating an application definition, select the product feature for which analysis is required, choosing from currently installed products (Alliance, Estimate, or Recap). You can perform a Recap analysis on an application defined for Alliance, if Recap is also installed, and vice versa.

To define a new application in Estimate, select Add Application Node on the Enterprise View.

3 Press Enter to return to the primary screen. A message displays indicating the status of the application (e.g, `APPLICATION WAS SUCCESSFULLY CREATED`).

To create an application description

To further describe an application, you can type an optional long description (up to 40 characters) to describe the application's business function.

1. Select Edit ▸ Application description on the primary product screen or the Application Maintenance Facility (AMF) screen. The Edit - Application Description pop-up displays with the user ID of the person that created the application displayed in the Description field, as the default description.

2. Type over existing text in the Description field to change the default description.

3. Press Enter to save the description and return to the primary screen.

Defining Application Components

After creating the application in the AKR, specify the libraries and members to be included in the application definition.

Note:

See "The Application Maintenance Facility" on page 103 for information about defining application components using the AMF.

Defining COBOL Source Libraries

Before specifying the COBOL programs to include in the application definition, you must define the libraries in which the programs are located.

To define COBOL libraries

1. Select Edit ▸ COBOL Definition on the primary product screen to display the Edit - COBOL Definition pop-up.

2. Select Libraries.

   If this is the first library to be defined, the Add Source Library pop-up displays. If at least one library is already defined to the application, the Application - COBOL Libraries pop-up displays. Press PF6 (Add) to access the Add Source Library pop-up.

3 In the Add Source Library pop-up, enter the appropriate information:

Field Name	Specifies the...
Data set name field	Name of the dataset where the source resides. If the high-level qualifier is different than your TSO prefix or user ID, enter the name in single quotes (''). For an Endevor source manager, enter the Endevor location inventory name as a dataset name. For more information about specifying Endevor source managers, see "Specifying an Endevor Source Manager" on page 33.
Source manager field	Number selection of the dataset's source manager. You must specify a source manager for each library.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

4 Press PF6 to add the library.

A message displays to indicate that the dataset was added. The Add Source Library pop-up remains open so you can define additional libraries.

Note:

If the source manager is CA-Librarian, the Add Library - Librarian pop-up displays for you to specify the Librarian attributes, if any.

5 Press PF3 to exit the Add Source Library pop-up when you have defined all the COBOL libraries. The Application - COBOL Libraries pop-up displays, listing the COBOL libraries defined to the application.

You can specify library attributes, modify or remove libraries, or add additional libraries to the definition.

Modifying the COBOL Version

The COBOL version that displays in the Type field of the Application - COBOL Libraries pop-up indicates the default COBOL version specified during product installation.

To modify the COBOL version for a program

1. Type any character in the selection column next to the library.
2. Press PF4 (Options) to display the COBOL Library Options pop-up.
3. Select the Analysis parms option to specify the correct version.

After defining the libraries, you are ready to specify the attributes for each library. These attributes include analysis parameters, COPY libraries, IDMS information, and procedure libraries.

Specifying an Endevor Source Manager

You must supply the location inventory name for each application to correspond to the hierarchical category names used within Endevor/MVS. For example, Endevor/MVS stores elements by type, in stages within the subsystem, within the system, and within the environment. The location inventory name must be defined as a dataset name for each unique combination of environment, system, subsystem, type, and stage.

Endevor/MVS controlled applications have these operation and customization considerations:

- The location inventory name cannot be checked for accuracy or validity against the Endevor hierarchy names. An invalid DSN (for example, an invalid Endevor System name) results in a `No members found` message or an aborted analyze.
- You must specify the Endevor CONLIB dataset in the installation parameter ENDEVOR-CONLIB.
- You must specify the Endevor loadlib dataset name in the installation option ENDEVOR-STEPLIB.

Note:

For more installation information, see the *ASG-Center Installation Guide*.

To specify an Endevor source manager

1. Access the Add Source Library pop-up.

2. Enter the dataset name with five nodes, one for each hierarchical name, in descending hierarchical order, as shown in this example.

   Format:

   '<Environment>.<System>.<Subsystem>.<Type>.<Stage Number ID>'

   Example:

   `'VIATEST.FINANCE.PAYROLL.COBOL.T'`

   Make sure that each node of the location inventory name DSN exactly matches its Endevor/MVS counterpart. From the DSN format, the product extracts the environment (VIATEST), system (FINANCE), subsystem (PAYROLL), type (COBOL), and stage (T) names, and uses them to access the Endevor/MVS database.

   If you use a numeric Stage number instead of an alphabetic stage ID, append the number 1 or 2 to the literal prefix STAGE#, as shown in this example:

   Format:

   '<Environment>.<System>.<Subsystem>.<Type>.<Stage Number ID>'

   Example:

   `'VIATEST.FINANCE.PAYROLL.COBOL.STAGE#1'`

3. Type 5 in the Source Manager field to select Endevor.

4. Press Enter to define the dataset. The message `DATASET ADDED` displays in the short message area.

Specifying COBOL Library Attributes

For each COBOL library defined to the application, you can specify these attributes:

- Analysis (compile) parameters
- Copy libraries
- IDMS information
- Procedure libraries

These attributes are used to analyze the application. Attributes specified at the library level apply to every program defined in the library, unless overridden at the program level, as described in "Specifying COBOL Program Attributes" on page 40.

Specifying Analysis Parameters (COBOL Libraries)

You must specify analysis parameters for a library when any of these situations apply:

- An incorrect COBOL version for the library displays in the Application - COBOL Libraries pop-up.
- The library members to be selected contain DB2 statements.
- Execution parameters are used by the source manager or compiler to successfully execute members (for Librarian and Panvalet).
- The compile/link JCL contains a procedure that is cataloged.

To specify analysis parameters

1. Access the Application - COBOL Libraries pop-up.

2. Select the desired library by typing any character in the selection column and pressing PF4 (Options) to display the COBOL Library Options pop-up.

   You can specify the same attributes for several libraries by selecting multiple libraries.

3. Select Analysis parms and press Enter to display the Library Options - Analysis Parms pop-up.

   Specify that information be extracted from the fields in the compile JCL using the extract function. You can also enter the information manually.

   If you choose extraction, this information is extracted and placed in the appropriate fields:

Field	Description
COBOL Version	Displays the site default initially; if this is correct, it is not changed.
DB2 STDSQL	Specifies that if the compile/link JCL contains a DB2 preprocessor step, this field value is extracted from the JCL; otherwise, the site default displays.

Field	Description
Src Mgr Exec Parms	Specifies that execution parameters are used by the source manager.
Compile Exec Parms	Specifies that execution parameters are used by the compiler.

In addition to the above information, any copy libraries found in the compile/link JCL are extracted and added as attributes of the selected library or libraries.

Note:

If you receive a JCL error, see Defining Cataloged JCL Procedures.

Defining Cataloged JCL Procedures

If you receive a JCL error when you perform an extract, an ISPF browse screen displays with information about the error.

To define a cataloged procedure

1. Review the JCL in error to locate the undefined procedure.
2. Identify the procedure library that contains the procedure.
3. Press PF5 in the Library Options - Analysis Parms pop-up to add the procedure library to the application definition.
4. Repeat the extract procedure introduced in step 2 on page 35.
5. Press Enter or PF3 to save the information in the Library Options - Analysis Parms pop-up and return to the COBOL Library Options pop-up.

   A message displays to indicate whether any copy libraries were extracted.

Specifying Copy Libraries (COBOL Libraries)

If any members in the COBOL library use copy members, and the copy libraries were not extracted in the Library Options - Analysis Parms pop-up or if you want to add additional copy libraries, you can use the Maintain copylibs option to specify the copy libraries.

To specify copy libraries

1. Access the Application - COBOL Libraries pop-up.
2. Select the desired library by entering any character in the selection column and pressing PF4 (Options) to display the COBOL Library Options pop-up.

3 Select Maintain Copylibs if no copy libraries have been defined for the selected library. The Add Copylib pop-up displays. If at least one copy library is defined, the Library Options - Maintain Copylibs pop-up displays. Press PF6 from this pop-up to access the Add Copylib pop-up.

The copy libraries should have the same concatenation order as in your compile JCL SYSLIB.

You can specify the same attributes for several libraries by selecting multiple libraries.

Note:

To add a copy library before a particular library already listed in the Library Options - Maintain Copylibs pop-up, select the library the new copy library is to precede and press PF6 to add the new copylib. The new copy library is listed before the selected copy library in the Library Options - Maintain Copylibs pop-up.

4 In the Add Copylib pop-up, enter the appropriate information:

Field Name	Specifies the...
Data set name field	Name of the library that contains the copy member.
Source manager field	Number selection of the copy library's source manager.
Usage field	Where the SYSLIB dataset can be found in the compile JCL. This information is used to determine when the copy library is processed during the analyze: during COBOL copy library processing, during DB2 include processing, or during both processes.
Subsystem field	Any character if the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The SUBSYS parameter must have been previously defined during product installation.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

5 Press PF6 to add the copy library as an attribute of the selected library. The Add Copylib pop-up remains open so you can add additional copy libraries.

6 Press PF3 when you have defined the last copy library to return to the Library Options - Maintain Copylibs pop-up.

7 Press PF3 twice to return to the Application - COBOL Libraries pop-up.

Specifying IDMS Information (COBOL Libraries)

If any members in a library use IDMS, you need to specify IDMS information for the library.

To specify IDMS information

1. Access the Application - COBOL Libraries pop-up.

2. Select the desired library and press PF4 to display the COBOL Library Options pop-up. You can specify the same attributes for several libraries by selecting multiple libraries.

3. Select IDMS information and press Enter to display the Library Options - IDMS Information pop-up. Specify information to be extracted from the compile JCL, or enter the information manually, such as STEPLIB, SYSCTL, DICTDB, DLODDB, and SYSIDMS. If any other DD names needed by IDMS are in the specified JCL, press PF6 to add them to the list of DDs.

4. Type the compile/link JCL dataset and member in the JCL DSN to Extract field and press PF4 (Extract JCL) to extract the IDMS DD and dataset names and the IDMSDMLC execution parameters.

   Note:

   If you get a JCL error when you perform the extract, see "Defining Cataloged JCL Procedures" on page 36.

5. Press PF3 to save the information on the Library Options - IDMS Information pop-up and return to the COBOL Library Options pop-up. If no additions or deletions were made to the default DD/DSN information, the Default DD Confirmation pop-up displays to confirm that you want to use the default DD information.

6. Select Exit from the COBOL Library Options pop-up to return to the Application - COBOL Libraries pop-up. From this pop-up, press PF3 repeatedly to return to the primary screen.

Adding COBOL Programs

After you have defined COBOL libraries, you can select the programs in these libraries to be included in the application definition.

To select programs to be included in the application definition

1. Select Edit ▸ COBOL Definition on the primary product screen to display the Edit - COBOL definition pop-up.

2. Select Members and press Enter to display one of these windows:

If you defined	Result
Only one COBOL library and no programs from that library are defined	The Add Member pop-up displays.
Only one COBOL library and that library contains at least one defined program	The Application - COBOL Library Member List pop-up displays. Press PF6 (Add) to access the Add Member pop-up.
At least two COBOL libraries	The Application - COBOL Library Selection pop-up displays. To define programs for one of the libraries listed, select the library name by typing any character in the selection column and pressing Enter.

3. Type a name or pattern to filter the list of programs available for selection in the Add Member pop-up. For example, to select from a list of all programs in the library that begin with an M, enter `M*` in the Name field. To view a list of all programs in the library, leave the Name field blank.

4. Press PF5 (Directory) to display the Members - Selection List pop-up.

5. Select the desired programs by typing any character in the selection column. Previously-selected members are marked with an asterisk (*) and cannot be selected. To select all programs in the library, press PF5 (Select All). Scroll the list if you need to view all members.

6. Press PF3 to return to the Add Member pop-up. You can enter another name pattern and repeat step 4 through step 6 to add more programs.

7. When you have finished defining programs, press PF3 to return to the Application - COBOL Library Member List pop-up, which lists the programs defined to the application.

Modifying the COBOL Version

The COBOL version that displays in the Type field of the Application - COBOL Library Member List pop-up indicates the default COBOL version specified for the entire library.

To modify the COBOL version for a program

1. Type any character in the selection column next to the program.
2. Press PF4 (Options) to display the COBOL Member Options pop-up.
3. Select the Analysis parms option to specify the correct version and press PF3 to exit.

Specifying COBOL Program Attributes

For each COBOL program in the application definition, you can specify these attributes to analyze the COBOL program:

- Analysis (compile) parameters
- Copy libraries
- DMS information
- Procedure libraries

You only need to specify attributes at the program level when a program has different attributes than those specified for the library. For example, if a program uses a copy library that was not defined at the library level, you must specify the copy library attribute for the program.

The attributes defined at the program level override any attributes specified for the library containing the program.

Specifying Analysis Parameters (COBOL Programs)

Analysis parameters must be defined for a program only if the parameters are different than those specified for the library containing the program in these situations:

- The COBOL version for the program is different than the one listed in the Application - COBOL Library Member List pop-up.
- The program contains DB2 statements.
- Execution parameters are used by the source manager or compiler to successfully execute the program.
- The compile/link JCL contains a procedure that is cataloged (this applies only if you are extracting compile JCL to identify the three items above).

To specify analysis parameters

1. Access the Application - COBOL Library Member List pop-up.

2. Select the desired program and press PF4 to display the COBOL Member Options pop-up. You can specify the same attributes for several programs by selecting multiple programs.

3. Select Analysis parms and press Enter to display the Member Options - Analysis Parms pop-up.

4. Specify that information be extracted for the fields from the compile JCL using the extract function, or enter the information manually. The format of the Member Options - Analysis Parms pop-up is identical to the Library Options - Analysis Parms pop-up, except for the title.

   This information is extracted and placed in the appropriate fields:

Field	Description
COBOL Version	Displays the site default initially; if this is correct, it is not changed.
DB2 STDSQL	Specifies that if the compile/link JCL contains a DB2 preprocessor step, this field value is extracted from the JCL; otherwise, the site default displays.
Src Mgr Exec Parms	Specifies if execution parameters are used by the source manager.
Compile Exec Parms	Specifies if execution parameters are used by the compiler.

   In addition to this information, any copy libraries found in the compile/link JCL are extracted and added as attributes of the selected program.

5. Type the compile/link JCL dataset and member name in the Compile JCL DSN field and press PF4 (Extract JCL) to extract the information for the Member Options - Analysis Parms pop-up.

   Note:

   If you receive a JCL error when you perform the extract, see "Defining Cataloged JCL Procedures" on page 36.

6 Press PF3 to save the information in the Member Options - Analysis Parms pop-up and return to the COBOL Member Options pop-up. A message displays to indicate whether any copy libraries were extracted.

7 Press PF3 again to return to the Application - COBOL Library Member List pop-up.

Specifying Copy Libraries (COBOL Programs)

If the program uses copy members that are different than those defined for the library and the copy libraries were not extracted, or if you want to add copy libraries, you can use the Maintain copylibs option in the COBOL Member Options pop-up to specify the copy libraries.

To specify copy libraries

1 Access the Application - COBOL Library Member List pop-up.

2 Select the desired program by entering any character in the selection column and press PF4 (Options) to display the COBOL Member Options pop-up. You can specify the same attributes for several programs by selecting multiple programs.

3 Select Maintain copylibs.

4 If no copy libraries have been defined for the selected member, the Add Copylib pop-up displays. If at least one copy library is defined, the Member Options - Maintain Copylibs pop-up displays. The copy libraries listed in the Member Options - Maintain copylibs pop-up are in the same concatenation order used in your compile JCL SYSLIB. Press PF6 from this pop-up to access the Add Copylib pop-up.

Note:

To add a copy library before a particular library already listed in the Member Options - Maintain Copylibs pop-up, select the library the new copy library is to precede and press PF6 to add the new copylib. The new copy library is listed before the selected copy library in the Member Options - Maintain Copylibs pop-up.

If you want to copy the copylib list from the library level to the selected member, press PF5.

Or

5 Enter the appropriate information in the Add Copylib pop-up:

Note:

The format of the Add CopyLib pop-up for program options is identical to the Add Copylib pop-up for library options.

Field	Specifies the...
Data set name field	Name of the library that contains the copy member.
Source manager field	Number selection of the copy library's source manager.
Usage field	Where the SYSLIB dataset can be found in the compile JCL. This information is used to determine when the copy library is processed during the analyze: during COBOL copy library processing, during DB2 include processing, or during both processes.
Subsystem field	Any character if the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The SUBSYS parameter must have been previously defined during product installation.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

6 Press PF6 to add the copy library as an attribute of the selected library.

The Add Copylib pop-up remains open to enable you to enter additional copy libraries.

7 Press PF3 when you have defined the last copy library to exit the Add Copylib pop-up and return to the Member Options - Maintain Copylibs pop-up.

Specifying IDMS Information (COBOL Programs)

If a program uses IDMS statements that are different than those used by the library containing the program, you need to specify IDMS information for the program.

To specify IDMS information

1. Access the Application - COBOL Library Member List pop-up.

2. Select the desired program and press PF4 to display the COBOL Member Options pop-up. You can specify the same attributes for several programs by selecting multiple programs.

3. Select IDMS information and press Enter to display the Member Options - IDMS Information pop-up.

4. Specify that information is to be extracted from the fields in the compile JCL using the extract function, or enter the information manually, such as STEPLIB, SYSCTL, DICTDB, DLODDB, and SYSIDMS. If any other DD names needed by IDMS are in the specified JCL, press PF6 to add them to the list of DDs.

   Note:

   The format of the Member Options - IDMS Information pop-up is identical to the Library Options - IDMS Information pop-up, except for the title.

5. Type the compile/link JCL dataset and member in the JCL DSN to Extract field and press PF4 (Extract JCL) to extract the IDMS DD and dataset names and the IDMSDMLC execution parameters.

   Note:

   If you receive a JCL error when you perform the extract, see "Defining Cataloged JCL Procedures" on page 36.

6. Press PF3 to save the information on the Member Options - IDMS Information pop-up and return to the Member Options pop-up. If you did not make any additions or deletions to the default DD/DSN information, the Default DD Confirmation pop-up displays for you to confirm that you want to use the default DD information.

7. Select Exit from the COBOL Member Options pop-up to return to the Application - COBOL Library Member List pop-up. From this pop-up, press PF3 repeatedly to return to the primary screen.

Removing COBOL Library Attributes

It may become necessary to remove attributes specified at the program level, (i.e., analysis (compile) parameters, copy libraries, IDMS information, and procedure libraries), and revert to the attributes specified at the library level. For example, a program used a copy library not defined at the library level but no longer does because of revisions to the code. When you remove program level attributes, the selected program(s) contain the attributes specified at the library level.

To remove library options

1. Select Edit ▸ COBOL Definition on the primary product screen to display the Edit - COBOL Definition pop-up.

2. Select Members and press Enter to display one of these pop-ups:

If you defined...	Result
Only one library and no other libraries are defined	The Application - COBOL Library Member List pop-up displays.
Only one library and at least one library is already defined	The Application - COBOL Library Selection pop-up displays.

3. Type any character in the selection column next to the program whose attributes are to be removed and press Enter. If the same attributes are to be removed from more than one program, you can select multiple programs.

4. Press PF4 (Options). The COBOL Member Options pop-up displays.

5. Select Restore library options and press Enter to display the Restore Library Options pop-up, which lists the name of the selected library, the first program selected, and the attributes that will be removed.

6. Press PF10 (Remove) to remove the attributes. If you selected multiple programs, the message `ATTRIBUTES REMOVED` displays in the short message area and the next selected program name is displayed. Press PF10 (Remove) for each program selected. When the attributes of the last program are removed, the Application - COBOL Library Member List pop-up redisplays.

7. Press PF3 to return to the Application - COBOL Library Selection pop-up.

Adding Load Libraries

You must define the libraries in which Load Module members are located before specifying the load module members to include in the application definition.

To add a load library

1. Select Edit ▸ Load Module definition on the primary product screen to display the Edit - Load Module Definition pop-up.

2. Select Libraries and press Enter to display one of these pop-ups:

If you defined...	Result
Only one library and no other libraries are defined	The Add Load Library pop-up displays.
Only one library and the application contains at least one library already defined	The Application - Load Library pop-up displays.

3. Type the name of the dataset for the load library in the Data set name field. If the high-level qualifier is different than your TSO prefix or user ID, enter the name in single quotes (').

4. Press PF6 to add the library. A message indicates the dataset is added. The pop-up remains open to define additional libraries.

5. Press PF3 when you are finished to return to the Application - Load Libraries pop-up. From this pop-up, you can specify, modify, or remove a library, or add additional libraries to the application definition.

Adding Load Module Members

After you define a load library, you can select members in that library to be included in the application definition.

To add load module members

1. Select Edit ▸ Load Module definition on the primary product screen to display the Edit - Load Module Definition pop-up.

2. Select Members and press Enter to display one of these pop-ups:

If you defined...	Result
Only one load library and no members from that library are defined	The Add Member pop-up displays.
Only one load library and that library contains at least one defined member	The Application - Load Library Member List pop-up displays. Press PF6 (Add) to open the Add Member pop-up.
At least two load libraries	The Application - Load Library Selection pop-up displays. To define members for one of the libraries listed, select the library name by typing any character in the selection column and pressing Enter.

3. Type a name or pattern to filter the list of programs available for selection in the Add Member pop-up. For example, to select from a list of all programs in the library that begin with an M, enter `M*` in the Name field. To view a list of all programs in the library, leave the Name field blank.

4. Press PF5 (Directory) to display the Members - Selection List pop-up.

5. Select the desired programs by typing any character in the selection column.

6. Previously-selected members are marked with an asterisk (*) and cannot be selected. To select all programs in the library, press PF5 (Select All). Scroll the list if you need to view all the members.

7. Press PF3 to return to the Add Member pop-up. You can enter another name pattern and repeat step 4 through step 6 to add more programs.

8. Press PF3 when you have finished defining members to return to the Application - Load Library Member List pop-up, listing the programs defined to the application.

Adding JCL Libraries

You must define the libraries in which JCL members are located before specifying the JCL members to include in the application definition.

To add a JCL library

1. Select Edit ▸ JCL Definition on the primary product screen to display the Edit - JCL Definition pop-up.

2. Select Libraries and press Enter to display one of these pop-ups:

If you defined...	Result
Only one library and no other libraries are defined	The Add JCL Library pop-up displays.
Only one library and the application contains at least one library already defined	The Application - JCL Library pop-up displays. Press PF6 (Add) to access the Add JCL Library pop-up.

3. In the Add JCL Library pop-up, enter the appropriate information:

Field	Specifies the...
Data set name field	Name of the dataset where the source resides. If the high-level qualifier is different that your TSO prefix or user ID, enter the name in single quotes ('').
Source manager field	Number selection of the dataset's source manager. You must specify a source manager for each library.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

4. Press PF6 to add the library. A message displays to indicate that the dataset was added. The Add JCL Library pop-up remains open to define additional libraries.

5. Press PF3 when you have defined all the JCL libraries to exit the Add JCL Library pop-up. The Application - JCL Libraries pop-up displays, listing the JCL libraries defined to the application.

6. From this pop-up, you can specify library attributes, modify, or remove a library, or add additional libraries to the definition.

Specifying JCL Library Attributes

For each JCL library defined to the application, you can specify procedure libraries that are used to analyze the JCL members of the specified library, source manager execution parameters, and user exit parameters.

Attributes defined at the library level apply to every member defined in the library, unless overridden at the program level. For more information about member attributes, see "Specifying JCL Member Attributes" on page 52.

Specifying Source Manager Execution Parameters (JCL Libraries)

If the members within a JCL library have Librarian or Panvalet as source manager and use execution parameters, you need to define them for that library.

To specify source manager execution and user exit parameters

1. Access the Application - JCL Libraries pop-up.

2. Select the desired library by typing any character in the selection column and pressing PF4 (Options) to display the JCL Library Options pop-up.

   You can specify the same source manager execution and user exit parameters for several libraries by selecting multiple libraries.

3. Select Analysis parms and press Enter to display the Library Options - Analysis Parms pop-up.

4. Type the execution parameters required for the source manager or the user exit parameters.

5. Press PF3 to apply the parameters and return to the JCL Library Options pop-up. If you selected multiple libraries, the attributes specified apply to each one.

6. Select Exit to return to the Application - JCL Libraries pop-up.

Specifying Procedure Libraries (JCL Libraries)

If the members within a JCL library use procedure libraries that have not been defined during product installation, you need to specify the procedure libraries for that library. To verify the procedure libraries specified during product installation, consult your systems administrator.

To specify procedure libraries

1. Access the Application - JCL Libraries pop-up.

2. Select the desired library by entering any character in the selection column and pressing PF4 (Options) to display the JCL Library Options pop-up. You can specify the same attributes for several libraries by selecting multiple libraries.

3. Select Maintain proclibs to display the Add Proclib pop-up.

   Note:

   If at least one proclib is defined for the library, the JCL Library - Maintain Proclibs pop-up displays when you press PF4 in step 2. Press PF6 on this pop-up to access the Add Proclib pop-up.

4. Type the dataset name for the procedure library in the Data set name field.

5. Press PF6 to add the procedure library as an attribute of the selected JCL library. The Add Proclib pop-up remains open to enter additional procedure libraries.

6. Press PF3 when you have defined the last procedure library to exit the Add Proclib pop-up and return to the JCL Libraries Options pop-up. You may need to press PF3 more than once.

7. Select Exit to return to the Application - JCL Libraries pop-up. Press PF3 twice to return to the primary screen.

Adding JCL Members

When you have defined a JCL library, you can select members in that library to be included in the application definition.

To select members to be included in the application definition

1. Select Edit ▸ JCL Definition on the primary product screen to display the Edit - JCL Definition pop-up.

2. Select Members and press Enter to display one of these pop-ups:

If you defined...	Result
Only one JCL library and no members from that library are defined	The Add Member pop-up displays.
Only one JCL library and that library contains at least one defined member	The Application - JCL Library Member List pop-up displays. Press PF6 (Add) to access the Add Member pop-up.
At least two JCL libraries	The Application - JCL Library Selection pop-up displays. To define members for one of the libraries listed, select the library name by typing any character in the selection column and pressing Enter.

3. Type a name or pattern to filter the list of members available for selection in the Add Member pop-up. For example, to select from a list of all members in the library that begin with an M, enter `M*` in the Name field. To view a list of all members in the library, leave the Name field blank.

4. Press PF5 (Directory) to display the Members - Selection List pop-up.

5. Select the desired members by typing any character in the selection column.

6. Previously-selected members are marked with an asterisk (*) and cannot be selected. To select all members in the library, press PF5 (Select All). Scroll the list if you need to view all members.

7 Press PF3 to return to the Add Member pop-up. You can enter another name pattern and repeat step 4 on page 51 through step 6 on page 51 to add more members.

8 Press PF3 when you have finished defining members to return to the Application - JCL Library Member List pop-up, listing the members defined to the application.

Specifying JCL Member Attributes

For each JCL member in the application, you can specify procedure library attributes, user exit parameters, and source manager execution parameters for Panvalet or Librarian libraries.

These attributes are used to analyze the JCL members. Attributes defined at the member level override any attributes specified at the library level.

You only need to specify attributes at the member level when a member has different attributes from those specified for the library. For example, if a member uses a procedure library that was not defined at the library level, you must specify the procedure library attribute for the member.

Specifying Source Manager Execution and User Exit Parameters (JCL Members)

If the members within a JCL library have Librarian or Panvalet as source manager and use execution parameters that were not specified as library attributes, you need to define them for that member. You may also define user exit parameters.

To specify source manager execution parameters

1 Access the Application - JCL Library Member List pop-up.

2 Select the desired member by typing any character in the selection column and pressing PF4 (Options) to display the JCL Member Options pop-up. You can specify the same source manager execution and user exit parameters for several members by selecting multiple members.

3 Select Analysis parms and press Enter to display the Member Options - Analysis Parms pop-up.

4 Type the execution and or user exit parameters required for the source manager.

5 Press PF3 to apply the parameters and return to the JCL Member Options pop-up. If you selected multiple members, the attributes specified apply to each one.

6 Select Exit to return to the Application - JCL Library Member List pop-up.

Specifying Procedure Libraries (JCL Members)

If a member within a JCL library uses procedure libraries that have not been defined during product installation or specified as library attributes for the library containing the member, you need to define the procedure libraries for that member. To verify the procedure libraries specified during product installation, consult your systems administrator.

To specify procedure libraries

1. Access the Application - JCL Library Member List pop-up.

2. Select the desired member by entering any character in the selection column and pressing PF4 (Options) to display the JCL Member Options pop-up. You can specify the same attributes for several programs by selecting multiple programs.

3. Select Maintain Proclibs to access the Add Proclib pop-up.

   Note:

   If at least one Proclib is defined for the member, the JCL Member - Maintain Proclibs pop-up displays. Press PF6 in this pop-up to access the Add Proclib pop-up.

   If you want to copy the proclib list from the library level to the selected member, press PF5.

   Or

4. Type the dataset name for the procedure library in the data set name field.

5. Press PF6 to add the procedure library as an attribute of the selected JCL member. The Add Proclib pop-up remains open to enter additional procedure libraries.

6. Press PF3 when you have defined the last procedure library to exit the Add Proclib pop-up and return to the JCL Member Options pop-up.

7. Select Exit to redisplay the Application - JCL Library Member List pop-up. Press PF3 repeatedly to return to the primary screen.

Removing JCL Library Attributes

It may become necessary to remove attributes specified at the member level, and revert to the attributes specified at the library level. For example, a JCL member used a procedure not defined at the library level; i.e., it is defined at the member level. Now, the JCL member doesn't use that procedure anymore or that procedure is now defined on the library level. When you remove JCL member level attributes, the selected member or members contain the attributes specified at the library level.

To remove library options

1. Select Edit ▸ JCL Definition on the primary product screen to display the Edit - JCL Definition pop-up.

2. Select Members and press Enter to display the Application - JCL Library Selection pop-up.

3. Type any character in the selection column next to the library you want to select and press Enter. The Application - JCL Library Member List pop-up displays.

4. Type any character in the selection column next to the member whose attributes are to be removed and press PF4 (Options). The JCL Member Options pop-up displays. If the same attributes are to be removed from more than one member, you can select multiple members.

5. Select Restore library options and press Enter to display the Restore Library Options pop-up, which lists the name of the selected library, the first member selected, and the attributes that will be removed.

6. Press PF10 (Remove) to remove the attributes. If you selected multiple members, the message `ATTRIBUTES REMOVED` displays in the short message area and the next selected program name is displayed. Press PF10 (Remove) for each member selected. When the attributes of the last member have been removed, the Application - JCL Library Member List pop-up redisplays.

7. Press PF3 to return to the Application - JCL Library Selection pop-up.

Adding CICS CSD Files

CICS CSD (CICS System Definition Dataset) files contain a resource definition record for every resource defined to CICS with RDO (Resource Definition Online), which is used to create entries in CSD.

You must specify any CSD files and group names to be included in the application definition. Consult the systems administrator for location information.

To add CSD files

1. Select Edit ▸ CICS Definition on the primary product screen to display the Edit - CICS Definition pop-up.

2. Select CSD and press Enter to display the Edit - CSD Definition pop-up.

3. Select Libraries and press Enter to display the Add CICS CSD Data Set pop-up. If at least one CSD file is defined to the application, the Application - CICS CSD Datasets pop-up displays. To add another CSD file, press PF6 (Add) to access the Add CICS CSD Data Set pop-up.

4. Type the CSD file name where the source resides in the Data set name field. If the high-level qualifier is different than your TSO prefix or user ID, enter the name in single quotes (').

5. Press PF6 to add the CSD file. A message displays to indicate that the file was added. The Add CICS CSD Dataset pop-up remains open to enable you to define additional files.

6. Press PF3 when you have defined all the CSD files to exit the Add CICS CSD Dataset pop-up. The Application - CICS CSD Datasets pop-up displays listing the CSD files defined to the application.

7. Press PF3 repeatedly to return to the primary screen.

Specifying CSD Group Names

After you have defined CICS CSD files, you can select the groups in those files to be included in the application definition. If you do not specify any group names, then all groups in the CSD are analyzed. If you specify one or more group names, then only those groups are analyzed.

To specify CSD group names

1. Select Edit ▸ CICS Definition on the primary product screen to display the Edit - CICS Definition pop-up.

2. Select CSD and press Enter to display the Edit - CSD Definition pop-up.

3. Select Group names and press Enter to display one of these pop-ups:

If you defined...	Result
Only one CSD file and no group names from that CSD are defined	The Add Group Name pop-up displays for that file.
Only one CSD file and that file contains at least one defined group name	The Application - CSD Library Group Name List pop-up displays.
Multiple CSD files	The Application - CSD Library Selection pop-up displays. Type any character in the selection column next to the applicable file you want to select and press Enter.

4. Type the name of the group in the Group name field of the Add Group Name pop-up.

5. You can use in the group name. Use a question mark (?) as the single character substitute, and an asterisk (*) as the multiple character substitute. For example, specify abc* to add all group names beginning with abc to the CSD files defined to the application.

6. Press PF6 to add the group name. A message displays indicating that the name was added. The Add Group Name pop-up remains open to enable you to define more groups.

7. Press PF3 when you have defined all the group names for the CSD file to exit the Add Group Name pop-up. The Application - CSD Library Group Name List pop-up displays listing the groups defined to the file.

8. Press PF3 repeatedly to return to the primary screen.

Adding CICS Libraries for FCT, PCT, PPT Tables, and BMS Maps

You must define the CICS libraries in which the members are located before specifying the members to include in the application definition. Consult the systems administrator for location information.

To add a library

1. Select Edit ▸ CICS Definition on the primary product screen to display the Edit - CICS Definition pop-up.

2. Select BMS/FCT/PCT/PPT and press Enter to display the CICS Definition pop-up.

3. Select Libraries and press Enter to display the Add CICS Library pop-up. If at least one CICS library is defined to the application, the Application - CICS Libraries pop-up displays. Press PF6 (Add) to access the Add CICS Library pop-up.

4. In the Add CICS Library pop-up, enter the appropriate information:

Field Name	Specifies the...
Data set name field	Name of the library that contains the CICS member.
Source manager field	Number selection of the CICS library's source manager. You must specify a source manager for each library.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

5. Press PF6 to add the library. A message displays to indicate that the dataset was added. The Add CICS Library pop-up remains open to enable you to define additional libraries.

   Note:
   If the source manager is CA-Librarian, the Add Library - Librarian pop-up displays for you to specify the Librarian attributes, if any.

6. Press PF3 when you have defined all the CICS libraries to exit the Add CICS Library pop-up. The Application - CICS Libraries pop-up displays, listing the CICS libraries defined to the application. You can specify library attributes, modify or remove a library, or add additional libraries to the definition.

Specifying CICS Library Attributes

For each CICS library defined to the application, you can specify maclib attributes. For Panvalet or Librarian libraries, you can also specify source manager execution parameters. These attributes are used to analyze the CICS members.

Attributes defined at the library level apply to every member defined in the library, unless overridden at the program level. For more information about member attributes, see "Specifying CICS Member Attributes" on page 61.

Specifying Source Manager Execution and User Exit Parameters (CICS Libraries)

If the members within a CICS library have Librarian or Panvalet as source manager and execution parameters are used, you need to define them for that library. You may also define user exit parameters.

To define source manager execution parameters

1. Access the Application - CICS Libraries pop-up.
2. Select the desired library by typing any character in the selection column and pressing PF4 (Options) to display the CICS Library Options pop-up. You can specify the same source manager execution and user exit parameters for several libraries by selecting multiple libraries.
3. Select Analysis parms and press Enter to display the Library Options - Analysis Parms pop-up.
4. Type the execution parameters required for the source manager or user exit parameters.
5. Press PF3 to apply the parameters and return to the CICS Library Options pop-up. If you selected multiple libraries, the attributes specified apply to each one.
6. Select Exit to return to the Application - CICS Libraries pop-up.

Specifying Maclibs (CICS Libraries)

If a member within a CICS library contains included source codes or macros, you need to specify maclibs for that library.

To specify maclibs

1. Access the Application - CICS Libraries pop-up.

2. Select the desired library by typing any character in the selection column and pressing PF4 (Options) to display the CICS Library Options pop-up. You can specify the same attributes for several libraries by selecting multiple libraries.

3. Select Maintain maclibs and press Enter to display the Add Maclib pop-up.

   Note:
   If at least one maclib is defined for the library, the CICS Library - Maintain Maclibs pop-up displays. Press PF6 in this pop-up to access the Add Maclib pop-up.

4. On the Add Maclib Library pop-up, enter the appropriate information:

Field Name	Specifies the...
Data set name field	Name of the library that contains the CICS member.
Source manager field	Number selection of the CICS library's source manager. You must specify a source manager for each library.
Subsystem	Any character if the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The SUBSYS parameter must have been previously defined during product installation.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

5. Press PF6 to add the maclib as an attribute of the selected CICS library. The Add Maclib pop-up remains open to enter additional procedure libraries.

6. Press PF3 when you have defined the last maclib, to exit the Add Maclib pop-up. Press PF3 repeatedly until you return to the primary screen.

Adding CICS FCT, PCT, and PPT Tables, and BMS Map Members

After you define the CICS libraries, you can select members in that library to be included in the application definition.

To add CICS members

1. Select Edit ▸ CICS Definition on the primary product screen to display the Edit - CICS Definition pop-up.

2. Select BMS/FCT/PCT/PPT and press Enter to display the CICS Definition pop-up.

3. Select Members and press Enter to display one of these pop-ups:

If you defined...	Result
Only one CICS library and no members from that library are defined	The Add Member pop-up displays.
Only one CICS library and that library contains at least one defined member	The Application - CICS Library Member List pop-up displays. Press PF6 to access Add Member pop-up.
At least two CICS libraries	The Application - CICS Library Selection pop-up displays. To define members for one of the libraries listed, type any character in the selection column next to the library name you want and press Enter.

4. Type a name or pattern to filter the list of members available for selection in the Add Member pop-up. For example, to select from a list of all members in the library that begin with an M, enter `M*` in the Name field. To view a list of all members in the library, leave the Name field blank.

5. Press PF5 (Directory) to display the Members - Selection List pop-up.

6. Select the desired members by typing any character in the selection column next to the member name. Previously-selected members are marked with an asterisk (*) and cannot be selected. To select all members in the library, press PF5 (Select All). Scroll the list if you need to view all members.

7. Press PF3 to return to the Add Member pop-up. You can enter another name pattern and repeat step 5 through step 7 to add more members.

8. Press PF3 when you have finished defining members to return to the Application - CICS Library Member List pop-up, listing the members defined to the application.

Specifying CICS Member Attributes

For each CICS member in the application, you can specify maclib attributes, and for Panvalet or Librarian libraries, source manager execution parameters. These attributes are used to analyze the CICS members.

You only specify attributes at the member level when a member has different attributes from those specified for the library. For example, if a member uses a maclib that was not defined at the library level, you specify the maclib attribute for the member.

Attributes defined at the member level override any attributes specified at the library level. You may have to consult your systems administrator for the location information.

Specifying Source Manager Execution and User Exit Parameters (CICS Members)

If the members within a CICS library have Librarian or Panvalet as a source manager and use execution parameters that were not specified as library attributes, you need to define them for that member. You may also define user exit parameters.

To define source manager execution parameters

1. Access the Application - CICS Library Member List pop-up.
2. Select the desired member by typing any character in the selection column and pressing PF4 (Options) to display the CICS Member Options pop-up. You can specify the same source manager execution or user exit parameters for several members by selecting multiple members.
3. Select Analysis parms and press Enter to display the Member Options - Analysis Parms pop-up.
4. Type the execution parameters required for the source manager.
5. Press PF3 to apply the parameters and return to the CICS Member Options pop-up. If you selected multiple members, the attributes specified apply to each one.
6. Select Exit to return to the Application - CICS Library Member List pop-up.

Specifying Maclibs (CICS Members)

Define the maclibs for a member within a CICS library that uses included source code or macros, if the maclibs containing the source code and macros have not yet been defined for the library containing the member.

To define maclibs

1. Access the Application - CICS Library Member List pop-up.

2. Select the desired member by typing any character in the selection column and pressing PF4 (Options) to display the CICS Member Options pop-up. You can specify the same attributes for several members by selecting multiple members.

3. Select Maintain maclibs and press Enter to display the Add Maclib pop-up.

   Note:
   If at least one maclib is defined for the member, the CICS Member - Maintain Maclibs pop-up displays. Press PF6 in this pop-up to access the Add Maclib pop-up.

4. Press PF5 to copy the maclib list from the library level to the selected member.

   On the Add Maclib pop-up, enter the appropriate information:

Field Name	Specifies the...
Data set name field	Name of the library that contains the CICS member.
Source manager field	Number selection of the CICS library's source manager. You must specify a source manager for each library.
Subsystem	Any character if the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The SUBSYS parameter must have been previously defined during product installation.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

5 Press PF6 to add the maclib as an attribute of the selected CICS member. The Add Maclib pop-up remains open to enter additional maclibs.

6 Press PF3 when you have defined the last maclib to exit the Add Maclib pop-up and return to the CICS Member Options pop-up. Press PF3 again to return to the Application - CICS Library Member List pop-up. Press PF3 repeatedly to return to the primary screen.

Removing CICS Library Attributes

It may become necessary to remove attributes specified at the member level and revert to the attributes specified at the library level. For example, a CICS member used a maclib not defined at the library level but no longer does. When you remove CICS member attributes, the selected member or members contain the attributes specified at the library level.

To remove library options

1 Select Edit ▸ CICS Definition on the primary product screen to display the Edit - CICS Definition pop-up.

2 Select BMS/FCT/PCT/PPT and press Enter to display the CICS Definition pop-up.

3 Select Members and press Enter to display the Application - CICS Library Selection pop-up (if you have more than one library defined).

4 Type any character in the selection column next to the library you want to select and press Enter. The Application - CICS Library Member List pop-up displays.

5 Type any character in the selection column next to the member whose attributes are to be removed and press PF4 (Options). The CICS Member Options pop-up displays. If the same attributes are to be removed from more than one member, you can select multiple members.

6 Select Restore library options and press Enter to display the Restore Library Options pop-up, which lists the name of the selected library, the first member selected, and the attributes that will be removed.

7 Press PF10 (Remove) to remove the attributes. If you selected multiple members, the message `ATTRIBUTES REMOVED` displays in the short message area and the next selected member name is displayed.

8 Press PF10 (Remove) for each member selected. When the attributes of the last member have been removed, the Application - CICS Library Member List pop-up redisplays.

9 Press PF3 to return to the Application - CICS Library Selection pop-up.

Adding IMS Libraries

Define the libraries in which IMS members are located before specifying the members containing the IMS DBDGEN, PSBGEN, DFSMDA, MFS maps, and Stage 1 library definitions. Consult your systems administrator for the location information.

To add IMS libraries

1. Select Edit ▸ IMS Definition on the primary product screen to display the Edit - IMS Definition pop-up.

2. Select Libraries and press Enter.

   If this is the first library to be defined, the Add IMS Library pop-up displays. If at least one library is already defined to the application, the Application - IMS Libraries pop-up displays. Press PF6 (Add) to access the Add IMS Library pop-up.

3. On the Add IMS Library pop-up, enter the appropriate information:

Field Name	Specifies the...
Data set name field	Name of the library that contains the IMS member.
Source manager field	Number selection of the IMS library's source manager. You must specify a source manager for each library.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

4. Press PF6 to add the library. A message displays to indicates that the library was added. The Add IMS Library pop-up remains open to define additional libraries.

   Note:
   If the source manager is CA-Librarian, the Add Library - Librarian pop-up displays for you to specify the Librarian attributes, if any.

5. Press PF3 when you have defined all the IMS libraries to exit the Add IMS Library pop-up. The Application - IMS Libraries pop-up displays, listing the IMS libraries defined to the application. You can specify library attributes, modify or remove libraries, or add additional libraries to the definition.

After defining the libraries, you are ready to specify the attributes for each library. These attributes include maclibs, source manager execution, and user exit parameters.

Specifying IMS Library Attributes

You can specify maclib attributes for each IMS library defined to the application. For Panvalet or Librarian libraries, you can also specify source manager execution parameters. These attributes are used to analyze the IMS members. You can also specify user exit parameters.

Attributes defined at the library level apply to every member defined in the library unless overridden at the member level. For more information about member attributes, see "Specifying IMS Member Attributes" on page 68.

Specifying Source Manager Execution and User Exit Parameters (IMS Libraries)

If the members within an IMS library have Librarian or Panvalet as a source manager and execution parameters are used, you need to define them for that library.

To specify source manager execution parameters

1. Access the Application - IMS Libraries pop-up.
2. Select the desired library by typing any character in the selection column and pressing PF4 (Options) to display the IMS Library Options pop-up. You can specify the same source manager execution or user exit parameters for several libraries by selecting multiple libraries.
3. Select Analysis parms and press Enter to display the Library Options - Analysis Parms pop-up.
4. Type the execution parameters required for the source manager or user exit parameters.
5. Press PF3 to apply the parameters and return to IMS Library Options pop-up. If you selected multiple libraries, the attributes specified apply to each one.
6. Select Exit to return to the Application - IMS Libraries pop-up.

Specifying Maclibs (IMS Libraries)

If the members within an IMS library contain included source code or macros, you need to define the maclibs for that library.

To specify maclibs

1. Access the Application - IMS Libraries pop-up.

2. Select the desired library by entering any character in the selection column and pressing PF4 (Options) to display the IMS Library Options pop-up. You can specify the same attributes for several libraries by selecting multiple libraries.

3. Select Maintain maclibs and press Enter to display the Add Maclib pop-up.

   Note:
   If at least one maclib is defined for the library, the IMS Library - Maintain Maclibs pop-up displays. Press PF6 on this pop-up to access the Add Maclib pop-up.

4. Press PF5 to copy the maclib list from the library level to the selected member.

   Or

   In the Add Maclib pop-up, enter the appropriate information:

Field Name	Specifies the...
Data set name field	Name of the library that contains the IMS member.
Source manager field	Number selection of the IMS library's source manager. You must specify a source manager for each library.
Subsystem	Any character if the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The SUBSYS parameter must have been previously defined during product installation.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

5. Press PF6 to add the maclib as an attribute of the selected IMS library. The Add Maclib pop-up remains open to enter additional maclibs.

6. Press PF3 when you have defined the last maclib to exit the Add Maclib pop-up and return to the IMS Libraries - Maintain Maclibs or the IMS Library Options pop-up. Press PF3 repeatedly to return to the primary screen.

Adding IMS Members

After you define the IMS libraries, you can select members in that library to be included in the application definition. Consult your systems administrator for the location information.

To add IMS members

1. Select Edit ▸ IMS Definition on the primary product screen to display the Edit - IMS Definition pop-up.

2. Select Members and press Enter to display one of these pop-ups:

If you defined...	Result
Only one IMS library and no members from that library are defined	The Add Member pop-up displays.
Only one IMS library and that library contains at least one defined member	The Application - IMS Library Member List pop-up displays. Press PF6 to access the Add Member pop-up.
At least two IMS libraries	The Application - IMS Library Selection pop-up displays. To define members for one of the libraries listed, type any character in the selection column next to the library name you want to select and press Enter.

3. Type a name or pattern to filter the list of members available for selection in the Add Member pop-up. For example, to select from a list of all members in the library that begin with an M, enter `M*` in the Name field. To view a list of all members in the library, leave the Name field blank.

4. Press PF5 (Directory) to display the Members - Selection List pop-up.

5. Select the desired members by typing any character in the selection column next to the member name. Previously-selected members are marked with an asterisk (*) and cannot be selected. To select all members in the library, press PF5 (Select All). Scroll the list if you need to view all members.

6. Press PF3 to return to the Add Member pop-up.

7 You can enter another name pattern and repeat step 5 on page 67 through step 6 on page 67 to add more members.

8 Press PF3 when you have finished defining members, to return to the Application - IMS Library Member List pop-up, listing the members defined to the application.

Specifying IMS Member Attributes

You can specify maclib attributes for each IMS member in the application. For Panvalet or Librarian libraries, you can specify source manager execution parameters. These attributes are used to analyze the IMS members.

You only specify attributes at the member level when a member has different attributes from those specified for the library. For example, if a member uses a maclib that was not defined at the library level, you specify the maclib attribute for the member.

The attributes defined at the member level override any attributes specified at the library level.

Specifying Source Manager Execution and User Exit Parameters (IMS Members)

If the members within an IMS library have Librarian or Panvalet as a source manager and use execution parameters that were not specified as library attributes, you need to define them for that member. You may also specify user exit parameters.

To define source manager execution and user exit parameters

1 Access the Application - IMS Library Member List pop-up.

2 Select the desired member by typing any character in the selection column and pressing PF4 (Options) to display the IMS Member Options pop-up. You can specify the same source manager execution or user exit parameters for several members by selecting multiple members.

3 Select Analysis parms and press Enter to display the Member Options - Analysis Parms pop-up.

4 Type the execution parameters required for the source manager or user exit parameters.

5 Press PF3 to apply the parameters and return to the IMS Member Options pop-up. If you selected multiple members, the attributes specified apply to each one.

6 Select Exit to return to the Application - IMS Library Member List pop-up.

Specifying Maclibs (IMS Members)

Define the maclibs for a member if a member within an IMS library uses included source code or macros, and the maclibs containing the source code and macros have not yet been defined for the library containing the member.

To define maclibs

1. Access the Application - IMS Library Member List pop-up.

2. Select the desired member by typing any character in the selection column and pressing PF4 (Options) to display the IMS Member Options pop-up. You can specify the same attributes for several members by selecting multiple members.

3. Select Maintain maclibs and press Enter to access the Add Maclib pop-up.

   Note:
   If at least one maclib is defined for the member, the IMS Member - Maintain Maclibs pop-up displays. Press PF6 in this pop-up to access the Add Maclib pop-up.

4. Press PF5 to copy the maclib list from the library level to the selected member.

   Or

   In the Add Maclib pop-up, enter the appropriate information:

Field Name	Specifies the...
Data set name field	Name of the library that contains the IMS member.
Source manager field	Number selection of the IMS library's source manager. You must specify a source manager for each library.
Subsystem	Any character if the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The SUBSYS parameter must have been previously defined during product installation.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected.

5. Press PF6 to add the maclib as an attribute of the selected IMS member. The Add Maclib pop-up remains open to enter additional maclibs.

6 Press PF3 when you have defined the last maclib to exit the Add Maclib pop-up and return to the IMS Member Options pop-up. Press PF3 repeatedly to return to the primary screen.

Removing IMS Library Attributes

It may become necessary to remove attributes specified at the member level and revert to the attributes specified at the library level. For example, a IMS member used a maclib not defined at the library level; i.e., it is defined at the member level. Now, the IMS member doesn't use that procedure anymore or that procedure is now defined on the library level. When you remove IMS member attributes, the selected member or members contain the attributes specified at the library level.

To remove library options

1 Select Edit ▸ MS Definition on the primary product screen to display the Edit - IMS Definition pop-up.

2 Select Members and press Enter to display the Application - IMS Library Selection pop-up.

3 Type any character in the selection column next to the library name you want to select and press Enter. The Application - IMS Library Member List pop-up displays.

4 Type any character in the selection column next to the member whose attributes are to be removed and press Enter. If the same attributes are to be removed from more than one member, you can select multiple members.

5 Press PF4 (Options). The IMS Member Options pop-up displays.

6 Select Restore library options and press Enter to display the Restore Library Options pop-up, which lists the name of the selected library, the first member selected, and the attributes that will be removed.

7 Press PF10 (Remove) to remove the attributes. If you selected multiple members, the message `ATTRIBUTES REMOVED` displays in the short message area and the next selected member name is displayed. Press PF10 (Remove) for each member selected. When the attributes of the last member have been removed, the Application - IMS Library Member List pop-up redisplays.

8 Press PF3 to return to the Application - IMS Library Selection pop-up.

Reviewing the Application Definition

Review your application definition before you analyze the application to verify that you have specified the correct source libraries, programs, and other application attributes.

Note:

See "The Application Maintenance Facility" on page 103 for more information about modifying the application definition.

To review the application definition

1. Select View ▸ Application Definition View on the primary product screen.

   The Application Definition View screen displays with the application name and a list of all libraries defined to the application, as shown in Figure 5. The libraries that are highlighted contain members included in the application definition.

   Figure 5 • Application Definition View Screen

```
  File  View  Options  Help
--------------------------------------------------------------------------------
                         Application Definition View
Command ===> _____________________________________________ Scroll ===> CSR

                                                                     1 OF 4
BILLING                                                         APPLICATION
|--> YIAKAS.TRAIN.JCLLIB                                            PDS\JCL
|--> YIAKAS.TRAIN.LOADLIB                                          PDS\LOAD
|--> YIAKAS.TRAIN.SRCLIB                                            PDS\COB
****************************** BOTTOM OF DATA ********************************
```

2. Select View ▸ Zoom In All and press Enter to view the members defined to the libraries. The Application Definition View screen expands to show all members defined to each library in the definition.

3. Select View ▸ Attributes and press Enter to review attribute information. The message `PLACE CURSOR` displays in the upper-right corner of the Application Definition View screen.

4. Move the cursor to the line containing the desired application name and press Enter. The Application Attributes pop-up displays.

5. Select View ▸ Attributes and press Enter to review the library and member. The message `PLACE CURSOR` displays in the upper-right corner of the Application Definition View screen.

6. Move the cursor to the desired library or member name and press Enter. The Library or Member Attributes pop-up displays.

Information tags on the right of the Application Definition View screen indicate the source manager and type of each component. These are the valid component types:

COB	CICS	ASM	JCL
LOAD	IMS	NATURAL	PL/I

Note:

You can also view application, library, and member attribute information by placing your cursor on the appropriate name in the Application Definition View screen and pressing PF4.

7 Press PF3 when you have finished your review to return to the primary product screen.

Modifying the Application Definition

If you discovered errors while reviewing the application definition, you can use the instructions listed in this section as a guide to help you correct the errors. You can change an application definition at any point. This procedure can include adding and removing libraries or members from the application, modifying library or member information, and changing library or member attributes. In each case, the starting point is assumed to be the Application - Member List pop-up.

Note:

See "The Application Maintenance Facility" on page 103 for more information about modifying the application definition.

Modifying Library Information

This section describes how to add and remove a library, modify library information (source manager, password, volume serial, etc.), and change the attributes of a library.

To modify a library

1. Select the definition action for the library type you want to modify in the Edit pull-down. For example, to modify a COBOL library, select COBOL definition. The Edit - Definition pop-up displays.

2. Select Libraries to access the Application - Libraries pop-up for the specified library type.

To add a library

- From the Application - Libraries pop-up, press PF6 to open the Add Library pop-up for the selected library type. For information about using the Add Library pop-up see one of these items:

Library Type	See...
COBOL	"Adding COBOL Programs" on page 39
Load	"Adding Load Libraries" on page 46
JCL	"Adding JCL Members" on page 51
CICS CSD files	"Adding CICS CSD Files" on page 55
CICS	"Adding CICS Libraries for FCT, PCT, PPT Tables, and BMS Maps" on page 57
IMS	"Adding IMS Libraries" on page 64

To remove a library

1. Access the Application - Libraries pop-up.

2. Select the library (or libraries) by typing any character in the selection column next to the library's dataset name.

3. Press PF10 to display the Remove Library pop-up.

4. The name of the library you specified for removal displays in the Dataset name field. If you selected multiple libraries, the name of the first library (alphabetically) displays in the initial Remove Library pop-up.

5 Press PF10 (Remove) to confirm the removal. Press any other key to cancel the removal, preferably PF3 (End).

6 If you selected multiple libraries, the Remove Library pop-up remains open and the name of the next library displays in the Dataset name field until all selected libraries have been processed (removed or kept). When you have processed the last library selected, the Application - Libraries pop-up redisplays.

Modifying Library Attributes

Library attributes, including DSN, source manager, password, and volume serial number are specified in the Add Library pop-up when the library is added to the application definition.

To modify library attributes

1 Access the Application - Libraries pop-up.

2 Select the library by typing any character in the selection column next to the dataset name for the library.

3 Press PF5 to display the Modify Library pop-up, which lists the name of the library that you selected for modification in the Dataset name field.

4 Change one or more of these dataset attributes:

- Dataset name
- Source manager
- Volume serial number
- Password

5 Press Enter to implement the changes and PF3 to exit.

To change Librarian source manager attributes

1. Access the Application - Libraries pop-up.
2. Select the library by typing any character in the selection column next to the dataset name for the library.
3. Press PF5 to display the Modify Library pop-up.
4. Press PF4 to display the Modify Library - Librarian pop-up.
5. Change Librarian attributes as needed and press Enter to exit.

To change the attributes for a library

1. Access the Application - Libraries pop-up.
2. Type any character in the selection column of the dataset name of the library.
3. Press PF4 (Options). Depending on the library type, one of these displays:

Library Type	Pop-up/Action	See Page
COBOL	COBOL Library Options pop-up displays. Specify the attributes for the library.	39
JCL	JCL Library Options pop-up displays. Add or remove proclibs, or specify Librarian or Panvalet source manager execution parameters.	51
CICS	CICS Library Option pop-up displays. Add or remove maclibs, or to specify CA-Librarian or Panvalet source manager execution parameters.	55
IMS	IMS Library Option pop-up displays. Add or remove maclibs, or to specify CA-Librarian or Panvalet source manager execution parameters.	64

Modifying Member Information

This section describes how to add or remove a program or member and modify the dataset attributes of a program or member.

To modify a member

1. Select the definition action for the library type containing the member you want to add or remove on the Edit pull-down. For example, to remove a COBOL program, select File ▸ COBOL Definition. The appropriate Edit - Definition pop-up displays.

2. Select Members and press Enter. Depending on the library type, one of these displays:

If you defined...	Result
Only one library of the selected type	The Application - Member List pop-up displays.
Multiple libraries	The Application - Library Selection pop-up displays. Type any character in the selection column next to the dataset name of the library that contains programs or members you want to add or remove or whose attributes you want to modify and press Enter. The Application - Member List pop-up displays.

To add a member

1. Access the Application - Member List pop-up.

2. Press PF6 to display the Add Member pop-up.

   Note:
   If the library containing the members listed in the Application - Member List pop-up is a password-protected CA-Librarian dataset, press PF6 in the Application - Member List pop-up to access the Add Member - Librarian pop-up.

3. Type the name of the member to be added in the Name field. The member must be a member of the library selected in the Application - Member List pop-up.

4. Press PF6. If the member resides in the specified library, it is added to the list of members defined to the application. The Add Member pop-up remains open to add more members.

5 Press PF3 when you have added the last member to exit the Add Member pop-up and return to the Application - Member List pop-up.

To remove a member

1 Access the Application - Member List pop-up.

2 Select the member by typing any character in the selection column next to the name of the member.

3 Press PF10 to open the Remove Member pop-up. The member that you selected for removal displays in the Name field. If you selected multiple members, the name of the first member (alphabetically) displays in the initial Remove Member pop-up.

4 Press PF10 to confirm the removal. Press PF3 to cancel the removal. If you selected multiple members, the Remove Program or Remove Member pop-up remains open and the next member displays in the Name field until all the selected members have been processed (removed or kept). When you have processed the last member, the Application - Member List pop-up returns.

To change the attributes for a member

1 Access the Application - Member List pop-up.

2 Type any character in the selection column next to the name of the member.

3 Press PF4 (Option). Depending on the member type, one of these pop-ups displays:

Library Type	Pop-up/Action	See Page
COBOL	COBOL Member Options pop-up displays. Add or remove copylibs; modify COBOL version, source manager, compile execution parameters, and IDMS information.	40
JCL	JCL Member Options pop-up displays. Add or remove proclibs, or specify Librarian or Panvalet source manager execution parameters.	52
CICS	CICS Member Option pop-up displays. Add or remove maclibs, or specify CA-Librarian or Panvalet source manager execution parameters.	61
IMS	IMS Library Member pop-up displays. Add or remove maclibs, or specify CA-Librarian or Panvalet source manager execution parameters.	68

Importing the Application Definition

The Import facility enables you to import an existing application definition into Recap, Alliance, or Estimate. For more information about importing application definitions, see the appropriate product's user guide.

Exporting the Application Definition

You can export both the application definition and the information produced from an analyze job. The exported definition can be stored in a sequential file or a PDS member. The exported application analysis information can be loaded into DB2 tables or used as input to other database managers. For more information about exporting applications, see the appropriate product's user guide.

5 Analyzing the Application

This chapter provides detailed instructions about analyzing and refining an application definition and contains these sections:

Section	Page
Introduction	79
Performing Application Analysis	82
Refining the Definition	95

Introduction

The AAE analyzes the application definition to provide information that is used by other ESW products to determine the scope of software change requirements.

After defining the application libraries and members, and specifying library and member attributes, you can analyze the application.

In the analysis process, the AAE gathers information about each entity in the application and its relationships. Output from the analyze job is stored in the AKR where it can be retrieved using online queries, impact analysis, or exported to a database manager.

After an application has been defined, the analyze process consists of opening the application, selecting an analysis type, and then setting up and executing a batch job. The analyze process is similar to a COBOL compile. The process has three primary inputs:

- Source members (including COBOL, Load module, JCL, CICS, and IMS source)
- JCL used to compile, link, and execute the COBOL programs
- Options and features that tailor the analyze steps

Analyze Input Description

An application definition is the analyze input. The analysis process gathers information about the application definition and stores it in the AKR. This table shows the application definition components used in the analysis process:

Component	Specifies...
COBOL copy members	The copy members included by the COBOL programs in the application.
Execution JCL	JCL referenced within the application.
PROCs	JCL Procedures called within the application.
CICS CSD, FCT, PCT and PPT tables, and BMS maps	CICS tables and maps accessed by the application.
IMS MFS, PSB, DBD, DFSMDA and Stage 1 library members	IMS specific information accessed by the application.
Load libraries	Load libraries accessed by the application.
COBOL source programs	Like the compiler, the analyze process requires these basic program standards: • The COBOL language, as specified in the *IBM COBOL II* and the *IBM COBOL/370 Language Reference* guides, is accepted by the analyze job. It correctly processes any program that can be compiled without errors by the IBM COBOL II compiler. COBOL for MVS and VM are accepted by the analyze job. • COBOL II and COBOL/370 programs that receive error (E), severe (S), or unrecoverable (U) messages from the IBM compiler cannot be successfully analyzed.
Assembler source programs	When the Assembler language option is installed. These languages are supported: • Assembler H • High-level Assembler For information about Assembler language support, see "Other Language Support," on page 177.

Component	Specifies...
Assembler maclib members	When the Assembler language option is installed, the maclib members included by the assembled programs in the application.
PL/I programs	PL/I versions 1.5, 2.3, MVS, and VM.
NATURAL objects	NATURAL versions 1.2, 2.1, 2.25, 2.2R.

Analyze Resource Estimates

This table contains the application analyzer resource estimates used to process applications of various sizes.

Pgms	Virtual Memory Size	XA Memory Size	CPU Time MM:SS	AKR Blocks	VIAUT2 Cyls	VIAUT3 Cyls
75	600K	20000K	15:00	4000	50	30
150	600K	25000K	25:00	8000	50	60
250	600K	30000K	35:00	10000	50	65
1000	600K	36000K	140:00	40000	50	265

These timing and estimated resource requirements are based on average program length of 2,000 lines of expanded source.

This information results from running the application analyzer under this criteria:

Version	ASG-Center V*x.x*
CPU Type	3090-600 running MVS/ESA
Disk Type	3390
Analyze Parms	BUFMAXK=4096KB
Compiler Parms	BUF=256K,SIZ=1024KB

Performing Application Analysis

To open the application

1. Select File ▸ Open application on the primary product screen to display the File - Open Application pop-up.

2. In the File - Open Application pop-up, enter the appropriate information.

Field Name	Specifies the...
Data set name field	Name of the AKR. If the high-level qualifier of the name is different from your TSO prefix or user ID, enclose the AKR name in single quotes (‘).
Application name	Name of the application, up to ten characters.
Volser field	Volume serial number, if the dataset is not cataloged.
Password field	Password, if the dataset is password protected. You may need to consult your system administrator to find out if this information is required for the specified AKR.

3. Press Enter to open the application. If the application was opened, the message `APPLICATION [name of application] SUCCESSFULLY OPENED` displays.

   The Application Maintenance Facility (AMF) screen displays.

   Note:
   If someone else has the application open, the application displays in read-only mode and you cannot update the definition or analyze the application.

Selecting the Analysis Type

A Full analysis selects every member in the application for analysis. A Full analysis can require a large amount of CPU resources. If only a limited number of members need to be analyzed, choose Incremental or Auto analysis. For resource estimates, see "Analyze Resource Estimates" on page 81 .

Note:
If you are performing a Recap analyze, Full analysis creates a new version of the metrics.

Select Incremental if you want to manually select the members to be analyzed. Any member can be reanalyzed. An incremental analysis analyzes only members that you select.

An Auto Analysis is like an incremental analysis except the analyzer automatically selects members for analysis that have not been analyzed, have bad analyze return codes, or have been modified since the last analyze.

To select the type of analysis you want to perform

1. Select File ▸ Analyze on the primary product screen or AMF screen and press Enter to display the File - Analyze pop-up.

2. Choose Select analysis type and press Enter to display the Analyze - Select Analysis Type pop-up.

3. Choose one of these analysis types:

Option Name	Results
Full analysis	Displays the Full Analysis Confirmation pop-up to confirm submission of all members in the application for analysis. Press Enter to exit the Full Analysis Confirmation pop-up and return to the File -Analyze pop-up.
Incremental...	Displays the Select Analysis Library Type pop-up for selection of the library type containing the libraries and members to be submitted for analysis.
Auto Analysis	The batch analysis job automatically selects all members that have not been analyzed, have bad analyze return codes, or have been modified since the last analyze.

4. When you have selected the members for analysis, press PF3 to return to the File - Analyze pop-up.

Submitting the Analyze Job

The Analyze Features field on the Analyze - Submit application pop-up screen lets you specify which product-specific analyses you want to run, depending on your site's installation. You can only change the analyze features on a full analyze. Otherwise, the Analyze Features field is inactive.

To analyze the application definition

1 Access the File - Analyze pop-up, and select Submit application analysis batch job. The Analyze - Submit Application displays. Depending on the products currently installed at your site, one or more products display on this screen.

Figure 6 shows the Analyze features field when Alliance, Recap, and Estimate are installed and a full analysis has been selected.

Figure 6 • Analyze - Submit Application Pop-up

```
                        Analyze - Submit Application              BILLING
Command ===> ____________________________________________________________

Specify option and jobcard information.  Then press PF key for action.
AKR: 'YIAKAS.ALLIANCE.AKR'

Analyze features (Y/N):
   ASG-Recap     :  Y
   ASG-Alliance  :  Y
   ASG-Estimate  :  Y

Do Semantic Linking (Y/N)?  Y

Job statement information:
  //NAME     JOB (ACCOUNT),NAME,
  //         MSGCLASS=A
  //*   INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
  //*

                 PF4=Options  PF5=Edit JCL  PF6=Submit JCL
```

Some analyze options require a specific type of analysis. For a listing of analysis options, see "Analyze Options and Parameters," on page 145.

Note:

If you are running a Recap analysis, an option displays to run reports. See the reports section of the *ASG-Recap User's Guide* for more information.

2 In the Do Semantic Linking (Y/N) field, specify whether you want the analyze job to perform semantic linking.

Semantic linking is the phase of the analyze job that establishes the relationships between the entities collected. This phase is required before you can use the AKR for impact assessments.

Note:

In the Application Definition process, semantic linking is used to generate dead, discovered, and missing components. Specify N in the Do Semantic Linking (Y/N) field while you are refining the application definition (resolving compile and syntactical errors). Specify Y in the Do Semantic Linking (Y/N) field, and select the Lite Link option by pressing PF4 (Options) to limit semantic linking to generating dead, discovered, and missing components, enabling you to finish defining your application. Once the definition is complete, run semantic linking without the Lite Link option.

3 On the Analyze - Submit Application pop-up, make any necessary changes to the Jobcard in the Job statement information field. This information is added to the beginning of the JCL that is submitted to analyze the application.

4 Press F4 to specify analyze execution options, if needed.

5 Press PF6 to submit the analyze job.

A message may appear prompting you to close the application because an analyze cannot be performed if the application remains open. You can disable this prompt by changing a product parameter.

Note:

If you need to edit the JCL for the analyze job, press PF5. You are prompted to close the application. After editing the JCL, issue a TSO SUBMIT command to submit the analyze job.

If the analyze abends, a restart pop-up displays when an attempt is made to open the application. Information about restarting the analysis is located in "Restarting the Analyze" on page 86.

Restarting the Analyze

CPU time limitations, space allocations, or other operating system-related issues can occasionally cause an application analyze to abend. You can restart the analyze when an abend occurs if the Recovery switch on the Analyze - Execution Options pop-up is set to Y.

To set the Recovery switch

1. Select File ▸ Analyze on the primary product screen. Ensure that an analysis type is selected.

2. In the Analyze - Submit Application pop-up, press PF4 (Options) to display the Analyze - Execution Options pop-up.

3. Set recovery to Flag Y.

   When an analyze has abnormally ended, an attempt to open the application within the product causes the Analysis Restart Warning pop-up to appear, as shown in Figure 7. This pop-up contains a message indicating the type of abend and enables the option to restart the analyze.

Figure 7 • Analysis Restart Warning Pop-up

```
                    Analysis Restart Warning

Prior analysis did not complete correctly.

Abend information:

Library(Mbr) : 'USER.TEST.COBOL(USERPROG)'

Library Mgr  : PDS

Analysis Type: COBOL

Abend cause  : EXCEEDED CPU TIME

Press PF key to begin the restart process and adjust necessary
parms, or to cancel the process.

        PF5=Restart analysis  PF7=AKR utility  PF12=Cancel restart
```

To resubmit the application for analysis

1. Access the Analysis Restart Warning pop-up, and press the Restart analysis PF key.
2. The Analyze - Submit Application pop-up displays for you to resubmit the analyze for completion.
3. In the Do Semantic Linking (Y/N) field, specify whether you want the analyze job to perform semantic linking.
4. Make any necessary changes to the Jobcard in the Job statement information field and press PF6 to submit the analyze job.

For more information about submitting the application for analysis, see the instructions in "Submitting the Analyze Job" on page 83.

Scheduling an Application Analyze Through Batch Submission

Members of an application can be selected and analyzed using a single batch job. This enables you to bypass the online facilities to perform regularly-scheduled analyses of applications, and submit an unlimited number of programs defined within an application for analysis.

In addition to allowing you to analyze applications for Alliance, Recap, and Estimate, this facility also allows you to analyze all of the COBOL programs within the application for Insight. The process analyzes selected members and creates entries in the same AKR as the application, based on the PROGRAM_ID of the COBOL member. Once the analysis is complete, the COBOL member entries on the AKR are available for all Insight functions. For additional information, see the *ASG-Insight User's Guide*.

Note:

For Recap and Insight only, the MEMNAME option is available. The MEMNAME option uses the member name rather than the program-ID as the program name. For more information, see "Analyze Options and Parameters," on page 145.

The members to be selected are specified using a card image input file. This input stream identifies the library name that contains the members to be analyzed, as well as the kind of entity selected.

Creating the Analyze JCL

The JCL generated from an online Alliance, Estimate, or Recap analyze is used to create the batch JCL.

To capture the generated JCL

1. Select File ▸ Analyze on the primary product screen or AMF screen and press Enter to display the File - Analyze pop-up.
2. Choose Select analysis type and press Enter to display the Analyze - Select Analysis Type pop-up. Select one of the analysis types.
3. After you have selected members for analysis, press PF3 to return to the File - Analyze pop-up.
4. Select Submit application analysis batch job on the File - Analyze pop-up to display the Analyze - Submit Application pop-up. Press PF5 to edit the analyze JCL.
5. From the TSO ISPF edit session, enter the TSO ISPF CREATE command to save the JCL.
6. Enter a TSO ISPF END command to end the TSO ISPF edit session.
7. Exit the product.

The JCL that was saved to a dataset can now be edited for batch submission.

Editing the Batch Analyze JCL

The batch analyze JCL must be edited to include the ddname for the card image input file. The AKR containing the application can also be changed. Use your standard editor to change the analyze JCL by adding a DD statement for the ddname VIASANIN. This ddname defines card image input file. The statement can be placed anywhere in the job stream.

This is an example of the DD statement for VIASANIN:

```
//VIASANIN    DD  DSN=USER.BATCH.ANALYZE,DISP=SHR
```

Note:

If you select Auto Analysis, the VIASANIN DD is already part of the analyze JCL.

The AKR containing the application is defined by the ddname VIAAKR and can be changed if necessary.

Creating the Input File for VIASANIN

The VIASANIN input file format is a subset of the format used by the product's Import Facility.

The input statements are hierarchical in nature. Multiple members can be specified within the member list. Multiple entity kinds (each with a corresponding member list) can be specified for each library. Multiple libraries (each with corresponding entity kinds and member lists) can be specified within the input file. The input file can include blank records and comments. A comment is identified by an asterisk in column one.

Input File Control Statements

The input file control statements can start anywhere on a line and are enclosed by these tags; (<) and (>). Each control statement must be coded on a separate line and each item associated with a control statement must also be coded on a separate line. Figure 8 shows input file control statements for a full analyze and for an incremental analyze.

Figure 8 • Input File Control Statements - Incremental Analyze

```
(A)  <CLEAR SELECTIONS>
(B)  <SEMANTIC LINK>
           NO
(C)  <ANALYSIS TYPE>
           INCREMENTAL
(D)  <LIBRARY>
              user.test.cobol
(E)  <OBJECT KIND>
             PLI
(F)  <MEMBER LIST START>
             membername
(G)  <MEMBER LIST END>
```

This table describes these valid control statements:

Control Statement	Description
(A) <CLEAR SELECTIONS>	Optional. Indicates which members currently selected to be analyzed should be deselected. If this control statement is specified, it must be the first input statement.
(B) <SEMANTIC LINK>	Indicates semantic linking (YES or NO). Not valid for Insight.
(C) <ANALYSIS TYPE>	Indicates that type of analyze (Full, Incremental, or Auto). Not valid for Insight.

Control Statement	Description
(D) <LIBRARY>	Identifies the next card as a fully-qualified library dataset name statement.
(E) <OBJECT KIND>	Identifies the next card as an entity kind selection statement. These are the valid entity kinds: COBOL COBOL PROGRAM BMS CICS CSD IMS JCL LOAD LOAD MODULE ALTERNATE LANGUAGE ASM PL/I **Note:** The only valid entity kind for Insight is COBOL.
(F) <MEMBER LIST START>	Indicates that these statements are member names (the name of the member selected to be analyzed). This is the name of an entity of the current entity kind within the current library. Only one member name can be specified on a line. However, any number of member name statements can be specified. The member names continue until the `<MEMBER LIST END>` control statement is encountered. You can use wildcards (*) or (?) to select members in the library to be analyzed. The (?) wildcard is used as a single character substitute, while the (*) wildcard is used as a multiple character substitute. For example, abc* would select all members beginning with abc.
(G) <MEMBER LIST END>	This control statement identifies the end of the member names for the current entity kind and library name combination.

Valid entries after <MEMBER LIST START> include a list of the member names within the specified library that you want to analyze, or <SELECT ALL> if you want to analyze all COBOL members within the specified library. Use <SELECT ALL> when selecting CSD names for this process, as shown in Figure 9.

Figure 9 • Analysis Example

```
//VIASANIN DD *
 <CLEAR SELECTIONS>
 <LIBRARY>
 VIAAL10.TEST1.COBOL
 <OBJECT KIND>
 COBOL
 <MEMBER LIST START>
 <SELECT ALL>
 <MEMBER LIST END>
 /*

OR

 //VIASANIN DD *
 <CLEAR SELECTIONS>
 <LIBRARY>
 VIAAL10.TEST1.COBOL
 <OBJECT KIND>
 COBOL
 <MEMBER LIST START>
 VIAFDEMO
 VIAIDEMO
 <MEMBER LIST END>
 /*
```

VIASYSTM DD

The VIASYSTM DD specifies the application name and these analyze features:

- AL - Alliance
- RC - Recap
- TM - Estimate
- IN - Insight

For example:

```
//VIASYSTM DD *
APPL=(your application),FEATURES=(AL,RC,TM)
```

Analyze Summary Report

Application analysis data is placed in the AKR when the analyze job completes. A summary report of the run-time statistics and diagnostic messages is also produced.

Figure 10 shows the Analyze Summary Report. For field descriptions, see figure callouts (A) through (E) in "Analyze Summary Report Field Descriptions" on page 93.

Figure 10 • Analyze Summary Report

```
ASG-CENTER-OS(ESA) Rx.x LVLnnn   APPLICATION ANALYTICAL ENGINE
(A)
VIASYSTM INPUT RECORD(S)

APPL=(EMPPAY),FEATURES=(AL)
VIA1731I *** END OF VIASYSTM ***
(B)
PARAMETERS PASSED:
OPTIONS IN EFFECT: BUFMAXK=4096K,LINECNT=55,FEATURES=(ALLIANCE)

DD-MM-YYYY HH:MM:SS  JCL ANALYZE STARTED.
JCL ANALYZE HAS ENDED.
(C)
JCL ANALYZE SUMMARY FOR APPLICATION: EMPPAY

INPUT JCL LIBRARY: USRXYZ.EMPPAY.JCL

JCL MEMBER (RETURN CODE) ANALYZE LIST
JOB1 (0)        JOB2 (0)
JOB3 (0)        JOB4 (0)

DD-MM-YYYY HH:MM:SS  LOAD MODULE ANALYZE STARTED.
DD-MM-YYYY HH:MM:SS  LOAD MODULE ANALYZE HAS ENDED.

LOAD MODULE ANALYZE SUMMARY FOR APPLICATION: EMPPAY

INPUT LOAD MODULE LIBRARY: USRXYZ.EMPPAY.LOADLIB

LOAD MODULE MEMBER (RETURN CODE) ANALYZE LIST
CALCPAY (0)    CALCTAX (0)    PAYREPT (0)    PAYROLL (0)    PRNTCHCK (0)

DD-MM-YYYY HH:MM:SS  COBOL ANALYZE STARTED.
DD-MM-YYYY HH:MM:SS  COBOL ANALYZE HAS ENDED.

COBOL ANALYZE SUMMARY FOR APPLICATION: EMPPAY

INPUT SOUCE LIBRARY: USRXYZ.EMPPAY.COBOL

COBOL MEMBER/PROGRAM-ID (RETURN CODE) ANALYZE LIST
CALCPAY/CALCPAY (0)     CALCTAX/CALCTAX (0)     PAYREPT/PAYREPT (0)
PAYROLL/PAYROLL (4)     PRNTCHCK/PRNTCHCK (0)

DD-MM-YYYY HH:MM:SS  PL/I ANALYZE STARTED.
DD-MM-YYYY HH:MM:SS  PL/I ANALYZE HAS ENDED. NO PL/I ENTITIES WERE SPECIFIED FOR ANALYSIS.

DD-MM-YYYY HH:MM:SS  ASM ANALYZE STARTED.
DD-MM-YYYY HH:MM:SS  ASM ANALYZE HAS ENDED. NO ASM ENTITIES WERE SPECIFIED FOR ANALYSIS.

DD-MM-YYYY HH:MM:SS  ALTERNATE LANGUAGE ANALYZE STARTED.
DD-MM-YYYY HH:MM:SS  ALTERNATE LANGUAGE ANALYZE HAS ENDED. NO ALTERNATE LANGUAGE ENTITIES
WERE
SPECIFIED FOR ANALYSIS.

DD-MM-YYYY HH:MM:SS  IMS ANALYZE STARTED.
DD-MM-YYYY HH:MM:SS  IMS ANALYZE HAS ENDED. NO IMS ENTITIES WERE SPECIFIED FOR ANALYSIS.

DD-MM-YYYY HH:MM:SS  CICS ANALYZE STARTED.
DD-MM-YYYY HH:MM:SS  CICS ANALYZE HAS ENDED. NO CICS ENTITIES WERE SPECIFIED FOR ANALYSIS.

(D)
ANALYZE RETURN CODE SUMMARY
                  RC=0
                ------
     JCL MEMBER       2
    LOAD MODULE       5
  COBOL PROGRAM       5
                ------
          TOTAL      12
(E)
DD-MM-YYYY HH:MM:SS  SEMANTIC LINKING STARTED.
DD-MM-YYYY HH:MM:SS  SEMANTIC LINKING HAS ENDED.

DD-MM-YYYY HH:MM:SS  POST PROCESSING STARTED.
DD-MM-YYYY HH:MM:SS  POST PROCESSING HAS ENDED.

END OF PROCESSING:  DD-MM-YYYY HH:MM:SS
```

Analyze Summary Report Field Descriptions

Field	Description
(A) VIASYSTM INPUT RECORD(S)	Lists the name of the application submitted for analysis and selected features.
(B) PARAMETERS PASSED	Lists all of the analyze options specified for this analyze job.
(C) ANALYZE SUMMARY	Lists the results of the analyze by input source library for each application component (JCL, Load Module, COBOL, etc.). After each library name, the summary includes a list of all members in the library analyzed and the return code generated. These are the codes and their descriptions: 0 - Normal conclusion. 4 - Warnings have been detected, but member was analyzed. Analyze is considered successful. You should note that warnings are not listed. 8 - Error messages have been listed and recovered by the AAE. Analyze is considered successful, unless the member is an Assembler program. For Assembler programs, a return code of 8 is an analyze failure. 12 - Severe errors have occurred and error messages have been listed. Execution is impossible and the analyze has failed. 16 - Terminal errors have occurred and error messages have been listed. Execution is impossible and the analyze has failed. **Note:** For instructions on resubmitting all members with a specified return code for analysis, see "Restarting the Analyze" on page 86.

Field	Description
(D) ANALYZE RETURN CODE SUMMARY	Lists the component type (for example, COBOL program) and the number of members analyzed for each return code reported.

(E) SEMANTIC LINKING	Indicates whether semantic linking is active. If semantic linking was requested (Y for YES) on the Analyze - Submit Application pop-up, this text is displayed:
	`SEMANTIC LINKING WAS STARTED.`
	`SEMANTIC LINKING HAS ENDED.`
	If semantic linking was not requested (N for NO), this text is displayed:
	`SEMANTIC LINKING WAS SKIPPED.`
	During the semantic linking process, the AAE attempts to associate segment references found within programs and the IMS components of the application.
	If the program does not have access to the segment through a PSB, the segment references cannot be associated. This text is displayed on the Analyze Summary Report (under SEMANTIC LINKING STARTED):
	`UNABLE TO ASSOCIATE SEGMENT REFERENCE`
	`NAME: CUSTOMER`
	`THE UNRESOLVABLE REFERENCES ARE IN THE`
	`FOLLOWING COMPONENTS`
	`XXX.SOURCE.COBOL(PROGRAM1)`
	`XXX.SOURCE.COBOL(PROGRAM2)`
	`XXX.SOURCE.COBOL(PROGRAM3)`
	To associate segment references, see the NOPSBCHECK and NODBDCHECK options in "Analyze Options and Parameters," on page 145.

Refining the Definition

Reviewing the Analyze Results

After the analyze job has completed, you can review the Analyze Summary Report in the output dataset VIAARPT for a brief listing of the analyze results. For information about this report, see "Analyze Summary Report" on page 91.

Select ▸ Discovered Components to view components that have been referenced in other components but that are not defined in the application. You can then determine whether to add them to the application definition.

Review the Missing Components and Dead Components views for components that have been removed from the application definition since the last analyze and components that appear to be unnecessary in the application definition.

Resolving Analysis Errors

The first step in resolving analysis errors is to evaluate the individual return codes for the analysis of each item in the application definition. Items with return codes of 4 through 16 have an entry in the SYSPRINT DD of the analysis job. Once the errors have been corrected, reanalyze the application using incremental or auto analysis.

This table contains some common problems and resolutions:

Problem	Resolution
Missing include dependency causing unresolvable include statements and undefined data items.	Add the correct library to the include dependency, (i.e., copy library or include library, for the member or library containing the member). The compile JCL for the item is the best source of information for include dependency.
Compile errors.	Correct the syntax error. This may also indicate the wrong library has been added to the definition (i.e., Test or Development vs. Production).

Once the items have been analyzed correctly, you are ready to resolve dead, discovered, and missing components.

Researching Dead, Discovered, and Missing Components

The AAE produces lists these types of questionable components:

- Dead
- Discovered
- Missing

To refine the application definition, review each item identified to determine if any action should be taken. For example, before resubmitting the definition for analysis, you may want to add discovered components to the definition, redefine, or remove references to missing components, or decide whether dead components should be removed from the definition.

Dead Components

Dead components are items that are part of the application definition, but are not used by any other component of the application. A dead component indicates one of these two situations:

- The component is not part of this application and should be removed from the definition.
- A component that would reference the dead component has not been defined to the application. In the second case, the component that has the reference to the dead component needs to be added to the definition.

Discovered Components

When the AAE finds a reference to a component that has not been defined to the application, the component has been discovered. Discovered components can be resolvable or unresolvable.

Resolvable Discovered Components

When the context of the reference to a discovered component allows the AAE to determine the location of the source for that component, it is labeled a resolvable discovered component. You can add the component's defining source to the definition. For example, a load library is marked as resolvable and discovered if it is found in a JCL steplib or joblib and has not been defined to the application. It is resolvable because the load library name can be added to the definition. No further information is needed.

Unresolvable Discovered Components

When the context of the reference to a discovered component does not allow the AAE to determine the location of the source for that component, it is labeled an unresolvable discovered component. In this case, you must manually determine the source defining the component. For example, a load module is marked as unresolvable and discovered if a JCL member executes the load module and the load module is not defined. It is unresolvable because the AAE does not necessarily know which load library contains the load module.

Missing Components

An application definition item is marked as missing when it is not found by the AAE. This indicates that the item was either deleted from the target system, or has been renamed or moved after it was added to the definition. These items need to be either removed from the application definition or renamed to match the current item.

Identifying Dead Components

The analyze process detects components that are part of the definition but that are not used by any other component of the application. These components are referred to as dead components. Dead components are components that are unused in the application under review. Review the list of dead components to determine if the components are completely unused (in any application) or only in the application you are analyzing. If the component is completely unused, you may consider removing it from the system. Proceed with the caution that applies to any deletion.

To view dead components

1 Select View ▸ Dead components to display the Dead Components Selection screen, as shown in Figure 11.

Figure 11 • Dead Components View

```
  File  Edit  Options  Help
--------------------------------------------------------------------------------
                          Dead Components View                        BILLING
Command ===> _____________________________________________ Scroll ===> CSR

                                                                     1 of 8
                                                                        >>>

ENTITY KIND                ENTITY NAME (WITH QUALIFICATION)
-------------------------- ----------------------------------------------------
********************************* TOP OF DATA *********************************
COBOL MEMBER               YIAKAS.TRAIN.SRCLIB(YIASTP04)
COBOL MEMBER               YIAKAS.TRAIN.SRCLIB(YIASTP07)
COBOL MEMBER               YIAKAS.TRAIN.SRCLIB(YIASTP08)
COBOL MEMBER               YIAKAS.TRAIN.SRCLIB(YIASTP09)
COBOL MEMBER               YIAKAS.TRAIN.SRCLIB(YIASTP10)
COBOL MEMBER               YIAKAS.TRAIN.SRCLIB(YIASTP11)
COBOL MEMBER               YIAKAS.TRAIN.SRCLIB(YIASTP12)
COBOL MEMBER               YIAKAS.TRAIN.SRCLIB(YIASTP13)
******************************** BOTTOM OF DATA ********************************
```

2 Select the dead components you want to view and press Enter to display the Dead Components View screen. You can scroll the Dead Components View screen to the right to allow for long qualifications of entity names.

Identifying Discovered Components

Discovered components are listed as either resolvable or unresolvable. The resolvable discovered components feature identifies and lists libraries, members, or attributes referenced in the JCL, but missing from the application definition. To gather the most complete and accurate information during the analyze process, review the components and attributes identified and determine whether to add them to the application definition.

The unresolvable discovered components feature shows components that were referenced in the application, but not present in the application definition. Research these components and add the libraries and members containing their definition to the application definition.

After reviewing the discovered components list and adding the necessary components and attributes to the definition, resubmit the application for analysis. For instructions, see "Submitting the Analyze Job" on page 83.

To review discovered components

1. Select File ▸ Open application on the primary product screen and specify the name of application you want to open. Press Enter to display the AMF screen and press PF3 to return to the primary product screen.

2. Select Edit ▸ Discovered components and press Enter.

   The Edit - Discovered Components pop-up displays. The analyze process categorizes the discovered components and attributes in two categories: resolvable ("Resolvable Discovered Components" on page 96) and unresolvable ("Unresolvable Discovered Components" on page 97).

3. Use the information in the Resolvable Discovered Components pop-up to add the component or attribute.

4. Use the information in the Unresolvable Discovered Components pop-up to determine the library or attribute type to add to the application definition.

To view resolvable discovered components

1. Select Edit ▸ Discovered components on the primary product screen to display the Edit - Discovered Components selection pop-up.

2. Select Resolvable and press Enter to display the Resolvable Discovered Components Selection pop-up, as shown in Figure 12.

Figure 12 • Resolvable Discovered Components Selection Pop-up

```
          Resolvable Discovered Components Selection
Command ===> _______________________________________________

Select the desired Discovered Components.  Then press Enter.

/  Load Libraries
/  Program Source Libraries
/  CICS CSD Files
/  MFS Source Libraries

            PF5=Select All  PF6=Unselect All
```

3 Select the component(s) and press Enter to display the Resolvable Discovered Components pop-up, as shown in Figure 13.

Figure 13 • Resolvable Discovered Components Pop-up

```
                    Resolvable Discovered Components
Command ===> ____________________________________________ Scroll ===> CSR

Add these discovered components to the application                1 of 1
definition, if desired.

Action:  A=Add  B=Browse

  Discovered Component  Component Kind   Reference Location          >
_ --------------------  ---------------  ----------------------------->
_ DEV.LOADLIB           LOAD LIBRARY     VIAAL10.ABBEY.COPYLIB(CDSJCL)
************************** BOTTOM OF DATA **************************
```

To add resolvable discovered components to the definition

1 From the Resolvable Discovered Components pop-up select the component name and press PF6 (Add) or type `A` to open the appropriate Add Library pop-up.

Note:

Optionally, you can also use the Edit pull-down to add the component using the same procedure you did for the full application definition.

2 After reviewing the resolvable discovered components list and adding the necessary components and attributes to the definition, you can resubmit the application for analysis.

To view unresolvable components

1 From the Edit - Discovered Components pop-up select Unresolvable and press Enter to display the Unresolvable Discovered Selection pop-up, as shown in Figure 14.

Figure 14 • Unresolvable Discovered Components Selection Pop-up

```
          Unresolvable Discovered Components Selection
Command ===> _____________________________________________________

Select the desired Discovered Components.  Then press Enter.

/  Program SYSLIB Libraries     /  IMS Stage 1 Transactions
/  Program CSECTs               /  IMS PSBs
/  Called Programs              /  IMS Programs
/  Load Modules                 /  IMS DBD DDs
/  JCL DDs                      /  IMS DFSMDA DDs
/  CICS Program PPTs            /  IMS Formats
/  CICS File Names              /  IMS Segments
/  CICS Transaction IDs         /  IMS PSBGEN DBDs
/  CICS Mapsets                 /  IMS DFSMDA DBDs
/  CICS Maps                    /  NATURAL Map
/  CICS Programs

             PF5=Select All  PF6=Unselect All
```

2 Select component(s) and press Enter to display the Unresolvable Discovered Components pop-up, as shown in Figure 15.

Figure 15 • Unresolvable Discovered Components Pop-up

```
                    Unresolvable Discovered Components
Command ===> ____________________________________________ Scroll ===> CSR

Use the standard definition facilities to add these              1 of 51
components to the application definition, if desired.

Action:  B=Browse

  Discovered Component  Component Kind   Reference Location           >
_ --------------------  ---------------  ----------------------------->
_ VIASTP08              LOAD MODULE      VIASTP08
_ VIASTP11              LOAD MODULE      VIASTP11
_ VIASTP12              LOAD MODULE      VIASTP12
_ VIASTP13              LOAD MODULE      VIASTP13
_ 'DC99DATE'            CALLED PROGRAM   VIAKAS.TRAIN.SRCLIB(VIASTP08)
_ 'DCPRA028'            CALLED PROGRAM   VIAKAS.TRAIN.SRCLIB(VIASTP08)
_ 'DCPRA005'            CALLED PROGRAM   VIAKAS.TRAIN.SRCLIB(VIASTP08)
_ 'DCMRA001'            CALLED PROGRAM   VIAKAS.TRAIN.SRCLIB(VIASTP08)
_ 'DCPRL001'            CALLED PROGRAM   VIAKAS.TRAIN.SRCLIB(VIASTP08)
_ 'DCSPA011'            CALLED PROGRAM   VIAKAS.TRAIN.SRCLIB(VIASTP09)

                          PF4=Print
```

3 Review the list of components to determine which component needs to be added to the application definition.

4 Determine the library or attribute type of any component to be added.

5 From the Edit pull-down, choose the appropriate definition action to add the component to the application definition.

For example, if you identify one of the components listed as an IMS PSB:

a Exit the Unresolvable Discovered Components pop-up.

b Select Edit ▸ IMS Definition on the primary product screen and press Enter to display the Edit - IMS pop-up.

c Add the discovered component using the procedures outlined in "Adding IMS Members" on page 67.

Identifying Missing Components

Application definitions can change over time due to renaming, movement, or deletion of source files. During the analyze process, components that were present in the last analyze but are no longer present are detected and reported through the Missing Components option on the View pull-down.

To view missing components

1 Select View ▸ Missing components on the primary product screen to display the Missing Components View screen, as shown in Figure 16.

Figure 16 • Missing Components View Screen Example

```
  File  Edit  Worklist  Options  Help
--------------------------------------------------------------------------------
                         Missing Components View                        ALEX01
Command ===> _____________________________________________ Scroll ===> CSR
                                                                        1 of 1

ENTITY KIND               ENTITY NAME (WITH QUALIFICATION)
------------------------- ----------------------------------------
********************************** TOP OF DATA **********************************
SOURCE LIBRARY            VIAJAZ.REL.CNTL
******************************** BOTTOM OF DATA *********************************
```

2 Review the missing components, if any are detected, to determine whether deleted components are still required for a complete definition. They may have been deleted or may no longer be a part of the application definition.

3 If the missing components were deleted by mistake, restore them in the proper source library. If they were deleted because they are no longer required by the application, remove them from the definition.

6 The Application Maintenance Facility

This chapter describes the Application Maintenance Facility (AMF) available in Alliance, Estimate, and Recap, and contains these sections:

Section	Page
Introduction	103
Opening the AMF	103
Editing the Application Definition	109

Introduction

This chapter provides a detailed description of the AMF that is used in Alliance, Estimate, and Recap. The AMF allows you to create and maintain application definitions; add source libraries, members, and copylib information to the application definition; and scan the existing source libraries to find new members, missing members or libraries, and modified source or include members.

Opening the AMF

Alliance and Recap

There are three ways to open the AMF when using Alliance or Recap:

- When you open an application, the AMF panel automatically displays. AMF reads and displays the application definition without scanning the source libraries for changes.
- Selecting File ▸ Scan application for changes from the main menu.

AMF compares the application definition to the underlying source libraries to look for members that are missing, new, or changed. The scanning process can take anywhere from a few seconds to several minutes, depending on the number of libraries and members.

- Selecting Facility ▸ Application Maintenance from the main menu.

  AMF reads and displays the application definition without scanning the source libraries for changes. This option is the fastest method for entering AMF.

Figure 17 shows the main screen for the AMF using either Alliance or Recap.

Figure 17 • The Application Maintenance Facility Screen in Alliance or Recap

```
  File  Edit  View  Facility  Select  Zoom  Options  Help
 ------------------------------------------------------------------------------
                         Application Maintenance Facility
 Command ===> _____________________________________________ Scroll ===> CSR

 Action: A=Anlz B=Brws C=Clr D=Del H=Hide M=Mod N=New O=Opt S=Sel X=Excl Z=Zoom
 1 lines hidden                                                        1 of 2
   Definition Entity                            Src  Type        Status      >
 - ------------------------------------------- ---- ----------- ------------->
 _  RW-ONLY                                          APPLICATION
 _  |-->+VIASOFT.COBOL.PANLIB                   PAN  COBOLII R3
********************************* BOTTOM OF DATA ********************************
```

Estimate

In Estimate, you enter the AMF by selecting Edit ▸ Application Definition while in the Enterprise View or the Impact View. With this option, the AMF does not scan the source libraries for changes. After entering the AMF, you can scan the application for changes by selecting File ▸ Scan from the File menu.

Figure 18 shows the Main Screen for the AMF using Estimate.

Figure 18 • The Application Maintenance Facility Screen in Estimate

```
  File  Edit  View  Select  Zoom  Options  Help
-------------------------------------------------------------------------------
                        Application Maintenance Facility
Command ===> _____________________________________________ Scroll ===> CSR

Action: A=Anlz B=Brws C=Clr D=Del H=Hide M=Mod N=New O=Opt S=Sel X=Excl Z=Zoom
1 lines hidden                                                          1 of 2
  Definition Entity                            Src  Type        Status       >
- ------------------------------------------- ---- ----------- -------------->
_  RW-ONLY                                          APPLICATION
_  |-->+VIASOFT.COBOL.PANLIB                   PAN  COBOLII R3
********************************* BOTTOM OF DATA *********************************
```

The AMF displays a hierarchical view of the application definition. You can scroll to the right to see other columns for each line, including the analyze return codes, analyze dates, and modification dates. You can change the order and width of the columns to fit more data into the view and execute various line commands that are described at the top of the screen, on the appropriate menu options, and in the help screen. A plus sign (+) next to a definition entity indicates that the line can be expanded.

Line Commands and Menu Options in AMF

Line Commands

These are the line commands available in AMF:

Line Command	Description
<Blank>	Pressing Enter with the cursor on a line toggles the selection/highlight of a line.
A	Selects a member to be analyzed.
B	Browses a member.
C	Unselects a member to be analyzed.
D	Deletes an item.
G	Adds a CSD group name.
H	Hides a line from view.
M	Modifies a library.
N	Adds a new member.

Line Command	Description
O	Displays the Edit options.
S	Selects a line.
U	Unexcludes a member.
X	Excludes a member.
Z	Zooms in.

AMF Menu Options

These are the options unique to the AMF pull-downs that apply to Alliance, Estimate, and Recap:

Menu Option	Pull-down	Used to
Scan application for changes	File	Rescan libraries.
Application Description	Edit	Create/edit the application description.
Edit Options	Edit	Edit library or member attributes.
Exclude from definition	Edit	Exclude member name from definition.
Unexclude from definition	Edit	Reverse the exclude operation.
Set PENDING ANALYZE status	Edit	Mark a member to be analyzed.
Clear PENDING ANALYZE status	Edit	Remove PENDING ANALYZE status from a member.
Include	View	Select statuses to include in view.
Format display	View	Change order and width of columns.
Initial Zoom	Zoom	Select the default zoom.
Zoom to library level	Zoom	Show source libraries and members.

Menu Option	Pull-down	Used to
Zoom to member level	Zoom	Show source libraries and members, but not attributes, such as copy libraries.
Zoom in all	Zoom	Show all lines.

These are the options unique to the AMF pull-downs that apply to Recap:

Menu Option	Pull-down	Used to
Reports	File	Generate reports from inventory analysis data in Recap.
Function Points	Edit	Modify function point parameters.
Enterprise Metrics	Facility	Specify an AKR name for which to view enterprise metrics.
Program Metrics	Facility	Specify an AKR name for which to view program metrics.
FP Task Manager	Facility	Modify function point parameters.

These are the options unique to the AMF pull-downs that apply to Alliance:

Menu Option	Pull-down	Used to
Query	Facility	Generate queries about an application.
Impact	Facility	Determine the impact of changing data elements.
System Cross-Reference Information	View	View information about application entity relationships, such as programs using CICS file, programs using CICS map, etc.
Program Cross-Reference Information	View	View specific information such as called program, calling programs, copybooks in program, etc.

Status Fields

The status displayed in the Status Field of the AMF screen depends on the state of the application definition and the outcome of the optional library scan.

Status	Specifies...
<Blank>	Good status, member is present, analyze return code is good, and the member has not been modified since its last analyze.
NOT ANALYZED	The member has not been analyzed.
BAD ANALYZE	The last analyze for the member was not successful.
PENDING ANALYZE	The member is selected for analyze.
MISSING `<item>`	The definition entity was not found during the AMF scan. Where `<item>` can be on of these: LIB, MEM, COPYLIB, or COPYMEM
MODIFIED	The source member or include member has been modified since the last analyze for the member.
NOT IN DEF	The library member is not currently in the definition.
EXCLUDED	The member has been excluded from the definition.

AMF determines the MODIFIED status by checking the ISPF statistics for the library member and comparing the modification time stamp to the last analyze timestamp in the AKR. If the ISPF statistics are not present for the member, AMF attempts to determine if the module has changed by calculating a CHECKSUM value for the member. The CHECKSUM value is calculated by reading each record in the member. During each analyze, a CHECKSUM value is also stored for the members and include members in the AKR. When a member is changed, its CHECKSUM value changes.

Note:

The CHECKSUM process may take a very long time for source libraries that do not have ISPF statistics. Select Options ▸ Options - Product Parameters to turn off the CHECKSUM calculation when ISPF statistics are missing. However, without the CHECKSUM or ISPF statistics, AMF is not able to determine if a member has changed. Another approach is to run the CHECKSUM process only during an auto-analysis batch job, described later in this chapter.

Editing the Application Definition

Edit the application by using the options under the Edit menu. Many of the options work with the set of selected lines in the view.

To add a library to the definition

1 Select Edit ▸ Add Library on the main view to display the Select Language Type pop-up, as shown in Figure 19.

Figure 19 • The Select Language Type Pop-up

```
  File  Edit  View  Facility  Select  Zoom  Options  Help
-        Select Language Type            ------------------------------------
C                                        nance Facility
                                                          Scroll ===> CSR
 Language Type:                          M=Mod N=New O=Opt S=Sel X=Excl Z=Zoom
A                                                                   1 of 4
3   1.  COBOL...                               Src  Type        Status       >
-   2.  LOAD MODULE...                      ----- ---- ----------- ------------->
_   3.  JCL...                                     APPLICATION
_   4.  CICS...                                PDS  IMS
_   5.  CSD...                                 PDS  JCL
_   6.  IMS...                                 PDS  COB MVS VM
_   7.  ASM...                              DATA ********************************
**  8.  PL/I...
    9.  NATURAL...
   10.  ALTERNATE...
```

2 Type the appropriate number for the language desired. The Add Source Library pop-up displays.

3 Follow the instructions in "Defining Application Components" on page 31 to add new libraries to the application definition.

To add a member to the definition

1 From the Application Maintenance Facility screen, highlight a library and select Edit ▸ Add New Member, as shown in Figure 20, or type `N` and press Enter to display the Add Member definition screen.

Figure 20 • Add Member Screen

```
File  Edit  View  Facility  Select  Zoom  Options  Help
 - .-------------------------------------------------------------. ------------
   |                          Add Member                         |
 C | Command ===> ______________________________________________ | 11 ===> CSR
   |                                                             |
 A | Specify member name.  Then press PF key for action.         | =Excl Z=Zoom
 1 | To get directory list, specify wild card characters or      |       1 of 2
   | leave it blank, then press Directory key.                   | atus        >
 - |                                                             | ------------>
 _ | Library . : 'VIATST.DEM.CNTL'                               |
 _ | Name  . . . __________                                      |
** |                                                             | *************
   |             PF5=Directory  PF6=Add  PF11=Browse             |
   '-------------------------------------------------------------'
```

Note:

The member must have NOT IN DEF status. If the library has multiple language types, AMF first prompts for the language type before displaying the Add Member screen.

2 Follow the instructions in "Defining Application Components" on page 31 to add new members to the definition. The status of newly-added members is NOT ANALYZED.

To edit options

1. Select either a library or member from the application definition.

2. Select Edit ▸ Edit Options on the main view, or type O on the command line and press Enter.

3. If a library is selected for editing, AMF displays the Library Options pop-up, as shown in Figure 21.

Figure 21 • Sample Library Options Screen

```
File  Edit  View  Facility  Select  Zoom  Options  Help
 ------------------------------------------------------------------------------
                         Application Maintenance Facility
 Command ===> VIAEDIT OPTIONS                                  Scroll ===> CSR

 Action: A=Anlz B=Brws C=Clr D=Del H=Hide M=Mod N=New O=Opt S=Sel X=Excl Z=Zoom
 1 lines hidden                                                        1 of 2
   Definition Entity                            Src  Type        Status       >
 - ---------------------------------------------- ---- ----------- ------------->
 _  JCLTEST                                            APPLICATION
 _  |-->+VIATST.DEM.CNTL                           PDS  JCL
******************************** BOTTOM OF DATA ********************************
                                     .-------------------------------------.
                                     |         JCL Library Options         |
                                     |                                     |
                                     | 1   1.  Analysis parms...           |
                                     |     2.  Maintain proclibs...        |
                                     |     3.  Exit                        |
                                     |                                     |
                                     '-------------------------------------'
```

If you select a member for editing, AMF displays the Member Options pop-up, as shown in Figure 22.

Figure 22 • Sample Member Options Screen

```
          COBOL Member Options

1   1.  Analysis parms...
    2.  Maintain copylibs...
    3.  IDMS information...
    4.  Restore library options...
    5.  Exit
```

4. Follow the instructions in "Defining Application Components" on page 31 to edit the options for a source library or member, such as analysis parameters or copy libraries.

To modify a library

1. Select a library from the application definition.
2. Select Edit ▸ Modify on the main view, or type `M` next to the library and press Enter.
3. For source libraries, AMF displays the Modify Library screen shown in Figure 23.

Figure 23 • Modify Library Screen

```
File  Edit  View  Facility  Select  Zoom  Options  Help
                          Modify Library
Command ===> ____________________________________________________
Modify data set attributes.  Then press PF key for action.
Data set name   'VIAM1J.C112224.PDS'
Source manager  1   1.  PDS
                    2.  Panvalet
                    3.  Librarian
                    4.  Sequential
                    5.  Endevor
                    6.  User
Volser  . . . . ______    (if volume not cataloged)
Password  . . .           (if password protected)
                  PF4=Librarian  PF5=Modify
```

For copy/include libraries, AMF displays a Modify Copylib screen shown in Figure 24.

Figure 24 • Example of Modify Copylib Screen

```
                          Modify Copylib
Command ===> ____________________________________________________
Modify data set attributes.  Then press PF key for action.
Data set name : 'VIAXE30.FP.SOURCE'
Source manager 1   1.  PDS           Usage 1   1.  COBOL compiler
                   2.  Panvalet                   2.  DB2 preprocessor
                   3.  Librarian                  3.  COBOL and DB2
Subsystem . . . _         (if panvalet or librarian)
Volser  . . . . ______    (if volume not cataloged)
Password  . . .           (if password protected)
                              PF5=Modify
```

4. Follow the instructions in "Defining Application Components" on page 31, to edit the options (such as analysis parameters or copy libraries) for a source library or member.

Excluding and Unexcluding Members from the Definition

The AMF displays members in source libraries that are not currently in the definition to highlight potentially new members. You have the ability to exclude member names from the definition so that the names are not presented in the future as potential new members. These excluded names are stored in the application definition for each source library.

To exclude members from the definition

1. Select a member from the application definition.

2. Select Edit ▸ Exclude definition on the main view, or type `X` next to the member and press Enter.

   Note:
   Members must have the status of NOT IN DEF to be excluded. The status changes to EXCLUDED after being excluded.

3. To unexclude members from the definition, select Edit ▸ Unexclude definition on the main view, or type `U` next to the member and press Enter.

   Note:
   Members must have the status of EXCLUDED to be unexcluded. The status changes to NOT IN DEF after being unexcluded. You can choose whether the AMF displays members with a status of NOT IN DEF or EXCLUDED by selecting View ▸ Include (see "To change AMF viewing options" on page 116).

To set/clear PENDING ANALYZE status

1. Select Edit ▸ Set PENDING ANALYZE on the main view, or type `A` next to the member, library, or application line and press Enter.

2. The status of the selected members are changed to PENDING ANALYZE. They will be analyzed during the next analyze job.

Follow these guidelines when selecting members to mark for analysis:

To mark	Select
All members in the application	The application line
All members within a library	The library line
One or more specific members	Each specific member

3 To clear PENDING ANALYZE status from members, select Edit ▸ Clear PENDING ANALYZE on the main view, or type `C` next to the member, library, or application line and press Enter.

The Auto-Analysis Feature

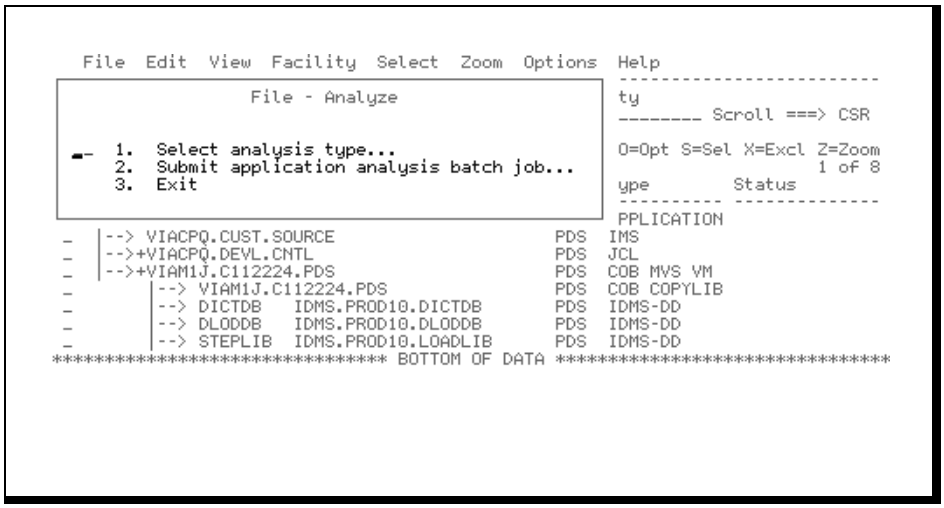
The AMF provides an Auto-Analysis option to make it easier to keep AKR applications up-to-date.

When you select the auto-analysis option, the AMF finds all members that have been modified since the last analyze, as well as all members that have not been analyzed or have bad analyze return codes. Each member is marked for an incremental analyze.

To perform an auto analysis

1 Select File ▸ Analyze to display the File - Analyze pop-up, as shown in Figure 25.

Figure 25 • File - Analyze Pop-up



2 Select File ▸ Select analysis type from the File - Analyze pop-up to display the Analyze - Select Analysis Type pop-up, as shown in Figure 26.

Figure 26 • Analyze - Select Analysis Type Pop-up

```
  File  Edit  View  Facility  Select  Zoom  Options  Help
                                                    ------------------------
                 File - Analyze                     ty
                                                    _ Scroll ===> CSR
            Analyze - Select Analysis Type
 1                                                  =Sel X=Excl Z=Zoom
                                                                 1 of 8
   Select the desired type of analysis                  Status
   1   1.  Full analysis...  all libraries/members      --- --------------
       2.  Incremental... select libraries/members      ION
       3.  Auto analysis...  auto-selected members
       4.  Exit                                         VM
                                                        LIB
       --> DICTDB   IDMS.PROD10.DICTDB       PDS  IDMS-DD
       --> DLODDB   IDMS.PROD10.DLODDB       PDS  IDMS-DD
       --> STEPLIB  IDMS.PROD10.LOADLIB      PDS  IDMS-DD
******************************** BOTTOM OF DATA ********************************
```

See "Analyzing the Application" on page 79 for more information about performing an application analysis.

You can set up a batch auto-analysis job to be run at regular intervals by adding this code to an application analyze job:

```
//VIASANIN DD *
<AUTOMATIC ANALYSIS>
<SEMANTIC LINK>
YES
/*
```

See "Scheduling an Application Analyze Through Batch Submission" on page 87 for more details.

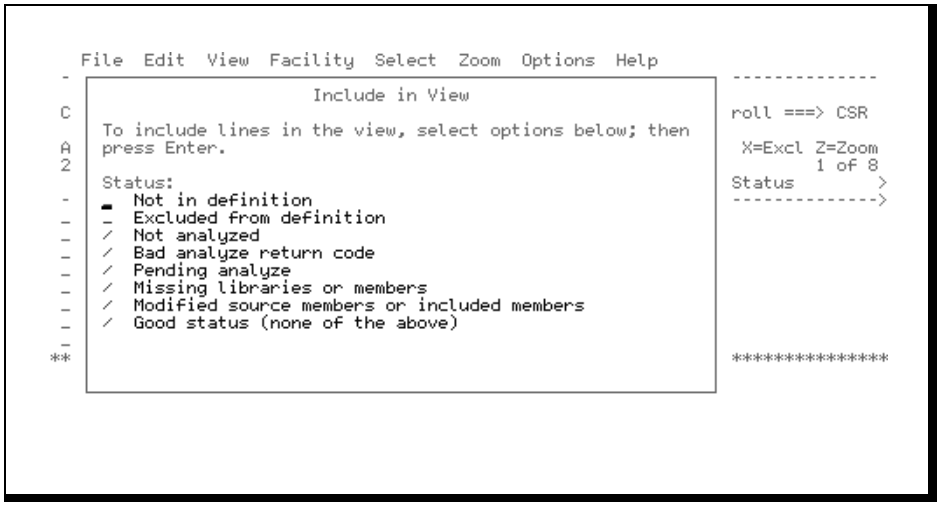
Changing the AMF View Include Options

You can adjust the parts of the definition that appear in the AMF. Lines that are included in the view may be shown or be hidden by using the zoom and hide commands. Lines that are not included in the display are always hidden.

To change AMF viewing options

1 Select View ▸ Include in the main view to display the Include in View pop-up, as shown in Figure 27.

Figure 27 • Include in View Pop-up



2 To include lines in the view, select the appropriate menu options and press Enter.

These are some of the commonly used options:

To see	Perform this
Library members not currently in the definition	Select the Not in definition option.
The entire definition	Select all statuses except Not in definition and Excluded from definition.
Members requiring attention only	Unselect the Good status option, which indicates the member has a good analyze return code, is still present in the source library, and has not been modified since the last analyze.

To change the main view display format

1 Select View ▸ Format in the main view to display the Format Display, as shown in Figure 28.

Figure 28 • Format Display Pop-up

```
File  Options  Help
------------------------------------------------------------------------
                              Format Display
Command ===> ___________________________________________________________

Column Name              Display order   Display width
--------------------     -------------   -------------              1 of 9
Definition Entity            1               44
Type                         3               11
Status                       4               15
Anlz RC                      5               7
Anlz Date                    6               11
Anlz Time                    7               9
Mod Date                     8               11
Mod Time                     9               8
Src Mgr                      2               4
***************************** BOTTOM OF DATA ******************************
```

2 Adjust the column order and width in the AMF view.

Note:

The Definition Entity column must remain in column 1. It remains locked in place when you scroll to the right, making it easier to line up the analyze return codes, analyze dates, and modification dates with the list of members.

To select lines in the AMF view

1 From the AMF main view screen, choose Select ▸ Select matching to display the Select Matching pop-up, as shown in Figure 29.

Figure 29 • Select Matching Pop-Up

```
                Select Matching - Application Maintenance Facility

Choose selection options and press Enter.
In order to match, an entry must match criteria in all the groups
selected.

   Select by Type:            Select by Status:
   _  COBOL                   _  Not in definition
   _  Load Module             _  Excluded from definition
   _  JCL                     _  Not analyzed
   _  CICS                    _  Bad analyze return code
   _  IMS                     _  Pending analyze
   _  PL/I                    _  Missing libraries or members
   _  ASM                     _  Modified source members or included
   _  Natural                    members
   _  Alternate               _  Good status (none of the above)

   Select by String:
      which contains:       ------------------------------
```

2 Select a group of lines based on specific criteria. Lines that match all of the specified criteria are highlighted in the display.

To set the default zoom upon entry to AMF

- Select Zoom ▸ Initial Zoom in the main view to display the Initial Zoom pop-up, as shown in Figure 30.

Figure 30 • Initial Zoom Pop-Up

```
File  Edit  View  Facility  Select  Zoom  Options  Help
                          Initial Zoom
Select one of the following for the initial zoom; then
press Enter.

Initial Zoom:
/  Zoom to library level
-  Zoom to member level
_  Zoom in all
```

This table contains the zoom options and their functions:

Zoom option	Description
Initial Zoom	Selects the default zoom option.
Zoom in	Expands a line (also accessible by pressing the PF6 key).
Zoom out	Collapses a line (also accessible by pressing the PF5 key).
Zoom to library level	Shows source libraries only.
Zoom to member level	Displays libraries and members, but not attributes such as copy libraries.
Zoom in all	Shows all lines.

To change the Zoom option

- You can expand and collapse the lines on the screen by using the Zoom options. Lines that can be expanded have a plus (+) sign to the left of the definition entity name.

Appendix A
Online and Batch AKR Utilities

This appendix describes the online AKR and batch AKR utilities. The online AKR utility lets you view and manipulate the contents of an AKR. The batch AKR utility lets you maintain an AKR without using ISPF.

Online Utility

The online AKR utility includes these pop-ups:

Pop-up Name	Performs
File - AKR Utility	• Renames or deletes a member. • Displays the AKR directory.
File - AKR Directory	• Displays all members in an AKR. • Renames or deletes a group of members. • Displays statistics about the AKR.
File - AKR Allocate/Expand	• Allocates a new AKR. • Expands an existing AKR.

When you use the Allocate/Expand utility, the default AKR organization type is applied. For example, if the default AKR type for your site is sequential, any new AKR allocated is a sequential file, and any non-sequential AKR you expand is reorganized as a sequential file.

For more information, see online help or the specific user guide for the product you are using.

Batch Utility

The batch AKR utility is used to maintain the AKR without using ISPF. This table lists the commands available in the batch AKR utility.

Command	Function	Page
ANLZSTAT	Produces the Analysis Status Report for one or more AKRs. **Note:** Applies to application-level products only.	125
COMMENTS	Includes comments with the command.	126
CONVERT	Converts selected members from a previous product release level to the current release level.	126
COPY	Copies selected members from one AKR to another.	128
DELETE	Deletes selected members from the AKR.	129
EXPORT	Creates metrics and function point CDF files. **Note:** Applies to Recap only.	130
HELP	Prints the AKR Utility Help Report.	131
INIT	Formats a previously-defined dataset into an AKR format.	132
MOVE	Copies selected members from one AKR to another and deletes them from the original AKR.	133
PRINT	Prints AKR directory information, or COBOL source listings for selected AKR members.	134
PUNCH	Produces a file that contains the AKR directory information, or the COBOL source code for selected AKR members.	135

Job Control Statements

The batch AKR utility uses these JCL statements:

- The VIAAKRIN and VIAAKROT DD statements describe AKRs that are used for AKR utility processing.
- The VIASYSIN DD control cards consist of the necessary batch AKR commands described in the previous section. See the description of each command to determine affected DD statements.

Figure 31 shows an example of batch AKR JCL statements.

Figure 31 • Batch AKR JCL Statements

```
//UTILITY  EXEC PGM=VIASAKRU,REGION=3000K,PARM=''
//STEPLIB  DD   DISP=SHR,DSN=(ASG load library)
//VIAAKRIN DD   DISP=SHR,DSN=(Input AKR)
//VIAPUNCH DD   SYSOUT=B (Punch file description)
//VIALOG   DD   SYSOUT=A (Log file)
//VIASYSIN DD   *
<control cards>
```

Control Cards

Commands are passed to the batch AKR utility with the control cards following the VIASYSIN DD statement. Control cards must conform to these standards:

- Command information must be contained in columns 1 through 72 of the control card.
- Only one command can be entered on each control card.
- Only one control card may be used per command.

All control cards with command disposition and command summaries are printed to the VIALOG AKR Utility Log file. Blank control cards are ignored.

Command Formats

Commands that use member names accept special characters to specify generic names. An asterisk (*) represents zero or more characters. A question mark (?) represents one character. For example:

Character	Description
DBA*	Specifies all members that begin with DBA and end with any other characters.
D?A*	Specifies all members that begin with D followed by one character, followed by an A, then followed by any other characters.
DBA???	Specifies all members that begin with DBA and end with any three characters.

Command Syntax

Each batch AKR utility command is described in this chapter. The descriptions include the format and a brief explanation of these command parameters.

Command	Description
ABBREViations	Command abbreviations are shown in uppercase letters; lowercase letters in the command are optional.
lowercase	Lowercase values indicate user-supplied variable information.
Underline	The default value of an operand is underlined.
| Vertical Bar	Separates synonymous commands or operands.
——►◄—	Right and left ending arrows indicate the end of the command syntax.
—— required ——	An operand or keyword appearing on the Command line is required.
┬┴optional┴┬	An operand or keyword appearing below the Command line is optional.
┬┬choice1┬┬ └└choice2┴┴	Operands stacked below the main line show a choice of optional items.

ANLZSTAT Command

```
ANLZSTAT ──────────────────────────────────────────────────────────────────►◄
```

Function

Produces the Analysis Status Report for one or more AKRs submitted with the job. The report provides this information:

- Analysis composition and status information for each AKR.
- Analysis composition and status information for applications within each AKR.
- Detailed analysis information about the components within each application.

Operand

None.

However the ANLZSTAT statement must be followed by one or more AKR dataset names, as shown in this example:

```
//VIAAKRU.VIASYSIN DD *
ANLZSTAT
VIAALxx.DEVL.SAMPAKR
ASG.VIACENxx.AKR
```

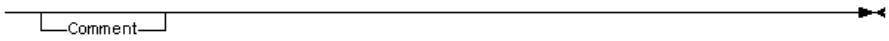
Usage Notes

The ANLZSTAT command is specified in the input card for the VIASAKRU batch AKR job. The VIAAKRIN file is not referenced by this command. ANLZSTAT must be followed by one or more AKR dataset names. Each AKR dataset name must be on a separate line.

A report is written to the VIASARPT DD name. In the default JCL, the DD name is defined as a SYSOUT file. You may override this DD statement and make it a permanent file; the permanent file must be deleted or a new file specified for each job run (see "Job Control Statements" on page 123).

The ANLZSTAT command looks at each component in each of the AKRs submitted in the job. The analysis status report is generated for application level components only, (Alliance, Recap, and Estimate applications). An analysis status report is not generated for program level components (for example, Encore and Insight programs).

COMMENTS Command



Function:

Includes a comment with the commands.

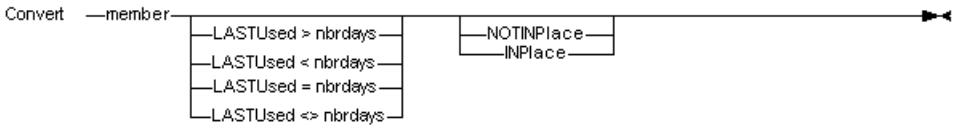
Operand:

Operand	Description
Comment	User-supplied text.

Usage notes:

Blank control cards are ignored.

CONVERT Command



Function:

Converts selected members that were analyzed by a prior release of ESW products to the current release level.

Note:

If you are converting from Recap 1.0 to 2.0 or a later release, use the procedure in the Recap Application Conversion section of the *ASG-Recap User's Guide*. These Operands and Usage Notes pertain to products other than Recap.

Operands

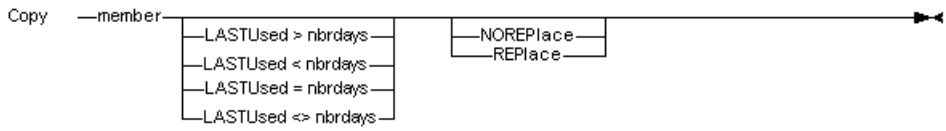
Operand	Description
member	Specifies a specific or generic name.
LASTUsed	Selects members based on the number of days since they were last used.
NOTINPlace	Specifies that a member with the same name on the receiving AKR should be replaced with the member from the sending AKR.
INPlace	Specifies that a member is to be converted and kept within the AKR named in the VIAAKRIN DD statement. **Note:** This option should be used with caution. Consult your systems administrator or the ASG Service Desk.

Usage Notes

Members are copied from the AKR specified in the VIAAKRIN DD statement to the AKR specified in the VIAAKROT DD. See "Job Control Statements" on page 123.

The LASTUsed, NOTINPlace, and INPlace operands are not valid for Recap.

COPY Command



Function

Copies selected members from one AKR to another.

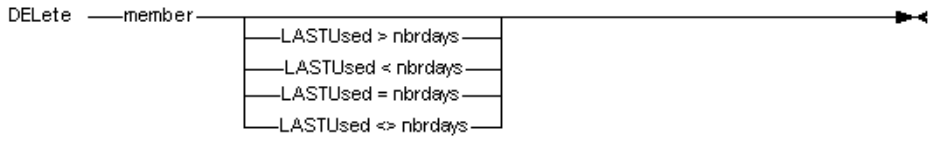
Operands

Operand	Description
member	Specifies a specific or generic name.
LASTUsed	Selects members based on the number of days since they were last used.
NOREPlace	Prevents existing members from being replaced by members with the same name. This is the default.
REPlace	Replaces members that have the same name on the receiving AKR.

Usage Notes

Members are copied from the AKR specified in the VIAAKRIN DD statement to the AKR specified in the VIAAKROT DD statement (see "Job Control Statements" on page 123).

DELETE Command



Function

Erases selected members from the AKR.

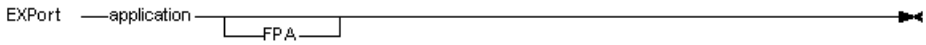
Operands

Operand	Description
member	Specifies a specific or generic name.
LASTUsed	Selects members based on the number of days since they were last used.

Usage Notes

Members are deleted from the AKR specified in the VIAAKRIN DD statement (see "Job Control Statements" on page 123). Members that begin with VIA cannot be deleted using this command. All ESW test members begin with VIA. If these members must be deleted, use the online AKR utility function described in the appropriate product user guide.

EXPORT Command



Function

Creates metrics and function point CDF files.

Operands

Operand	Description
application	Specifies a specific or generic name.
FPA	Generates only function point information. If FPA is not specified, both metrics and function point information are generated.

Usage Notes

EXPORT is available only for Recap users.

HELP Command

```
Help | ?  ──────────────────────────────────────────────────────────────────────────────────────►◄
```

Function

Prints a description of the batch AKR utility and the commands that can be used.

Operands

None.

Usage Notes

The HELP command has no operands.

A question mark (?) can be used as an alternate command.

The Help report is printed to the SYSOUT specified in the VIAPRINT DD statement (see "Job Control Statements" on page 123).

INIT Command

Function

Initializes a new AKR. This internal command is used by the online AKR utility allocation function described in the appropriate product's user guide.

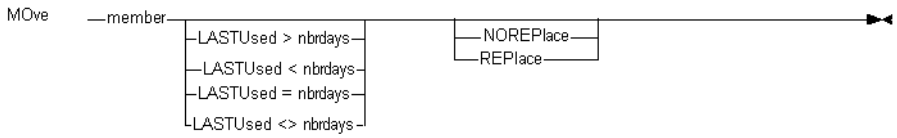
Operand

Operand	Description
DSname(dsname)	Specifies the dataset name for the new AKR.

Usage Notes

The AKR dataset to be initialized must be created prior to the initialization. The AKR that is initialized can be described in the VIAAKRIN DD statement. The VIAAKRIN DD statement is ignored if the DSNAME parameter is specified (see "Job Control Statements" on page 123).

MOVE Command



Function

Moves selected members from one AKR to another. Specified members are copied to the receiving AKR and erased from the sending AKR.

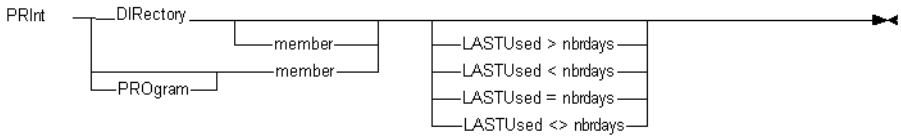
Operands

Operand	Description
member	Specifies a specific or generic name.
LASTUsed	Selects members based on the number of days since they were last used.
NOREPlace	Prevents existing members from being replaced by members with the same name. This is the default.
REPlace	Replaces members that have the same name on the receiving AKR.

Usage Notes

Members are moved from the AKR specified in the VIAAKRIN DD statement to the AKR specified in the VIAAKROT DD statement (see "Job Control Statements" on page 123).

PRINT Command



Function

Prints AKR directory information for the entire AKR, for a specified member, or for the source code for a specified member.

Operands

Operand	Description
blank	Specifies that if the PRINT batch AKR command is entered with no operand, the AKR directory information is printed.
DIRectory	Prints AKR directory information. This is the default. If a member is specified, the AKR directory information for that member only is printed.
PROgram	Prints the COBOL source for the specified AKR member. The generated COBOL source listing contains expansions of all copybooks or includes, as well as the results of any source preprocessors, such as CICS macro expansion (for program-level products only).
member	Specifies a specific or generic name.
LASTUsed	Selects a member based on the number of days since it was last used.

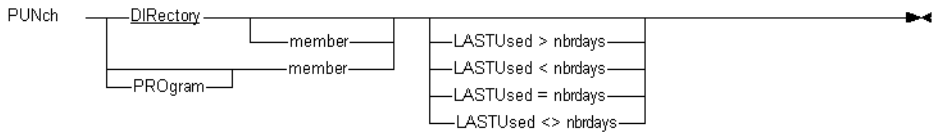
Note:

The output for the PRINT command does not include the source for the application product analyze types. In other words, you cannot use the PROGRAM option for application-level products (Alliance, Estimate, and Recap).

Usage Notes

Directory information or COBOL source is extracted from the AKR specified in the VIAAKRIN DD statement and is printed to the SYSOUT specified in the VIAPRINT DD statement (see "Job Control Statements" on page 123).

PUNCH Command



Function

Produces a file that contains AKR directory information for the entire AKR, for a specified member, or for the source code for a specified member.

Operands

Operand	Description
blank	Specifies that if the PUNCH batch AKR command is entered with no operand, a file is produced containing the AKR directory information.
DIRectory	Produces a file that contains directory information. This is the default. If a member is specified, the AKR directory information for that member only is printed.
PROgram	Produces a file that contains COBOL source code for the specified AKR member. The generated COBOL source listing contains expansions of all copybooks or includes, as well as the results of any source preprocessors, such as CICS macro expansion (for program-level products only).
member	Specifies a specific or generic name.
LASTUsed	Selects a member based on the number of days since it was last used.

Note:

The output for the PRINT command does not include the source for the Application product analyze types. In other words, you cannot use the PROGRAM option for application-level products (Alliance, Estimate, and Recap).

This table shows the format of the file produced by the PUNCH DIRECTORY command:

Description	Length	Format
Member name	10	Character
Number of source lines	6	Right justified
Days since last used	4	Right justified
Analyze date	9	DD MMM YY
Analyze job name	8	Character
Analyze CPU	4	Character
Analyze product level	8	Character
Last reference date	9	DD MMM YY
Last reference user ID	8	Character
Last reference CPU	4	Character
Analyze feature	2 and above	Character

Usage Notes

Directory information or COBOL source is extracted from the AKR specified in the VIAAKRIN DD statement and is written to the file specified in the VIAPUNCH DD statement. The file that is produced is in standard IBM IEBUPDTE Utility format. ADD control cards are produced for each logical entity.

The NAME parameter contains this information:

- The member name for COBOL source PROGRAM operand.
- AKRDIR`nn` for DIR operand, where `nn` is a consecutively assigned number (see "Job Control Statements" on page 123).

Batch AKR Reports

These reports are described in this section:

- AKR Utility Directory Report, as shown in Figure 32.
- AKR Utility Log (see Figure 33 on page 138).
- File produced by the PUNCH DIRECTORY command (see Figure 34 on page 139).
- "Analysis Status Report" on page 139.

AKR Utility Directory Report

The AKR Utility Directory report, shown in Figure 32, lists the results of a PRINT DIRECTORY command and is written to the VIAPRINT DD file. The title line contains the component level information, title, date, and time the job was executed.

The report lists the AKR dataset used, the command used to produce the report, and the directory information for the selected members.

Figure 32 • AKR Utility Directory Report

```
ASG CENTER-OS Rx.x LVLnnn              AKR UTILITY - DIRECTORY        DD MMM YY  HH:MM:SS  Page  1


    AKR: ASG.VIACENnn.AKR
Command: PRINT DIRECTORY * LASTUSE > 7

  Member          Last  ---- Analyzed --------v--- -- Last Referenced ----
   Name    Lines  Use     Date       Time      Job   CPU   Level     Date      Time     Job   CPU

 ACTG0018     40  16   DD MMM YY HH:MM:SS  SRENFKKA CPUA IN030000 DD MMM YY HH:MM:SS SRENCTY CPUA
 PYRL0085     17  16   DD MMM YY HH:MM:SS  SRENFKKA CPUA IN030000 DD MMM YY HH:MM:SS SRENDEM CPUA
 PYRL0105     17   8   DD MMM YY HH:MM:SS  SRENDEMA CPUA IN030000
 SR0005A     493   8   DD MMM YY HH:MM:SS  SRENJDRA CPUC IN030000
 W550044      66  12   DD MMM YY HH:MM:SS  SRENFKKA CPUC IN030000 DD MMM YY HH:MM:SS SRENCTY CPUA
 XRSCL070   1714  12   DD MMM YY HH:MM:SS  SRENDEMA CPUA IN030000 DD MMM YY HH:MM:SS SRENCTY CPUA
 XRSCL100     15   9   DD MMM YY HH:MM:SS  SRENFKKZ CPUC IN030000
 XRSCL200     41  10   DD MMM YY HH:MM:SS  SRENCTYZ CPUC IN030000 DD MMM YY HH:MM:SS SRENCTY CPUA

*** End of Directory Report ***
```

AKR Utility Log

The AKR Utility Log, shown in Figure 33, provides a summary of the commands issued to the batch AKR utility, and contains this information:

- Comments
- Commands
- Completion messages
- Short summary of commands processed

The heading includes the ASG component-level information, and the date and time the job was executed. Comments are enclosed in a box comprised of asterisks. The second page contains the log summary.

Figure 33 • AKR Utility Log

```
ASG-CENTER-03 Rx.x LVLnnn              AKR UTILITY LOG                DD MMM YY  HH:MM:SS  Page 1


*************************************************************          00140000
* PRODUCE A REPORT CONTAINING DIRECTORY INFORMATION FOR ALL  *         00150000
*   MEMBERS OF ASG.RENAISSA.AKR (VIAAKRIN) THAT HAVE         *         00160000
*   NOT BEEN REFERENCED IN THE LAST 7 DAYS.                  *         00170000
*************************************************************          00180000
*                                                                      00190000
PRINT DIRECTORY * LASTUSE > 7                                          00200000

  VIA1289I 8 DIRECTORY ENTRIES SUCCESSFULLY PRINTED.


                                                                       00210000
*************************************************************          00220000
* PRODUCE A FILE CONTAINING DIRECTORY INFORMATION FOR ALL    *         00230000
*   MEMBERS OF ASG.RENAISSA.AKR (VIAAKRIN) THAT HAVE         *         00240000
*   NOT BEEN REFERENCED IN THE LAST 7 DAYS.                  *         00250000
*************************************************************          00260000
*                                                                      00270000
PUNCH DIRECTORY * LASTUSE > 7                                          00280000

  VIA1290I 8 DIRECTORY ENTRIES SUCCESSFULLY PUNCHED.


  VIA1314I *** END OF VIASYSIN ***


ASG-CENTER-03 Rx.x LVLnnn              AKR UTILITY LOG - SUMMARY      DD MMM YY  HH:MM:SS  Page 2


  VIA1301I         8 DIRECTORY ENTRIES PRINTED     0 FAILED.
  VIA1302I         8 DIRECTORY ENTRIES PUNCHED     0 FAILED.

  VIA1315 *** END OF SUMMARY REPORT ***
§
```

AKR Punch Directory File

The AKR Punch Directory, shown in Figure 34, is written to the VIAPUNCH DD file when the PUNCH DIRECTORY command is processed. The file is formatted in standard IBM IEBUPDTE Utility format.

The first card, `./ADD...`, is an IEBUPDTE control card that indicates that the following cards are to be added to a partitioned dataset specified in the NAME parameter. The cards are in the format described in the PUNCH DIRECTORY command description. The last card is an IEBUPDTE control card that indicates the end of the control cards.

For the format of the AKR Punch Directory file, see the PUNCH batch AKR command.

Figure 34 • AKR Punch Directory File

```
./ ADD NAME=AKRDIR1,LIST=ALL
ACTG0018      40  16DD MMM YYSRENFKKACPUAIN030000DD MMM YYSRENCTY CPUA
PYRL0085      17  16DD MMM YYSRENFKKACPUAIN030000DD MMM YYSRENDEM CPUA
PYRL0105      17   8DD MMM YYSRENDEMACPUAIN030000
SR0005A      493   8DD MMM YYSRENJDRACPUCIN030000
W55004        66  12DD MMM YYSRENFKKACPUCIN030000DD MMM YYSRENCTY CPUA
XRSCL070    1714  12DD MMM YYSRENDEMACPUAIN030000DD MMM YYSRENCTY CPUA
XRSCL100      15   9DD MMM YYSRENFKKZCPUCIN030000
XRSCL200      41  10DD MMM YYSRENCTYZCPUCIN030000DD MMM YYSRENCTY CPUA
./ ENDUP
```

Analysis Status Report

The analysis status report provides a list of AKR names and statistics about the applications in each AKR, as the result of submitting an ANLZSTAT command. The report is written to the VIASARPT file and provides this level of information:

- AKR summary and detail
- Application summary and detail
- Library summary and detail

At each reporting level, information is given on the number of these items:

- Members
- Members successfully analyzed, unsuccessfully analyzed, and not analyzed
- Lines of source code in analyzed COBOL or Assembler programs

At the library level, this additional detailed information is provided for each member:

- Application component type (COBOL, JCL, LOADMODULE, etc.)
- Detailed analysis information (i.e., time and date, return code)

AKR Summary Information

The analysis status report begins with a cover page listing the source information for the report. The first section of the report provides AKR summary information, for example:

- (A) - the AKR's input for analysis.
- (B) - the number of applications and members in each AKR) and member analysis information, as shown in Figure 35.

Each AKR is identified with a number and name. That number is then used to reference the AKR.

Figure 35 • Analysis Status Report - AKR Summary

```
                         ANALYSIS STATUS REPORT                    PAGE:  1
                           DDMMMYYYY  HH:MM:SS

ANALYSIS REPORT SUMMARY

AKR COMPOSITION SUMMARY

                                                   COUNT(% OF TOTAL)
                                                 ------------------------------
 #  AKR                                         APPLICATIONS     MEMBERS
--- ------------------------------------------- -------------- ---------------
(A)1 VIAALxx.DEVL.SAMPAKR                            1(100)        22(100)
                                                 -------------- --------------
TOTAL                                                1(100)        22(100)
                                               ============== ==============


AKR ANALYSIS SUMMARY

          MEMBER COUNT(% OF TOTAL IN AKR)
       ---------------------------------------------
 #     SUCCESSFUL    UNSUCCESSFUL   NOT ANALYZED  LINES OF CODE(% OF TOTAL)
----- -------------- -------------- -------------- --------------------------
(B) 1       21( 95)        1(  5)        0(  0)              17,067(100)
      -------------- -------------- -------------- --------------------------
TOTAL       21( 95)        1(  5)        0(  0)              17,067(100)
    ============== ============== ============== ========================
```

AKR Detail Information

The second section of the analysis status report, shown in Figure 36, provides this information:

- (A) - a breakdown of the applications within each AKR.
- (B) - member analysis information.

Each application is identified with a number and name. That number is then used to reference the application.

Figure 36 • Analysis Status Report - AKR Detail

```
                          ANALYSIS STATUS REPORT                                PAGE:  2
                           DDMMMYYYY  HH:MM:SS

AKR DETAIL INFORMATION

AKR:  VIAALxx.DEVL.SAMPAKR

NUMBER OF APPLICATIONS:                    1
NUMBER OF LIBRARIES:                       3
NUMBER OF MEMBERS:                        22
LINES OF CODE:                        17,067


AKR COMPOSITION DETAIL

                                                      COUNT(% OF TOTAL)
                                                 ----------------------------
  #   APPLICATION DESCRIPTION                     LIBRARIES      MEMBERS
----- ----------- ------------------------------ -------------- --------------
(A) 1 ACCTS-PYBL  VIAALxx SAMPLE ACCTS-PAYBL APPLI        3(100)       22(100)
                                                 -------------- --------------
TOTAL                                                     3(100)       22(100)
                                                ============== ==============


AKR ANALYSIS DETAIL

       MEMBER COUNT(% OF TOTAL IN APPLICATION)
      ---------------------------------------------
  #    SUCCESSFUL    UNSUCCESSFUL   NOT ANALYZED  LINES OF CODE(% OF TOTAL)
----- -------------- -------------- -------------- -------------------------
(B) 1        21( 95)        1(  5)         0(  0)              17,067(100)
      -------------- -------------- -------------- -------------------------
TOTAL        21( 95)        1(  5)         0(  0)              17,067(100)
     ============== ============== ============== =========================
```

Application Detail Information

The last section of the analysis status report provides this information:

- A breakdown of the composition of each application and the members per library analysis information, as shown in Figure 37.
- Analysis/composition detail information for each member analyzed (see Figure 38 on page 143).

Each library is identified by a number and its name. This number is then used to reference the library.

Figure 37 • Analysis Status Report - Application Detail

```
                         ANALYSIS STATUS REPORT                    PAGE:  3
                           DDMMMYYYY  HH:MM:SS

APPLICATION DETAIL INFORMATION

APPLICATION:      ACCTS-PYBL

DESCRIPTION:      VIAALxx SAMPLE ACCTS-PAYBL APPLICATION
AKR:             VIAALxx.DEVL.SAMPAKR
TOTAL LIBRARIES:                 3
TOTAL MEMBERS:                  22
LINES OF CODE:             17,067

MEMBER ANALYSIS SUMMARY - MEMBERS(% OF TOTAL IN APPLICATION)
   SUCCESSFUL:             21( 95)
   UNSUCCESSFUL:            1(  5)
   UNANALYZED:              0(  0)


APPLICATION COMPOSITION SUMMARY

                                                   COUNT(% OF TOTAL)
                                              ------------------------------
 #   LIBRARY                                    MEMBERS       LINES OF CODE
---  ---------------------------------------- -------------- ----------------
(A)1 VIAALxx.SAMPAPPL.CNTL                          1(  5)          0(  0)
   2 VIAALxx.SAMPAPPL.COBOL                        15( 68)     17,067(100)
   3 VIAALxx.SAMPAPPL.LOADLIB                       6( 27)          0(  0)
                                              -------------- ----------------
TOTAL                                               22(100)     17,067(100)
                                              ============== ================
-
```

Figure 38 • Analysis Status Report - Composition Detail

```
APPLICATION ANALYSIS/COMPOSITION DETAIL

                                                    LAST ANALYSIS
            LIBRARY  LINES OF CODE                 --------------
 MEMBER     NUMBER    (% OF TOTAL)      TYPE  TIME    DATE     RC
-------    -------    -----------      -----  ----   -----    ----
CD05         (B) 3           0(  0) LOAD MOD MM:SS 12FEB1995  0
CD05APGM         2       2,525( 15) COBOL    MM:SS 12FEB1995  4
CD05BPGM         2         563(  3) COBOL    MM:SS 12FEB1995  0
CD05CPGM         2         632(  4) COBOL    MM:SS 12FEB1995  0
CD10             3           0(  0) LOAD MOD MM:SS 12FEB1995  0
CD10APGM         2       1,661( 10) COBOL    MM:SS 12FEB1995  0
CD10BPGM         2         680(  4) COBOL    MM:SS 12FEB1995  0
CD11             3           0(  0) LOAD MOD MM:SS 12FEB1995  0
CD110PGM         2       1,248(  7) COBOL    MM:SS 12FEB1995  0
CD15             3           0(  0) LOAD MOD MM:SS 12FEB1995  0
CD15APGM         2       2,597( 15) COBOL    MM:SS 12FEB1995  4
CD15BPGM         2       1,447(  8) COBOL    MM:SS
CD20             3           0(  0) LOAD MOD MM:SS 12FEB1995  0
CD20APGM         2         735(  4) COBOL    MM:SS 12FEB1995  0
CD20BPGM         2         934(  5) COBOL    MM:SS 12FEB1995  0
12FEB1995  0
CD18             3           0(  0) LOAD MOD MM:SS 12FEB1995  0
CD180PGM         2       1,295(  8) COBOL    MM:SS 12FEB1995  0
CD20CPGM         2         983(  6) COBOL    MM:SS 12FEB1995  4
CD20DPGM         2         863(  5) COBOL    MM:SS 12FEB1995  4
CD20EPGM         2         556(  3) COBOL    MM:SS 12FEB1995  0
CD200PGM         2         348(  2) COBOL    MM:SS 12FEB1995  8
DAILY            1           0(  0) JCL      MM:SS 12FEB1995  0
                       -------
TOTAL                   17,067(100)
                       ===============
```

Appendix B
Analyze Options and Parameters

This appendix has tables of information describing options and parameters available to analyze your application definition.

Analyze Options

Default options for the analyze process are established during product installation. Options that are to be overridden are specified when submitting the analyze job. This table lists allowable options. Unless otherwise specified, these options are valid for all products.

Option	Description
ALTIMP	Enables the use of the alternate assignments options during analysis with Alliance and Estimate. A full analysis must be performed.
BUF(*nnnnn*K) BUF=*nnnnn*K	Overrides the dynamically allocated amount of main storage to buffers and internal tables. Use this option if an override is necessary. Minimum value is 20K and the maximum is 20000K.
DB2LIB=*xxxxxx.xxxxx.xxxxxx* (*xxxxxx.xxxxx.xxxxxx* is DB2 load library dataset name)	Specifies the load library used to invoke on-site DB2 preprocessor.
DB2PLAN=*xxxxxxxx*	Specifies the application plan created at installation by the VIASBIND job. Use this option to override default plan name.
DETIMP	Enables the use of the Detailed Impact feature during analysis with Alliance. A full analysis must be performed.

Option	Description
DONOTHALT	Enables you to continue the analyze process after encountering a corrupted application to determine what other components are also failing. If you need to use this parameter, you have a corrupted permanent AKR for the application in question. After finding and fixing errors, export the current application definition, delete the application from the AKR, import the definition, and then rerun the analyze without the DONOTHALT parameter.
DYNcall NODYNcall	Specifies whether commands look up called programs by using the data names in dynamically CALLed programs. The default is DYNcall.
fLAGI fLAGW fLAGE fLAGS fLAGU fLAG(x)	Specifies the types of messages to be listed for the analyze job. Flag Level: • I Informational • W Warning • E Error • S Severe • U Unrecoverable **Note:** Some informational messages are produced regardless of the flag setting.
Input(*x*,*x*,...*x*) Input=*x*	Lists the CALLed programs that contain INPUT statements. When commands that search for INPUT are issued, statements that CALL these programs are shown in the command results. Specified programs are in addition to those specified at installation.
IO(*x*,*x*...*x*) IO=*x*	Lists CALLed programs that contain INPUT and OUTPUT statements. When commands that search for INPUT and OUTPUT are issued, statements that CALL these programs are shown in the command results. Specified programs are in addition to those specified at installation.

Option	Description
LIBERPCT=100	Specifies the percentage of each member type (COBOL, CICS, IMS,JCL, LOAD module) that can fail in a library during the analyze. When the percentage is reached for a specific member type, the analyze stops processing the library for that member type. A list of members not analyzed prints in the VIASARPT file in the analyze job. Default percentage is set at installation.
LIBMCD(*xxxx*) *xxxx* is a 4-digit security code.	Lists source members stored on a CA-Librarian master file that require specification of the MCD security code to retrieve the source from the master file.
LineCNT=60	Specifies number of lines, 1 to 99, to print per page for the source listing. The default value is 60.
LITELINK	Specifies that semantic linking should not build lists of global synonyms and homonyms. This saves execution time when the current task does not require global synonyms and homonyms.
MAIN	Overrides the standard exit processing. EXIT PROGRAM statements in COBOL programs are treated as GOBACKS by the analyze job because the program is considered a CALLed subprogram. If the program is the main program, the MAIN option treats the EXIT program as a fallthrough.
MBRERCNT=4000	Specifies the maximum number of analysis errors (1 to 4000) allowed per member during analyze before the analyze terminates processing for that member. The default is set at installation.
MEMNAME	Specifies that member name is used as a program name instead of the program ID. Members defined in libraries with the same program ID for every member are distinguished separately. All references to program ID are the member name. **Note:** Recap and Insight only. Recap batch reports show the member name instead of the program ID in reports that display the program ID. For a batch Insight analyze, the member name is used as the AKR member name.
NOFDREDEF	Stops records within FDs from redefining each other. This parameter speeds up synonym calculation.

Option	Description
NOInput(*x,x,…x*) NOInput=*x*	Overrides the installation default list of CALLed programs containing INPUT statements. Specified programs are deleted from the default list.
NOIO	Overrides the installation default list of CALLed programs containing INPUT and OUTPUT statements. Specified programs are deleted from the default list.
NOOutput(*x,x,…x*) NOOutput=*x*	Overrides the installation default list of CALLed programs containing OUTPUT statements. Specified programs are deleted from the default list.
NOPSBCHECK NODBDCHECK	Allows the AAE to associate segment references, when an application allows segment access outside the normal IMS access rules. • NOPSBCHECK associates all valid segment references with the same valid name regardless of whether the program has access to the PSB. If more than one DBD contains segments of the same name, the association does not occur. • Used with NOPSBCHECK, NODBDCHECK associates all valid segment references with the same valid name even if more than one DBD contains segments of the same name. • NODBDCHECK must be used with NOPSBCHECK. Used alone, it has no effect. NODBDCHECK affiliates the segment references; no reporting occurs. An evaluation of the segments defined to the application allows you to determine which of the options, if any, are needed. If named segments of different DBDs represent different data types, use NOPSBCHECK only. If named segments of different DBDs represent the same data type, consider using NODBDCHECK with NOPSBCHECK. Possible indicators of equivalent data types include equivalent segfield definitions or if the containing DBDs are associated to the same physical data file.
NORETurn(*x,x,...x)* NORETurn=*x*	Lists additional programs or entry points that are not to return when CALLed. When any of these programs are CALLed by the program being analyzed, they are treated as non-returning CALLs. Specified programs are in addition to the system defaults for programs that do not return when CALLed.

Option	Description
Output(*x,x,...x*) Output=*x*	Lists the CALLed programs that contain OUTPUT statements. When commands that search for OUTPUT are issued, statements that CALL these programs are shown in the command results. Specified programs are in addition to those specified at installation.
PANLOG	Specifies PANLOG DD statement is added to the analyze JCL. If the site Panvalet implementation requires the PANLOG DD statement, PAN#1 (Panvalet extract utility program) abends if this analyze option is not specified.
PANSYSIN (*xxxxx.xxxx.xxxx*)	Provides the name of a PDS that contains Panvalet control statement to extract the Panvalet source member into a temporary file. If PANSYSIN is not specified, a ++WRITE statement extracts the member. The PDS dataset name is used during the analyze as the SYSIN DD for invocation of the Panvalet extract utility program (PAN#1). The PDS member name is the name of the source member being extracted from the Panvalet dataset. If the Panvalet source member exceeds 8 characters, then characters 9 and 10 are truncated to form the PDS member name.
RETurn(*x,x,...x*) RETurn=*x*	Overrides the installation list of programs or entry points that do not return when CALLed. The system defaults are overridden by listing the desired programs or entry points that are to return when CALLed.
SOUrce NOSOUrce	Specifies whether the source program is to be listed. The SOURCE option is specified if a full program listing is desired at analyze time. Default is NOSOURCE.
spACE1|2|3	Specifies spacing for source listing generated when SOURCE option is used. Default is single.
SQLID=*x* SQLID(*x,x,...x*) *X* is 8 character authorization ID or owner used by the analyze process.	Specifies authorization ID or owner used by the analyze process to qualify unqualified table and view references in the program. The SQLID option can also be specified in the application definition Analysis Parms - Compile Exec Parms field, either at the library or member level.
SUBSYS = *xxxxxxxx* is name of subsystem or location of DBMS.	Specifies subsystem or location then designates the DBMS where tables accessed by a specified program are stored. SUBSYS overrides the name provided at installation.

Option	Description
SYNOPT1	Specifies that dataitems that are direct assignments must be exactly the same size to be recorded as synonyms. Default is to process any direct assignment as synonyms. **Note:** A full analysis must be performed if you specify SYNOPT1. For application-level products only.
SYNOPT2	Prints the list of data items that would have been recorded as synonyms if SYNOPT1 had not been specified. SYNOPT2 has no impact if SYNOPT1 is not specified.
SYNOPT3	Eliminates renames and redefinitions as synonyms. **Note:** A full analysis must be performed if you specify SYNOPT3. For application-level products only.
XLIVE	Used as an override only with programs that contain live exits. Live exits are exits from perform ranges that are left dangling by imbedded PERFORMs or GOTOs in the original PERFORMed paragraph. If XLIVE is not used, code that is unprocessed because of the live exit is ignored. If it is used, unprocessed code is saved.
XMEM	Overrides the current IO and CPU usage settings. If a program is extremely large (in excess of 30,000 source lines) and there is still insufficient memory after increasing region space, enter the XMEM option. Although this results in more disk I/O and additional CPU usage, there is less memory consumption.

Analyze Parameters

The AAE makes use of compiler and source manager parameters that are entered on the application definition panels. You can specify these parameters at the library or member level. For COBOL members, analyze parameters are entered in the Source manager exec parms field on the Library Options - Analysis Parms pop-up or the Member Options - Analysis Parms pop-up. For non-COBOL members, analyze parameters are entered in the Source mgr exec parms field on the Library Options - Analysis Parms pop-up or the Member Options - Analysis Parms pop-up.

Parameters	Description
AUXINC	For Librarian, specifies that up to eight AUXINCnn COPY libraries in the definition can be allocated using the DD names AUXINC01 through AUXINC08. When the AUXINC parameter is used, it must be entered with the parameter string `OEX=AUXINC,NRJS,NJTA`.
COPYDD = `<name>`	Allows you to specify a DD name so that the copylib list can be available to the user exit.
COPYMAST	Specifies that the first COPY library found, that is Librarian and specified with `LAM=NO/FALSE`, is dynamically allocated as DD name COPYMAST. For the COPYMAST parameter to operate correctly, the installation option `Librarian-Extract=NO` must be specified. **Note:** To use COPYMAST and AUXINC together, enter the parameter string `COPYMAST,OEX=AUXINC,NRJS,NJTA`. In this case, the first COPY library found, that is Librarian and specified with `LAM=NO/FALSE`, is dynamically allocated as DD name COPYMAST. The next eight copy libraries found that meet the same criteria are allocated as AUXINC01 through AUXINC08.
INDD = `<name>`	Allows you to override the default user exit input DD name, such as VIASCOBI for COBOL.
OUTDD = `<name>`	Allows you to override the default user exit output DD name, such as VIASCOBO for COBOL.

Parameters	Description
SQLID=x SQLID (x,x,...x) X is 8 character authorization ID or owner used by the analyze process.	Specifies authorization ID or owner used by the analyze process to qualify unqualified table and view references in the program. The SQLID option can also be specified in the application definition Analysis Parms - Compile Exec Parms field, either at the library or member level.
USEREXIT = <*name*>	Allows you to override the default user exit name, such as VIASCOBO for COBOL.

User exit name	Specifies the name of a user exit to be executed by the application analyzer for the purpose of processing source members before they are analyzed. This parameter must be the module name of the user exit in use at each site. For example, the ASG-supplied COBOL user exit has the name VIACOBU. To execute this exit, enter the parameter VIACOBU. There may be user exits for each type of source member that can be defined:
	`COBOL VIASCOBU`
	`IMS VIASIMSU`
	`CICS VIASCCSU`
	`JCL VIASJCLU`
	`Assembler VIASASMU`
	`PL/I VIASPLIU`
	Note:
	The user exit program must be compiled and linked into a load library accessible to the application analyzer. You can specify the load library with the parameter Analyze - Steplib - Libraries.

Appendix C
Import Language Reference

This appendix describes how to prepare an application for import into an AKR.

Import Language

Import Language Syntax

Import statements may start anywhere on the line. Keywords are enclosed by the tag <>. Each keyword must be coded on a separate line. Each item associated with a keyword must also be coded on a separate line as shown in Figure 39.

Figure 39 • Import Data Item Syntax

```
<AKR>
  USER.TEST.AKR
<APPLICATION NAME>
  MY-APPL
```

Import File Structure

The Import file for the application definition contains three major sections:

Section Name	See page
Application information	154
Default information	172
Library and Member Definition	174

Note:

The easy way to create an application import file is to export an existing application definition. For more information, see the appropriate product user guide.

Application Information Section

The Application Information section specifies the name of the AKR where the application is stored along with the application name and application description. This table shows the application information that should be specified:

Entry	Description
AKR	Identifies the dataset name of the AKR where the application is stored. This entry is required.
Product	Indicates whether the application definition is used in Recap, Alliance, or Estimate. If more than one product is specified, separate the entries by a comma. Valid entries are ASG-RECAP, ASG-ALLIANCE, ASG-ESTIMATE.
Application name	Specifies a 1 through 10 character name for the application you are defining. This entry is required.
Application description	Specifies a 1 through 40 character description. This entry is optional.

Figure 40 shows the code for the Application Information section of an import file.

Figure 40 • Application Information Section Example

```
<AKR>
 USER.TEST.AKR
<PRODUCT>
 RECAP
<APPLICATION NAME>
 IMP-FULL
<APPLICATION DESCRIPTION>
 COMPLETE APPLICATION TEST
```

Default Dataset and Attribute Information Section

The Default Information section is used to specify the default information for the libraries defined to the application. All defaults are optional. The defaults specified in the import file override any corresponding site defaults that were specified during the installation of Recap or Alliance. The default section must begin with the keyword DEFAULT START and end with the keyword DEFAULT END. This table contains the default information that can be specified:

Default Information	Description
FILEORG	Identifies the default source manager for the library dataset specified. If not specified, the FILEORG defined in the default section is used. Valid entries are PDS, PAN, LIB, VSAM, SEQ, ENDV, or USER.
COBOL VERSION	Identifies the default COBOL version for COBOL libraries and members defined to the application. Only one version can be specified. If the COBOL version is not specified, the COBOL version specified during product installation is used by default. Valid entries are COB68, COB74, COB2, COB2R3, COB370, COBMVSVM, COBOS390.
COMPILE EXECUTION PARM	Identifies the compile batch execution parameters used for COBOL libraries defined to the application.
COBOL SOURCE EXTRACTION EXECUTION PARM	Identifies the default source library manager batch execution parameters used for COBOL libraries defined to the application.

Default Information	Description
JCL SOURCE EXTRACTION EXECUTION PARM	Identifies the default source library manager batch execution parameters used for JCL libraries defined to the application.
CICS SOURCE EXTRACTION EXECUTION PARM	Identifies the default source library manager batch execution parameters used for CICS libraries defined to the application.
IMS SOURCE EXTRACTION EXECUTION PARM	Identifies the default source library manager batch execution parameters used for IMS libraries defined to the application.
PLI VERSION	Identifies the default PL/I version for PL/I libraries and members defined to the application. Only one version can be specified. If the PL/I version is not specified, the PL/I version specified during product installation is used by default. Valid entries are 1.5, 2.x, MVS&VM.
PLI COMPILE EXECUTION PARM	Identifies the compile batch execution parameters used for PL/I libraries defined to the application.
PLI PROCESS OPTION	Identifies the dataset name and member defined to the application.
PLI SOURCE EXTRACTION EXECUTION PARM	Identifies the default source library manager batch execution parameters used for PL/I libraries defined to the application.
ALTERNATE LANGUAGE SOURCE EXTRACTION EXECUTION PARM	Identifies the default source library manager batch execution parameters used for alternate language libraries defined to the application.
ALTERNATE LANGUAGE OTHER EXECUTION PARM	Identifies the compile batch execution parameters used for other alternate language libraries defined to the application.
ASM SOURCE EXTRACTION EXECUTION PARM	Identifies the default source library manager batch execution parameters used for Assembler libraries defined to the application.
STDSQL(86)	Identifies the default SQL usage found in the JCL. If the DB2 STDSQL(86) parm is found in the compile/link JCL, the default is YES. Valid entries are YES and NO.

Default Information	Description
COPYLIB | INCLUDE | ALTERNATE LANGUAGE INCLUDE	Identifies the default copy library attributes for COBOL libraries and members defined to the application. Valid entries are names of valid copy libraries. Each library name must appear on a separate line. These are attributes of the default copylib. Each attribute must appear on a separate line after the valid default copy library entry.
COPY FILEORG | INCLUDE FILEORG | ALTERNATE LANGUAGE INCLUDE FILEORG	Identifies the source manager that is used for the specified copy/include library. Only one source manager can be specified. Valid entries are PDS, PAN, or LIB.
COPY SUBSYSTEM | INCLUDE SUBSYSTEM | ALTERNATE LANGUAGE INCLUDE SUBSYSTEM	Indicates whether the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The copy subsystem is only valid for Panvalet and Librarian copy libraries. Valid entries are YES and NO.
COPY PASSWORD | INCLUDE PASSWORD | ALTERNATE LANGUAGE INCLUDE PASSWORD	Identifies the password for the specified copy library. If the library is password protected, enter the password.
COPY VOLSER | INCLUDE VOLSER | ALTERNATE LANGUAGE VOLSER	Identifies the volume serial number for the specified copy library. If the library is not cataloged, enter the volume serial number.
COPY USAGE | INCLUDE USAGE | ASM MACLIB USAGE | ALTERNATE LANGUAGE INCLUDE USAGE	Identifies the location of the SYSLIB dataset in the compile JCL. Usage information is used to determine when the AAE needs to use these copy libraries defined to the application: during COBOL | PL1 | ASM | ALTERNATE LANGUAGE library processing, during DB2 include processing, or during both types of processing. Only one COPY USAGE | INCLUDE USAGE | ASM MACLIB USAGE can be specified. Dataset — Valid Entries COPY USAGE — COB, DB2, BOTH INCLUDE USAGE — PLI, DB2, BOTH ASM MACLIB USAGE — ASM, DB2, BOTH ALTERNATE LANGUAGE INCLUDE USAGE — ALTERNATE, DB2, BOTH

Default Information	Description
CA-LIBRARIAN MEMBER COPYDD	Indicates whether the CA-Librarian master file for CA-Librarian members in the application definition requires the CopyDD feature. If this value is not specified, the default value is NO. Valid entries are YES and NO.
CA-LIBRARIAN MEMBER PSWD EXISTS	Indicates whether the CA-Librarian members in the application are password-protected. If this value is not specified, the default value is NO. Valid entries are YES and NO.
PREPROCESSOR OPTION	Identifies preprocessor defined PL1 | ASM library. Any combination of entries (except NONE) is accepted by specifying each option on a separate line. Valid entries are ADMS, DB2, CICS, and NONE.
IMS MACLIB	Identifies the default maclibs for IMS libraries included in the application definition. Valid entries are names of valid maclibs. Each library name must appear on a separate line. These are attributes of the default maclib for IMS members. Each attribute must appear on a separate line after the entry for the valid default maclib.
IMS MACLIB FILEORG	Identifies the source manager that is used for the specified maclib. Only one source manager can be specified. Valid entries are PDS, PAN, and LIB.
IMS MACLIB SUBSYSTEM	Indicates whether the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The maclib subsystem is only valid for Panvalet and Librarian libraries. Valid entries are YES and NO.
IMS MACLIB PASSWORD	Identifies the password for the specified macro library. If the library is password protected, enter the password.

Default Information	Description
IMS MACLIB VOLSER	Identifies the volume serial number for the specified macro library. If the library is not cataloged, enter the volume serial number.
CICS MACLIB	Identifies the default maclibs for CICS libraries included in the application definition. Valid entries are names of valid maclibs. Each library name must appear on a separate line. These are attributes of the default maclib for CICS members. Each attribute must appear on a separate line after the entry for the valid default maclib.
CICS MACLIB FILEORG	Identifies the source manager that is used for the specified maclib. Only one source manager can be specified. Valid entries are PDS, PAN, LIB, and ENDV.
CICS MACLIB SUBSYSTEM	Indicates whether the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. The maclib subsystem is only valid for Librarian and Panvalet libraries. Valid entries are YES and NO.
CICS MACLIB PASSWORD	Identifies the password for the specified macro library. If the library is password protected, enter the password.
CICS MACLIB VOLSER	Identifies the volume serial number for the specified macro library. If the library is not cataloged, enter the volume serial number. For more information about specifying maclibs in the application definition, see "Specifying Maclibs (CICS Libraries)" on page 59 and "Specifying Maclibs (CICS Members)" on page 62.
PROCLIB	Identifies the default procedure libraries for JCL libraries and members included in the application definition. Valid entries are names of valid procedure libraries; each library name must appear on a separate line.

Default Information	Description
ASM VERSION	Identifies the default Assembler version for Assembler (ASM) libraries and members defined to the application. Only one version can be specified. Valid entries are ASMF, ASMH, and HLASM.
ASSEMBLER EXECUTION PARM	Identifies the default Assembler batch execution parameters used for ASM libraries defined to the application.
ASM MACLIB	Identifies the default maclibs for ASM libraries included in the application definition. Valid entries are names of valid maclibs. Each library name must appear on a separate line. These are attributes of the default maclib for ASM members. Each attribute must appear on a separate line after the entry for the valid default maclib.
ASM MACLIB FILEORG	Identifies the source manager that is used for the specified maclib. Only one source manager can be specified. Valid entries are PDS, PAN, and LIB.
ASM MACLIB SUBSYSTEM	Indicates whether the SYSLIB DD statement in the assemble JCL makes use of the SUBSYS parameter. Valid entries are YES and NO.
ASM MACLIB PASSWORD	Identifies the password for the specified macro library. If the library is password protected, enter the password.
ASM MACLIB VOLSER	Identifies the volume serial number for the specified macro library. If the library or member is not cataloged, enter the volume serial number.

Figure 41 shows the code for the Default section of an Import file.

Figure 41 • Default Information Section Example

```
<DEFAULT START>
   <FILEORG>
      PDS
   <COBOL VERSION>
      COB2
   <COPYLIB>
      SYS1.COBLIB
   <IMS MACLIB>
      IMSESA.MACLIB
   <CICS MACLIB>
      CICS330.MACLIB
   <PROCLIB>
      SYS1.PROCLIB
      SYS2.PROCLIB
   <COMPILE EXECUTION PARM>
      LIB,QUOTE
   <CA-LIBRARIAN MEMBER PSWD EXISTS>
      NO
   <CA-LIBRARIAN MEMBER COPYDD>
      NO
<DEFAULT END>
```

Library Member and Definition Section

The Definition section is used to specify these types of application components:

- COBOL
- Load module
- JCL
- CICS CSD files
- CICS PPT, PCT, and FCT tables
- CICS BMS maps
- IMS DBD, PSB, DFSMDA, MFS, and Stage 1 libraries
- ASM, PL/I, NATURAL, and ALF

Each library must be defined separately and any members defined for that library must be specified immediately following the library definition. These keywords are used to define the various components of the library definitions.

Keyword	Description
LIBRARY	Identifies the dataset name for the COBOL, ASM, Load, JCL, PL/I, Alternate, CICS, IMS library, or CICS CSD file being defined. This keyword must precede each library definition. Only one library can be defined at a time. Valid entries are a valid dataset name.
FILEORG	Identifies the source manager that is used for all COBOL, ASM, JCL, CICS, IMS, PL/I, NATURAL, and alternate language datasets included in the application definition. Only one source manager can be specified. If not specified, the FILEORG defined in the default section is used. Valid entries are PDS, PAN, LIB, VSAM, SEQ, ENDV, and USER.
OBJECT KIND (required)	Identifies the library type for the dataset specified. This entry is required for each library defined to the application. Only one library type can be specified at a time. Valid entries are CICS, COBOL, IMS, JCL, LOAD, CSD, PL/I, ASM, and Alternate.
COBOL VERSION	Identifies the COBOL version for the COBOL library and members defined to the application. Only one version can be specified. Valid entries are COB68, COB74, COB2, COB2R3, COB370, COBMVSVM, and COBOS390.
COMPILE EXECUTION PARM	Identifies the default compile batch execution parameters used for COBOL libraries defined to the application. Compile execution parms can only be specified within a COBOL library definition or in the MEMBER OVERRIDE information for COBOL member(s).
COBOL SOURCE EXTRACTION EXECUTION PARM	Identifies the source library manager batch execution parameters used for COBOL libraries defined to the application.

Keyword	Description
JCL SOURCE EXTRACTION EXECUTION PARM	Identifies the source library manager batch execution parameters used for JCL libraries defined to the application.
CICS SOURCE EXTRACTION EXECUTION PARM	Identifies the source library manager batch execution parameters used for CICS libraries defined to the application.
IMS SOURCE EXTRACTION EXECUTION PARM	Identifies the source library manager batch execution parameters used for IMS libraries defined to the application.
STDSQL(86)	Identifies the SQL usage found in the JCL. If the DB2 STDSQL(86) parm is found in the compile/link JCL, the default is YES. Valid entries are YES and NO.
PASSWORD	Identifies the password for the specified COBOL, JCL, CICS, IMS, PL/I, ASM, or Alternate library or member, if the library or member(s) is password protected. Enter the password.
VOLSER	Identifies the volume serial number for the specified COBOL, JCL, CICS, IMS, PL/I, or Alternate library or member(s) if the library or member is not cataloged. Enter the volume serial number.
CA-LIBRARIAN MEMBER PASSWORD	Identifies the password for a CA-Librarian member if that member in the application is password protected. The member is password protected if the <CA-LIBRARIAN MEMBER PSWD EXISTS> is set to YES.
COPYLIB	Identifies the copy library for the specified COBOL library or member(s). Each copy library must appear on a separate line. Copy library information can only be specified within a library definition or in the MEMBER OVERRIDE information. For each copy library, the information listed in the next table can be specified.
COPY FILEORG	Identifies the source manager that is used for the specified copy library. Only one source manager can be specified. Valid entries are PDS, PAN, and LIB.

Keyword	Description
COPY SUBSYSTEM	Indicates whether the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. Valid entries are YES and NO.
COPY USAGE	Identifies the location of the SYSLIB dataset in the compile JCL. Usage information is used to determine when the application Analytical Engine needs to use the copy libraries defined to the application: during COBOL library processing, during DB2 include processing, or during both types of processing. Only one COPY USAGE can be specified. Valid entries are COB, DB2, PLI, ASM, and BOTH. BOTH means COB & DB2, PLI & DB2, and ASM & DB2.
COPY PASSWORD	Identifies the password for the specified copy library. If the library is password protected, enter the password.
COPY VOLSER	Identifies the volume serial number for the specified copy. If the library or member is not cataloged, enter the volume serial number.

Figure 42 shows the code to define copy library information for MY.FIRST.COBOL.LIB.

Figure 42 • Copy Library Attribute Definition

```
<LIBRARY>
   MY.FIRST.COBOL.LIB
   <OBJECT KIND>
      COBOL
   <COBOL VERSION>
      COB68
   <PASSWORD>
      SRCPSWD
   <VOLSER>
      SRC001
      <COPYLIB>
       MY.FIRST.COPY.LIB
         <COPY FILEORG>
          PDS
         <COPY PASSWORD>
          CPYPSWD
         <COPY VOLSER>
          SRC001
         <COPY USAGE>
          COB
```

Keyword	Description
IMS MACLIB	Identifies the maclib attributes for IMS libraries and members included in the application definition. Valid entries are names of valid macro libraries. Each library name must appear on a separate line.
IMS MACLIB FILEORG	Identifies the source manager that is used for the specified maclib. Only one source manager can be specified. Valid entries are PDS, PAN, or LIB.
IMS MACLIB SUBSYSTEM	Indicates whether the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. Valid entries are YES and NO.
IMS MACLIB PASSWORD	Identifies the password for the specified macro library. If the library is password protected, enter the password.
IMS MACLIB VOLSER	Identifies the volume serial number for the specified macro library. If the library or member is not cataloged, enter the volume serial number.
CICS MACLIB	Identifies the maclib attributes for CICS libraries and members included in the application definition. Valid entries are names of valid macro libraries. Each library name must appear on a separate line.
CICS MACLIB FILEORG	Identifies the source manager that is used for the specified maclib. Only one source manager can be specified. Valid entries are PDS, PAN, or LIB.
CICS MACLIB SUBSYSTEM	Indicates whether the SYSLIB DD statement in the compile JCL makes use of the SUBSYS parameter. Valid entries are YES and NO.

Keyword	Description
CICS MACLIB PASSWORD	Identifies the password for the specified macro library. If the library is password protected, enter the password.
CICS MACLIB VOLSER	Identifies the volume serial number for the specified macro library. If the library or member is not cataloged, enter the volume serial number.
PROCLIB	Identifies the procedure library attributes for the specified JCL library or member. Each procedure library must be entered on a separate line.
IDMS INFORMATION START and IDMS INFORMATION END	Marks the beginning and end of the IDMS attribute information. IDMS information can be specified in a COBOL, CICS, or DB2 library definition, or in the MEMBER OVERRIDE information for COBOL, CICS, or DB2 members. For each IDMS attribute, the additional information listed in this table can be specified. If an item is not specified, the default specified in the Default section of the import file is used. If the default is not specified in the import file, the site default specified at installation is used.
IDMSDMLC EXECUTION PARM	If you are using IDMS release 10, specifies the IDMSDMLC parameters for the IDMS preprocessor.
IDMS DDNAME/DATA SET NAME	Specifies the IDMS DD name and the dataset name used by the library or member.

Figure 43 shows the code used to define IDMS information for MY.FIRST.COBOL.LIB.

Figure 43 • IDMS Attribute Definition Example

```
<LIBRARY>
  MY.FIRST.COBOL.LIB
   <OBJECT KIND>
      COBOL
   <COBOL VERSION>
      COB68
   <IDMS INFORMATION START>
     <IDMSDMLC EXECUTION PARM>
         PARM GOES HERE
     <IDMS DDNAME/DATA SET NAME>
         STEPLIB IDMS.DEVL.STEPLIB
         STEPLIB IDMS.PROD.STEPLIB
         SYSCTL IDMS.PROD.SYSCTL
         DLODDB IDMS.PROD.DLODDB
         DICTDB IDMS.DEVL.DICTDB
         DICTDB IDMS.PROD.DICTDB


   <IDMS INFORMATION END>
```

Keyword	Description
MEMBER LIST START and MEMBER LIST END	After a library and its attributes have been defined, these tags identify the beginning and end of the list of members within the library that are to be included in the definition.

Figure 44 shows the code required to include members in the library, COBOL.SOURCE.LIBRARY.

Figure 44 • Member Definition

```
<LIBRARY>
  COBOL.SOURCE.LIBRARY
  <OBJECT KIND>
    COBOL
  <MEMBER LIST START>
       Membername1
       Membername2
  <MEMBER LIST END>
```

Keyword	Description
MEMBER OVERRIDE INFORMATION START and MEMBER OVERRIDE INFORMATION END	Overrides member attributes. If you want to specify member attributes that are different than those defined at the library level, these keywords identify the beginning and end of the override information. The keyword should be followed by the list of members that use the specified attributes.

Figure 45 shows the code to define the member override information for Membername1 and Membername2.

Figure 45 • Member Override Definition

```
<MEMBER OVERRIDE INFORMATION START>
   <MEMBER LIST START>
       Membername1
       Membername2
   <MEMBER LIST END>

   <FILEORG>
        LIB
   <COBOL VERSION>
       COB68
   <COPYLIB>
        SYS1.COPYLIB2

<MEMBER OVERRIDE INFORMATION END>
```

Keyword	Description
GROUP LIST START and GROUP LIST END	After a CSD file has been defined, these tags identify the beginning and end of the list of group names within the file that are to be included in the definition. Wildcard characters (*) and (?) are accepted. Each group name should be listed on a separate line.
ASM VERSION	Identifies the default Assembler version for Assembler (ASM) libraries and members defined to the application. Only one version can be specified. Valid entries are ASMF, ASMH, and HLASM.
ASSEMBLER EXECUTION PARM	Identifies the default assemble batch execution parameters used for ASM libraries defined to the application.
ASM MACLIB	Identifies the default maclibs for ASM libraries included in the application definition. Valid entries are names of valid maclibs. Each library name must appear on a separate line. These are attributes of the default maclib for ASM members. Each attribute must appear on a separate line after the entry for the valid default maclib.
ASM MACLIB FILEORG	Identifies the source manager that is used for the specified maclib. Only one source manager can be specified. Valid entries are PDS, PAN, or LIB.
ASM MACLIB SUBSYSTEM	Indicates whether the SYSLIB DD statement in the assemble JCL makes use of the SUBSYS parameter. The maclib subsystem is only valid for Librarian and Panvalet libraries. Valid entries are YES and NO.
ASM MACLIB PASSWORD	Identifies the password for the specified macro library. If the library is password protected, enter the password.
ASM MACLIB VOLSER	Identifies the volume serial number for the specified macro library. If the library is not cataloged, enter the volume serial number.

Sample Import File

Figure 46 shows the Application Information section and Default Information section of an import file for Alliance, Recap, and Estimate.

Figure 46 • Import File Example (1 of 2)

```
*********************************
*                               *
*   APPLICATION INFORMATION     *
*                               *
*********************************

<AKR>
 USER.TEST.AKR
<PRODUCT>
 RECAP, ALLIANCE, TEAM
<APPLICATION NAME>
 IMP-FULL
<APPLICATION DESCRIPTION>
 COMPLETE APPLICATION TEST

*******************************
*                             *
*   DEFAULT INFORMATION       *
*                             *
*******************************

<DEFAULT START>
   <FILEORG>
      PDS
   <COBOL VERSION>
      COB2
   <COPY USAGE>
      COB
   <COPYLIB>
      SYS1.COPYLIB
   <IMS MACLIB>
      IMSESA.MACLIB
   <CICS MACLIB>
      CICS330.MACLIB
   <PROCLIB>
      SYS1.PROCLIB
      SYS2.PROCLIB
   <COMPILE EXECUTION PARM>
      LIB,QUOTE
   <CA-LIBRARIAN MEMBER PSWD EXISTS>
      NO
   <CA-LIBRARIAN MEMBER COPYDD>
      NO
<DEFAULT END>
```

Figure 47 shows the Application Definition section of an import file for Recap, Alliance, and Estimate.

Figure 47 • Import File Example (2 of 2)

```
*******************************
*                             *
*    DEFINITION INFORMATION   *
*                             *
*******************************

<LIBRARY>
 MY.FIRST.COBOL.LIB
  <OBJECT KIND>
    COBOL
  <COBOL VERSION>
    COB68
  <COPYLIB>
     MY.FIRST.COPY.LIB
     <COPY FILEORG>
       PDS
     <COPY USAGE>
       COB
  <COPYLIB>
     MY.FIRST.DB2.INCLUDE.LIB
     <COPY USAGE>
       DB2
  <COPYLIB>
     MY.FIRST.DB2.COPY.LIB
     <COPY USAGE>
       BOTH
  <IDMS INFORMATION START>
    <IDMSDMLC EXECUTION>
       PARM GOES HERE
    <IDMS DDNAME/DATA SET NAME>
       STEPLIB IDMS.DEVL.STEPLIB
       STEPLIB IDMS.PROD.STEPLIB
       SYSCTL IDMS.PROD.SYSCTL
       DLODDB IDMS.PROD.DLODDB
       DICTDB IDMS.DEVL.DICTDB
       DICTDB IDMS.PROD.DICTDB
  <IDMS INFORMATION END>
    <MEMBER LIST START>
     BATMAN
     ROBIN
     HARDY
    <MEMBER LIST END>
    <MEMBER OVERRIDE INFORMATION START>
    <MEMBER LIST START>
     OZZIE
     HARRIET
    <MEMBER LIST END>
     <COBOL VERSION>
       COB370
     <COPYLIB>
       MY.FIRST.COPY.LIBX
     <MEMBER OVERRIDE INFORMATION END>
```

Import Language Keywords

This section lists the keywords for each section of the import file.

Application Information Section

AKR
APPLICATION NAME
APPLICATION DESCRIPTION
PRODUCT

Default Section

DEFAULT START
DEFAULT END
FILEORG
COBOL VERSION
COMPILE EXECUTION PARM
COBOL SOURCE EXTRACTION EXECUTION PARM
JCL SOURCE EXTRACTION EXECUTION PARM
CICS SOURCE EXTRACTION EXECUTION PARM
IMS SOURCE EXTRACTION EXECUTION PARM
STDSQL(86)
COPYLIB COPY FILEORG COPY SUBSYSTEM COPY PASSWORD COPY VOLSER
COPY USAGE

IMS MACLIB IMS MACLIB FILEORG IMS MACLIB SUBSYSTEM IMS MACLIB PASSWORD IMS MACLIB VOLSER
CICS MACLIB CICS MACLIB FILEORG CICS MACLIB SUBSYSTEM CICS MACLIB PASSWORD CICS MACLIB VOLSER
PROCLIB
CA-LIBRARIAN MEMBER COPYDD
CA-LIBRARIAN MEMBER PSWD EXISTS
ASM VERSION
ASSEMBLER EXECUTION PARM
ASM MACLIB ASM MACLIB FILEORG ASM MACLIB SUBSYSTEM ASM MACLIB PASSWORD ASM MACLIB VOLSER

Library and Member Definition Section

LIBRARY
- OBJECT KIND
- ALTERNATE LANGUAGE
  - ALTERNATE LANGUAGE NAME

- PLI
  - PLI VERSION
  - PLI COMPILE EXECUTION PARM
  - INCLUDE
  - INCLUDE FILEORG
  - INCLUDE USAGE
  - PREPROCESSOR OPTION

- ASM
  - PREPROCESSOR OPTION
  - CICS INFORMATION START
  - CICS DDNAME/DATA SET NAME
  - CICS LOAD MODULE NAME
  - CICS INFORMATION END
  - DB2 INFORMATION START
  - DB2 LOAD MODULE NAME
  - DB2 DDNAME/DATA SET NAME
  - DB2 INFORMATION END

- FILEORG

- COBOL VERSION

- COMPILE EXECUTION PARM

- COBOL SOURCE EXTRACTION EXECUTION PARM

- JCL SOURCE EXTRACTION EXECUTION PARM

- CICS SOURCE EXTRACTION EXECUTION PARM

- IMS SOURCE EXTRACTION EXECUTION PARM

- STDSQL(86)

COPYLIB COPY FILEORG COPY USAGE COPY SUBSYSTEM COPY VOLSER COPY PASSWORD
CA-LIBRARIAN MEMBER COPYDD
CA-LIBRARIAN MEMBER PSWD EXISTS
PASSWORD
VOLSER
IDMS INFORMATION START IDMSDMLC EXECUTION PARM IDMS DDNAME/DATA SET NAME
IDMS INFORMATION END
MEMBER LIST START (must appear after library attributes)
MEMBER LIST END MEMBER OVERRIDE INFORMATION START MEMBER OVERRIDE INFORMATION END
IMS MACLIB IMS MACLIB FILEORG IMS MACLIB SUBSYSTEM IMS MACLIB PASSWORD IMS MACLIB VOLSER
CICS MACLIB CICS MACLIB FILEORG CICS MACLIB SUBSYSTEM CICS MACLIB PASSWORD CICS MACLIB VOLSER
PROCLIB

CA-LIBRARIAN MEMBER PASSWORD IDMS INFORMATION START IDMS DDNAME/DATA SET NAME IDMSDMLC EXECUTION PARM IDMS INFORMATION END
GROUP LIST START
GROUP LIST END
ASM VERSION
ASM EXECUTION PARM
ASM MACLIB ASM MACLIB FILEORG ASM MACLIB SUBSYSTEM ASM MACLIB PASSWORD ASM MACLIB VOLSER

Appendix D
Other Language Support

This appendix discusses application definition and analysis support of other languages.

Assembler Language Support

All levels of the Assembler (ASM) language are supported, including EXEC CICS, EXEC DL/I, ASMTDLI, SQL, and IDMS. This enhanced language support enables this functionality for ASM:

- Homonym calculation and results
- Included as impact analysis starting points and results
- Included as search starting points and results
- Included in the query facility results and filters

ASM Support Limitations

There are certain limitations to the ASM support related to macro support, use of constants, synonyms, and source contents. This section addresses these limitations.

Macro Support

Macros are treated the same as copys, which causes line references in the Macro to be potentially incorrect. Macro support is limited to these system provided macros, which are analyzed and their impact on control and data flow is reported.

Macro	Definition
DCB	Creates an FD in the program (PROGRAM HAS FD).
CALL, LINK, LINKX, XCTL, XCTLX	Creates an external CALL. (Program CALLs Entry).

Macro	Definition
GET, PUT, PUTX, READ, WRITE	Creates CRUD information for the FD (DCB) and the data area referenced by the DCB. Options specified on the OPEN macro are also used to determine the CRUD (Create, Read, Update, Delete).
OPEN, CLOSE	Provides information regarding the activities of IO in the program that impact.

Using Constants

The Analyzer traces and keeps track of the contents of variables or registers, and uses this information to improve program understanding. For example, if a CALL is coded with a variable and the value is set prior to the CALL site, the analyzer traces the activity on the variable to find the name of the CALLed program.

Synonyms

The Assembler language option does not create synonyms for Assembler programs. Therefore, in the Impact facility or Query facility, the Search-Synonym feature is not operational for Assembler data items.

The impact analysis accepts Assembler entities as targets, however, because there are no synonyms, dataflow through the program is not traced. Impact analysis results are limited to answers that can be determined from the relationships that are populated.

For example, if a file is read, the record is moved to another record and the second record is written to a second file. Without synonyms, the impact analysis on a field in the second record does not show that a corresponding field in the first record is impacted, nor does it show that there is potential impact to both files. The impact analysis would report that the record, program, step, and job are impacted.

Source Contents

Source contents follow these conditions:

- The program to be assembled must contain executable code, or it fails to analyze.
- The program to be assembled must contain a valid CSECT, or it fails to analyze.
- Programs that use conditional assembly can produce incorrect line references.
- The PRINT NOGEN option is supported only for high-level Assembler.

Defining Assembler Components

Assembler components are defined and analyzed in the same manner as COBOL components. Assembler components would include any Assembler libraries and members, and attributes such as analysis parameters, macro libraries, and preprocessor information. Assembler libraries, members, and attributes are defined using the Edit pull-down and subsequent definitions pop-ups.

To define Assembler components

Note:

This section provides only the basic procedure for defining Assembler components.

1. Select File ▸ ASM Definition from the primary screen and press Enter to display the ASM Definition pop-up, as shown in Figure 48.

Figure 48 • Edit Pull-down - ASM Definition

```
  1. Application description...
  2. COBOL definition...
  3. Load module definition...
  4. JCL definition...
  5. CICS definition...
  6. IMS definition...
  7. PL/I definition...
_ 8. ASM definition...
  9. Natural definition...
 10. Alternate language definition...
 11. Discovered components...
```

2. Select ASM definition and press Enter to display the Edit - ASM Definition pop-up, as shown in Figure 49.

Figure 49 • Edit - ASM Definition Pop-up

```
     Edit - ASM Definition

1    1.  Libraries...
     2.  Members...
     3.  Exit
```

3. Define libraries and members.

4 If no library has been previously defined, use the Add ASM Library pop-up to add library information, as shown in Figure 50.

Figure 50 • Add ASM Library Pop-up

```
                         Add ASM Library
Command ===> ____________________________________________________

Specify library information.  Then press PF key for action.

Data set name   _____________________________________________

Source manager  1   1.  PDS
                    2.  Panvalet
                    3.  Librarian
                    4.  Sequential
                    5.  Endevor
                    6.  User

Volser  . . . . ______     (if volume not cataloged)
Password  . . .            (if password protected)

               PF4=Members  PF6=Add  PF11=Browse
```

5 If a library has been previously defined, review the information in the Application - ASM Libraries pop-up, as shown in Figure 51.

Figure 51 • Application - ASM Library Pop-up

```
                      Application - ASM Libraries              ABBEY300
Command ===> ___________________________________________ Scroll ===> CSR

Select desired data sets.  Then press PF key for action.           1 of 1

  Data set name                                MGR  Def mbrs Type
_ ------------------------------------------  ---- -------- -------------
_ YIAAL10.ABBEY.COBOL                          PDS         0 ASMH
****************************** BOTTOM OF DATA ****************************



        PF4=Options  PF5=Modify  PF6=Add  PF10=Remove  PF11=Members
```

6 Use the PF key options to perform these functions:

- Define library options, including analysis parameters, macro libraries, and preprocessor information.
- Modify, add, or remove a library.
- Add a member.

7 If no members have been previously defined, open the Add Member pop-up to provide member information, as shown in Figure 52.

Note:

You can also define members by selecting Members on the Edit - ASM Definition pop-up.

Figure 52 • Add (ASM) Member Pop-up

```
                          Add Member
Command ===> ____________________________________________

Specify member name.  Then press PF key for action.
To get directory list, specify wild card characters or
leave it blank, then press Directory key.

Library . : 'YIAAL10.ABBEY.COBOL'
Name  . . . _________

            PF5=Directory  PF6=Add  PF11=Browse
```

8 If a member has been previously defined, review the information in the Application - ASM Members pop-up, as shown in Figure 53.

Figure 53 • Application - ASM Member Pop-up

```
            Application - ASM Library Member List
Command ===> ______________________________ Scroll ===> CSR

Select desired members.  Then press PF key            1 of 17
for action.  To add a member press add key.

Library: 'YIAAL10.ABBEY.COBOL'

                                  ------ Last Analysis ------
  Name        Type               Date         Time        RC
- ----------  ----------------  -----------  --------  ----
* CD05APGM    COBOLII R3        *** ANALYZE  FAIL ***    16
* CD05BPGM    COBOLII R3        18-APR-1997  14:59:14     0
* CD05CPGM    COB MYS YM        18-APR-1997  14:59:24     0
* CD10APGM    COBOLII R3        18-APR-1997  14:59:38     0
* CD10BPGM    COBOLII R3        18-APR-1997  14:59:49     0
* CD110PGM    COBOLII R3        18-APR-1997  15:00:01     0
* CD15APGM    COBOLII R3        18-APR-1997  15:00:20     0
* CD15BPGM    COBOLII R3        18-APR-1997  15:00:40     0
* CD180PGM    COBOLII R3        18-APR-1997  15:00:54     0

            PF4=Options  PF6=Add  PF10=Remove  PF11=Browse
```

9 Use the PF key options to perform these functions:

- Define member options, including analysis parameters, macro libraries, preprocessor information, or restoration of original attributes.
- Add, remove, or browse members.

PL/I Language Support

PL/I support is consistent with the level of support currently provided for COBOL. The enhanced support enables this functionality for PL/I:

- Local and global synonym calculation and results
- Homonym calculation and results
- Included as impact analysis starting points and results
- Included as search starting points and results
- Included in the query facility results and filters

PL/I Support Limitations

PL/I support is limited by the ability of the static analyzer to handle pointer or address based assignments. When programs use something other than a name based assignment, the static analyzer assignment chain is incomplete. This causes the calculations dependent on that assignment chain to generate incomplete answers. This limitation manifests itself in incomplete answers in the synonym, impact analysis, and constant propagation calculations.

This example shows how this occurs:

```
DCL     P1, P2    POINTER
DCL     X         CHAR(6)
DCL     Y         CHAR(6) BASED(P2)
P1      =         ADDR(DATE)
P2      =         P1
X       =         Y
```

In this case, `Y` should be a synonym of DATE. However, since a pointer assignment was used to establish the relationship between them, the PL/I analyzer loses the connection.

Defining PL/I Components

PL/I components are defined and analyzed in the same manner as COBOL components. PL/I components include any PL/I libraries and members, and attributes such as analysis parameters, macro libraries, and preprocessor information.

Analyzing NATURAL Applications for Alliance and Estimate

Unlike programs written in traditional languages, NATURAL programs and data definitions are kept in NATURAL databases. This section describes the manual process for extracting a NATURAL application from a NATURAL database and analyzing the application under Center.

Overview

The process of analyzing a NATURAL application can be divided into these steps:

- "Step 1 - Extracting the Application from the Database" on page 183.
- "Step 2 - Transferring The Extracted File To MVS (Optional)" on page 187.
- "Step 3 - Splitting Up the Extracted File Into Individual Components" on page 187.
- "Step 4 - Defining the Application Under Alliance or Estimate" on page 188.
- "Step 5 - Submitting the Analysis Job" on page 189.

These steps are described in more detail in the next sections.

Step 1 - Extracting the Application from the Database

NATURAL applications are kept in NATURAL databases, which may reside on non-MVS platforms. Center does not have the capability to read program source code and data definitions directly from a NATURAL database. To prepare a NATURAL application for analysis, you must first extract it from its database, as described in this section.

NATURAL Version 2.1 or Later

NATURAL V2.1 or later comes with a utility to unload and load a NATURAL database. This utility is called SYSTRANS and can be run either online or from JCL for the MVS platform.

Figure 54 shows the sample JCL used to run SYSTRANS under NATURAL V2.2:

Figure 54 • Sample JCL to Run SYSTRANS Under NATURAL V2.2

```
//NAT2BAT  EXEC PGM=NATBAT22,REGION=4096K, <--------------> See Note 1
//         PARM=('DBID=160,FNAT=(001,051),FUSER=(,029)',<-> See Note 2
//         'FDIC=(150,162),FSEC=(150,164)',
//         'MADIO=0,MAXCL=0,IM=D,OBJIN=N')
//STEPLIB  DD DISP=SHR,DSN=site.NATURAL.LOAD <------------> See Note 3
//         DD DISP=SHR,DSN=site.ADABAS.LOAD
//CMPRINT  DD SYSOUT=*
//CMWKF01  DD DSN=ASG.SYSTRANS.OUTPUT, <--------------> See Note 4
//         DISP=(NEW,CATLG,DELETE),
//         DCB=(LRECL=96,BLKSIZE=6240,RECFM=FB),
//         UNIT=SYSDA,SPACE=(6240,(100,200),RLSE)
//CMWKF03  DD DSN=&&WORK3,
//         DISP=(NEW,DELETE,DELETE),
//         DCB=(LRECL=43,BLKSIZE=6450,RECFM=FB),
//         UNIT=SYSDA,SPACE=(6450,(100,200),RLSE)
//CMSYNIN  DD *
GLOBALS PS=62 <-----------------------------------------> See Note 5
LOGON SYSTRANS
MENU
U <-----------------------------------------------------> See Note 6
N,N,N,Y,Y,N,N <------------------------------------------> See Note 7
N <-----------------------------------------------------> See Note 8
pgmlib,*,*,CMWKF01 <-------------------------------------> See Note 9
M <-----------------------------------------------------> See Note 10
maplib,*,Y,N,CMWKF01 <-----------------------------------> See Note 11
V <-----------------------------------------------------> See Note 12
*,,CMWKF01 <---------------------------------------------> See Note 13
FD <----------------------------------------------------> See Note 14
7,125,0,0 <----------------------------------------------> See Note 15
FIN
/*
//*
```

Notes

Note Number	Description
1	The PGM name for executing NATURAL in batch mode is different depending on the version of NATURAL you are using. Check with the NATURAL DBA to see what the PGM name should be and if there is a standard JCL PROC you are supposed to use to run NATURAL from a JCL.
2	These are only sample parameters. Check with the NATURAL DBA to see what the settings should be for the NATURAL database you are working with. The most important parameter is IM=D.
3	The load libraries needed to run NATURAL in batch depend on the site. Check with the DBA for the correct load libraries.
4	This is the output from SYSTRANS. If you are running this JCL multiple times, specify a different DSN each time. You may need to adjust the parameters for space allocation if the application contains many programs and data definitions.
5	This parameter was needed at a NATURAL site for the rest of the commands to work.
6	U specifies the unload command.
7	These set the options for the unload command. The settings are: • N - Do not perform EBCDIC to ASCII conversion. • N - Do not use user-defined conversion table. • N - Do not substitute line references. • Y - Produce a report listing the objects that were unloaded. • Y - Include line numbers in the output file. • N - Do not take input for the unload function from a work file. • N - Do not display a list for selecting the objects to be unloaded.
8	N selects NATURAL objects.

Note Number	Description
9	These are the parameters for selecting NATURAL objects: • From Library - Specify the name of the library where the objects are located. • Object Name - Use * to select all objects. • Object Type - Use * to select all types of objects. • To Library - Use CMWKF01.
10	M selects NATURAL maps.
11	These are the parameters for selecting NATURAL maps: • From Library - Specify the name of the library where the maps are located. • Map Name - Use * to select all maps. • PREDICT Rules - Use Y to include PREDICT rules in the map source. • Separate Free Rules - Use N to keep the free rules with the map source. • To Library - Use CMWKF01.
12	V selects NATURAL DDMs.
13	These are the parameters for selecting NATURAL DDMs: • DDM Name - Use * to select all DDMs. • From Library - Specify the name of the library where the DDMs are located; not used on MVS. • To Library - Use CMWKF01.
14	FD selects ADABAS FDTs.
15	These are the parameters for selecting ADABAS FDTs: • Source DBID - Specify the database ID of the ADABAS FDT. • Source FNR - Specify the file number of the ADABAS FDT. • Target DBID - Use 0. • Target FNR - Use 0.

Caution! This sample JCL and its associated notes are based on ESW's current understanding of SYSTRANS. Because the syntax for the commands are slightly different from one release of NATURAL to another, you may have to experiment with the JCL and the options to get SYSTRANS to work correctly. You may find more up-to-date information about running SYSTRANS from the database administrator (DBA) at your site.

NATURAL Version 2.1 and Earlier

If your NATURAL version is older than 2.1, contact the ASG Service Desk for a set of NATURAL programs that can be used to unload your NATURAL programs and data definitions.

Step 2 - Transferring The Extracted File To MVS (Optional)

If the platform for your NATURAL applications is not MVS, you have to transfer the extracted file to MVS for analysis, since that is where your ESW products are installed. You may do this either through a tape or a file transfer program. These are things to watch for:

- The dataset on MVS must have a record length (LRECL) no less than 96. The dataset can be either variable blocked (VB) or fixed blocked (FB).
- If you are transferring the file from a platform that uses ASCII as its character set, you must specify ASCII-to-EBCDIC translation in your file transfer. If so, make sure the translation for the carat character (^) in ASCII is the not symbol (¬) in EBCDIC.

Step 3 - Splitting Up the Extracted File Into Individual Components

The output from SYSTRANS is a single file containing all the programs and data definitions from the NATURAL database. Before defining the application to Alliance or Estimate for analysis, you must split up the SYSTRANS output file into individual programs and data definitions. Center R4.4 and above provide a utility program called the NATURAL Splitter for this purpose.

To use NATURAL Splitter

1 Allocate a partitioned dataset (PDS) with these characteristics:

```
RECFM=FB,LRECL=96

BLKSIZE=6240
```

Use the size of the SYSTRANS output file to estimate the space and directory blocks needed for the PDS.

2 Copy the VIASNASP JCL from the CNTL library of your ESW product installation to your own dataset. Edit the JCL. Change the SYSIN DD to point to the SYSTRANS output file, and the VIASPGMS DD to the PDS you allocated in step 1 on page 187.

3 Save and submit the JCL.

The NATURAL Splitter utility takes as its input an output file from SYSTRANS and splits it into individual members in a PDS. Each member may be a program, a map, a subprogram, a subroutine, help routine, or a data definition (i.e.such as DDM, FDT). Since PDS member names are limited to 8 characters in length and NATURAL objects may have names up to 32 characters in length, the NATURAL Splitter has to shorten the name of any object whose name is longer than 8. A list of the objects in the SYSTRANS file and their corresponding member names in the PDS is written to a separate member under the name of $$VIAINV, which is also known as the inventory file. This file must be present in the PDS for the analysis to work.

Step 4 - Defining the Application Under Alliance or Estimate

Once the SYSTRANS file has been split up and stored in a PDS, you may start defining your NATURAL application to Alliance or Estimate.

To define your NATURAL application

1 Start the product.

2 For Alliance, select File ▸ New Application.

Or

For Estimate, select Edit ▸ Add Application Node.

3 For Alliance, enter a dataset name for the AKR and a new application name and press Enter.

Or

For Estimate, select Edit ▸ Application Definition.

4 Select Edit ▸ Natural Definition.

5 Type a / next to either NAT1.2, NAT2.2S, or NAT2.2R, depending on whether the programs you want to include in the application are NATURAL V1.2, V2.2S, or V2.2R. Use NAT2.2S or NAT2.2R for programs that are created under NATURAL V2.1 or NATURAL V2.2.

6 Select Libraries on the next screen.

7. Enter a dataset name on the next screen. Make sure the source manager is set to PDS.

8. Press PF6 to add the library.

9. Press PF4 to add members.

10. Press PF5 on the Add Member screen to display the directory.

11. Type a / next to any member that represents a program, a map, a subprogram, a subroutine, or a help routine. Scroll down if necessary to add all such members. You may want to keep a hardcopy of the inventory file handy to help you decide what each member represents, since the file shows not just the original name of every member but also what kind of object is in the member.

12. Press PF3 to add the member(s) selected in step 11.

13. Press PF3 again to exit the Add Member screen.

14. The Application - Alternate Library Member List screen displays, showing all the members that have been added to the application. Press PF3 to exit this screen.

15. Press PF3 again to exit the Add Alternate Library screen.

16. The Application - Alternate Libraries screen displays, showing the library that has just been added and the number of members selected from the library. Press PF3 to exit this screen.

17. Select Exit to exit the Edit - Alternate Language Definition screen.

18. If the application contains a mixture of different versions of NATURAL, select another version by repeating step 5 on page 188 through step 17.

The application is now defined to Alliance or Estimate.

Step 5 - Submitting the Analysis Job

After your NATURAL programs have been defined in an application definition, analyzing the application is identical to analyzing a non-NATURAL application. For more details, see "Step 5 - Submitting the Analysis Job" on page 189.

Index

A

AAE 1
AKR
- defining application components 5
- relationship to applications 29
- Utility Directory Report 137
- Utility Log example 138

Alliance
- accessing from ESW screen xiv
- description xi
- linking xiv

analysis
- full 8
- incremental 8

analysis restart warning 86
analysis status report
- AKR detail 141
- AKR summary 140
- application detail 142

analysis type, incremental analysis 83
analyze
- a COBOL program 79
- analyze process 79
- features 84
- Recap/Alliance 84
- recovery/restart 86
- requirements 79
- resource estimates 81
- restarting the analyze 86
- semantic linking 84
- source program 80
- submitting 84
- summary report 91
- timings and estimated resource requirements 81
- using a batch job 87

analyzing an application
- overview 8
- using a batch job to analyze 79
- using the batch application analysis selection 79

application
- analyze job 79
- batch analyze 87
- creating 29
- opening 82

Application Analytical Engine 1
application definition
- adding members 76
- analyzing 79
- changing member attributes 77
- COBOL program attributes 40
- COBOL version 33, 40
- CSD files 55
- CSD group names 56
- defined 6
- defining COBOL libraries 31
- defining COBOL programs 39
- defining IMS DBDGEN, PSBGEN, DFSMDA, MFS map and stage 1 library 64
- defining IMS members 67
- defining JCL libraries 48
- defining JCL members 51
- defining libraries for BMS maps 57
- defining libraries for CICS FCT, PCT, and PPT tables 57
- hierarchy 27
- JCL error on extract 41, 44
- providing an application description 30
- removing members 76–77
- reviewing for accuracy 71
- steps for modifying 72
- view example 71
- viewing 71

Application Inventory Worksheets 5
Application Knowledge Repository (AKR) 5
ASM components, defining 179

Assembler language option
defining Assembler components 179
limitations 177
macro support 177
assessing your task overview 4
attributes
COBOL program 40
restoring CICS library attributes 63
restoring COBOL library attributes 45
restoring IMS library attributes 70
restoring JCL library attributes 54
reviewing 71
auto analysis 8
AutoChange
accessing from ESW screen xiv
description xi

B

batch AKR utility
AKR utility log 138
analysis status report 139
command formats 124
command syntax 124
comments 126
control cards 123
CONVERT command 126
JCL 123
print directory example 137
print directory report 137
punch directory 139
punch directory example 139
reports 137
batch analyze
creating the analyze JCL 88
input file control statements 89
selecting members 79
batch, analyze of an application 87
Bridge
accessing from ESW screen xiv
description xi
building and analyzing an application
analyzing the application 79
CICS libraries for CICS FCT, PCT and PPT tables and BMS maps 57
COBOL library attributes 34
creating an application 29
CSD group names 56
IMS libraries 64
IMS library attributes 65
IMS members 67
specifying analysis parameters 35
specifying CICS FCT, PCT and PPT tables and BMS map members 60
specifying CICS library attributes 58
specifying CICS member attributes 61
specifying COBOL program attributes 40
specifying COBOL program IDMS information 44
specifying COBOL programs 39
specifying copy libraries 36
specifying IDMS information 38
specifying JCL libraries 48
specifying JCL library attributes 49
specifying JCL member attributes 52
specifying JCL members 51
specifying load libraries 46
specifying load module members 47
specifying maclibs 59
specifying procedure libraries 49

C

Center, description xi
changing the concatenation sequence of copy libraries 37, 42
CICS libraries
defining FCT, PCT, and PPT tables and BMS maps 57
specifying maclib attributes 58
CICS library attributes
removing attribute specified for member 63
restoring 63
CICS members
defining BMS maps 60
defining FCT, PCT, and PPT tables 60
COBOL libraries
defining 31
specifying analysis parameters 35
specifying attributes 34
specifying copy libraries 36
specifying IDMS information 38
COBOL library attributes
removing attribute specified for program 45
restoring 45
COBOL programs
defining 39
specifying analysis parameters 40
specifying attributes 40
specifying copy libraries 42
specifying IDMS information 44

COBOL version 33, 40
compile parameters
how to extract from JCL (for programs) 41
how to extract from JCL (for source library) 35
components
dead 96
discovered 96
discovered, missing , dead, overview 10
missing 96
overview 96
conventions page xvii
copy libraries
defining for COBOL libraries 36
extracting from JCL (for programs) 41
extracting from JCL (for source libraries) 35
specifying for COBOL programs 42
creating an application 29
CSD files, defining 55
CSD group names, defining 56

D

dead components
action 97
defined 96–97
defining an application 29
defining application components
BMS maps 57
CICS CSD files 55
CICS FCT, PCT and PPT tables 57
CICS FCT, PCT, and PPT tables and BMS map members 60
CICS library attributes 58
CICS member attributes 61
CSD group names 56
IMS libraries 64
IMS library attributes 65
IMS members 67
maclibs 59
overview 6
specifying COBOL program attributes 40
specifying COBOL programs 39
specifying copy libraries 36
specifying IDMS information 38
specifying JCL libraries 48
specifying JCL library attributes 49
specifying JCL members 51
specifying load libraries 46
specifying load module members 47
specifying procedure libraries 49
defining ASM components 179
defining your requirements, overview 5
discovered components
resolvable 99
unresolvable 99

E

Encore
accessing from ESW screen xiv
description xii
Endevor support, location inventory name as dataset name 34
Estimate
accessing from ESW screen xiv
description xii
ESW
description x
invoking products xiii
product integration xiv
extract, JCL error 44

G

gathering information about related application components, overview 5

I

IDMS information, how to extract from JCL (for source libraries) 38
import data item syntax 153
import language reference
application information 154
default section 155
definition section 161
definition section example 171–172
file structure 154
keyword list 172
sample import file 170
syntax 153
IMS libraries
defining 64
specifying maclib attributes 65
IMS library attributes
removing attribute specified for member 70
restoring 70
IMS members
defining DBDGEN, PSBGEN, DFSMDA, MFS maps and stage 1 libraries 67
specifying maclib attributes 68

incremental analysis
defined 8
described 83
incremental analyze 83
input file control statements 89
Insight
accessing from ESW screen xiv
description xii
using analysis functions xiv

J

JCL error, in extract 41
JCL libraries
defining 48
specifying procedure libraries for attributes 49
specifying procedure library attributes 49
JCL library attributes
removing attribute specified for member 54
restoring 54
JCL members
defining 51
defining procedure library attributes 52

L

Librarian, specifying source manager execution parameters 58
libraries, modifying dataset attributes 74
load modules
defining libraries 46
defining members 47

M

missing components, action 102
modifying application definition 72
modifying library information 74

N

NATURAL
defining the application 188
Submitting the analysis job 189

O

opening an application 82

P

Panvalet libraries, specifying source manager execution parameters 58
procedure libraries, specifying for JCL libraries 49
product integration xiv

R

Recap
accessing from ESW screen xiv
description xii
refining the application definition, overview 9
removing CICS member level attributes, procedure 63
removing COBOL program level attributes, procedure 45
removing IMS member level attributes, procedure 70
removing JCL member level attributes, procedure 54
report, analysis status 139
reports
VIAAPRT 8
VIALOG 9
resolvable discovered components 99
resolving analysis errors, overview 9
restoring
CICS library attributes 63
COBOL library attributes 45
IMS library attributes 70
JCL library attributes 54
reviewing application definition, through online facility 71
reviewing the application definition 71

S

semantic linking
defined 8
definition 84
SmartDoc
accessing from ESW screen xiv
description xii
SmartEdit
accessing from ESW screen xiv
description xiii
SmartTest
accessing from ESW screen xiv
description xiii
source manager execution parameters, specifying 58
submitting the analyze 84

U

unresolvable discovered components 99

V
VIAARPT, defined 8
VIALOG, defined 9