

# ASG-Center™ Installation Guide

Version 6.0

Publication Number: CNX0300-60

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.

©1987-2002 Allen Systems Group, Inc. All rights reserved.

All names and products contained herein are the trademarks or registered trademarks of their respective holders.







# ASG 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***

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

## ***Non-Business Hours - Emergency Support***

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

---

<b>Preface</b> .....	<b>ix</b>
<b>About this Publication</b> .....	<b>ix</b>
<b>Related Publications</b> .....	<b>x</b>
<b>ASG-Existing Systems Workbench (ASG-ESW)</b> .....	<b>xi</b>
<b>Invoking ESW Products</b> .....	<b>xiv</b>
<b>ESW Product Integration</b> .....	<b>xv</b>
<b>Examples</b> .....	<b>xvi</b>
<b>Publication Conventions</b> .....	<b>xviii</b>
<b>1 Overview &amp; Planning</b> .....	<b>1</b>
<b>ASG Service Pack</b> .....	<b>1</b>
<b>Overview</b> .....	<b>1</b>
<b>Planning</b> .....	<b>2</b>
Environment .....	<b>2</b>
Compilers/Languages .....	<b>3</b>
Product Compatibility .....	<b>3</b>
BDAM AKR Considerations .....	<b>3</b>
Prerequisites .....	<b>4</b>
<b>2 ESW Installation and Customization</b> .....	<b>9</b>
<b>Overview</b> .....	<b>9</b>
<b>Center Installation Checklist</b> .....	<b>11</b>
Installing ESW .....	<b>11</b>
Customizing ESW .....	<b>11</b>
Customizing Common Options and Update Your TSO/ISPF Environment .....	<b>12</b>
<b>Installing ESW</b> .....	<b>13</b>
Performing a New Installation .....	<b>13</b>
Installing Into an Existing Center Installation .....	<b>19</b>

Customizing ESW .....	25
Customizing Common Options and Update Your TSO/ISPF Environment.....	35
<b>3 Endeavor Installation and Customization.....</b>	<b>49</b>
Enabling Endeavor/MVS Support.....	50
Endeavor Customization and Operation .....	57
<b>4 Alternate Language Facility .....</b>	<b>59</b>
Step 1 - Providing the ALF Option Passwords for Download .....	60
Step 2 - Modifying ALF Installation Options.....	60
Step 3 - Building the Required Datasets for the ALF .....	61
Analyzing IDEAL Applications.....	61
Step 1 - Preparing the Source for Analysis .....	61
Step 2 - Defining the Application .....	61
Step 3 - Submitting the Analyze Job.....	62
Analyzing NATURAL Applications .....	62
Step 1 - Extracting the Application from the Database.....	62
Step 2 - Transferring the Extracted File to MVS (Optional) .....	65
Step 3 - Splitting the Extracted File into Individual Components .....	66
Step 4 - Defining the Application .....	67
Step 5 - Submitting the Analyze Job.....	67
Analyzing Model 204 Applications .....	67
Step 1 - Extracting the Application from the Database.....	67
Step 2 - Splitting the Extracted File into Individual Components .....	68
Step 3 - Defining the Application .....	69
Step 4 - Submitting the Analyze Job.....	69
Analyzing Easytrieve Applications .....	69
Step 1 - Defining the Application .....	69
Step 2 - Submitting the Analyze Job.....	70
Analyzing Fortran Applications.....	70
Step 1 - Preparing the Source for Analysis .....	70
Step 2 - Defining the Application .....	70
Step 3 - Submitting the Analyze Job.....	70
<b>5 AKR Management.....</b>	<b>71</b>
Maintaining AKRs .....	71
Defining Additional AKRs and Expanding Existing AKRs .....	72
Restrictions .....	73

<b>Batch AKR Utility</b> .....	<b>74</b>
Job Control Statements .....	74
Control Cards .....	76
Command Format .....	76
Command Syntax .....	77
ANLZSTAT Command .....	78
comment Command .....	79
CONVERT Batch AKR Command .....	80
COPY Batch AKR Command .....	81
DELETE Batch AKR Command .....	82
EXPORT Batch AKR Command .....	83
HELP Batch AKR Command .....	84
INIT Batch AKR Command .....	85
MOVE Batch AKR Command .....	86
PRINT Batch AKR Command .....	87
PUNCH Batch AKR Command .....	88
Reports .....	89
AKR Utility Log .....	89
AKR Utility Directory Report .....	90
Punch Directory File .....	91
<b>6 Analyze Submit Facility</b> .....	<b>93</b>
<b>Introduction</b> .....	<b>93</b>
<b>Program-level Analyze Input Descriptions</b> .....	<b>94</b>
<b>The Analyze Process</b> .....	<b>96</b>
Using the ESW - Prepare Program Pop-up .....	97
Using ISPF .....	100
Using ISPF/PDF Edit .....	100
<b>Automatic JCL Modifications for COBOL Procedures</b> .....	<b>106</b>
<b>Analyze Summary Report</b> .....	<b>119</b>
<b>Adding Analyze Facilities to a Standard Compile Mechanism</b> .....	<b>122</b>
CLIST Compile Mechanism .....	122
ISPF Compile Mechanism .....	123
<b>Assembler Analyzer (SmartTest Only)</b> .....	<b>123</b>
<b>Assembler Analyzer Input</b> .....	<b>123</b>
<b>Assembler Analyze JCL</b> .....	<b>125</b>

<b>7 Processing Considerations</b> .....	<b>127</b>
Security Software .....	127
Creating a Site-specific Default Profile .....	129
ISPF/PDF Edit Environment .....	129
Analysis Space Requirements .....	131
Batch Execution .....	133
Reblocking and/or Copying the ESW Load Library .....	133
<b>8 Command Usage Facility</b> .....	<b>135</b>
Introduction .....	135
Command Usage Information .....	136
CUF Log File Record Format .....	136
Enabling the Command Usage Facility .....	138
Step 1 - Modifying the CUF Installation Options .....	138
Step 2 - Compiling the Sample CUF Report Program - VIASCUFRR .....	140
Step 3 - Running the Sequential File Utility (Optional for CUF Log Files) .....	140
Step 4 - Modifying the User Exit - VIASEXT4 (Optional for SMF Records) .....	141
Command Usage Facility Limitations .....	141
<b>9 Problem Diagnosis</b> .....	<b>143</b>
<b>General Problems</b> .....	<b>143</b>
DBCS Terminals Displaying in Lowercase Characters .....	144
Program Analyzer Problems .....	144
Program Query Problems .....	144
<b>Service Desk Commands</b> .....	<b>144</b>
PRODLVL Command .....	144
<b>Reporting Internal Errors</b> .....	<b>145</b>
Log File Contents .....	145
Batch and Online Error Logs .....	146
<b>Printing AKR Data for Debugging</b> .....	<b>148</b>

**10 Application Analyzer**

**Execution Monitor Support..... 149**

**Execution Monitor Support ..... 149**

**Execution Monitor Description File ..... 150**

Monitor monitor-name ..... 150

Options: option-name(option-value),...; ..... 151

syntax-rule...; ..... 151

End monitor-name; ..... 153

Example. .... 154

**Appendix A**

**Installation Options ..... 157**

**File Format ..... 158**

**Syntax Specification ..... 158**

**CNTL Member Parameters ..... 159**

VIA\$PRMB – Alliance ..... 159

VIA\$PRMM – AutoChange ..... 159

VIA\$PRMG – Bridge ..... 159

VIA\$PRMU – AutoChange and Bridge ..... 159

VIA\$PRMS – Center ..... 159

VIA\$PRMA – Encore ..... 167

VIA\$PRMJ – Estimate ..... 167

VIA\$PRMD – SmartDoc ..... 168

VIA\$PRME – SmartEdit ..... 169

VIA\$PRMP – SmartTest ..... 170

VIA\$PRMX – Recap ..... 170

**Changing an Installation Option Default Value..... 171**

**Installation Options..... 172**

**Appendix B**

**CNTL, Skeleton, and CLIST Members ..... 223**

**Center CNTL Members ..... 223**

**Center Skeleton (SLIB) Members ..... 229**

**Center CLIST Members..... 230**

<b>Appendix C</b>	
<b>VIASCUST Customization CLIST</b> .....	<b>235</b>
<b>VIASCPRM Parameters</b> .....	<b>236</b>
<b>CLIST Invocation</b> .....	<b>239</b>
<b>Appendix D</b>	
<b>Maintenance</b> .....	<b>241</b>
<b>Problems or Questions</b> .....	<b>241</b>
<b>Applying Interim Modifications</b> .....	<b>241</b>
<b>PTF Display Facility</b> .....	<b>242</b>
<b>Appendix E</b>	
<b>Abend Codes</b> .....	<b>245</b>
<b>Appendix F</b>	
<b>Listing Allocated Datasets</b> .....	<b>251</b>
<b>Primary Commands</b> .....	<b>251</b>
<b>Line Commands</b> .....	<b>252</b>
<b>Appendix G</b>	
<b>User Exit Changes</b> .....	<b>253</b>
<b>Appendix H</b>	
<b>DBCS Addendum</b> .....	<b>255</b>
<b>Introduction</b> .....	<b>255</b>
<b>Downloading the Product Tape</b> .....	<b>255</b>
Copying the Message File .....	256
Modifying VIA\$PRMS .....	256
Adding ESW Products to the TSO/ISPF Environment .....	257
<b>Logon Procedure Samples</b> .....	<b>260</b>
Sample 1: Implementing DBCS (Non-Japanese) .....	260
Sample 2: Implementing DBCS (Japanese) .....	261
<b>Message File Allocations</b> .....	<b>262</b>

**Detecting and Handling DBCS Terminals** ..... 263  
VTAM Queriable Terminals ..... 263  
ISPF Terminal Type of 3278KN ..... 263  
Language=JAPANESE Parameter in VIA\$PRMS ..... 263

**Appendix I**

**Using Process Support** ..... 265

**Overview** ..... 265

**Listing AKRs** ..... 266

Viewing an AKR ..... 269

Analyzing an Enterprise Member ..... 274

Analyzing an Application Member ..... 276

Analyzing a Program Member ..... 277

Changing an AutoChange or Bridge Member ..... 278

**Creating Application Definitions** ..... 278

**Index** ..... 281



---

## Preface

---

This *ASG-Center Installation Guide* provides a guide for the installation, customization or migration of ASG-Existing Systems Workbench (ASG-ESW) products. Before installing any ASG-ESW (herein called ESW) product, it is necessary to first install the shared libraries provided by ASG-Center (herein called Center).

Center is the common platform for ESW and provides the common Analytical Engine used to analyze and store application information in the Application Knowledge Repository (AKR). Center provides a homogeneous environment in which all ESW products work synergistically.

Allen Systems Group, Inc. (ASG) provides professional support to resolve any questions or concerns regarding the installation or use of any ESW 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 ESW product.

## About this Publication

This publication consists of these chapters:

- [Chapter 1, "Overview & Planning,"](#) provides an overview of Center and describes the planning process involved with installing the Center product.
- [Chapter 2, "ESW Installation and Customization,"](#) describes the Center installation and customization process (including passwords), JCL downloads, and migration procedures.
- [Chapter 3, "Endevor Installation and Customization,"](#) describes the Endevor/MVS tape management system installation and customization process.
- [Chapter 4, "Alternate Language Facility,"](#) describes ASG-ESW's alternate language support, which allows customers to analyze several different alternate languages for the ASG-Estimate and ASG-Alliance products.
- [Chapter 5, "AKR Management,"](#) describes the AKR file organization, online and batch processing, and AKR statistical generation.

- [Chapter 6, "Analyze Submit Facility,"](#) describes the methods used to analyze programs for ASG-Insight, ASG-SmartTest, ASG-SmartDoc, ASG-AutoChange, and ASG-Encore.
- [Chapter 7, "Processing Considerations,"](#) describes the interface to CA-Panvalet's IPNEXIT security module for additional security. IPNEXIT allows security at the CA-Panvalet dataset level to include member level security.
- [Chapter 8, "Command Usage Facility,"](#) describes the ESW Command Usage Facility (CUF), which provides a mechanism for analyzing command usage in ESW products. When the CUF is active, it records each instance of primary command execution within an ESW product in a CUF log. This information can be used to create reports for managing ESW products.
- [Chapter 9, "Problem Diagnosis,"](#) describes how to diagnose general and specific problems associated with ESW products.
- [Chapter 10, "Application Analyzer Execution Monitor Support,"](#) describes the Analytical Engine's JCL Analyzer, which recognizes values supplied to PGM parameters in JCL EXEC statements and executable programs for the ASG-Alliance, ASG-Estimate, and ASG-Recap application-level products.

## **Related Publications**

The documentation library for ASG-ESW consists of all publications for ASG-ESW products because ASG-Center is the common platform shared by all other products. For additional installation information, see the installation guide for the specific product(s).



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 xv](#) 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 <b>Or</b> 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.



**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(UIAPCOBC)'

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).
<i>lowercase italic monospace</i>	Information that you provide according to your particular situation. For example, you would replace <i>filename</i> 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  <u>N</u> ).

---

# 1

## Overview & Planning

---

This chapter provides an overview of Center and contains these sections:

Topic	Page
<a href="#">ASG Service Pack</a>	<a href="#">1</a>
<a href="#">Overview</a>	<a href="#">1</a>
<a href="#">Planning</a>	<a href="#">2</a>

### ASG Service Pack

Verify whether you received an ASG Service Pack for this product. If so, read the instructions for installing the Service Pack before proceeding with the product installation. The installation instructions are located in a text file on the distribution media included with the Service Pack. If you have any problems with the Service Pack, contact the ASG Service Desk.

### Overview

ESW products facilitate significantly improved productivity in all aspects of the maintenance programmer's job. Before installing any ESW product, it is necessary to first install the shared libraries provided by Center because Center contains the code that is common to all ESW products.

Center provides the common Analytical Engine used to analyze and store application information in the AKR, as well as a homogeneous environment in which all ESW products work synergistically.

For detailed information about installing a specific ESW product, review the installation guide for that product. If you are interested in acquiring additional ESW software, contact your ASG representative.

**Note:** \_\_\_\_\_

In this installation guide, the term application-level products refers to Alliance, Recap, and Estimate.

---

## **Planning**

### **Environment**

These are the environment requirements for ESW products:

- MVS/ESA or OS/390
- MVS ISPF versions 3.5 through 4.8
- BATCH region size of 4096 KB or larger
- TSO logon region size of 2048 KB or larger below 16 MB
- Direct access storage
- 3270 type terminals; Models 2, 3, 4, or 5
- VSAM (required if any AKR is allocated as VSAM)

For information about needed storage space, see ["Step 2 - Confirming Availability of Direct Access Storage Space" on page 14.](#)

## Compilers/Languages

ESW products support these compilers and programming languages:

- COBOL II
- COBOL/370
- COBOL for MVS and VM
- COBOL for OS/390 and VM
- OS PL/I versions 1.5 and 2.3
- PL/I MVS & VM
- High-level Assembler

**Note:** \_\_\_\_\_

Object-oriented extensions added to COBOL for MVS and VM and COBOL for OS/390 and VM are not supported.

---

## Product Compatibility

Earlier releases of ESW products are not compatible with this release of Center. If you have ESW products currently installed, install this software into separately managed libraries.

## BDAM AKR Considerations

Center provides support for VSAM and BDAM AKRs. The AKR-DSORG-VSAM parameter in the VIA\$PRMS CNTL member determines the file organization of all AKRs allocated at your site. The AKR-DSORG-VSAM parameter is set during installation of the Center shared libraries.

**Caution!** Hardware and software compression routines and programs that release free space should not be run against BDAM AKRs. AKR data is compressed using its own internal routines. Releasing free space or running compression routines against AKRs corrupts the AKR data. If free space has been released from an AKR, reallocating the AKR to a size equal to or larger than its original size, before any additional processing, should salvage the data.

BDAM AKRs provide support for secondary extents so that smaller AKRs can be allocated and later expanded as required. In addition, BDAM AKRs do not require formatting as required for VSAM RRDS AKRs. Security, backup, restore, and migration facilities may be more flexible for BDAM AKRs than for VSAM RRDS AKRs. If the AKR support for your site is set to BDAM, for example AKR-DSORG-VSAM=NO, existing VSAM AKRs continue to work correctly with ESW products. When these AKRs are expanded, they are converted from VSAM RRDS to BDAM file organization automatically unless you have saved the expansion JCL for the VSAM AKR and submit that JCL directly.

If you want to convert existing AKRs from VSAM RRDS to BDAM file organization, use the AKR utility COPY command.

## **Prerequisites**

### **Dataset Reservesdataset**

ESW products use ENQ and RESERVE macros to serialize access to the AKR and the SmartTest Batch queue file. To make this serialization effective across multiple systems in a shared DASD environment, it may be necessary to adjust the configuration of your global resource manager.

Each ENQ or RESERVE macro specifies a QNAME and an RNAME, that together identify the resource to be serialized. If your site has multiple systems that share DASD, then it may also be necessary to configure your global resource manager to recognize the QNAMEs used by ENQ and RESERVE macros. The information below is provided to help you with this configuration. For specific details, consult your global resource manager documentation.

A common QNAME (VIASOFT) is used for the serialization of AKRs created by Center and the SmartTest Batch queue file. The global resource manager needs to be configured to recognize the QNAME VIASOFT.

### **IBM Global Resource Serialization Configuration (GRS)**

If your global resource manager is IBM Global Resource Serialization Configuration (GRS), it may not be necessary to make GRS configuration changes. By default, GRS considers an ENQ with SCOPE=SYSTEMS to be identifying a global resource. You can use either of these configuration techniques:

- Leave Reserve-MACRO=YES in VIA\$PRMS, and add a generic entry to the RESERVE Conversion RNL for QNAME VIASOFT. For example:

```
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(VIASOFT)
```

**Or**

- Set Reserve-MACRO=NO in VIA\$PRMS. No GRS configuration changes should be necessary.

These are the QNAMEs, RNAMEs, and the usage for each of the VIASOFT ENQ and RESERVE macros:

QNAMEs	RNAMEs	Usage
VIASOFT	2-byte type code (A) + 44-byte AKR dataset name.	Serialization of the program directory in an AKR.
VIASOFT	2-byte type code (P, U, or C) + 44-byte AKR dataset name + 10-byte program name.	Serialization of a particular program in an AKR.
VIASOFT	8-byte user ID + 3-byte constant (VIA) + 1-byte type code (A, W, R, P, S, V, X, Y, or Z) + 4-byte relative queue position.	Serialization of the SmartTest Batch queue file.
VIASOFT	8-byte user ID + 8-byte impact list name + 44-byte AKR dataset name.	Serialization of the Alliance impact lists.

### Other Resource Managers: Multi-image Manager

If your global resource manager is Multi-Image Manager (MIMS), or another resource manager product, these configuration changes are required:

- If the Multi-Image Manager processing mode is PROCESS=ALLSYSTEMS (or P=A), set the Reserve-MACRO to NO and do not change the qname list.
- If the Multi-Image Manager processing mode is PROCESS=SELECT, add the VIASOFT QNAME to the qname list. The VIASOFT QNAME is used for an AKR created by Center and for the SmartTest Batch queue file. This should be the corresponding QNAME list entry:

```
VIASOFT GDIF=YES SCOPE=SYSTEMS EXEMPT=NO RESERVES=CONVERT
```

- If the Multi-Image Manager EDIF facility is being used to automatically enqueue datasets, add this EDIPARMS exclusion entry to exclude AKRs:

```
SUFFIX NAME=.AKR OPTION (NOENQUEUE, NOWAIT)
```

Multi-Image Manager must not generate enqueues for AKR processing.

#### Note:

Failure to add the exclusion entry may cause mainframe processing to wait on the resource EDIDSN, and eventually fail with an S522 abend code.

### IBM Maintenance for ISPF, DFP, and MVS

These IBM APARs need to be applied to the ISPF/PDF system. ESW products do not abnormally terminate without these APARs; however, various processing restrictions do occur.

#### CICS 4.1

APAR	PTF	Description
PQ07234	UQ08600	Customer is running an application that is a long-running task and it has a GLUE exit that issues EXEC CICS INQUIRE STORAGE. This causes an accumulation of storage in the CICS region when using Bridge-CICS.
PN89788	UN97278	CSD analyze fails with an ASG6975E and a DFH5201 Command Not Valid message. The problem can be circumvented by entering BCSD=(.20.) as an analyze option.

#### ISPF 3.5

APAR	PTF	Description
OY61762	UY92530 - R302 UY92531 - R402 UY92532 - R502	DSNAME LEVEL ERROR MESSAGE (ISRU172) occurs when an asterisk (*) is used in the dataset name on Option 3.4 and SmartEdit is installed as the default editor.
OW00474	UW01257	When using the Program Summary feature and selecting the FILES option, an ISPF SUBTASK ABEND may occur. This abend usually occurs the first time this option is selected and if no other items in the Program Summary List were previously selected. The problem is usually associated with Model 2 type terminals.

**DFP 2.3**

APAR	PTF	Description
OY10535	UY16961	System 05C, return code 804 (hexadecimal) when trying to allocate multiple load libraries.  Prerequisite PTFs: UY14243, UY16278, and UY14182

**DFP 2.4**

APAR	PTF	Description
OY14850	UY23239	System 05C, return code 804 (hexadecimal) when trying to allocate multiple load libraries.

**DFP 3.1**

APAR	PTF	Description
OY23850	UY40930	Users running DFP3.1 with Model 13 controllers may experience IOS error messages and hardware errors when attempting to update or initialize an AKR.

**DFP 3.2.0**

APAR	PTF	Description
OY33195	UY53287	Doing a LIST MOD in SmartTest with SMS and non-SMS datasets concatenated results in a system 001-1abend.
OY57093 OY57448 OY57768	UY84655	0C1 and 0C4 abends occur upon entry to ESW components. Analyze ends with return code 2816. If these PTFs are not applied, do not change the LBLKSIZ parameter in the ESW installation JCL. If you want to reblock the ESW library, use the ESW CNTL member VIASCOPY to reblock after the libraries are unloaded to disk.

**DFP 3.3**

APAR	PTF	Description
OY57093	UY84656	0C1 and 0C4 abends occur upon entry to ESW components. Analyze ends with return code 2816. If these PTFs are not applied, do not change the LBLKSIZ parameter in the ESW installation JCL. If you want to reblock the ESW library, use the ESW CNTL member VIASCOPY to reblock after the libraries are unloaded to disk.
OY57448		
OY57768		

---

---

# 2

## ESW Installation and Customization

---

This chapter describes how to install and customize Center and related ESW products, and includes these sections:

Topic	Page
<a href="#">Overview</a>	<a href="#">9</a>
<a href="#">Center Installation Checklist</a>	<a href="#">11</a>
<a href="#">Installing ESW</a>	<a href="#">13</a>
<a href="#">Customizing ESW</a>	<a href="#">25</a>
<a href="#">Customizing Common Options and Update Your TSO/ISPF Environment</a>	<a href="#">35</a>

### Overview

All ESW products are distributed on tape. Each product requires a unique product key that defines the licensing characteristics of the product and any options that are included in the licensing agreement. The product keys are contained in the letter that accompanied the product tapes. However, if you cannot locate the product keys, you can obtain them by contacting ASG Order Processing in the same way you would contact the ASG Service Desk. See ["Step 3a - Installing the Product Keys" on page 27](#) for instructions on installing the product key.

**Note:** \_\_\_\_\_

If you are installing ESW products into an existing Center installation, proceed to ["Installing Into an Existing Center Installation" on page 19](#).

---

All ESW products share a common set of libraries called Center. A separate product key for the Center libraries includes any licensed Center options. You must supply the Center product key in addition to the product keys for each individual ESW product. If you are installing individual ESW products on top of a previously installed Center, you may not need to resupply the Center product key; however, you must ensure that the installed version of Center is appropriate for the products you are installing. If it is not, you need to install a newer version of Center.

ESW products have several separately licensable options. These options are specified in the product key that is obtained from ASG Order Processing.

The product options listed for each of these ESW products are included in the product key for each product:

- Alliance
  - DB2 Export
  - ASM
  - PL/I
- AutoChange
- Bridge
  - CICS
    - CIC/DL/I
  - Euro
  - IMS
- Estimate
  - ASM
  - PL/I
- SmartTest
  - ASM
  - APS
  - CICS
  - DB2 Stored Procedure Option
  - IMS
  - PL/I
  - Test Coverage Analysis Option (TCA)

Before installing any ESW products, read each product's Installation Guide. If you already downloaded the products you are installing, proceed to ["Step 6 - Editing the Installation Options in VIA\\$PRMS" on page 31](#).

**Note:** \_\_\_\_\_

If there are discrepancies between the JCL examples shown in this manual and the JCL on the product tape, the JCL on the product tape should be considered current.

---

DBCS Installations - See [Appendix H, "DBCS Addendum," on page 255](#), for special information regarding installation in a DBCS environment.

## Center Installation Checklist

### Installing ESW

**Note:** \_\_\_\_\_

Detailed instructions start on page [13](#).

---

- \_\_\_\_\_ Step 1 – Download the installation JCL.
- \_\_\_\_\_ Step 2 – Confirm availability of direct access storage space.
- \_\_\_\_\_ Step 3 – Modify the installation JCL and download selected products.

### Customizing ESW

**Note:** \_\_\_\_\_

Detailed instructions start on page [25](#).

---

- \_\_\_\_\_ Step 1 – Add the ESW CLIST library to SYSPROC.
- \_\_\_\_\_ Step 2 – Modify VIASCPRM and execute VIASCUST to tailor the CNTL and CLIST libraries.
- \_\_\_\_\_ Step 3 – Verify the parameter values in VIASBASJ and execute it to identify the parameter file library.
- \_\_\_\_\_ Step 4 – (Optional) Migrate prior Center installation options.
- \_\_\_\_\_ Step 5 – Identify and accommodate special preprocessing requirements.

\_\_\_\_\_ Step 6 – (Optional) Install the Alternate Language Facility.

\_\_\_\_\_ Step 7 – Edit the installation options in VIA\$PRMS.

**Note:** \_\_\_\_\_

See [Appendix A, "Installation Options," on page 157](#) for a complete description of all installation option fields.

---

\_\_\_\_\_ Step 8 – (Optional) Configure global resource managers for ESW products after reviewing the Dataset Reserves section in ["Overview & Planning" on page 1](#).

### **Customizing Common Options and Update Your TSO/ISPF Environment**

**Note:** \_\_\_\_\_

Detailed instructions start on page [35](#).

---

\_\_\_\_\_ Step 1 – Modify and execute the appropriate CNTL library members needed for your environment.

\_\_\_\_\_ Step 2 – (Optional) Modify user exits for the analyze process.

\_\_\_\_\_ Step 3 – Modify Skeleton library members (ISPSLIB).

\_\_\_\_\_ Step 4 – Modify CLIST library members.

\_\_\_\_\_ Step 5 – (Optional) Add Center load modules to MLPA/PLPA.

\_\_\_\_\_ Step 6 – Add ESW products to the TSO/ISPF environment.

\_\_\_\_\_ Step 7 – Prepare ESW products for use.

\_\_\_\_\_ Step 8 – Add Analyze Submit to ISPF/TSO environment.

\_\_\_\_\_ Step 9 – (Optional) Add the Command Usage Facility.

## Installing ESW

### Performing a New Installation

#### Step 1 - Downloading the Installation JCL

The installation JCL for all ESW products is contained in the first file on the tape, which is ASG.INSTALL.JCL. Download the installation JCL using this bootstrap JCL:

Do not change the spacing in this JCL.

```
//LOADJCL EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=ASG.INSTALL.JCL,DISP=SHR,UNIT=TAPE,
// VOL=SER=xxxxxxx,LABEL=(1,SL,EXPDT=98000),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUT2 DD DSN=user.lib(VIASINST),DISP=SHR
```

where:

*xxxxxxx* is the VOLSER on your product tape and *user.lib* is the name of a partitioned dataset where you want to store the installation JCL. The installation JCL is copied to the *user.lib* PDS and renamed VIASINST.

**Note:** \_\_\_\_\_

If you use this JCL, preallocate *user.lib* (VIASINST).

If your site uses a tape management system that cannot use labeled tapes, you must use a different version of the installation JCL. Download the Bypass Label Processing (BLP) version of the installation JCL with this bootstrap JCL:

```
//LOADJCL EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=ASG.INSTBLP.JCL,DISP=SHR,UNIT=TAPE,
// VOL=SER=xxxxxxx,LABEL=(5,BLP,EXPDT=98000)
//SYSUT2 DD DSN=user.lib(VIASINST),DISP=SHR
```

The differences in the Standard Label version of the installation JCL and the BLP version are minor; however, they occur on each tape file reference.

## Step 2 - Confirming Availability of Direct Access Storage Space

This table contains direct access storage space requirements for the ESW product libraries. When all ESW products are installed, the total is approximately 451 cylinders on a 3380 device.

Installation Member Name	Cylinders Allocated
ASG.VIACEN <sub>xx</sub> .LOADLIB	187
ASG.VIACEN <sub>xx</sub> .CNTL	30
ASG.VIACEN <sub>xx</sub> .CLIST	3
ASG.VIACEN <sub>xx</sub> .VIAMSGS	5
ASG.VIACEN <sub>xx</sub> .PTF	1
ASG.VIACEN <sub>xx</sub> .ISPLIB	45
ASG.VIACEN <sub>xx</sub> .ISPSLIB	1
ASG.VIACEN <sub>xx</sub> .ISPTLIB	1
ASG.VIACEN <sub>xx</sub> .MLSYNRWD	2
ASG.VIACEN <sub>xx</sub> .MLSLYNTBL	20
ASG.VIACEN <sub>xx</sub> .PLIMSGS	1
ASG.VIACEN <sub>xx</sub> .ALFIRLS.DATA	155
Total	451

Additional DASD space is required for one or more AKRs. Sites may create one common AKR, or users may each create their own individual AKRs. All ESW products, except SmartEdit, use an AKR.

The space needed for an AKR depends on the size and the number of programs to be placed in it, as well as on the Analyze features selected. Allocation of AKRs may be deferred until execution of the first selected product's installation verification process.

See [“Analysis Space Requirements” on page 131](#) to estimate your AKR space requirements. See the *ASG-Bridge Installation Guide* for Bridge-related space and resource analysis specifications.

### Step 3 - Modifying the Installation JCL and Downloading Selected Products

The JCL downloaded in [“Step 1 - Downloading the Installation JCL” on page 13](#) (VIASINST) creates ESW libraries or adds to existing ones. These are the core libraries that are created or modified:

- Product Libraries
  - CNTL
  - CLIST
  - ISPPLIB
  - ISPSLIB
  - ISPTLIB
  - LOADLIB
  - PTF
  - VIAMSGS
- Training Libraries
  - SETRNLIB (source code, copy libs, etc.)

Depending on the options you choose, these libraries are created or modified:

- MLSYNRWD and MLSYNTBL If installing Alternate Language Facility (ALF)
- PLIMSGS If installing ASG Application-PL/I

#### *To create new ESW libraries or add products to existing libraries*

- 1 Edit VIASINST after reviewing the Readme section in the JCL. Change the jobcard to a valid one for your site, including the TIME parameter. Adjust or delete the ROUTE and SETUP statements as required. See [step 4 on page 20](#) for a partial listing of the installation JCL.
- 2 Type `FIND INSTALL 3` to locate the invocation of the VIAINSTL procedure.
- 3 Specify these parameters in JCL overrides:

Parameter	Description
VIASOFT	Represents the high-level node for the ESW libraries (node length <=8, default=ASG)
CENTER	Represents the mid-level node for the ESW libraries (node length <=8, default=VIACENxx)

- 4 If necessary, adjust these parameters in JCL overrides:

Parameter	Description
TAPEUNT	Specifies the tape drive unit name.
SYSOUT	Specifies the SYSOUT class to be used for SYSPRINT datasets, if * is not appropriate.
SYSDA	Specifies the unit name to be used for allocating temporary datasets, if SYSDA is not appropriate. The amount of temporary space needed is approximately equal to the total for all products listed in <a href="#">"Step 2 - Confirming Availability of Direct Access Storage Space" on page 14</a> .

- 5 If space for the ESW libraries has already been allocated (i.e., you are reinstalling or upgrading Center and individual ESW products), uncomment the line with ALLOC='LE'.

If space for the ESW libraries has not yet been allocated, choose the appropriate instructions below depending on whether your site uses System Managed Storage (SMS).

### SMS Sites

To use SMS to allocate space for Center and individual ESW product libraries, adjust these parameters:

Parameter	Description
PRMMGMT	If your site uses SMS-managed datasets, specify the SMS Management Class to be used to allocate the new ESW libraries.
PRMSTOR	If your site uses SMS managed datasets, specify the SMS Storage Class to be used for allocating the new ESW libraries.
TMPSTOR	If your site has temporary datasets managed by SMS, specify the SMS Temporary Storage Class for the temporary datasets needed during the installation process.

**Non-SMS Sites**

*To allow the VIASINST JCL to allocate space for Center and individual product libraries*

- 1 Adjust these parameters:

Parameter	Description
PERMUNT	Specify the unit name for allocating the libraries.
PERMVOL	Replace xxxxxx with the desired volume serial. If you do not want the libraries to be allocated on a specific volume, comment out the line with PERMVOL='xxxxxx'. If you comment out the PERMVOL line, the libraries are allocated on any available volume within the unit type PERMUNT.

**Note:**

DBCS Installations - See [Appendix H, "DBCS Addendum," on page 255](#) for special information regarding installation in a DBCS environment.

- 2 Uncomment the line with FBCLIST='YES' if you want the CLIST library to have RECFM=FB (fixed-length blocked records) instead of RECFM=VB (variable-length blocked records). This also causes CBLKSIZ (block size for CLIST library) to be changed from 1680 to 3120.

**Note:**

If you concatenate the CLIST library with SYSPROC, first ensure the CLIST blocking matches the existing SYSPROC blocking.

- 3 If there are existing members in the ESW libraries, and you want them to be replaced, uncomment the line with REPLACE='YES'.

- 4** If you do not want to allocate specific product libraries, uncomment the appropriate product ALLOC statements at the bottom of the VIASINST JCL.

Parameter	Description
ALLOCML='LE'	Prevents allocation of Multiple Language Support (MLS) libraries.
ALLOCTR='LE'	Prevents allocation of training libraries.

- 5** Specify the products to download from the tape. Product selection is accomplished by changing the default keyword from NO to YES for each product and option listed in the letter that accompanied the product tape. Make these changes in the DOWNLOAD.VIAIN override at the end of the VIASINST JCL.

Specifying the keyword YES constitutes a request to download the corresponding set of libraries. Keywords for products you are not installing should be left as NO.

The product key letter contains the keywords required to activate the products after installation. Per seat licenses are included as part of the product key. These product keys are described in detail in ["Step 3a - Installing the Product Keys" on page 27](#).

**Note:**

\_\_\_\_\_

If you install a product option for a trial period, an override key is provided.

\_\_\_\_\_

If the passwords have not been provided, contact ASG Order Processing in the same way you would contact the ASG Service Desk.

- 6** Submit the JCL. The job asks for the ESW product tape and installs the libraries for the selected products.

## **Installing Into an Existing Center Installation**

### **Software Products**

When upgrading an ESW product or adding new products, you may need to reinstall other ESW products, for example:

- If you are reinstalling Center, you must reinstall all your ESW products.
- The product you are installing must use the same Center libraries as those already installed. Each ESW product uses Center libraries from a specific Center release. Installed products must all use the same Center libraries. Ensure that the release number of the Center libraries needed by each product matches the release number of the Center libraries already installed.

You must also ensure that the maintenance levels match. To determine if the maintenance levels match, log on to ESW and type `TSO PRODINFO`. Compare the displayed Center maintenance level to the level listed on the product tape for the products to be installed.

If the release and maintenance levels do not match with those required by installed ESW products, you must bring the installed Center libraries up to the release and maintenance level required. If the Center release and maintenance levels match with those required by installed ESW products, you can perform a single product install.

### **To perform a single product install**

- 1 Uncomment the `ALLOC=` parameter in the `VIASINST JCL`.

```
//*          ALLOC='LE', TO SKIP ALLOCATION OF LIBRARIES
```

- 2 Specify the keyword `YES` for the specific product(s) and product option(s) to be installed. Change all other keywords to `NO` or comment out the lines by placing an asterisk (\*) in column 1.
- 3 Verify that the `REPLACE=` parameter in the install JCL specifies `NO`.
- 4 Submit the job after making the appropriate changes in the install JCL (or a copy of it). The job will ask for the ESW product tape and installs libraries for the selected products.

## ESW Application-MLS

If you are installing ESW Application-MLS into an existing Center installation, make sure that the release and maintenance level of the installed Center shared libraries matches the release and maintenance level of the ESW Application-MLS you are installing. Then allocate space for the ESW Application-MLS libraries as defined in the Installation JCL on the product tape.

---

```
//ALLOCMLS EXEC PGM=IEFBR14,COND=(0,&ALLOCML)
//VIANTBL DD DSN=&VIASOFT..&CENTER..MLSYNTBL,DISP=(,CATLG,DELETE),
//          UNIT=&PERMUNT,VOL=SER=&PERMVOL,
//          DCB=(RECFM=FB,LRECL=2000,BLKSIZE=6000),
//          MGMTCLAS=&PRMMGMT,
//          STORCLAS=&PRMSTOR,
//          SPACE=(CYL,(20,1,25))
//*
//VIANRWD DD DSN=&VIASOFT..&CENTER..MLSYNRWD,DISP=(,CATLG,DELETE),
//          UNIT=&PERMUNT,VOL=SER=&PERMVOL,
//          DCB=(RECFM=FB,LRECL=54,BLKSIZE=3240),
//          MGMTCLAS=&PRMMGMT,
//          STORCLAS=&PRMSTOR,
//          SPACE=(CYL,(2,1,25))
```

---

### To install the Application-MLS libraries

- 1 Uncomment the ALLOCML='LE' parameter in the VIASINST JCL:

```
//* ALLOCML='LE', PREVENT ALLOCATE MLS LIBRARIES
```

- 2 Specify only the APPLICATION-MLS keyword as YES. Make sure that all other keywords are set to NO or comment out all product selection lines by placing an asterisk (\*) in the first column of each.

- 3 Verify that the REPLACE= parameter in the install JCL specifies NO.

- 4 After editing VIASINST, submit the job. The job asks for the ESW product tape and installs the libraries for APPLICATION-MLS. This examples shows the applicable parts of the JCL:

```
//ASG      JOB ( ),'ASG INSTALL',TIME=(1440),REGION=6M
//*      INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
/*SETUP ** NEED TAPE VOL=SER=ESW601 ASG-CENTER TAPE
//*
//*****
//*
//*      ASG, INC.          ASG-CENTER
//*
//*      ALLOCATE ASG LIBRARIES AND DOWNLOAD PRODUCTS FROM TAPE
//*
//*      NOTE: SEE THE "README FIRST" DIRECTLY BELOW FOR
//*            INSTRUCTIONS FOR CUSTOMIZING THIS JCL.
//*
//*
```

```

/**      PLEASE MAKE YOUR SYMBOLIC OVERRIDE CHANGES AT THE BOTTOM      *
/**      OF THIS JCL IN THE VIAINSTL STEP EXECUTION JCL.                *
/**      *                                                                *
/**      *****                                                        *
/**      >>>> README FIRST >>>> README FIRST <<<<< README FIRST <<<<< *
/**      *****                                                        *
/**      *                                                                *
/**      *                                                                *
/**      THIS JCL WILL ALLOCATE THE ASG LIBRARIES AND DOWNLOAD THE      *
/**      PRODUCTS TO DISK. MORE DETAILED INSTRUCTIONS ARE IN THE      *
/**      TECHNICAL SUPPORT MANUALS. ALL SYMBOLIC OVERRIDES SHOULD BE  *
/**      DONE IN THE "INSTALL" STEP EXECUTION JCL AT THE BOTTOM OF THIS *
/**      MEMBER.                                                         *
/**      *                                                                *
/**      1. JOB CARD                                                      *
/**      *                                                                *
/**      SPECIFY A VALID JOB CARD AND VERIFY A TIME=1440 IS ALSO      *
/**      SPECIFIED                                                       *
/**      *                                                                *
/**      2. LIBRARY NAMES:                                              *
/**      *                                                                *
/**      ENTER THE CORRECT VALUES FOR THE HIGH LEVEL NODE AND THE    *
/**      MIDDLE NODE OF THE ASG LIBRARIES IN 'INSTALL' STEP          *
/**      EXECUTION JCL AT THE BOTTOM OF THIS MEMBER:                  *
/**      *                                                                *
/**      VIASOFT='ASG'          HIGH LEVEL NODE OF ASG LIBRARIES      *
/**      CENTER='VIACENXX'      MIDDLE NODE OF ASG LIBRARIES          *
/**      *                                                                *
/**      *                                                                *
/**      3. TAPE, SYSOUT AND SYSDA OVERRIDES:                          *
/**      *                                                                *
/**      ADJUST THE TAPEUNT, SYSOUT, AND SYSDA PARAMETERS AS NEEDED.  *
/**      *                                                                *
/**      *                                                                *
/**      4. LIBRARY ALLOCATION:                                          *
/**      *                                                                *
/**      FOR ALLOCATING THE LIBRARIES, ENTER THE CORRECT VALUES FOR  *
/**      PERMUNT AND PERMVOL. IF YOU DO NOT WANT TO USE A SPECIFIC    *
/**      VOLUME, BUT INSTEAD WANT THE LIBRARIES TO BE ALLOCATED ON    *
/**      ANY AVAILABLE VOLUME WITHIN THE UNIT TYPE PERMUNT, THEN      *
/**      COMMENT OUT THE LINE WITH PERMVOL='XXXXXX'.                  *
/**      *                                                                *
/**      IF THE LIBRARIES ARE ALREADY ALLOCATED AND YOU DO NOT WANT   *
/**      TO RE-ALLOCATE ANY LIBRARIES THEN UNCOMMENT THE LINE        *
/**      WITH ALLOC='LE' IN THE 'INSTALL' STEP EXEC JCL AT THE      *
/**      BOTTOM OF THIS MEMBER.                                         *
/**      *                                                                *
/**      *                                                                *
/**      IF THE SPECIFIC PRODUCT LIBRARIES ARE ALREADY ALLOCATED AND  *
/**      YOU DO NOT WANT TO REALLOCATE THE PRODUCT LIBRARIES THEN,   *
/**      UNCOMMENT THE SPECIFIC PRODUCT ALLOC'S IN THE 'INSTALL' STEP *
/**      EXECUTION JCL AT THE BOTTOM OF THIS MEMBER.                  *
/**      *                                                                *
/**      *                                                                *
/**      - PREVENTS ALLOCATION OF MLS LIBRARIES    - ALLOCML='LE'      *
/**      - PREVENTS ALLOCATION OF TRAINING LIBS    - ALLOCTR='LE'      *
/**      - PREVENTS ALLOCATION OF ASG FDP PROGRAMS - ALOCFDP='LE'      *
/**      *                                                                *
/**      THIS JOB WILL DELETE ANY LIBRARIES THAT ARE EMPTY WHEN IT    *
/**      COMPLETES. PLEASE DO NOT EDIT THE JCL OTHER THAN MAKING     *
/**      MODIFCATIONS TO THE PARAMETER OVERRIDES.                    *
/**      *                                                                *
/**      *                                                                *

```

```
/** 5. NON-SMS DISK OVERRIDES: *
/** *
/** IF YOU ARE INSTALLING THIS PRODUCT IN NON-SYSTEMS MANAGED *
/** STORAGE (SMS), THEN EDIT THE FOLLOWING PARAMETERS *
/** FOR YOUR ENVIRONMENT IN THE 'INSTALL' STEP EXEC JCL: *
/** *
/** PERMUNIT='' - UNIT TYPE PARAMETER FOR NON SMS *
/** PERMVOL='' - NON SMS PARAMETER FOR WHAT DASD DEVICE *
/** OR STRING THE ASG SOFTWARE WILL BE INSTALLED. *
/** IF LEFT BLANK, YOUR SYSTEM WILL DESIGNATE WHERE *
/** THE SOFTWARE IS INSTALLED. *
/** *
/** *
/** 6. SMS DISK MANAGEMENT OVERRIDES: *
/** *
/** IF YOU ARE INSTALLING THIS PRODUCT USING SYSTEM MANAGED *
/** STORAGE (SMS), THEN EDIT THE FOLLOWING PARAMETERS FOR *
/** YOUR SMS ENVIRONMENT IN THE 'INSTALL' EXEC JCL AT THE *
/** BOTTOM OF THIS MEMBER: *
/** *
/** PRMMGMT='' - SMS MGMTCLAS FOR PERMANENT LIBRARIES *
/** PRMSTOR='' - SMS STORCLAS FOR PERMANENT LIBRARIES *
/** TMPSTOR='' - SMS STORCLAS FOR TEMPORARY LIBRARIES *
/** *
/** *
/** 7. PRODUCT SELECTION: *
/** *
/** PRODUCT SELECTION KEYWORDS MUST BE SPECIFIED FOR THE EACH *
/** PRODUCT YOU HAVE LICENSED. SPECIFY THE KEYWORD 'YES' FOR *
/** EACH PRODUCT AND PRODUCT OPTION LISTED IN YOUR PRODUCT KEY *
/** LETTER. PRODUCT KEYS ARE ENTERED INTO MEMBER VIASAUTH *
/** IN THE CNTL LIBRARY IN A LATER STEP. PRODUCT KEYS MUST *
/** BE INSTALLED PRIOR TO EXECUTING ANY OF THE PRODUCTS. *
/** *
/** PRODUCTS THAT ARE NOT BEING INSTALLED CAN BE LEFT AS THEIR *
/** DEFAULT 'NO' VALUE OR CAN BE COMMENTED OUT BY PLACING AN *
/** ASTERISK IN COLUMN 1. *
/** *
/** NOTE: 'ASG-CENTER' MUST BE SELECTED UNLESS CENTER HAS *
/** ALREADY BEEN INSTALLED (FROM THE SAME RELEASE!) *
/** *
/** *
/** *****
/** >>> OPTIONAL SYMBOLIC INSTALLATION OVERRIDES FOLLOW <<< *
/** *****
/** *
/** *
/** 8. ALTERNATE PANEL AND MESSAGE FILES: (FOREIGN LANGUAGES) *
/** *
/** IF YOU NEED TO DOWNLOAD THE ALTERNATE PANEL LIBRARY AND *
/** MESSAGE FILE (FOR DBCS TERMINALS) IN ADDITION TO THE *
/** THE STANDARD ONES, UNCOMMENT THE LINE WITH ALTPM='NE'. *
/** *
/** *
/** 9. CLIST LIBRARY FIXED BLOCK OPTION: *
/** *
/** IF YOU WANT THE CLIST LIBRARY TO BE FIXED BLOCKED (RECFM=FB) *
/** INSTEAD OF VARIABLE BLOCKED (RECFM=VB), THEN UNCOMMENT THE *
/** LINE WITH FBCLIST='YES' THIS WILL ALSO SET THE CBLKSIZ *
/** *
/** NOTE: THIS WILL ALSO SET THE CBLKSIZ PARAMETER (BLOCK SIZE *
/** OF THE CLIST LIBRARY) TO '3120' FROM '1680'. *
```

```

/**
/**
/** 10.ADDING ADDITIONAL PRODUCTS OR MEMBER REPLACEMENT:
/**
/** IF THERE ARE EXISTING PRODUCTS OR MEMBERS IN THE ASG
/** LIBRARIES AND YOU ARE ADDING A NEW PRODUCT OR REPLACING
/** EXISTING MEMBERS FROM THE TAPE, UN-COMMENT THE FOLLOWING
/** SYMBOLIC IN THE 'INSTALL' STEP EXEC JCL AT THE BOTTOM.
/**
/** REPLACE='YES' - TO REPLACE ANY EXISTING MEMBERS
/**
/** NOTE: ANOTHER OPTION IS TO INSTALL INTO A SEPARATE
/** SET OF LIBRARIES AND COPY WITH TSO OR OTHER UTILITIES
/** TO THE PRODUCTION ASG LIBRARIES.
/**
/**
/**
/** 11. SUBMIT THIS JOB.
/**
/*******
/** >>>> END OF README <<<<<<
/*******
/**
/** *****
/** * IN-STREAM JCL PROCEDURE TO ALLOCATE ASG *
/** * LIBRARIES AND DOWNLOAD THE ASG PRODUCTS *
/** *****
/**
/**VIAINSTL PROC ALLOC='NE',
/** ALLOCML='NE',
/** ALTPM='LE',
/** ALLOCTR='NE',
/** ALOCFDP='NE',
/** BATNUM='350',
/** CBLKSIZ='1680',
/** CLRECL='255',
/** CRECFM='VB',
/** FBCLIST='NO',
/** PERMUNT='',
/** PERMVOL='',
/** PRMMGMT='',
/** PRMSTOR='',
/** REPLACE='NO',
/** SYSDA='SYSDA',
/** SYSOUT='*',
/** TAPEUNT='TAPE',
/** TAPEVOL='ESW601',
/** TMPSTOR='',
/** VIASOFT='ASG',
/** CENTER='ASGCENXX'
/**
/** .
/** .
/** .
/** *****
/** * INSTALL: VIAINSTL EXECUTE JCL *
/** *****
/**
/**INSTALL EXEC VIAINSTL, *** DOWNLOAD ASG PRODUCTS ***
/** VIASOFT='ASG', HIGH LEVEL NODE OF ASG LIBRARIES
/** CENTER='VIACENXX', MIDDLE NODE OF ASG LIBRARIES
/**

```

## ASG-Center Installation Guide

```

//      TAPEUNT='TAPE',      TAPE DRIVE UNIT NAME
//      SYSOUT='*',          SYSOUT CLASS FOR SYSPRINT DATA SETS
//      SYSDA='SYSDA',      UNIT NAME FOR TEMPORARY DATA SETS
//**
//**      ALLOC='LE',        TO SKIP ALLOCATION OF LIBRARIES
//**      ALLOCML='LE',     PREVENT ALLOCATION OF MLS LIBRARIES
//**      ALLOCTR='LE',     PREVENT ALLOCATION OF TRAINING LIBS
//**      ALOCFDP='LE',     PREVENT ALLOCATION OF FDP  OPT LIBS
//**
//      PERMUNT='',         UNIT NAME FOR ASG LIBRARIES
//      PERMVOL='',        SPECIFIC VOLUME SERIAL FOR LIBRARIES
//      PRMMGMT='',        SMS MGMTCLAS FOR PERMANENT LIBRARIES
//      PRMSTOR='',        SMS STORCLAS FOR PERMANENT LIBRARIES
//**
//      TMPSTOR=''         SMS STORCLAS FOR TEMPORARY LIBRARIES
//**      NOTE: ADD A CONTINUATION COMMA TO
//**      TMPSTOR= IF ANY OF THE FOLLOWING PARMS
//**      ARE USED
//**
//**      ALTPM='NE',        TO DOWNLOAD ALT PANELS/MESSAGES ALSO
//**
//**      FBCLIST TO CHANGE CLIST LIB TO FB
//**      FBCLIST='YES',CBLKSIZ='3120',CLRECL='80',CRECFM='FB',
//**
//**      REPLACE='YES'     TO REPLACE ANY EXISTING MEMBERS
//**
//**
//**      *****
//**      * PRODUCT SELECTION:                               *
//**      *                                                    *
//**      * CHANGE THE SELECTION KEYWORD FROM 'NO' TO 'YES' *
//**      * FOR EACH PRODUCT/PRODUCT OPTION YOU WISH TO   *
//**      * DOWNLOAD FROM THE TAPE.                         *
//**      *                                                    *
//**      * REMEMBER, ACCESS TO PRODUCTS IS CONTROLLED BY A *
//**      * PRODUCT KEY SUPPLIED IN THE VIASAUTH JCL        *
//**      *                                                    *
//**      *****
//**
//DOWNLOAD.VIAIN DD *
* SET KEYWORD TO YES FOR PRODUCTS/OPTIONS TO BE INSTALLED:
ASG-CENTER:          YES
ASG-CENTER-IDMS:    NO
ASG-ALLIANCE:       NO
ASG-ALLIANCE-EXPORT: NO
ASG-ESTIMATE:       NO
ASG-RECAP:          NO
ASG-APPLICATION-ASM: NO
ASG-APPLICATION-PLI: NO
ASG-APPLICATION-NAT: NO
ASG-APPLICATION-MLS: NO
ASG-ENCORE-WB:      NO
ASG-INSIGHT:        NO
ASG-SMARTDOC:       NO
ASG-SMARTEDIT:      NO
ASG-SMARTTEST:      NO
ASG-SMARTTEST-TSO:  NO
ASG-SMARTTEST-CICS: NO
ASG-SMARTTEST-ASM:  NO
ASG-SMARTTEST-APS:  NO
ASG-SMARTTEST-IMS:  NO
ASG-SMARTTEST-PLI:  NO

```

```
ASG-SMARTTEST-FS:      NO
ASG-SMARTTEST-TCA:     NO
ASG-SMARTTEST-DB2-SP:  NO
ASG-AUTO-CHANGE:       NO
ASG-BRIDGE:            NO
ASG-BRIDGE-CICS:       NO
ASG-BRIDGE-IDMS:       NO
ASG-BRIDGE-IMS:        NO
ASG-BRIDGE-DLI:        NO
ASG-BRIDGE-EURO:       NO
/*
//
```

## Customizing ESW

### Step 1 - Adding the ESW CLIST Library to SYSPROC

The ESW CLIST library must be permanently added to SYSPROC to support all ESW users. You may permanently allocate it through a TSO ALTLIB system level allocation or by a direct update to the SYSPROC concatenation in your ISPF procedure.

**Note:** \_\_\_\_\_

If you concatenate the CLIST library with SYSPROC, first ensure the CLIST blocking matches the existing SYSPROC blocking. If you are upgrading an existing Center installation, ensure that SYSPROC is pointing to the correct CLIST library.

---

If you cannot allocate the ESW CLIST library permanently at this time, you can choose to add it temporarily by using the VIASPROC CLIST. Remember, this procedure must be repeated each time you logon.

#### *To temporarily allocate the ESW CLIST library*

- 1 Review VIASPROC for invocation instructions. The VIASPROC CLIST offers both ADD and REMOVE features.
- 2 From the TSO commands screen, type this command (substituting the high-level qualifiers you have chosen):

```
EX 'ASG.VIACENxx.CLIST(VIASPROC)' 'ADD(ASG.VIACENxx.CLIST)'
```

## Step 2 - Modifying VIASCPRM and Executing VIASCUST to Tailor the CNTL and CLIST Libraries

Center includes several CLIST and CNTL library members. They contain jobcards, references to the ESW libraries, generic unit names, COBOL compile and run-time libraries, the name of the assembler to use, and so forth. By specifying resource names in VIASCPRM and running the VIASCUST CLIST, you can update all of these references to reflect your site's standards.

### To tailor the CNTL and CLIST libraries

- 1 Edit the SET parameters in the VIASCPRM CLIST member:

Parameter	Description
VIASOFT	Represents the high-level node of the ESW DSNs (node length <=8, default=ASG).
CENTER	Represents the mid-level node of the ESW DSNs (node length <= 8, default=VIACEN <sub>xx</sub> ).
LIBDEF	Indicates whether LIBDEF is being used. The valid values are YES and NO and the default is YES.
ALTLIB	Specifies the TSO ALTLIB used to ALLOC CENTER CLIST. The valid values are YES and NO, and the default is NO.
VIAWORK	Represents the high-level node of ESW MLS VSAM DSNs (length <=8, default=VIAWORK, users need update access).

**Caution!** Be careful when editing VIASCPRM. Any spelling or typographical errors are propagated to all the affected CLIST and CNTL library members. When you execute VIASCUST, all the CLIST and CNTL symbolic values are replaced by the values in VIASCPRM. To correct errors, you cannot re-edit VIASCPRM. You must restore a backup of the CLIST and CNTL libraries, re-edit VIASCPRM, and rerun VIASCUST.

- 2 Execute VIASCUST by typing `TSO VIASCUST` on any ISPF screen. VIASCUST may take 30 minutes or longer to run. As VIASCUST executes, it creates a sequential file named `ASG.CENTER.CUSTLOG`, writes return codes to it, and displays the file in the ISPF editor.

**Note:**

See [Appendix C, "VIASCUST Customization CLIST," on page 235](#) for detailed information about the parameters used by VIASCUST.

### Step 3 - Verifying and Executing the Parameter Values in VIASBASJ to Identify the Parameter File Library

Center uses parameter files to specify default execution options. The VIASBASJ job in the CNTL library creates a load module, VIASBASE, which contains the name of the library where the parameter files are stored. Executing the VIASCUST CLIST (see ["Step 2 - Modifying VIASCPRM and Executing VIASCUST to Tailor the CNTL and CLIST Libraries" on page 26](#)) makes the necessary updates to VIASBASJ.

#### *To identify the parameter file library*

- 1 Verify that these parameters in the VIASBASJ CNTL member contain the correct values for your site:
  - VIASOFT - The high-level node for the ESW libraries (node length <=8, default=ASG).
  - CENTER - The mid-level node for the ESW libraries (node length <=8, default=VIACEN<sub>xx</sub>).
- 2 Verify that the PARMBASE parameter in the VIASBASE macro contains the name of the library where all ESW installation option files are stored.
- 3 Edit the above parameters if necessary and submit the VIASBASJ job.

**Caution!** Failure to perform this step causes a user abend 955 at startup.

### Step 3a - Installing the Product Keys

The VIASAUTH member installs the product authorization key(s) into the load-module ASGPTBL. Product authorization key(s) are included in your installation package or are provided through e-mail, fax, or the ASG web site. You can copy and paste the site-specific information contained in your product authorization letter directly to VIASAUTH.

This JCL must reflect the authorization keys for all installed ESW products.

**Note:** \_\_\_\_\_

Additional installation instructions provided on the Product Authorization form supersede these instructions.

---

**To install the product keys**

- 1 Edit the VIASAUTH member in the product JCL library to conform to your site standards.
- 2 Place all product authorization codes between the ASGPRODS and ASGPRODE within VIASAUTH. For example:

```

ASGPRODS
FAC  ASGPROD 999999999,2001175,**2003-207,027319,1,D840,91D0,C511
      PRODVR1='9999'
FSW  ASGPROD 999999999,2001175,**2003-207,027319,1,D911,23FC,7609
      PRODVR1='9999'
ASGPRODE
    
```

- 3 Submit the edited JCL. You should receive a condition code of zero.

**VIASAUTH Key Field Descriptions**

JCL member VIASAUTH contains the assemble and link cards to create the ESW password module ASGPTBL. In this example, the key fields on the record are highlighted:

```

-----1-----2-----3-----4-----5-----6-----7-----+
CNX  ASGPROD 123456789,2000366,009672-ZZ7,**9876,3,648C,2674,2E22, X
      PRODVR1='999911'
    
```

This table describes these fields:

Field	Description
Pivotal Product Code	Specifies the Pivotal product code. This field is located in columns 1 through 6 and is 3 to 6 characters in length. The example value is CNX.
Pivotal Site ID	Specifies the Pivotal site ID number. This field is located in columns 18 through 26 and is 9 characters in length. The example value is 123456789.
Expiration Date	Specifies the expiration date of the product license. This field is located in columns 28 through 34 and is 7 characters in length in the format YYYYDDD. The example shows year 2000, day 366 (2000366).
CPU MODEL	Specifies the CPU model number. This field is located in columns 36 through 45 and is 10 characters in length. The example value is 009672-ZZ7.
CPU ID/SN	Specifies the CPU identification number. This field is located in columns 47 through 52 and is 6 characters in length. The example value is **9876.

Using these example values, the MVS command D M=CPU yields this output:

```
PROCESSOR STATUS
  ID  CPU              SERIAL
  0   +              0398769672
  1   +              1398769672
  2   +              2398769672
  3   +              3398769672
  4   +              4398769672
  5   +              5398769672

CPC ND = 009672.ZZ7.IBM.02.0000000049876
CPC SI = 9672.ZZ7.IBM.02.00000000000049876
CPC ID = 00
```

This is the password information provided by the ASG Product Distribution group:

CPU ID/SN: 039876  
CPU MODEL: IBM 9672 -ZZ7

PWD TYPE: Annual

NEW PASSWORD EXPIRES: 12/31/2002 (MM/DD/YYYY)

This is the password format for ESW Version 5.0 and later releases:

```
CNX      ASGPROD 123456789,2000366,009672-ZZ7,**9876,1,2DAA,8773,0C04, X
          PRODVR1='999910'
```

### Product Key Message Parameters

You can set these parameters in VIASPRMS to control the display of product key messages.

- ASG-Pkey-Message=YES|NO

Controls whether ASG APM error messages display when a product key is not found or is invalid. If this option is set to YES, messages display for all uninstalled products. If set to NO, these warning messages are suppressed.

- ASG-Pkey-Warning-Days=[5...45|45]

Controls the display of product expiration warning messages. The expiration message begins appearing on the *nn*th day prior to expiration. Allowable values for *nn* are 5 through 45 days.

## Product Option Flag Description

There are several product options that are referenced in the PRODVR1 statement of the password key. The example explains the binary codes that follow the 9999 on the key. Each digit represents a unique product option. For example, 1 indicates the product is enabled and 0 indicates that it is not enabled. The value 9999 in the PRODVR1 statement refers to the seat-based license. The 9999 value indicates unlimited seats.

**Caution!** The product key cannot be modified in any way. If the key changes, the internal checksums fail and access to all ESW products is denied.

```

Alliance: ALX      ASGPROD 900000000,2000366,***** ,***** ,3,1C6C,AD2C,1802, X
                  PRODVR1='999911111
Position 1:      Alliance DB2 Export Option
                2:      Alliance ASM Option
                3:      Alliance Natural Option
                4:      Alliance PL/I Option
                5:      Alliance TEF Export Option
Bridge:  BCX      ASGPROD 900000000,2000366,***** ,***** ,3,FCCB,0640,5F4B, X
                  PRODVR1='999911111
Position 1:      Bridge CICS/DLI sub-Option
                2:      Bridge CICS Option
                3:      Bridge Euro Option
                4:      Bridge IDMS Option
                5:      Bridge IMS Option
                6:      Bridge Y2K Option
AutoChange:CCX   ASGPROD 900000000,2000366,***** ,***** ,3,2CAF,6B54,431C, X
                  PRODVR1='99991
Center:  CNX      ASGPROD 900000000,2000366,***** ,***** ,3,648C,2674,2E22, X
                  PRODVR1='999911
Position 1:      Center Authorize Option
                2:      Center IDMS Option
SQ/CICS:  SQC     ASGPROD 900000000,2000366,***** ,***** ,3,1451,B2EC,5304
SQ/MVS:   SQM     ASGPROD 900000000,2000366,***** ,***** ,3,2493,1314,F412
SmartTest STX     ASGPROD 900000000,2000366,***** ,***** ,3,E55E,C3D1,1DD1, X
                  PRODVR1='9999111111111
Position 1:      SmartTest APS Option
                2:      SmartTest CICS Option
                3:      SmartTest IMS Option
                4:      SmartTest ASM Option
                5:      SmartTest DB2 Option
                6:      SmartTest File Services Option
                7:      SmartTest IDMS Option
                8:      SmartTest PL/I Option
                9:      SmartTest TCA Option
Encore:  ENX      ASGPROD 900000000,2000366,***** ,***** ,3,A399,F3FE,681D
Recap:   RCX      ASGPROD 900000000,2000366,***** ,***** ,3,8F48,38AF,0E8C
Insight: INX      ASGPROD 900000000,2000366,***** ,***** ,3,F6ED,12B4,C60D
SmartDoc: DCX     ASGPROD 900000000,2000366,***** ,***** ,3,4A2E,4EFE,0122
SmartEdit: SET    ASGPROD 900000000,2000366,***** ,***** ,3,9C7D,50D8,3845
Estimate: TMX     ASGPROD 900000000,2000366,***** ,***** ,3,4717,6CC7,4E1C, X
                  PRODVR1='9999111
Position 1:      Estimate ASM Option
                2:      Estimate Natural Option
                3:      Estimate PL/I Option

```

\*\*\* END OF REPORT \*\*\*

## Step 4 - Identifying and Accommodating Special Preprocessing Requirements

### Installing the ESW SVC to Enable Endeavor Support (Optional)

If you have Endeavor-based applications that are processed by ESW products, see ["Endeavor Installation and Customization" on page 49](#).

### Installing Application Analyzer User Exit (Optional)

See ["Step 2 - Modifying User Exits for the Analyze Process" on page 36](#), and [Appendix G, "User Exit Changes," on page 253](#) if you are installing one or more application-level products (Alliance, Recap, or Estimate) and your source code is stored in a format other than one of these:

- Standard ISPF/PDF source members and copybooks
- Librarian
- Panvalet
- Endeavor
- SCLM

## Step 5 - Installing the Alternate Language Facility (Optional)

Some ESW products, such as Alliance and Estimate, support alternate languages. To install ESW's Alternate Language Facility, see ["Alternate Language Facility" on page 59](#).

## Step 6 - Editing the Installation Options in VIA\$PRMS

The VIA\$PRMS CNTL library member contains two types of installation options. Some are used only by Center, while others are shared with one or more individual ESW products. When you ran VIASCUST, it adjusted the ASG-Hi-Level-Nodes parameter in VIA\$PRMS; however, you must now manually set the remaining installation options.

Before you edit VIA\$PRMS, you need to know the options appropriate for your site. The installation options listed in VIA\$PRMS are set to their default values. Verify that the default installation options are suitable for your site or adjust them as necessary.

**Note:** \_\_\_\_\_

All installation options, their defaults, and a description of each are contained in [Appendix A, "Installation Options," on page 157](#).

\_\_\_\_\_

For each individual ESW product you install, you can edit a product-specific installation option file contained in the ESW CNTL library. Parameter values in product-specific installation option files override parameter values contained in VIA\$PRMS.

These are the ESW CNTL library members containing product-specific default options:

CNTL Member	Product Option
VIA\$PRMA	Encore
VIA\$PRMB	Alliance
VIA\$PRMD	SmartDoc
VIA\$PRME	SmartEdit
VIA\$PRMG	Bridge
VIA\$PRMJ	Estimate
VIA\$PRMM	AutoChange
VIA\$PRMP	SmartTest
VIA\$PRMR	Encore
VIA\$PRMU	AutoChange and Bridge (shared)
VIA\$PRMX	Recap

For a list of options contained in a product-specific installation option file, see [Appendix A, "Installation Options," on page 157](#). All the above CNTL library members are shipped with duplicates named VIASRLS<sub>x</sub>, where <sub>x</sub> represents a letter denoting the specific product.

Individual users can override option settings in the product-specific installation option files and VIA\$PRMS with a file named VIAUPARM. VIAUPARM must be a preallocated PDS or sequential file defined to the user's TSO session with the DDNAME VIAUPARM.

To override VIA\$PRMS for Batch jobs, modify the SYSUT1 DD of the VIAUPARM IEBGENER step with the fully-qualified dataset name of the VIAUPARM dataset. For example:

```
//SYSUT1 DD DSN=ASG.VIACENxx.VIAUPARM(UPARM1),DISP=SHR
```

After loading the standard installation option files at runtime, ESW attempts to open VIAUPARM. If VIAUPARM exists for a user, installation options contained in it replace matching installation options already loaded. Consequently, VIAUPARM can contain only those installation options the user wants to override.

VIAUPARM cannot remove an installation option; however, it can remove the value of an installation option. For example, entering the installation option with a null value removes the value contained in the standard installation options file.

### Common Component Options

These are the common Center installation parameters. Where appropriate, the default is underlined. For a complete list of all installation options, see [Appendix A, "Installation Options," on page 157](#).

- AKR-Compression=YES|NO
- AKR-DSORG-VSAM=YES|NO
- DSN-Qualifier-Node=xxxxxxxxxx
- Online-List-Unit=SYSDA
- Online-Log-Unit=SYSDA
- Online-Perm-Unit=CYLS | TRKS | BLKS
- Online-Punch-Unit=SYSDA
- SMS=YES | NO
- SMS-Data-Class=
- SMS-Mgmt-Class=
- SMS-Storage-Class=
- SYSOUT-Class=\*
- ASG-Hi-Level-Nodes=ASG.VIACENxx
- Work-SYSDA=SYSDA

**Note:** \_\_\_\_\_

These parameters are used by Analyze, and are not needed by SmartEdit-only installations:

\_\_\_\_\_

- Analyze-Region-Size=4096
- Analyze-STEPLIB-Libraries=ASG.VIACENxxLOADLIB
- Analyze-Work-Unit=CYLS | TRKS | BLKS

**Note:** \_\_\_\_\_

If you change this option, you should also modify the Analyze-Work-Primary and Analyze-Work-Secondary options. For an Alliance or Encore install, Analyze-Work-Primary of 60 cylinders and Analyze-Work-Secondary of 20 cylinders is recommended.

\_\_\_\_\_

- JES-Proc=xxxxxxxx . xxxxxxxx . xxxxxxxx (xxxxxxxx)
- JOB-Proc=xxxxxxxx
- PROCLIB-JES-Alternates= (nnnnnnnn,d,xxx . xxx . xxxx,  
nnnnnnnn,d,xxx . xxx . xxxx,  
.  
.  
.  
nnnnnnnn,d,xxx . xxx . xxxx)
- PROCLIBs=SYS1.PROCLIB
- Translator-Work-Unit=CYLS | TRKS | BLKS

**Note:** \_\_\_\_\_

If you change this option, you should also change the Translator-Work-Primary and Translator-Work-Secondary options.

---

**Alliance, Estimate, and Recap Options**

- Appl-Work-Primary=50
- Appl-Work-Secondary=10
- Appl-Work-SYSDA=SYSDA
- Appl-Work-Unit=CYLS|TRKS|BLKS

**Note:** \_\_\_\_\_

If you change the Appl-Work-Unit option, you should also modify the Appl-Work-Primary and Appl-Work-Secondary options.

---

- Appl-Work-Volume=xxxxxx

**Step 7 - Configuring Global Resource Managers for ESW Products (Optional)**

If you are going to configure Global Resource Managers for ESW products, first review ["Dataset Reservesdataset" on page 4](#) .

## Customizing Common Options and Update Your TSO/ISPF Environment

### Step 1 - Modifying and Executing the Appropriate CNTL Library Members Needed for Your Environment

#### VIASBIND

If you have DB2 installed at your site, VIASBIND is required to analyze application-level and program-level analysis of DB2 programs. Modify and execute the VIASBIND JCL found in the ESW CNTL library. Correctly specify these parameters:

Parameter	Description
VIASOFT='ASG'	Represents the high-level node. The node length is <=8 and the default is ASG.
CENTER='VIACEN <sub>xx</sub> '	Represents the mid-level node. The node length is <=8 and the default is VIACEN <sub>xx</sub> , where <sub>xx</sub> is the Center release number.
DB2LIB='DB2.LOADLIB'	Specifies the name of the DB2 load library.

Modify these parameters under the //BIND.SYSTSIN DD \* and //BIND.SYSIN DD \* statements at the bottom of the JCL.

Parameter	Description
DSN SYSTEM	Specifies the name of the DB2 subsystem for this bind.
PLAN	<ul style="list-style-type: none"> <li>SYSTSIN - Specifies the name of the plan to be used for ESW access to the DB2 catalog on the BIND.</li> <li>SYSIN - Specifies the name of the plan to be used by DSNTIAD on the RUN (i.e., DSNTIA<sub>xx</sub> where <sub>xx</sub> is the DB2 release number).</li> </ul>
LIB	Specifies the name of the DB2 Load library where DSNTIAD is located.

#### *To accommodate more than one DB2 subsystem*

- 1 Locate the BIND step in the instream PROC. Repeat the BIND step for each DB2 subsystem at your site, changing BIND to BIND1, BIND2, etc. For example, if you have four subsystems, you should have four BIND steps: BIND, BIND1, BIND2, and BIND3.

- 2 Change the disposition of &&DBRMOUT from DISP=(OLD, DELETE) to DISP=(OLD,PASS) on all but the last BIND step.
- 3 Locate the //BIND.SYSTSIN DD and //BIND.SYSIN.DD statements. For each BIND step you added, repeat the //BIND.SYSTIN DD and //BIND.SYSIN DD statements, changing BIND to BIND1, BIND2, and so forth. In the SYSTEM(????) parameter, change ???? to represent the appropriate DB2 subsystems.

**Note:** \_\_\_\_\_

VIASBIND performs the BIND and grants PUBLIC access to the plan.

---

### VIASPAMJ

If Panvalet is installed at your site, edit the VIASPAMJ CNTL member and specify a valid jobcard and the correct values for the VIASOFT, CENTER, PANLIB, SYSOUT, and SYSDA parameters. Submit VIASPAMJ to assemble and link edit the Panvalet VIASPAM module.

**Note:** \_\_\_\_\_

Do not perform this step if you are using Panvalet R12 or above and the module PAM is available in LINKLST or LPA.

---

### VIASFAIJ

If Librarian is installed at your site, edit the VIASFAIJ CNTL member and specify a valid Jobcard and the correct values for VIASOFT, CENTER, LIBRLIB, LIBRMAC, SYSOUT, and SYSDA parameters. Submit VIASFAIJ to assemble and link edit the Librarian VIASFAIR module.

### Step 2 - Modifying User Exits for the Analyze Process

This step applies only if you are installing Alliance, Estimate, or Recap.

**Caution!** If you are upgrading your Center installation from a previous release and using a user exit, see [Appendix G, "User Exit Changes," on page 253](#).

ESW provides sample user exits that preprocess input source members for various source types and write output source to the output file. The user exits are delivered as shells that either copy the input source to the output source file or not copy the input source and set the return code to 4. The sample user exits are shipped with the copy code disabled and produce a return code of 4. CNTL library members are supplied as JCL to compile the default user exits.

The sample application analyzer user exits for various source types are included in the CNTL library in these members:

User Exit	Language	Compiler
VIASCOBU	COBOL	VIASCOBJ
VIASIMSU	IMS	VIASIMSJ
VIASCCSU	CICS	VIASCCSJ
VIASJCLU	JCL	VIASJCLJ
VIASASMU	Assembler	VIASASMJ
VIASPLIU	PL/I	VIASPLIJ
VIASALFU	Alternate Language	VIASALFJ

The sample application analyzer user exits include code to copy the source member from the input to the output file. In the sample user exit shipped on the product tape, the code to copy the source member is disabled. Instead of copying the source code, the sample user exit sets a return code of 4 and performs no other processing. It is the responsibility of a site's systems administrator to customize this user exit as required.

Compile and link the desired user exits into the ESW LOADLIB library with the member name identical to the exit member name. For example, VIASCOBU must be linked as member VIASCOBU. The exits are ATTACHED dynamically by the application analyzer.

Each JCL member used to compile a user exit uses the VIASCII PROC. If you have not already customized the PROC, you may need to do so in this step.

You can invoke the application analyzer user exit after the source member has been extracted in the analyze process. The user exit is given an input source (e.g., VIASCOBI, DD) and an output source (e.g., VIASCOBO, DD). If the user exit also expands copybooks to be returned to the analyzer, the user exit must store the expanded copybook using the DD VIAYINCL.

Three parameters are passed to the user exit specifying the name of the member being analyzed, the compiler parameters for the member or library, and the DSN of the library containing the member. This allows you to pass compiler parameter information to the user exit at both the library and member levels.

The user exits may be invoked during the analyze at either the Library or Member definition levels. To specify execution of the user exit, use its name as a parameter in the appropriate part of the application definition. For example, if the source is COBOL, type VIASCOBU as a parameter; if the source is IMS, type VIASIMSU as a parameter; and so forth. This indicates to the application analyzer that the user exits are executed for the library or member depending on where the parameter is defined to the application. To invoke the exit on an individual member, use the above definitions replacing Library Options with Member Options.

Specify the user exit parameters in the Analysis Params section of the definition, using the appropriate pop-up:

- For COBOL and PL/I source, enter the analyze parameter in the Compile exec parms field on the Library Options - Analysis Params pop-up.
- For Assembler source, enter the analyze parameter in the Assembly exec parms field.
- For non-COBOL source, enter the analyze parameter in the User exit parms field.

If you specify the library source manager type as USER during the definition of the application, the user exit for the source type is always invoked to retrieve the source from the source manager. In this case, the analyze parameter is not required in the application definition and all other user exit specifications remain as described above.

To specify that COPY libraries should be allocated to a specific DD within the user exit, enter the analyze parameter COPYDD=DDname on the same pop-ups listed above for the source member. This parameter overrides the installation option User-Exit-Copy-DD.

These are the default user exit specifications:

User Exit Specification	Description
Input DD	For VIASCOBU, the DDname VIASCOBI contains the original COBOL source. For VIASCCSU, VIASIMSU, VIASJCLU, VIASASMU, VIASALFU, and VIASPLIU, the DDname VIASYIN contains the original source.
Output DD	For VIASCOBU, the DDname VIASCOBO contains source modified by the user exit. For VIASCCSU, VIASIMSU, VIASJCLU, VIASASMU, VIASALFU, and VIASPLIU, the DDname VIASYOUT contains source modified by the user exit.

The output DD for COPY information is VIAYINCL.

Return codes indicate that the application analyzer should use the modified source in the DD VIASCOBO or VIASYOUT as input. Return codes 1 through 4 indicate that no errors were detected and that no modifications were made. The application analyzer uses the original source in DDname VIASCOBI or VIASYIN as input. Return codes greater than 4 indicate that an error occurred during the user exit processing. The application analyzer continues and uses the original source in DDname VIASCOBI or VIASYIN as input.

You can override the default user exit name and DDnames by specifying this options in the application definition:

```
USEREXIT=name, INDD=name, OUTDD=name
```

For example, if you have a COBOL preprocessor called ABC that expects INFILE as the input and OUTFILE as the output, you would type:

```
USEREXIT=ABC, INDD=INFILE, OUTDD=OUTFILE
```

The application analyzer passes the original source to ABC in the file INFILE and expects the user exit output in OUTFILE.

You can modify the sample user exits or write your own user exit. If any other DD is required, except for the input DD (e.g., VIASCOBI) and the output DD (e.g., VIASCOBO), you must dynamically allocate it in the user exit. For additional one-time DD allocations, you can also modify the user exits using the application-level JCL fragment installation parameter.

If the user exit expands copybooks for return to the analyzer, the expanded copybook must be written to the DD VIAYINCL as 80 byte records in this format:

```
BEGIN EXPANSION OF member name  
LIBRARY = library name  
START = starting line number of the copybook  
END = ending line number of the copybook  
KIND = kind
```

This control statement specifies the kind of copybook being expanded. If you do not specify a value for KIND=, COBOL\_COPY is assumed. These are the valid values for the KIND control statement:

- COBOL\_COPY COBOL copy statements
- LIB\_INC Librarian -INC statement
- LIB\_COPY Librarian copy statement
- PANVALET Panvalet ++INCLUDE statements
- SQL EXEC SQL INCLUDE statements

- IDMS COPY IDMS statements
- COPYDD DATACOM copy statements

```
TEXT = copy statement original text
CONTENTS = contents of the copybook<--- (Optional field)
END EXPANSION OF member name
```

The contents of the copybook can contain nested copies by inserting a BEGIN EXPANSION...END EXPANSION block of control statements. The contents of the copybook must be in fully preprocessed form, that is, any processing of the source performed by the user exit must also be applied to the copybook text in this file.

You can use VIAIMAST with a nested copy of VIAPHONE, for example:

```
BEGIN EXPANSION OF VIAIMAST
LIBRARY=VIACExx.CNTL
START=19
END=43
TEXT=COPY VIAIMAST
CONTENTS=
01 MASTER-IN.
   05 CLIENT-ID.
       10 DISTRICT-ID PIC 9(3).
       10 CUSTOMER-ID PIC 9(3).
   05 NAME PIC X(24).
   05 ADDRESS1 PIC X(24).
   05 CITY PIC X(20).
   05 STATE PIC X(2).
   05 ZIP.
       10 ZIP-CODE PIC 9(5).
       10 FILLER PIC 9(11).
BEGIN EXPANSION = VIAPHONE
LIBRARY=VIACExx.CNTL
START=29
END=32
TEXT=COPY VIAPHONE
CONTENTS =
   05 PHONE.
       10 AREA-CODE PIC 9(3).
       10 EXCHANGE PIC 9(3).
       10 PHONE-NUMBER PIC 9(4).
END EXPANSION=VIAPHONE
   05 LOAN-INFORMATION.
       10 PAYMENT-AMT PIC 9(7)V99.
       10 LOAN-AMT PIC 9(13)V99.
       10 LOAN-START-DATE PIC 9(6).
       10 LOAN-TYPE PIC 9(2).
       10 LOAN-BILL-DATE PIC 9(6).
       10 BILLING-DATE PIC 9(3).
       10 YEAR-TO-DATE-INTEREST PIC 9(13)V99.
END EXPANSION OF VIAIMAST
```

**Step 3 - Modifying Skeleton Library Members (ISPSLIB)**

**Note:** \_\_\_\_\_

If you are installing SmartEdit only, proceed to ["Step 4 - Modifying CLIST Library Members" on page 42.](#)

Edit these members and specify the correct values for the VIASOFT and CENTER parameters. Update authority is required for all users to access the AKR.

- VIASAKAP     Skeleton JCL for AKR initialization
- VIASAKXP    Skeleton JCL for AKR expansion

**Note:** \_\_\_\_\_

A complete list of all Skeleton members and their descriptions is contained in [Appendix B, "CNTL, Skeleton, and CLIST Members," on page 223.](#)

In the members previously listed, specify the correct value for these parameters:

Parameter	Description
VIASOFT	Specifies the high-level node where the ESW products are installed. The node length is <=8 and the default is ASG.
CENTER	Specifies the mid-level node where ESW products are installed (node length <=8, default=VIACEN <sub>xx</sub> ). If the ESW dataset names contain more than three nodes, specify all nodes except the first and last as CENTER (e.g., the dataset name of SYS3.VIACEN <sub>xx</sub> .NEW.LOADLIB would have VIASOFT=SYS3 and CENTER=VIACEN <sub>xx</sub> .NEW).

### Step 4 - Modifying CLIST Library Members

Review the VIASGBL CLIST member for modifications. This member contains global parameters for all other product CLISTs. A complete list of all CLIST members and their descriptions is available in [Appendix B, "CNTL, Skeleton, and CLIST Members," on page 223](#).

CLIST Member	Description
VIASOFT	Represents the high-level node where all ESW product datasets have been installed. The node length is <=8 and the default is ASG.
CENTER	Represents the mid-level node where all ESW product datasets have been installed. The node length is <=8 and the default is VIACEN <sub>xx</sub> .
LIBDEF	Specifies YES if ISPF R3.1 or above is installed and LIBDEF facilities are used for allocating datasets to ISPF sessions. Otherwise specify NO. The default is YES.
CMDLIST	Specifies whether to perform a trace of all the CLISTs as they execute. This member is for debugging purposes only. The default is NO.
SYSDA	Specifies the disk unit that should be used by ESW CLISTs to allocate datasets necessary during the execution of the products. The default is SYSDA.
USERDSQ	Allows a special qualifier name to be appended after the USERID and/or prefix for all product-generated datasets (i.e., Log, List, and Punch files). There is no default. The USERDSQ parm is overridden by the DSN-Qualifier-Node parameter (see <a href="#">"DSN-Qualifier-Node=" on page 197</a> ) located in the VIA\$PRMS CNTL member.
VIAJCL	Specifies the last node of the dataset name generated to contain the JCL that creates or expands the AKR. The default is VIAJCL.
PROF	Specifies the second and third qualifiers for ISPF profile dataset names.
WORKDSN	Creates a temporary work dataset name necessary to do the profile conversion.

**Step 5 - (Optional) Adding Center Load Modules to MLPA/PLPA**

In this step, the re-entrant load modules may be added to MLPA/PLPA. These Center load modules are re-entrant and linked AMODE(31),RMODE(ANY).

For the names of modules that are eligible for location in the MLPA or PLPA, see the Installation Guide(s) for any other ESW product(s) you have installed.

These modules are used extensively by all ESW products:

VIASONLP	VIASAKRP	VIASIOSP	VIASPAKP
VIASSTVP	VIASANLZ	VIASLCPP	VIASPKGT
VIASSYMB	VIASANLP	VIASLFXP	VIASRESP
VIASSTV2P	VIASGRMP	VIASLPOP	VIASSCMP
VIASXMIP	VIASSQL	VIASMISP	VIASSPNP
VIASAJCP			

These modules are used extensively by SmartEdit:

VIASEDTP	VIASEGNP	VIASEPRP	VIASEUTP
VIASED2P	VIASEPFP	VIASETVP	
VIASEFXP	VIASEPGP	VIASEUPP	

The VIASTSKP module is used extensively by Insight:

These modules are used extensively by SmartTest, Insight, and Encore:

VIASLMIP	VIASLPMP	VIASLTBP
VIASLM2P	VIASLSTP	VIASLTRP

These modules are used extensively by SmartDoc and Recap:

VIASDCOP	VIASMETP
----------	----------

These modules are used extensively by Alliance, Estimate, and Recap:

VIASYANZ	VIASYGMP	VIASYMDJ	VIASYBTP
VIASYIMS	VIASYMDK	VIASYCIC	VIASYJCL

VIASYMDL	VIASYCPY	VIASYLOD	VIASYONP
VIASYCSD	VIASYMDB	VIASYSHP	VIASYDBL
VIASYMDC	VIASYDSF	VIASYMDI	

These are the advantages of locating these modules in the MLPA or PLPA:

- A reduction of the memory requirement per user.
- An overall decrease in required swap space.
- Performance improvements.

Moving these modules to MLPA/PLPA is optional. It is recommended that the original ESW load library (from the installation tape) be kept as a staging library so that any required PTFs can easily be applied. The re-entrant modules can be copied to LPA and the non-reentrant modules copied to a separate user library. These steps also require changes to user logons or product allocations. The CNTL library contains two members, VIASLP AJ and VIASLPXJ, that you can use to perform these copy steps.

**Caution!** Do not use the ISPF 3.3 copy feature to copy these modules because some of them have aliases.

### **Step 6 - Adding ESW Products to the TSO/ISPF Environment**

DBCS Installations - See [Appendix H, "DBCS Addendum," on page 255](#) for special information regarding installation in a DBCS environment.

### **Modifying ISPF Logon Allocation Procedure or CLIST**

The logon allocations for ISPF must be modified to include these Center libraries:

- Load library
- Panel library
- Skeleton library
- Table library
- CLIST library

**Note:** \_\_\_\_\_

If the LIBDEF facility is used, you need to include only CLIST libraries in the ISPF logon allocations. However, if SmartEdit is made globally available as the standard editor, then using the LIBDEF facility is discouraged due to performance considerations. If SmartEdit is made available using the VIAFISPF CLIST, then you cannot use LIBDEF.

---

Defining the ESW libraries to ISPF requires that those libraries be concatenated with the other ISPF libraries. Below are examples of how this can be done by modifying the TSO logon procedure cards, or by modifying CLIST logon allocation cards. Use the appropriate values for ASG, VIACEN<sub>xx</sub>, and ISPFVER.

To use the LIBDEF facility, skip the allocations for ISPLLIB, ISPPLIB, ISPSLIB, and ISPTLIB. Continue with the CLIST allocations. Remember to keep the library with the largest block size first in the concatenation, or add a DCB=BLKSIZE parameter on the first DD card specifying the largest block size.

**Note:** \_\_\_\_\_

Some or all of the modifications below may have been done in a prior installation and may need no further customization.

---

### Load Library

TSO logon procedure cards:

```
//ISPLLIB DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR
// DD DSN=ISP.ISPFVER.ISPLLIB,DISP=SHR
```

CLIST allocation:

```
FREE DD(ISPLLIB)
ALLOC DD(ISPLLIB) DS('ASG.VIACENxx.LOADLIB' -
'ISP.ISPFVER.ISPLLIB') SHR
```

**Note:** \_\_\_\_\_

If SmartEdit is made available by aliasing VIAFPDF to ISPF and/or PDF, you must also allocate the ESW load library to STEPLIB before any ISPF system libraries are allocated in the TSO logon procedure:

---

```
//STEPLIB DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR
```

### Panel Library

TSO logon procedure cards:

```
//ISPPLIB DD DSN=ASG.VIACENxx.ISPPLIB,DISP=SHR
// DD DSN=ISP.ISPFVER.ISPPLIB,DISP=SHR
```

For DBCS terminals displaying Japanese, you should also change the ISPPALT DD:

```
//ISPPALT DD DSN=ASG.VIACENxx.ISPPALT,DISP=SHR
// DD DSN=ISP.ISPFVER.ISPPALT,DISP=SHR
```

For DBCS terminals displaying languages other than Japanese, you should also change the ISPPALT DD:

```
//ISPPALT DD DSN=ASG.VIACENxx.ISPPLIB,DISP=SHR
// DD DSN=ISP.ISPFVER.ISPPALT,DISP=SHR
```

CLIST allocation:

```
FREE DD(ISPPLIB)
ALLOC DD(ISPPLIB) DS('ASG.VIACENxx.ISPPLIB' -
'ISP.ISPFVER.ISPPLIB') SHR
```

For DBCS terminals displaying Japanese, you should also change the CLIST allocation:

```
FREE DD(ISPPALT)
ALLOC DD(ISPPALT) DS('ASG.VIACENxx.ISPPALT' -
'ISP.ISPFVER.ISPPALT') SHR
```

For DBCS terminals displaying languages other than Japanese, you should also change the CLIST allocation:

```
FREE DD(ISPPALT)
ALLOC DD(ISPPALT) DS('ASG.VIACENxx.ISPPLIB' -
'ISP.ISPFVER.ISPPALT') SHR
```

## **Skeleton Library**

TSO logon procedure cards:

```
//ISPSLIB DD DSN=ASG.VIACENxx.ISPSLIB,DISP=SHR
// DD DSN=ISP.ISPFVER.ISPSLIB,DISP=SHR
```

If used, you should also change the ISPSALT DD:

```
//ISPSALT DD DSN=ASG.VIACENxx.ISPSLIB,DISP=SHR
// DD DSN=ISP.ISPFVER.ISPSALT,DISP=SHR
```

CLIST allocation:

```
FREE DD(ISPSLIB)
ALLOC DD(ISPSLIB) DS('ASG.VIACENxx.ISPSLIB' -
'ISP.ISPFVER.ISPSLIB') SHR
```

If ISPSALT is used, you should also change the CLIST allocation:

```
FREE DD(ISPSLIB)
ALLOC DD(ISPSLIB) DS('ASG.VIACENxx.ISPSLIB' -
'ISP.ISPFVER.ISPSALT') SHR
```

## Table Library

TSO logon procedure cards:

```
//ISPTLIB DD DSN=ASG.VIACENxx.ISPTLIB,DISP=SHR
// DD DSN=ISP.ISPFVER.ISPTLIB,DISP=SHR
```

CLIST allocation:

```
FREE DD(ISPTLIB)
ALLOC DD(ISPTLIB) DS('ASG.VIACENxx.ISPTLIB' -
'ISP.ISPFVER.ISPTLIB') SHR
```

## CLIST Library

TSO logon procedure cards:

```
//SYSPROC DD DSN=ASG.VIACENxx.CLIST,DISP=SHR
// DD DSN=SYS1.COMDPROC,DISP=SHR
```

CLIST allocation:

```
FREE DD(SYSPROC)
ALLOC DD(SYSPROC) DS('ASG.VIACENxx.CLIST' -
'SYS1.COMDPROC') SHR
```

## Step 7 - Preparing ESW Products for Use

**Note:** \_\_\_\_\_

If only one ESW product is being installed, proceed to ["Step 8 - Adding Analyze Submit to ISPF/TSO Environment" on page 48.](#)

---

Use these panel definition cards to add the ESW products menu to the ISPF primary menu or another dialog menu. For example, to describe the ESW product selection option to the user, use this statement:

```
% V +ASG - Analysis and Testing Products
```

To invoke the ESW products menu based on the user selection of the letter V, use this statement:

```
V, 'CMD(%CENTER) NEWAPPL(VIAC) '
```

**Note:** \_\_\_\_\_

After updating the ISPF environment you may need to re-enter ISPF before the facilities are available.

---

To invoke ESW products with a CLIST, follow this step:

- ▶ Use the VIACEN CLIST to invoke the ESW products menu rather than using the Center CLIST. The VIACEN CLIST invokes the Center CLIST while specifying the NEWAPPL (VIAC) parameter to set the correct application ID. These CLISTs are presented in the Center CLIST Members section of [Appendix B, "CNTL, Skeleton, and CLIST Members," on page 223](#).

### **Step 8 - Adding Analyze Submit to ISPF/TSO Environment**

**Note:**

Bridge, SmartEdit, Estimate, Alliance, and Recap do not require the Analyze Submit facility. The ESW program products (Insight, SmartDoc, SmartTest, and Encore) require that the Analyze Submit facility be installed.

Most ESW products require the execution of a Batch analyze job. This job is similar to a compile and requires the compile JCL for its execution. If the normal method of compilation is through the use of the ISPF Batch compile facility, or through a site-defined screen facility, see ["Analyze Submit Facility" on page 93](#) for instructions on how to implement Analyze Submit at your site.

### **Step 9 - (Optional) Adding the Command Usage Facility**

The Command Usage Facility tracks command usage within products. For instructions on how to implement the Command Usage Facility, see ["Command Usage Facility" on page 135](#).

---

# 3

## Endevor Installation and Customization

---

This chapter describes how to install and customize Center for an Endevor/MVS environment, and includes these sections:

Topic	Page
<a href="#">Enabling Endevor/MVS Support</a>	<a href="#">50</a>
<a href="#">Endevor Customization and Operation</a>	<a href="#">57</a>

**Note:**

If you are not using Endevor/MVS as your source manager, you can skip this chapter.

ESW has added support for Endevor/MVS to application-level products. This support allows you to treat Endevor/MVS as another type of library management system. Endevor support in Alliance and Recap requires the installation of the ESW module VIASSVCE (initially located in the ESW LOADLIB) in the same authorized load library that contains the Endevor utility module NDVRC1. This module installs an SVC that enables Alliance or Recap to invoke the Endevor utility directly, and extract the required source elements (members). By default, the ESW SVC uses SVC 214. If your site already uses this number, use the ENDEVOR-User-SVC-Number to change to any user SVC number that is available at your site.

The ESW SVC is dynamically installed by the program VIASSVCE. This program is executed by the procedure VIASNDVR (found in the ESW CNTL library). This procedure must be placed in a PROCLIB that is normally searched for PROC resolution at your site. The procedure VIASNDVR is automatically inserted into the generated JCL by Alliance or Recap when the application contains an Endevor library (type USER). The SVC installation program first checks to see if the SVC is installed. If it is, the program terminates normally. If the SVC is not installed, the program attempts to install it, and if successful, terminates normally. If the program is unable to install the SVC for any reason, it displays an error message and terminates abnormally.

## Enabling Endeavor/MVS Support

### *To enable Endeavor/MVS support*

- 1** Install the Endeavor/MVS interface SVC. It is preferable to install this interface permanently, but you may install it temporarily, if necessary.
- 2** Modify the ESW Endeavor/MVS installation options to reflect the proper dataset names for your site.
- 3** Modify the Endeavor/MVS SCL templates to reflect your site conventions, if necessary.
- 4** (Optional) Modify the ESW-supplied Endeavor user exit or create your own user exit and compile it into the ESW load library if you need to process source modules after they have been extracted and before they are processed by the application analyzer.
- 5** Test the Endeavor/MVS interface.

### *Step 1 - Installing the ESW Endeavor SVC*

Endeavor/MVS support requires that you permanently install an ESW Type 3 User SVC (VIASSVC). This SVC allows dynamic execution of the Endeavor/MVS product from within application-level products. It permits the extraction of source elements and member lists from the Endeavor database.

### **Installing the SVC Permanently**

The default SVC number of 214 is specified in the parameter CNTL member VIA\$PRMS under the parameter name Endeavor-User-SVC-Number=214. Your installation may require a different SVC number if 214 is already in use.

**Note:** \_\_\_\_\_

The VIASNDVJ CNTL member provides a FIND function that identifies the SVC numbers not in use.

---

The ESW-supplied Endeavor SVC module is named VIASSVC in the ESW load library. Note that IBM has defined specific naming conventions for user SVCs. The name should be IGC00 $xy$ , where  $xx$  corresponds to the first two digits of the SVC number, and  $y$  is one of these values:

Last Digit of SVC Number	Y Value
0	{
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H
9	I

The IBM-specified name for SVC 214 is IGC0021D.

Before installing the SVC as number 214, make a copy of the VIASSVC module and rename the copy to IGC0021D. The SVC must then be stored in SYS1.LPALIB. If you are installing this module under a different SVC number, you must rename it to the appropriate IBM-specified name for that number.

Add the SVC to the SYS1.PARMLIB(IEASVC00) member with a command like:

```
SVCPARM 214,REPLACE,TYPE(3),APF(NO)
```

After installing the SVC, an IPL is required.

## Installing the Endeavor SVC Temporarily

ESW supplies a support program and procedure that is used to dynamically install a temporary version of the ESW SVC. An IPL is not required for the temporary version of the SVC.

### *To install the SVC support program and procedure*

- 1 Copy the module VIASSVCE from the ESW Load Library to the authorized load library at your site that contains the Endeavor utility module NDVRC1.
- 2 Modify member VIASNDVR in the ESW CNTL library. Modify the AUTHLIB parameter to reflect the authorized library DSN where you placed the module VIASSVCE.
- 3 If you are executing on MVS/ESA 4.1 or earlier, copy member VIASNDVR from the ESW CNTL library to a system PROCLIB that is normally searched for Procedure resolution at your site.
- 4 Submit the JCL VIASNDVJ to invoke the procedure VIASNDVR.

You can also execute the VIASNDVR procedure to remove the temporarily installed ESW SVC by specifying the REMOVE parameter.

```
//* *****
//* * ASG, INC.           ASG-CENTER *
//* * * * * * * * * * * * * * * * * *
//* * ASG PROC USED TO DYNAMICALLY INSTALL THE ENDEAVOR SVC *
//* * * * * * * * * * * * * * * * * *
//* * THIS PROC IS USED BY THE APPLICATION ANALYZER EXECUTION *
//* * JCL AND MUST BE PLACED IN AN ACCESSIBLE PROCLIB, OR THE *
//* * JCL MUST BE MODIFIED TO CONTAIN A PROCLIB STATEMENT. *
//* * THIS PROC IS ALSO INVOKED BY THE CNTL MEMBER "VIASNDVJ". *
//* * * * * * * * * * * * * * * * * *
//* * MODIFY THE PARAMETER 'AUTHLIB' TO POINT TO THE AUTHORIZED *
//* * LIBRARY THAT 'VIASSVC' AND 'VIASSVCE' WERE INSTALLED IN. *
//* * * * * * * * * * * * * * * * * *
//* * MODIFY THE ASG AND CENTER PARMS TO REFLECT THE LIBRARY*
//* * WHICH CONTAINS THE ASG INSTALLATION PARMS. *
//* * * * * * * * * * * * * * * * * *
//* * * PARAMETER FUNCTIONS: *
//* * * * * * * * * * * * * * * * * *
//* * * INSTALL: INSTALL THE ASG TEMPORARY SVC. *
//* * * REMOVE: REMOVE THE ASG TEMPORARY SVC. *
//* * * FIND: FIND/LIST THE SVC NUMBERS WHICH ARE NOT IN USE.*
//* * *****
//VIASNDVR PROC AUTHLIB='USER.AUTH.LOADLIB',
//          COMMAND=INSTALL,
//          VIASOFT='ASG',
//          CENTER='VIACENXX'
//*
//INSTALL EXEC PGM=VIASSVCE, PARM='&COMMAND'
//STEPLIB DD DSN=&AUTHLIB, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//PARMLIB DD DSN=&VIASOFT..&CENTER..CNTL(VIA$PRMS), DISP=SHR
```

### **Step 2 - Modifying the Endevor/MVS Installation Options**

These Endevor/MVS-specific installation options have been added to the VIA\$PRMS installation options file:

- ENDEVOR-Conlib
- ENDEVOR-Steplib
- ENDEVOR-User-SVC-Number
- ENDEVOR-TypeLength-NE80

Modify these installation options to specify the CONLIB used by Endevor/MVS, the Authorized Load Library from which Endevor/MVS is executed, and the ESW user SVC number for Endevor access. If ESW products will be processing custom Endevor TYPE members whose LRECL is not 80, then you must specify the option Endevor-Type Length-NE80.

### **Step 3 - Customizing the SCL Templates**

To provide the most user-friendly and flexible access to Endevor controlled members, ESW products generate Endevor SCL from the user-modifiable model SCL templates VIASSCLA (ADD), VIASSCLL (LIST), and VIASSCLR (RETRIEVE). These templates can be found in the CNTL library. The model SCL templates are shown in [Figure 6 on page 54](#), [Figure 7 on page 54](#), [Figure 8 on page 54](#), and [Figure 9 on page 55](#).

You may modify these templates, if necessary, to better suit your needs or your particular environment. For example, some sites require the use of CCID or comment options in the SCL.

Review the model Endevor SCL templates VIASSCLA, VIASSCLL, VIASSCLR, and VIASSCLN for any additional syntax requirements at your site. If necessary, modify these templates to satisfy your site requirements.

If you modify these templates, you must adhere to these rules:

- Each variable name must appear only once.
- Multiple variables may appear on the same line, space permitting. The model is formatted across multiple lines for readability only.
- Variables may appear on any line and in any order, subject only to the rules of SCL syntax.

- The variable &STAGENM must remain unquoted. If quotes are required by the syntax (e.g., in the case of a Stage ID), the Endeavor interface generator supplies them automatically.
- Any other valid SCL syntax may be added to the command model, as long as it doesn't violate the rules stated above.

**Figure 6 • VIASSCLA - Model Endeavor SCL Template for ADD**

```
ADD      ELEMENT  '&ELEMENT '  
        FROM DDNAME '&DDNAME '  
        TO ENV    '&ENVNAME '  
          SYS    '&SYSNAME '  
          SUBSYS '&SUBSYSN '  
          TYPE  '&TYPENAM '  
        OPTIONS  
          CCID   ASG  
          COMMENT '&COMMENT '  
          &UPDATE  
          OVERRIDE SIGNOUT  
          BYPASS GENERATE PROCESSOR  
.  
EOF.
```

**Figure 7 • VIASSCLL - Model Endeavor SCL Template for LIST**

```
LIST     ELEMENT  '&ELEMENT '  
        FROM ENV  '&ENVNAME '  
          SYS    '&SYSNAME '  
          SUBSYS '&SUBSYSN '  
          TYPE  '&TYPENAM '  
          STAGE  &STAGENM  
        TO DDNAME '&DDNAME '  
        OPTIONS  
          REPLACE  
.  
EOF.
```

**Figure 8 • VIASSCLL - Model Endeavor SCL RETRIEVE Template with Include Expansion**

```
RETRIEVE ELEMENT  '&ELEMENT '  
        FROM ENV  '&ENVNAME '  
          SYS    '&SYSNAME '  
          SUBSYS '&SUBSYSN '  
          TYPE  '&TYPENAM '  
          STAGE  &STAGENM  
        TO DDNAME '&DDNAME '  
        OPTIONS  
          NO SIGNOUT  
          EXPAND  
          REPLACE  
.  
EOF.
```

Figure 9 • VIASSCLL - Model Endeavor SCL RETRIEVE Template without Expansion

```

CLEAR OPTIONS
EXPAND INCLUDES
.

RETRIEVE ELEMENT '&ELEMENT'
FROM ENV '&ENVNAME'
SYS '&SYSNAME'
SUBSYS '&SUBSYSN'
TYPE '&TYPENAM'
STAGE &STAGENM
TO DDNAME '&DDNAME'
OPTIONS
NO SIGNOUT
REPLACE
.
EOF.

```

**Note:**

The templates shown above must remain available in the ESW CNTL library.

**Step 4 - (Optional) Installing the Endeavor/MVS User Exit**

If you are upgrading your Center installation from a previous release and using a user exit, see [Appendix G, "User Exit Changes," on page 253](#).

ESW supplies an Endeavor/MVS user exit that allows user processing of extracted source modules after they have been extracted and before they are processed by the application analyzer. The user exit, VIASNDVX is delivered as a skeleton COBOL program that has been functionally disabled. The CNTL library member VIASNDXJ is supplied as model JCL to compile the default user exit.

The user exit is ATTACHED by the application analyzer after the source member has been extracted from the Endeavor/MVS Library Manager. The user exit is given an input (NDVROUT) and an output (NDVRNEW) dataset. Two parameters are passed to the user exit specifying the module name and its Endeavor/MVS type. When the user exit is invoked, the application analyzer has dynamically allocated the input and the output datasets.

When enabled, the supplied example user exit copies the dataset NDVROUT to the dataset NDVRNEW without making any changes. You can modify this exit to change the extracted source or write a new module to make the change. The user exit may be written in any language so long as the module can be ATTACHED by the ESW modules.

These are the user exit specifications:

User Exit Specification	Description
Location	The exit module must be located in the ESW load library or in a library specified in the Analyze-Steplib-Libraries parameter.
Input DD	NDVROUT. This is preallocated with these parameters: DCB= (LRECL=80, BLKSIZE=16000, RECFM=FB) DISP= (NEW, DELETE)
Parameters	Register 1 points to a list containing these items: <ul style="list-style-type: none"><li>Name of the element being extracted</li><li>Type of the element being extracted</li></ul>
Output DD	NDVRNEW. This is preallocated with these parameters: DCB= (LRECL=80, BLKSIZE=16000, RECFM=FB) DISP= (NEW, DELETE)
Registers	Register 15 must contain one of these return codes when the user exit terminates: <ul style="list-style-type: none"><li>+<i>n</i> indicates that no change was performed. Use the original Endeavor/MVS output dataset (NDVROUT). The recommended values for this are 4, 8, 12, or 16.</li><li>0 indicates that the original Endeavor/MVS output has been modified. Use the modified dataset (NDVRNEW).</li><li>-<i>n</i> indicates that the user exit program failed to correctly change the Endeavor/MVS output dataset. Use the original Endeavor/MVS dataset (NDVROUT). The recommended values are -1 through -<i>n</i> as different reason codes.</li></ul>

The user exit is dynamically ATTACHED by VIASYANZ immediately after the source member has been extracted from the Endeavor/MVS library management system. In the event of a user exit abend, the Endeavor/MVS extraction is aborted.

### Step 5 - Testing the Endeavor/MVS Interface

For application-level products, test the implementation by setting up an Endeavor Application and selecting a program to be analyzed. See ["Endeavor Customization and Operation" on page 57](#) for more information.

The user exit must be available at all times for the Endeavor/MVS source extraction to work. The user exit is dynamically ATTACHED by VIASYANZ immediately after the source member has been extracted from the Endeavor/MVS library management system. If the user exit cannot be found, the Endeavor/MVS source extraction is aborted.

## Endeavor Customization and Operation

ESW's Endeavor support uses the Source Manager Type Endeavor. This support differs from other library management systems in that you must supply dummy (or pseudo) dataset names for each application in place of, or as a substitute for, the hierarchical category names used within Endeavor/MVS. For example, Endeavor/MVS stores elements by type in stages within the subsystem, within the system, and within the environment. For an Endeavor/MVS based application, you must specify a pseudo library dataset name for each unique combination of Environment, System, Subsystem, Type, and Stage. This dataset name consists of five nodes, one for each hierarchical name, in descending hierarchical order, as shown in this statement:

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

For example:

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

**Or**

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

From either of these pseudo library dataset name formats, application-level products extract the Environment (VIATEST), System (FINANCE), Subsystem (PAYROLL), Type (COBOL), and Stage names, and uses them to access the Endeavor/MVS database. Consequently, be sure that each node of these pseudo library dataset names matches exactly its Endeavor/MVS counterpart. The single exception to this requirement is the Stage node (or last node). If you choose to use a numeric Stage number instead of the alphabetic Stage ID, the numeric Stage number (1 or 2) must be appended to the literal prefix: STAGE# (see first example). Alternatively, you can choose to use the single character Stage ID as a number 1 character node (see second example).

## Considerations

Endevor/MVS controlled applications have these considerations:

- Member lists of newly added Endevor Library datasets are not available, and new members must be added manually.
- The BROWSE facility is not available for Endevor controlled members.
- Added member names cannot be verified against the Endevor library. New member names are checked against known members for duplication only.
- Application Product pseudo dataset names are not checked for accuracy or validity of the Endevor hierarchy names. An invalid dummy DSN (e.g., invalid Endevor System name) results in no members found or an aborted application analysis.
- Your ENDEVOR-CONLIB dataset name must be specified in the installation option ENDEVOR-CONLIB (see ["Step 2 - Modifying the Endevor/MVS Installation Options" on page 53](#)). An invalid or unspecified ENDEVOR-CONLIB installation option can result in no members found or an aborted application analysis.
- Your Endevor LOADLIB dataset name must be specified in the installation option Endevor STEPLIB (see ["Step 2 - Modifying the Endevor/MVS Installation Options" on page 53](#)). An invalid or unspecified ENDEVOR-Steplib installation option can result in no members found or an aborted application analysis.

---

# 4

## Alternate Language Facility

---

This chapter describes the Alternate Language Facility (ALF) support option, and contains these sections:

Topic	Page
<a href="#">Step 1 - Providing the ALF Option Passwords for Download</a>	<a href="#">60</a>
<a href="#">Step 2 - Modifying ALF Installation Options</a>	<a href="#">60</a>
<a href="#">Step 3 - Building the Required Datasets for the ALF</a>	<a href="#">61</a>
<a href="#">Analyzing IDEAL Applications</a>	<a href="#">61</a>
<a href="#">Analyzing NATURAL Applications</a>	<a href="#">62</a>
<a href="#">Analyzing Model 204 Applications</a>	<a href="#">67</a>
<a href="#">Analyzing Easytrieve Applications</a>	<a href="#">69</a>
<a href="#">Analyzing Fortran Applications</a>	<a href="#">70</a>

**Note:**

This option has been discontinued and is only provided for customers currently on maintenance.

ESW provides alternate language support that allows you to analyze several different alternate languages for Estimate. ALF also allows you to analyze NATURAL programs for Alliance.

If your site has acquired ALF, continue with this chapter.

These alternate languages are supported:

- IDEAL
- NATURAL
- Model 204 User Language
- Easytrieve and Easytrieve Plus
- Fortran

## Step 1 - Providing the ALF Option Passwords for Download

All required ALF load modules and CNTL members are downloaded when you complete ["Step 3 - Modifying the Installation JCL and Downloading Selected Products" on page 15](#).

## Step 2 - Modifying ALF Installation Options

The ALF installation options are contained in the VIA\$PRMS CNTL library member and are disabled by default. If you are using this feature in Estimate, you must complete these steps to modify the default values.

### *To set up the options needed for ALF support*

- 1 Edit the VIA\$PRMS member.
- 2 Delete the 3 lines containing Alternate-Lang-Defs, Alternate-Lang-Entities, and Alternate-Lang-Types. See [Appendix A, "Installation Options," on page 157](#) for more information about these parameters.
- 3 Append the ESW CNTL library member VIASALDF to the bottom of VIA\$PRMS.

**Caution!** VIASALDF contains specific, correct values for the Alternate-Lang-Defs, Alternate-Lang-Entities, and Alternate-Lang-Types installation options. All three installation options and their values are required for ALF support. They should not be modified. Any modification causes ALF to improperly function.

- 4 For Estimate, uncomment the appropriate ALTN entries in the Default-language-criteria parameter in VIA\$PRMJ. In VIAJLMNT, uncomment the language maintenance factors for languages loaded in VIA\$PRMJ. See the *ASG-Estimate Installation Guide* for more information.

## Step 3 - Building the Required Datasets for the ALF

### *To allocate the VSAM datasets required for ALF support*

- 1 Modify and execute the VIASMLSI JCL found in the ESW CNTL library. Be sure to specify these parameters correctly.

Parameter	Description
VIAWORK	Specifies the high-level node for ESW datasets.
VIACEN <sub>xx</sub>	Specifies the middle-level node for ESW datasets.
(xxxxxxx)	Specifies the volume where ESW datasets reside.

Because VIASMLSI does VSAM allocation while in edit, you must do a CHANGE ALL for the above values.

- 2 Submit the JCL VIASMLSI.

## Analyzing IDEAL Applications

If you purchased the MLS-IDEAL option, you can perform an Estimate analysis on CA-IDEAL source code that has been extracted from an IDEAL database.

### Step 1 - Preparing the Source for Analysis

For the ALF to analyze IDEAL source, the code must be extracted from the IDEAL database into a PDS. Consult your sites IDEAL DBA to determine the best way to do this. The Data Views, Programs, Panels, and Reports that make up the IDEAL application should be placed in separate members of a PDS with a record format of FB, a record length of 80, and a blocksize of 8000 (blocksize can be any multiple of 80). Support for expanding IDEAL COPY statements is not included. Source must be fully expanded prior to analysis, or a user exit such as VIASALFU must be used to expand COPY statements.

### Step 2 - Defining the Application

For instructions on defining IDEAL applications to a specific ESW product, consult the product's user's guide.

### **Step 3 - Submitting the Analyze Job**

Once your IDEAL programs are defined in an Application Definition, the process of analyzing the application is identical to that of analyzing COBOL programs.

For instructions on submitting the analyze job, consult the user's guide for the ESW product you are using.

## **Analyzing NATURAL Applications**

Unlike programs written in traditional languages, NATURAL programs and data definitions are kept in NATURAL databases. This section describes the manual process of extracting a NATURAL application from a NATURAL database and preparing to analyze the application.

### **Step 1 - Extracting the Application from the Database**

NATURAL applications are kept in NATURAL databases, which can reside on non-MVS platforms. ESW products installed along with Center R6.0 are unable 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 using SYSTRANS.

#### ***Using NATURAL Version 2.1 or Later***

NATURAL V2.1 and later comes with a utility to load and unload a NATURAL database. The utility is called SYSTRANS and can be run online or from JCL for the MVS platform.

[Figure 10](#) contains sample JCL used to run SYSTRANS under NATURAL V2.2.

**Figure 10 • Sample JCL used 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

- 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 determine the program name and if there is a standard JCL PROC you should use to run NATURAL from 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. Depending on your site's NATURAL configuration, you may be required to specify D as the value for the IM parameter.
- 3 The specific load libraries needed to run NATURAL in Batch depend on the NATURAL configuration at your 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, make sure you 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 may be needed to enable the rest of the commands to work.

- 6** Specifies the unload command.
- 7** These set the options for the unload command:
- 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** Selects NATURAL objects.
- 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** 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** 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** 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

The sample JCL in [Figure 10 on page 63](#) and its accompanying notes are based on ESW's current understanding of SYSTRANS. Because the documentation on SYSTRANS is very sketchy on some of the details, and because the syntax for the commands is slightly different from one release of NATURAL to another, you may need to experiment with the JCL and the options to get SYSTRANS to work correctly. You can find more up-to-date information about running SYSTRANS from the database administrator (DBA) at your site.

## **Step 2 - Transferring the Extracted File to MVS (Optional)**

If the platform for your NATURAL applications is not MVS, you must transfer the extracted file to MVS for analysis. You can transfer the NATURAL applications through a tape or a file transfer program. Make sure:

- The dataset on MVS has a record length (LRECL) no less than 96.
- The dataset is variable blocked (VB) or fixed blocked (FB).
- If you are transferring the file from a platform that uses ASCII as its character set, you specify ASCII-to-EBCDIC translation in your file transfer. If so, make sure the translation for the carat character (^) in ASCII is the logical not symbol (¬) in EBCDIC.

### Step 3 - Splitting 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 in an ESW product for analysis, you must split the SYSTRANS output file into individual programs and data definitions. You can run multiple SYSTRANS and concatenate them in the NATURAL splitter.

**Note:** \_\_\_\_\_

All information relating to an application must reside in one PDS. All associated DDM, LDA, GDA, and FDTS for program modules must be present.

\_\_\_\_\_

Center provides a utility program called NATURAL Splitter for this purpose.

#### *To use NATURAL Splitter*

**1** Allocate a partitioned dataset (PDS) with these characteristics:

- RECFM=FB,LRECL=96
- BLKSIZE=6240 (or any multiple of 96)

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.

**3** Edit the JCL and change the SYSIN DD to point to the SYSTRANS output file, and the VIASPGMS DD to the PDS you allocated in [step 1](#).

**4** 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 can be a program, a map, a subprogram, a subroutine, a help routine, or a data definition such as DDM, FDT, and so on. 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 shortens the name of any object whose name is longer than 8 characters. 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. \$\$VIAINV must be present in the PDS for the analysis to work.

### **Step 4 - Defining the Application**

Once the SYSTRANS file is split up and stored in a PDS, you may start defining the NATURAL application to an ESW product. The major difference with defining NATURAL applications is that there are no INCLUDE libraries and all DDM, FDT, LDA, and GDA are in the PDS containing the programs. A single PDS can contain both Structured and Reporting mode objects. Unlike other languages, you can define the file twice, once for each type.

**Note:** \_\_\_\_\_

Select only objects (programs, subroutines, maps, help, etc.) and not data definitions, such as DDM and LDA.

---

### **Step 5 - Submitting the Analyze Job**

Once your NATURAL programs have been defined in an application definition, analyzing the application is identical to analyzing a non-NATURAL application. For instructions on submitting the analyze job, see the user's guide for the product you are using.

## **Analyzing Model 204 Applications**

Unlike programs written in traditional languages, Model 204 programs are kept in a Model 204 database. This section describes the manual process for extracting a Model 204 application from the Model 204 database and preparing to analyze the application.

### **Step 1 - Extracting the Application from the Database**

Model 204 applications are kept in a Model 204 database. Center cannot read program source code directly from a Model 204 database. To prepare a Model 204 application for analysis, you must first extract it from its database. This is done by allocating an MVS sequential dataset, specifying use of that dataset for output, and outputting all procedures.

#### ***Allocating the Dataset***

Allocate a dataset such as ASG.MODEL204.PROC according to these specifications:

- Sequential
- Record Format: FB
- Record Length: 80
- Block Size: 8000 (or any multiple of 80)

The primary and secondary quantities for this dataset should conform to the size requirements of the procedures to be unloaded from the Model 204 database.

### Specifying Use of the Dataset

In Model 204, specify the use of the previously allocated dataset using these commands:

```
ALLOCATE OUTFILE DSNAME='ASG.MODEL204.PROC'  
USE OUTFILE
```

### Output All Procedures

In Model 204, list all procedures using this command:

```
DISPLAY PROCEDURE (LABEL,COMPACT) ALL
```

## Step 2 - Splitting the Extracted File into Individual Components

Output from the Model 204 DISPLAY command is a single file containing all programs from the Model 204 database. Before defining the application for analysis, you must split up this file into individual programs. Center provides a utility program called the Model 204 Splitter for this purpose.

### To use the Model 204 Splitter

**1** Allocate a partitioned dataset (PDS) with these characteristics:

- RECFM=FB
- LRECL=80
- BLKSIZE=8000 (or any multiple of 80)

Use the size of the DISPLAY output file to estimate the space and directory blocks needed for the PDS.

- 2** Copy the VIASM2SP JCL from the ESW CNTL library to your own dataset.
- 3** Edit the JCL and change the SYSIN DD (by changing the INFILE procedure parameter) to point to the sequential dataset, and the OUTPUT DD (by changing the OUTPDS procedure parameter) to the PDS you allocated in [step 1 on page 68](#).
- 4** Save and submit the JCL.

The Model 204 Splitter utility takes as its input an output file from the Model 204 DISPLAY command and splits it into individual members in a PDS. Each member contains an individual procedure.

Since PDS member names are limited to 8 characters and Model 204 procedure names may contain up to 255 characters, the Model 204 Splitter shortens the name of any object whose name is longer than 8 characters. A list of the programs in the DISPLAY output 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 3 - Defining the Application**

For instructions on defining Model 204 application to an ESW product, consult the product's user's guide.

### **Step 4 - Submitting the Analyze Job**

Once your Model 204 user language programs have been defined in an application definition, the process of analyzing the application is identical to that of analyzing a non-Model 204 application.

For instructions on submitting the analyze job, see the user's guide for the ESW product you are using.

## **Analyzing Easytrieve Applications**

If you purchased the MLS-Easytrieve option, you can perform an Estimate analysis on MVS Easytrieve and Easytrieve Plus source code.

### **Step 1 - Defining the Application**

ESW adds support for pulling Easytrieve macros from separate include libraries. Additionally, it includes alternate language support for third-party source managers such as Panvalet and Librarian, including support for expanding standard Easytrieve %macros from separate Panvalet or Librarian macro libraries. Support for Panvalet ++INCLUDE and Librarian -INC copybooks is also provided. Nine and ten character Panvalet member names are not supported for Easytrieve macros. Easytrieve Security ACCESS Codes and Library CONTROL Codes are also not supported. Endeavor support is provided for Easytrieve source libraries.

For instructions on defining Easytrieve applications to a specific ESW product, consult the product's user's guide.

## **Step 2 - Submitting the Analyze Job**

Once your Easytrieve programs are defined in an Application Definition, the process of analyzing the application is identical to that of analyzing COBOL programs.

For instructions on submitting the analyze job, consult the user's guide for the ESW product you are using.

## **Analyzing Fortran Applications**

If you purchased the MLS-Fortran option, you can perform an Estimate analyze on MVS IBM Fortran 77 source code.

### **Step 1 - Preparing the Source for Analysis**

For the Alternate Language Facility to analyze Fortran 77 source, the code must be in fully expanded format. Support for expanding Fortran INCLUDEs is not provided. Source must be fully expanded prior to analysis or a user exit such as VIASALFU must be used to expand INCLUDE statements. In addition, EQUIVALENCE statements are not currently supported and should be commented out prior to analysis.

### **Step 2 - Defining the Application**

For instructions on defining Fortran applications to a specific ESW product, consult the product's user's guide.

### **Step 3 - Submitting the Analyze Job**

Once your Fortran programs are defined in an Application Definition, the process of analyzing the application is identical to that of analyzing COBOL programs.

For instructions on submitting the analyze job, consult the user's guide for the ESW product you are using.

---

# 5

## AKR Management

---

This chapter describes how to define and maintain an AKR, and includes these sections:

Topic	Page
<a href="#">Maintaining AKRs</a>	<a href="#">71</a>
<a href="#">Defining Additional AKRs and Expanding Existing AKRs</a>	<a href="#">72</a>
<a href="#">Restrictions</a>	<a href="#">73</a>
<a href="#">Batch AKR Utility</a>	<a href="#">74</a>

See "[Analyze Submit Facility](#)" on page 93 for more information about performing an analyze. Also, see each product's installation guide for instructions on completing all necessary product installations prior to creating an AKR.

### Maintaining AKRs

You can allocate the AKR as a VSAM RRDS or as a BDAM file organization. Deleting or renaming members from the AKR can be done online or in Batch using the AKR Utilities facility. Statistics are also provided online for the AKR file, as well as for each member of the AKR.

If the default AKR organization is specified as BDAM for your site, existing VSAM RRDS AKRs continue to work correctly with ESW products. When an existing VSAM AKR is expanded, it is redefined as a BDAM AKR.

## Defining Additional AKRs and Expanding Existing AKRs

The File - AKR Allocate/Expand screen allows you to allocate and initialize AKRs online.

### To complete the File - AKR Utility screen

- 1 Select Manage AKR from the File pull-down to display the ASG-ESW - AKR Utility screen.
- 2 Type the name of the ESW AKR to be allocated or expanded.
- 3 Type A in the primary command field and press Enter. The File - AKR Allocate/Expand pop-up, shown in [Figure 11](#), displays with the ESW AKR name. Verify that it is correct.

Figure 11 • File - AKR Allocate/Expand Pop-up

```
File - AKR Allocate/Expand
Command ==> -----
          S - Submit JCL      E - Edit JCL      C - Specify Catalog
Expand existing AKR . . . NO          (Yes or No)
AKR data set name . . . . 'USER.TEST.AKR'
Volume . . . . . 12
Unit . . . . . 12          (Generic unit name)
Space units . . . . . RECORDS    (Records, Tracks or Cylinders)
Primary space . . . . . 4000     (Primary amount in above units)
Secondary space . . . . . 0      (Secondary amount in above units)

Job statement information:
//USER12 JOB (DEUIXS,283200),
//      MSGCLASS=A
//*   INSERT /*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
```

**Note:**

The exact format and fields on the File - AKR Allocate/Expand pop-up depend on the values for AKR-DSORG-VSAM and SMS in the Center installation option file, VIA\$PRMS.

- 4 If you are expanding an existing AKR, type YES in the Expand existing AKR field. Leave this value set to NO if you are allocating a new AKR.
- 5 Enter the SMS classes or the Volume and space information for this AKR.
- 6 For VSAM AKRs, enter the appropriate Unique parameter for the Volume.

- 7 If the AKR is to reside in a private catalog or requires a password, type C in the command input area to display the AKR Catalog Information pop-up, shown in [Figure 12](#), and enter a catalog dataset name and password.

Figure 12 • AKR Catalog Information Pop-up

```

Command ==> _____ AKR Catalog Information
Catalog DSN . . . . . _____
Password . . . . . _____ (Catalog password, if required)

```

- 8 Enter the appropriate Job statement information and submit the job by typing S in the command input area.

See the product validation chapter in each product installation guide for complete details.

The space needed for the AKR file depends on the size and the number of COBOL programs to be analyzed and placed in it. (A conservative rule in figuring AKR space is that for every 1,000 lines of COBOL code approximately eight 3390 tracks, ten 3380 tracks, or twenty-five 3350 tracks are required).

See "[Analysis Space Requirements](#)" on page 131 to estimate your AKR space requirements.

## Restrictions

**Caution!** Hardware and software compression routines and programs that release free space should not be run against BDAM AKRs. AKR data is compressed using its own internal routines. Releasing free space or running compression routines against AKRs corrupts the AKR data. If free space has been released from an AKR, reallocating the AKR to a size equal to or larger than its original size, before any additional processing, should salvage the data.

An AKR is theoretically limited to 1048575 records, but in practice it is limited by the size of the volume where it resides.

**Note:** \_\_\_\_\_  
An AKR may not span volumes.  
\_\_\_\_\_

## Batch AKR Utility

The Batch AKR Utility is used to maintain the AKR. Use the VIASAKRU JCL in the ASG.VIACEN<sub>xx</sub>.CNTL library to run this utility. These are the commands available in the Batch AKR Utility:

Command	Function
ANLZSTAT	Provides a summary of the composition and analysis status for one or more AKRs.
COMMENTS	Includes comments with the command.
CONVERT	Converts selected members that were analyzed with a previous release of ESW products to the current release level.
COPY	Copies selected members from one AKR to another.
DELETE	Deletes selected members from the AKR.
EXPORT	Creates metrics and function point CDF files.
HELP	Prints the AKR Utility Help report.
INIT	Formats a previously-defined dataset into an AKR format.
MOVE	Copies selected members from one AKR to another and deletes them from the original AKR.
PRINT	Prints AKR directory information, or COBOL source listings for selected AKR members.
PUNCH	Produces a file that contains the AKR directory information, or the COBOL source code for selected AKR members.

Output for the Batch AKR Utility commands PRINT and PUNCH excludes source for Alliance, Estimate, and Recap analyze types.

### 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 this section. See the description for each command to determine which DD statements are affected.

```

//ASG JOB ( ),'ASG-CENTER VIASAKRU'
/** INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
/**
/*******
/** ASG, INC.                ASG/CENTER                *
/**                          *                          *
/** MEMBER NAME: VIASAKRU  *                          *
/**                          *                          *
/** DESCRIPTION: UTILITY TO PERFORM VARIOUS PROCESSING WITH ASG *
/**                   APPLICATION KNOWLEDGE REPOSITORY (AKR)   *
/**                          *                          *
/** INSTRUCTIONS:        *                          *
/**                          *                          *
/** 1.  ADD A VALID JOB CARD. *                          *
/** 2.  CHANGE THE SYMBOLIC PARAMETERS AS NEEDED (SEE PRODUCT *
/**      REFERENCE GUIDE FOR DESCRIPTIONS OF PARAMETERS AND CONTROL *
/**      CARDS) *                          *
/** 3.  SUBMIT THE JOB *                          *
/**                          *                          *
/*******
/**
//VIASAKRU PROC SYSDA='SYSDA',    WORK FILE UNIT NAME
//          SYSOUT='*',          PRINTED OUTPUT MESSAGE CLASS
//          VIASOFT='ASG',       ASG HI-LVL NODES
//          CENTER='VIACENXX',   ASG MIDDLE NODES
//          AKRIN='ASG.VIACENXX.AKR', (DSN FOR INPUT AKR)
//          AKROUT='ASG.VIACENXX.AKR', (DSN FOR OUTPUT AKR)
//          PUNCH='NULLFILE',    (DSN FOR 'PUNCH' FILE)
//          METRIC='NULLFILE',   (DSN FOR METRICS EXPORT FILE)
//          FPA1='NULLFILE',     (DSN FOR FUNCTION POINT EXPORT FILE)
//          FPA2='NULLFILE',     (DSN FOR FUNCTION POINT EXPORT FILE)
//          PERMVOL='XXXXXX'    (VOLUME FOR 'PUNCH' FILE)
/**
//VIAAKRU EXEC PGM=VIASAKRU,REGION=2048K,
//          PARM=''
//STEPLIB DD DSN=&VIASOFT..&CENTER..LOADLIB,DISP=SHR
//VIAAKRIN DD DSN=&AKRIN,DISP=SHR
//VIAAKROT DD DSN=&AKROUT,DISP=SHR
//VIAPUNCH DD DSN=&PUNCH,
//          SPACE=(TRK,(1,1)),DISP=(NEW,CATLG,DELETE),
//          DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB),
//          VOL=SER=&PERMVOL,UNIT=&SYSDA
//VIASCDFM DD DSN=&METRIC,
//          SPACE=(TRK,(1,1)),DISP=(NEW,CATLG,DELETE),
//          DCB=(BLKSIZE=9900,LRECL=300,RECFM=FB),
//          VOL=SER=&PERMVOL,UNIT=&SYSDA
//VIASCDFS DD DSN=&FPA1,
//          SPACE=(TRK,(10,1)),DISP=(NEW,CATLG,DELETE),
//          DCB=(BLKSIZE=9750,LRECL=250,RECFM=FB),
//          VOL=SER=&PERMVOL,UNIT=&SYSDA
//VIASCDFP DD DSN=&FPA2,
//          SPACE=(TRK,(1,1)),DISP=(NEW,CATLG,DELETE),
//          DCB=(BLKSIZE=9750,LRECL=250,RECFM=FB),
//          VOL=SER=&PERMVOL,UNIT=&SYSDA
//VIASARPT DD SYSOUT=&SYSOUT
//VIALOG DD SYSOUT=&SYSOUT
//VIAPRINT DD SYSOUT=&SYSOUT
//VIASYSIN DD DDNAME=SYSIN
//SYSPRINT DD SYSOUT=&SYSOUT
//SYSUDUMP DD SYSOUT=&SYSOUT
/**

```

```
//          PEND
// *
//VIASAKRU EXEC VIASAKRU
// *
//VIAAKRU.VIASYSIN DD *
*
* INSERT ASG AKR UTILITY CONTROL CARDS HERE
*
HELP
// *
```

## 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 can be used per command.

All control cards with command disposition and command summaries are printed to the VIALOG AKR Utility Log file. Comments are entered by placing an asterisk (\*) before the text of the comment. Blank control cards are ignored.

## Command Format

Commands that use member names accept special characters to signify generic names. An asterisk (\*) represents zero or more characters and a question mark (?) represents one character. For example:

Example	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.

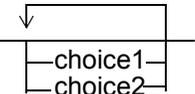
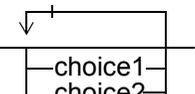
The LASTUsed parameter is used to provide the selection criteria for several commands. The specified number represents the number of days since the member was last referenced online, or the date the member was analyzed if it has not been referenced.

The REPLACE parameter is used to specify that members are to be replaced on the output AKR. The NOREPLACE parameter is used to prevent members from being replaced on the output AKR. NOREPLACE is the default value.

## Command Syntax

Each Batch AKR Utility command is described in this section. These descriptions include the format and a brief explanation of the command parameters. The command syntaxes use these format guidelines:

Item	Description
ABBREVIations	Illustrates the command abbreviation, which is shown in uppercase letters. Lowercase letters in the command are optional.
lowercase	Indicates user-supplied variable information.
UPPERCASE	Indicates commands or keywords.
<b>Bold</b>	Indicates operands that are available only if Center is installed and a Center analysis has been run on the COBOL program being tested.
Underline	Specifies the default value of an operand.
	Separates synonymous commands or operands.
—————>	Indicates that the command syntax is continued on the next line.
—>————	Indicates the command syntax is continued from the previous line.
—————✕	Indicates the end of the command syntax.
— required —	Indicates that the operand or keyword appearing on the main command line is required.
<pre> ┌ choice1 ─┐ ├ choice2 ─┤ └ choice3 ─┘ </pre>	Indicates that one operand is required.
<pre> ┌ optional ─┐ └           ┘ </pre>	Indicates that an operand or keyword appearing below the main command line is optional.
<pre> ┌ choice1 ─┐ ├ choice2 ─┤ └           ┘ </pre>	Indicates that operands are optional.

Item	Description
	Indicates that more than one operand can be chosen.
	Indicates that operands can be concatenated by placing a plus sign (+) between them.

## ANLZSTAT Command

ANLZSTAT ➤

### Function

Produces the Analysis Status report for one or more AKRs submitted with the job. The report provides:

- 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.

### Operands

None. However, the ANLZSTAT command must be followed by one or more AKR dataset names, for example:

```
//VIAAKRU.VIASYSIN DD *
ANLZSTAT
VIAALXX.DEVL.SAMPAKR
ASG.VIACENXX.AKR
```

### Usage Notes

Specify the ANLZSTAT command in the input card for the Batch AKR job, VIASAKRU. The VIAAKRIN file is not referenced by this command. ANLZSTAT must be followed by one or more AKR dataset names, each on a separate line.

A report is written to the DD name VIASARPT. In the default JCL, the DD name is defined as a SYSOUT file. You can 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.

The ANLZSTAT command looks at each component in each AKR submitted in the job. The Analysis Status report is generated for application-level products only (Alliance, Recap, and Estimate). An Analysis Status report is not generated for program-level components (e.g., SmartEdit, SmartTest, Insight, and Encore programs).

### **comment Command**



#### **Function**

Includes a comment with the commands.

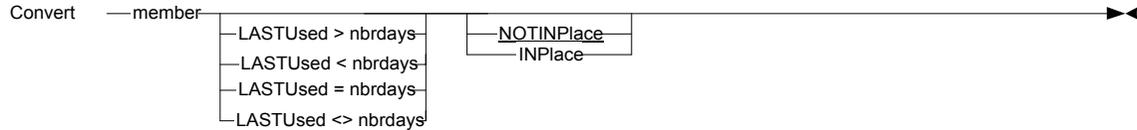
#### **Operands**

**comment.** Provides user-supplied text.

#### **Usage Notes**

Blank control cards are ignored.

## CONVERT Batch AKR Command



### Function

Converts selected members that were analyzed with a prior release of ESW products to the current release level.

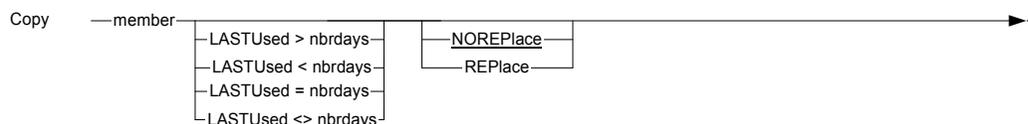
### Operands

Operand	Description
<i>member</i>	Specifies the member name and can be a specific or generic name as described in <a href="#">"Command Format" on page 76</a> . This operand is not valid for Recap.
LASTUsed	Selects members based on the number of days since they were last used, as described in <a href="#">"Command Format" on page 76</a> .
NOTINPlace	Specifies that a new member will be created in the receiving AKR, which is created by converting the contents from the sending AKR. If that member already exists in the receiving AKR, it is not replaced. This is the default.
INPlace	Specifies that a member is to be converted and kept within the AKR named in the VIAAKRIN DD statement. This option should be used with caution. Consult your systems programmer 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 statement, as described in ["Job Control Statements" on page 74](#).

## COPY Batch AKR Command



### Function

Copies selected members from one AKR to another.

### Operands

Operand	Description
<i>member</i>	Specifies the member name. This operand can be a specific or generic name as described in <a href="#">"Command Format" on page 76</a>
LASTUsed	Selects members based on the number of days since they were last used, as described in <a href="#">"Command Format" on page 76</a>
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, as described in ["Job Control Statements" on page 74](#).

If the AKR-DSORG-VSAM installation option parameter is set to NO, copying an existing VSAM AKR converts it to a BDAM AKR.

## DELETE Batch AKR Command

```
DELeTe  —member— |—LASTUsed > nbrdays—|  
              |—LASTUsed < nbrdays—|  
              |—LASTUsed = nbrdays—|  
              |—LASTUsed <> nbrdays—|
```

### Function

Erases selected members from the AKR.

### Operands

Operand	Description
<i>member</i>	Specifies the member name. This operand can be a specific or generic name as described in <a href="#">"Command Format" on page 76</a> .
LASTUsed	Selects members based on the number of days since they were last used, as described in <a href="#">"Command Format" on page 76</a> .

### Usage Notes

Members are deleted from the AKR specified in the VIAAKRIN DD statement.

Members that begin with VIA cannot be deleted using this command. All ESW test members begin with a VIA prefix. If these members must be deleted, use the AKR utility function described in ["Batch AKR Utility" on page 74](#).

## EXPORT Batch AKR Command

EXPort    application        

### Function

Creates metrics and function point CDF files.

### Operands

Operand	Description
<i>application</i>	Specifies the name of the application. This operand can be a specific or generic name as described in <a href="#">"Command Format" on page 76</a> .
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 Batch AKR Command

HELP | ? —————▶

### Function

Prints a description of the Batch AKR Utility and the commands that can be used.

### Operands

None.

### Usage Notes

- A question mark (?) can be used as an alternate command.
- The HELP command has no operands.

The Help report is printed to the SYSOUT specified in the VIAPRINT DD statement, as described in ["Job Control Statements" on page 74](#).

## INIT Batch AKR Command

INIT  

### Function

Initializes a new AKR. This internal command is used by the online AKR Utility Allocation function.

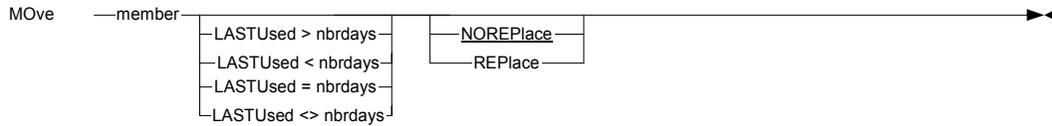
### Operand

**Dsname** (*dsname*). Specifies the dataset name for the new AKR.

### Usage Notes

You must create the AKR dataset to be initialized prior to the initialization. Describe the AKR that is initialized in the VIAAKRIN DD statement. The VIAAKRIN DD statement is ignored if you specify the DSNAME parameter.

## MOVE Batch AKR Command



### Function

Moves selected members from one AKR to another. Specified members are copied to the receiving AKR and erased from the sending AKR.

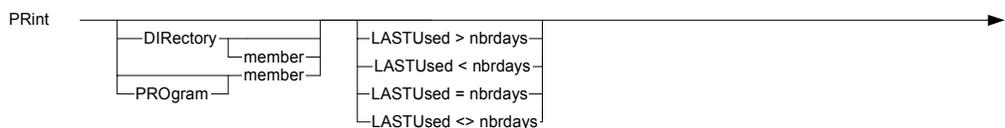
### Operands

Operand	Description
<i>member</i>	Specifies the member name. This operand can be a specific or generic name as described in <a href="#">"Command Format" on page 76</a> .
LASTUsed	Selects members based on the number of days since they were last used, as described in <a href="#">"Command Format" on page 76</a> .
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, as described in ["Job Control Statements" on page 74](#).

## PRINT Batch AKR Command



### Function

Prints AKR directory information for the entire AKR, a specified member, or the source code for a specified member.

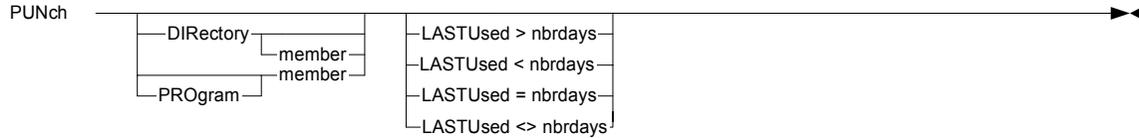
### Operands

Operand	Description
Blank	Prints the AKR directory information if the PRINT Batch AKR command is entered with no operand.
DIRectory	Prints AKR directory information. This is the default. If you specify a member, only the AKR directory information for that member is printed.
PROgram	Prints the COBOL source for the specified AKR member. The source printed is a copy of what is shown on the Source View or Program View screens.
	<p><b>Note:</b></p> <p>The output for the PRINT command does not include the source for the application-level product analyze types.</p>
<i>member</i>	Specifies the member name. This operand can be a specific or generic name as described in <a href="#">"Command Format" on page 76</a> .
LASTUsed	Selects a member based on the number of days since it was last used, as described in <a href="#">"Command Format" on page 76</a> .

### 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, as described in ["Job Control Statements" on page 74](#).

## PUNCH Batch AKR Command



### Function

Produces a file that contains AKR directory information for the entire AKR, a specified member, or the source code for a specified member.

### Operands

Operand	Description
Blank	Produces a file containing the AKR directory information if you enter the PUNCH command with no operand.
DIRectory	Produces a file containing directory information. This is the default. If you specify a member, only the AKR directory information for that member is printed.
PROgram	Produces a file that contains COBOL source code for selected AKR member. The source printed is a copy of what is on the Source View or Program View screens.  <b>Note:</b> _____ The output for the PUNCH command does not include the source for the application-level product analyze types.
<i>member</i>	Specifies the member name. This operand can be a specific or generic name as described in <a href="#">"Command Format" on page 76</a> .
LASTUsed	Selects a member based on the number of days since it was last used, as described in <a href="#">"Command Format" on page 76</a> .

### 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 the member name for COBOL source and AKRDIR $nn$  for directory information, where  $nn$  is a consecutively assigned number.

This is 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	DDMMYYYY
Analyze job name	8	Character
Analyze CPU	4	Character
Analyze product level	8	Character
Last reference date	9	DDMMYYYY
Last reference user ID	8	Character
Last reference CPU	4	Character

## Reports

- AKR Utility Log (see [Figure 13 on page 90](#))
- AKR Utility Directory report (see [Figure 14 on page 90](#))
- File produced by the PUNCH DIRECTORY command (see [Figure 15 on page 91](#))

## AKR Utility Log

The AKR Utility log contains:

- Comments
- Commands
- Completion messages
- Short summary of commands processed

The heading includes the ESW product 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 13 • AKR Utility Log

```

ASG-CENTER-OS RX.X LVLXXX          AKR UTILITY LOG          DDMMYYYY HH:MM:SS Page 1
*****
* PRODUCE A REPORT CONTAINING DIRECTORY INFORMATION FOR ALL *
* MEMBERS OF ASG.SMARTDOC.AKR (VIAAKRIN) THAT HAVE *
* NOT BEEN REFERENCED IN THE LAST 7 DAYS. *
*****
*
PRINT DIRECTORY * LASTUSE > 7
ASG1289I 8 DIRECTORY ENTRIES SUCCESSFULLY PRINTED.

*****
* PRODUCE A FILE CONTAINING DIRECTORY INFORMATION FOR ALL *
* MEMBERS OF ASG.RENAISSA.AKR (VIAAKRIN) THAT HAVE *
* NOT BEEN REFERENCED IN THE LAST 7 DAYS. *
*****
*
PUNCH DIRECTORY * LASTUSE > 7

ASG1290I 8 DIRECTORY ENTRIES SUCCESSFULLY PUNCHED.

ASG1314I *** END OF VIASYSIN ***

ASG-CENTER-OS RX.X LVL000          AKR UTILITY LOG - SUMMARY DDMMYYYY HH:MM:SS Page 2

ASG1301I      8 DIRECTORY ENTRIES PRINTED      0 FAILED.
ASG1302I      8 DIRECTORY ENTRIES PUNCHED     0 FAILED.

ASG1315I *** END OF SUMMARY REPORT ***
    
```

## AKR Utility Directory Report

The AKR Utility Directory report lists the results of a PRINT DIRECTORY command and is produced on the VIAPRINT AKR Utility Print file. The title line contains the ESW product 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 14 • AKR Utility Directory Report

```

ASG-CENTER-OS RX.X LVL000          AKR UTILITY - DIRECTORY DDMMYYYY HH:MM:SS Page 1

AKR: USER.TEST.AKR
Command: PRINT DIRECTORY * LASTUSE > 7

Member      Last      Analyzed      Last Referenced
Name        Lines   Use    Date      Time    Job   CPU   Level   Date      Time    Job   CPU
-----
ACTG0018    40    16    DDMMYYYY HH:MM:SS    ASGA  CPUT  INO30000 DDMMYYYY HH:MM:SS    ASG  CPUT
PYRL0085    17    16    DDMMYYYY HH:MM:SS    ASGA  CPUT  INO30000 DDMMYYYY HH:MM:SS    ASG  CPUT
PYRL0105    17     8    DDMMYYYY HH:MM:SS    ASGA  CPUT  INO30000
SR0005A     493    8     DDMMYYYY HH:MM:SS    ASGA  CPUC  INO30000
W550044     66    12    DDMMYYYY HH:MM:SS    ASGA  CPUC  INO30000 DDMMYYYY HH:MM:SS    ASG  CPUT
XRSCLO70   1714   12    DDMMYYYY HH:MM:SS    ASGA  CPUT  INO30000 DDMMYYYY HH:MM:SS    ASG  CPUT
XRSCLO100   15     9     DDMMYYYY HH:MM:SS    ASGA  CPUC  INO30000
XRSCLO200   41    10    DDMMYYYY HH:MM:SS    ASGA  CPUC  INO30000 DDMMYYYY HH:MM:SS    ASG  CPUT

*** End of Directory Report ***
    
```

## Punch Directory File

The Punch Directory file is created on the VIAPUNCH AKR Utility 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 cards are to be added to a partitioned dataset specified in the NAME parameter. The cards that follow 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.

**Figure 15 • AKR Punch Directory File**

```

./ ADD NAME=AKRDIR1, LIST=ALL
ACTG0018      40  16DDMMYYYYYASGACPUAINO30000DDMMYYYYYASG      CPUA
PYRL0085      17  16DDMMYYYYYASGACPUAINO30000DDMMYYYYYASG      CPUA
PYRL0105      17   8DDMMYYYYYASGACPUAINO30000
SR0005A       493  8DDMMYYYYYASGACPUCINO30000
W55004        66  12DDMMYYYYYASGACPUCINO30000DDMMYYYYYASG      CPUA
XRSCL070     1714 12DDMMYYYYYASGACPUAINO30000DDMMYYYYYASG      CPUA
XRSCL100      15   9DDMMYYYYYASGACPUCINO30000
XRSCL200      41  10DDMMYYYYYASGACPUCINO30000DDMMYYYYYASG      CPUA
./ ENDUP

```



---

# 6

## Analyze Submit Facility

---

This chapter describes the Analyze Submit facility, and contains these sections:

Topic	Page
<a href="#">Introduction</a>	<a href="#">93</a>
<a href="#">Program-level Analyze Input Descriptions</a>	<a href="#">94</a>
<a href="#">The Analyze Process</a>	<a href="#">96</a>
<a href="#">Automatic JCL Modifications for COBOL Procedures</a>	<a href="#">106</a>
<a href="#">Analyze Summary Report</a>	<a href="#">119</a>
<a href="#">Adding Analyze Facilities to a Standard Compile Mechanism</a>	<a href="#">122</a>
<a href="#">Assembler Analyzer (SmartTest Only)</a>	<a href="#">123</a>
<a href="#">Assembler Analyzer Input</a>	<a href="#">123</a>
<a href="#">Assembler Analyze JCL</a>	<a href="#">125</a>

**Note:** \_\_\_\_\_

Information in this chapter applies only to installations of Insight, SmartTest, SmartDoc, AutoChange, and Encore.

---

### Introduction

A program must be analyzed before ESW products can provide intelligent information about it. The analyze process gathers information about the program, including program relationships, logic, data, and execution paths, and stores this information in an AKR. After the analyze information is placed in the AKR, it is available to ESW products in online and Batch environments, where you can access the data to provide valuable information about the design and operation of user systems.

Application-level products (Alliance, Estimate, and Recap) use the Application Definition facility to prepare applications for analysis. Using the Application Definition facility is a process completely different than the process described in this chapter. See the *ASG-Application Definition and Analysis User's Guide* for more information about creating an Application Definition.

## Program-level Analyze Input Descriptions

The analyze process, shown in [Figure 16](#), is an enhancement to an ordinary COBOL compile. The process has three primary inputs:

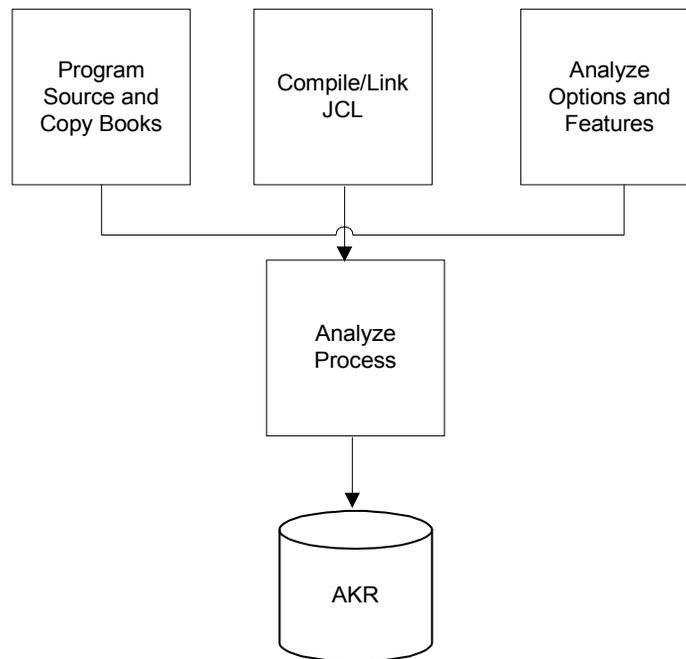
- Source COBOL program (including copybooks).
- JCL used to compile and link the COBOL program.
- Options and features that tailor the analyze steps.

**Note:** \_\_\_\_\_

The analyze does not affect the output load module. It does not add anything extra to the load module.

\_\_\_\_\_

**Figure 16 • Analyze Process**



### **COBOL Source Program**

Like the compiler, the analyze process requires some basic program standards. These are the required standards:

- The COBOL language as specified in the *IBM COBOL II Language Reference Manual* is accepted by the analyze job. It correctly processes any program that can be compiled without warnings or errors by the IBM COBOL II, COBOL/370, COBOL for MVS and VM, or COBOL for OS/390 and VM compilers.
- COBOL II, COBOL/370, COBOL for MVS and VM, and COBOL for OS/390 and VM programs that receive error (E), severe (S), or unrecoverable (U) messages from the IBM compiler cannot be successfully analyzed.
- Object-oriented extensions added to COBOL for MVS and VM and COBOL for OS/390 and VM are not supported.

### **Compile/Link JCL**

This JCL should be the complete JCL used to compile the program. Specifically, the JCL should contain steps to fetch the source from the source manager (such as CA-Librarian or CA-Panvalet), execute the preprocessor, invoke the compiler with the appropriate options and COPY libraries, and invoke the linkage editor.

### **Analyze Options and Features**

The analyze features indicate the type of analysis to be performed:

- An Encore analysis provides the information required for code extraction and execution flow capabilities. Encore analysis provides the ability to analyze and extract code based on the most frequently used re-engineering criteria. Encore allows you to identify, isolate, and extract logical business functions from COBOL programs and save them for later use (e.g., enhancement, reimplementation, and web enablement).
- A SmartTest analysis provides the testing and debugging information required by SmartTest.
- An Insight analysis provides logic and execution flow capabilities.
- An SmartDoc analysis provides the information required for SmartDoc reports.
- An Extended SmartDoc analysis provides data flow analysis.

**Note:** \_\_\_\_\_

Default options for the analyze process are established at installation time. Options that are to be overridden are specified when submitting the analyze job.

---

## The Analyze Process

The analyze process consists of setting up and executing a Batch job. There are three methods used to invoke the analyze process. The method to be used depends primarily on the environment from which the analyze process is invoked, but may depend on the access method containing the compile/link JCL. These are the three methods used to invoke the analyze process:

- Using Prepare Program, Analyze Option, or Command

From the File pull-down on the ESW primary screen, select Prepare Program; from the File pull-down on any ESW component primary screen, select Analyze; or, type ANALYZE from any screen. On the pop-up, enter the required input and output information and submit the job.

- ISPF

On any ISPF screen, execute the VIASUBDS CLIST. This CLIST is executed by typing `TSO VIASUBDS DSN parms`, where *DSN* is a PDS member or sequential dataset containing the compile JCL and *parms* represents any of the available execution parameters described in ["Using ISPF/PDF Edit" on page 100](#).

- ISPF/PDF Edit

Execute the VIASUB PDS edit macro. This edit macro is executed by typing `VIASUB parms`, where *parms* represents any of the available execution parameters described in ["Using ISPF/PDF Edit" on page 100](#).

The methods for executing an analyze job and when to use each method are described in this table:

Compile JCL is From	Method for Executing Analyze Job
PDS or sequential dataset	Analyze Submit screen, VIASUBDS CLIST, or VIASUB edit macro
CA-Librarian, CA-Panvalet, or other user source manager when editing the JCL with ISPF/PDF	VIASUB edit macro
Screen-driven submit facility that generates JCL	VIASUBDS CLIST

## Using the ESW - Prepare Program Pop-up

The ASG-ESW - Prepare Program pop-up, shown in [Figure 17](#), displays when you select File ► Analyze or type ANALYZE on any screen.

These are the defaults for the Analyze features field:

Understand	Y
Test	Y
Extended Analysis	Y
Document	N (except in SmartDoc guides it is Y)
Re-engineer	N (except in Encore guides it is Y)

**Note:** \_\_\_\_\_

See ["Fields" on page 98](#) for more information about these analyze types.

**Figure 17 • ASG-ESW - Prepare Program Pop-up**

```

                                ASG-ESW - Prepare Program
Command ==> _____
                                E - Edit JCL      S - Submit JCL      D - Doc Options

Compile and link JCL (PDS or sequential):
  Data set name 'USER.TEST.CNTL(YOURJCL) '

Analyze features (Y/N):
  Understand: Y  Test: Y  Extended Analysis: Y  Document: N
  Re-engineer: N
AKR data set name 'USER.TEST.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      (Test only)

```

## Options

Option	Description
E - Edit JCL	<p>Edits the JCL. Type <b>E</b> and press Enter to review or change the compile/analyze JCL, if necessary. The JCL to be edited is generated from the JCL member you specify in the Data set name field, applying the rules outlined in the <a href="#">"Automatic JCL Modifications for COBOL Procedures" on page 106</a>. The generated JCL is then displayed on the Edit screen.</p> <p>When you are finished editing, type <b>SUBMIT</b> to submit the edited JCL for execution. Optionally, you can save the edited JCL in a partitioned dataset by using the <b>CREATE</b> command. Otherwise, any changes made at this time are not saved.</p>
S - Submit JCL	<p>Submits the JCL and compile/analyze the specified program. Type <b>S</b> and press Enter. The JCL submitted is generated from the JCL member you specify in the Data set name field, applying the rules outlined in <a href="#">"Automatic JCL Modifications for COBOL Procedures" on page 106</a>.</p>
D - Doc Options	<p>Displays only if SmartDoc is installed on your system. Type <b>D</b> and press Enter to display the File - SmartDoc Options pop-up that is used to request an Extended SmartDoc analysis and to specify which reports (if any) are to be generated.</p>

## Fields

Field	Description
Data set name	<p>Specifies the PDS member or sequential dataset containing the JCL to compile and link the program. If the JCL resides in a source manager such as CA-Librarian or CA-Panvalet, use the VIASUB edit macro to submit the compile/analyze job.</p>
Understand	<p>Displays only if Insight is installed. This type of analysis provides the logic and program execution flow capabilities of Insight. If Insight is the only product installed, this field contains a Y value and cannot be changed. The default is Y.</p>
Test	<p>Displays only if SmartTest is installed. This type of analysis provides the testing and debugging information required by SmartTest. If SmartTest is the only product installed, this field contains a Y value and cannot be changed. The default is Y.</p>

Field	Description
Extended Analysis	Displays only if SmartTest is installed. This type of analysis provides comprehensive program analyzing capabilities for SmartTest FLOW and TRACE commands in addition to the testing and debugging of SmartTest. The default is Y.
Document	Displays only if SmartDoc is installed. This type of analysis provides the report information generated by SmartDoc. If SmartDoc is the only product installed, this field contains a Y value and cannot be changed. The default is N, unless you are accessing this feature from within SmartDoc.
Re-engineer	Displays only if Encore is installed. This type of analysis provides the logic and program execution flow capabilities of Encore. If Encore is the only product installed, this field contains a Y value and cannot be changed. The default is N, unless you are accessing this feature from within Encore.
AKR data set name	Specifies the AKR that will contain the information for the analyzed program.
AKR program name	Specifies an alias name used by the analyze process to save its results in the AKR. If a value is not entered in this field, the analyze job uses the program name from the PROGRAM-ID statement in the COBOL source as the name under which to save results in the AKR.  <b>Note:</b> _____ This field is only used for the AKR program name and does not change the COBOL program name in the source. _____
Analyze options	Specifies analyze options that are to be overridden. Default options for the analyze job are established at installation time. Analyze options that can be entered in this field are described in the user's guide for each ESW product.
Compile?	Indicates whether the program is to be compiled. A program need not be compiled if Insight, Encore, or SmartDoc are the only features specified. You can suppress the compile step by typing N in this field. This field is forced to a value of Y if SmartTest and/or Extended Analysis are selected.
Link load module reusable	Tests a program using SmartTest (non-CICS) that is dynamically loaded and will be tested with RUN NOMONITOR. It is necessary to mark the load module as reusable so that the Breakpoints are retained across calls. The default is Y.

## Using ISPF

Use the VIASUBDS CLIST on any ISPF screen to submit the analyze job. This is the syntax for VIASUBDS:

```
TSO VIASUBDS input.jcl.dsn parms
```

where:

*input.jcl.dsn* is the dataset containing the compile/link JCL. This dataset must be a sequential dataset or a member of a PDS.

*parms* is one or more parameters that control the operation of VIASUBDS. Typically, you enter the PANEL parameter to display the Analyze Submit Parameters screen for entry of any necessary parameters. The parameters are saved in the ISPF profile and used as defaults for the next analyze submission. [Using ISPF/PDF Edit](#) contains a list of these parameters, with default parameters underlined.

**Note:** \_\_\_\_\_

Using the VIASUBDS CLIST requires the ESW CLIST library to be available through the standard SYSPROC allocations.

---

## Using ISPF/PDF Edit

Use the VIASUB edit macro from the ISPF/PDF Edit screen to submit the analyze job. This is the syntax for VIASUB:

```
VIASUB parms
```

where *parms* is one or more parameters that control the operation of VIASUB. Typically, you enter the PANEL parameter to display the Analyze Submit Parameters screen for entry of any necessary parameters. The parameters are saved in the ISPF profile and used as defaults for the next analyze submission.

**Note:** \_\_\_\_\_

Using the VIASUB edit macro requires the ESW CLIST library to be available through the standard SYSPROC allocations.

---

These are the analysis type defaults used in the table:

- INS
- NOENS (except in Encore guides, it is ENS)
- NOSD (except in SmartDoc guides, it is SD)
- ST
- STX

Parameter	Description
AKR(#####)	<p>Indicates the AKR where the results of the analyze job are placed. The specified name must conform to the standard TSO dataset naming conventions. If the name requires quotes, you must use triple quotes. For example:</p> <pre>AKR ( ' ' 'ASG.VIACENxx.AKR' ' ' )</pre>
EN	Specifies that a Encore analysis is to be performed.
AOPT(#####)	<p>Specifies the options to be supplied to the analyze job. The COBOL II option is automatically added if the compiler specified in the input JCL is COBOL II. When specifying more than one analyze option, separate the options with commas and enclose in single quotes. For example:</p> <pre>AOPT ( ' XMEM, RECUR, SUBSYS=D239 ' )</pre>
CMPL NOCMPL	<p>CMPL indicates a COBOL compile and an analysis will be executed by the new JCL. NOCMPL indicates the new JCL is to bypass the compile step and only execute the analyze job. When NOCMPL is specified, a return code of 1000 (decimal) greater than the analyze return code is produced. This causes the subsequent job steps (e.g., a link edit) to be bypassed based on a successful compilation. You cannot specify NOCMPL if you are executing a SmartTest analysis.</p>
DSCHK NODSCHK	<p>DSCHK specifies that datasets needed by the resulting JCL are verified to ensure they exist. Specifically, the AKR and the load library containing VIASMNTR are checked. When NODSCHK is specified, the AKR and the load library need not exist at the time VIASUB or VIASUBDS is executed. NODSCHK is useful when the JCL is being prepared for submission on another system, or for delayed execution when an AKR does not yet exist. Note that the cataloged procedure libraries must exist and be accessible to VIASUBDS or VIASUB.</p>

Parameter	Description
EDIT	<p>EDIT specifies that the resulting JCL is not to be submitted for Batch processing. The PDF editor is invoked for the resulting JCL. Make any changes and type <code>SUBMIT</code> to submit the JCL. You must type <code>EDIT</code> each time it is needed.</p> <p><b>Note:</b></p> <p>Edits made to the JCL are not saved. Use the <code>CREATE</code> command to save the modified JCL elsewhere. The <code>EDIT</code> option is ignored if the Analyze Submit Parameters screen displays. In this case, type <code>E</code> to edit the JCL.</p>
INS NOINS	INS specifies that an Insight analysis is to be performed.
OUTPUT(#####)	Specifies that the resulting JCL is not to be submitted for Batch processing. The JCL is written to the specified dataset. The specified name must conform to the standard TSO dataset naming conventions. A dataset is created if it does not already exist. You must enter <code>OUTPUT</code> each time it is needed.
PANEL NOPANEL	PANEL indicates that the Analyze Submit Parameters screen is to be displayed for entry of parameters for the analyze job. The Analyze Submit Parameters screen displays even if a valid AKR name is specified as a parameter, or can be obtained from the ISPF profile when you specify <code>PANEL</code> .
PGM(#####)	Specifies a name to be used when storing the program in the AKR. This name overrides the program name in the <code>PROGRAM-ID</code> paragraph.
PROONLY	Indicates that the JCL contains only a cataloged procedure rather than a complete job. <code>PROONLY</code> suppresses the generation of the <code>VIAIN DD</code> statement and you must enter it each time it is needed.
<u>REUS</u>   NOREUS	<u>REUS</u> specifies that when the program is tested using SmartTest, it is dynamically loaded and will be tested with <code>RUN NOMONITOR</code> .
EN   <u>NOENS</u>	EN specifies that an Encore analysis is to be performed.
<u>SD</u>   NOSD	<u>SD</u> specifies that a SmartDoc analysis is to be performed.
<u>SDR</u>   NOSDR	<u>SDR</u> specifies that SmartDoc reports will be run.
<u>SDX</u>   NOSDX	<u>SDX</u> specifies that a SmartDoc Extended analysis is to be performed.

Parameter	Description
<u>ST</u>   NOST	ST specifies that a SmartTest analysis is to be performed.
<u>STX</u>   NOSTX	STX specifies that an Extended SmartTest analysis is to be performed. When you specify the INS and ST parameters, an Extended SmartTest analysis is automatically performed.

[Figure 18](#) shows the Analyze Submit Parameters screen. This screen displays when you specify the PANEL parameter when executing VIASUBDS or VIASUB, or when the NOPANEL option is used and an error condition is detected.

Use this screen as you would the ASG-ESW - Prepare Program pop-up described in ["Using the ESW - Prepare Program Pop-up" on page 97](#).

**Figure 18 • Analyze Submit Parameters Screen**

```

ASG-ESW Analyze Submit Parameters
Command ==> _____
      E - Edit JCL      S - Submit JCL      D - Doc Options
Analyze features (Y/N):
  Understand: Y  Test: Y  Extended Analysis: N  Document: N
  Re-engineer: N
AKR data set name 'USER.TEST.AKR'
AKR program name _____ (if overriding PROGRAM-ID)
Analyze options:
_____  

_____  

Compile? (Y/N) . . . . . N      (Y if needed by features)
Link load module reusable? (Y/N) N      (Test only)
Display this panel by default in the future? (Y/N) Y

```

## Options

Option	Description
E - Edit JCL	<p>Displays the compile/analyze JCL. Type E to review or change the JCL, if necessary. The JCL to be edited is generated from the JCL you specify when you invoke the VIASUBDS CLIST or VIASUB edit macro, applying the rules outlined in the <a href="#">"Automatic JCL Modifications for COBOL Procedures" on page 106</a>. The generated JCL then displays on the Edit screen.</p> <p>When you are finished editing, type SUBMIT to submit the edited JCL for execution. Optionally, you can save the edited JCL in a partitioned dataset by using the CREATE command. Otherwise, any changes made at this time are not saved.</p>
S - Submit JCL	<p>Submits the JCL. Type S to submit the JCL and compile/analyze the specified program. The JCL submitted is generated from the JCL you specify when you invoke the VIASUBDS CLIST or VIASUB edit macro, applying the rules outlined in <a href="#">"Automatic JCL Modifications for COBOL Procedures" on page 106</a>.</p>
D - Doc Options	<p>Displays only if SmartDoc is installed on your system. Type D to display the File - SmartDoc Options pop-up that is used to request an Extended SmartDoc analysis and to specify which reports (if any) are to be generated.</p>

## Fields

Field	Description
Understand	<p>Displays only if Insight is installed. This type of analysis provides the logic and program execution flow capabilities of Insight. If Insight is the only product installed, this field contains a Y value and cannot be changed. The default is Y.</p>
Test	<p>Displays only if SmartTest is installed. This type of analysis provides the testing and debugging information required by SmartTest. If SmartTest is the only product installed, this field contains a Y value and cannot be changed. The default is Y.</p>
Extended Analysis	<p>Displays only if SmartTest is installed. This type of analysis provides comprehensive program analyzing capabilities for SmartTest FLOW and TRACE commands in addition to the testing and debugging of SmartTest. The default is Y.</p>

Field	Description
Document	Displays only if SmartDoc is installed. This type of analysis provides the report information generated by SmartDoc. If SmartDoc is the only product installed, this field contains a Y value and cannot be changed. The default is N, unless you are accessing this feature from within SmartDoc.
Re-engineer	Displays only if Encore is installed. This type of analysis provides the logic and program execution flow capabilities of Encore. If Encore is the only product installed, this field contains a Y value and cannot be changed. The default is N, unless you are accessing this feature from within Encore.
AKR data set name	Specifies the AKR that will contain the information for the analyzed program.
AKR program name	Specifies an alias name used by the analyze process to save its results in the AKR. If a value is not entered in this field, the analyze job uses the program name from the PROGRAM-ID statement in the COBOL source as the name under which to save results in the AKR.  <b>Note:</b> This field is only used for the AKR program name and does not change the COBOL program name in the source.
Analyze options	Specifies analyze options that are to be overridden. Default options for the analyze job are established at installation time. Analyze options that can be entered in this field are described in the user's guide for each ESW product.
Compile?	Indicates whether the program will be compiled. A program need not be compiled if Insight, Encore, or SmartDoc are the only features specified. You can suppress the compile step by typing N in this field. This field is forced to a value of Y if SmartTest and/or Extended Analysis are selected.
Link load module reusable	Tests a program using SmartTest (non-CICS) that is dynamically loaded and will be tested with RUN NOMONITOR. It is necessary to mark the load module as reusable so that the Breakpoints are retained across calls. The default is Y.
Display this panel by default in the future?	Determines whether this screen displays whenever subsequent executions of VIASUBDS or VIASUB are invoked. Typing Y causes the ISPF profile to be updated to display this screen. If you specify No, this screen is not displayed unless an error condition is encountered.

## Automatic JCL Modifications for COBOL Procedures

The analysis process automatically modifies the JCL based on the specific parameters and analyze options. If problems arise, and until the problems can be resolved, you can manually code automatic modifications using these examples.

### *To modify the JCL, the compile procedure, or a copy of the compile procedure*

- 1 Replace the PGM= parameter in these compile step(s):

```
PGM=IKFCBL00 PGM=VIACOBVS
PGM=IGYCRCTL PGM=VIACOBII
PGM=CPXUPTSM PGM=VIAOPT3
PGM=CAOTSMON PGM=VIAOPTII
```

- 2 Add these DD statements to the compile step(s) for these ESW datasets:

```
//VIASDTC DD SYSOUT=*
//VIASDRPT DD SYSOUT=*
//VIALOG DD SYSOUT=*
//VIAMRPT DD SYSOUT=*
//VIAPRINT DD SYSOUT=*
//VIAAKR DD DSN=[specified AKR name],DISP=SHR
```

### *Encore DD Statement*

#### *To add an Encore DD statement before the VIASDTC DD statement*

```
//VIAUT2 DD UNIT=SYSDA,SPACE=(vspunit,(vspprim,vspsecd))
```

- 1 If the SYSIN DD statement contains FREE=CLOSE, change it to FREE=END.
- 2 Ensure that the ESW load libraries are available to the modified step by adding a //STEPLIB DD statement specifying the ESW load libraries, or by concatenating these libraries to an existing STEPLIB DD.
- 3 Ensure that the JOB and the modified STEP EXEC statements have a minimum of REGION=4096K.
- 4 Add a VIAIN DD statement that designates the features and options to be used during analysis. This is an example of the DD statement:

```
//VIAIN DD *
* ANALYZE FEATURES:
INS,ST,STX,SD,SDX,SDR,RNS
/*
```

You can also do this step manually by modifying the COBOL parameter string to include the appropriate parameter:

```
VPARM=(vopt,vopt,vopt...)
```

where *vopt* can have these values:

INS	Insight only analysis (no COBOL compile)
ST	SmartTest only analysis (no Extended analysis)
STX	Extended SmartTest analysis
SD	SmartDoc analysis
SDX	XExtended SmartDoc analysis
SDR	SmartDoc report generation
EN	Encore analysis (no COBOL compile)
<i>analyze parms</i>	Valid analyze parameters (using the standard analyze options)
CMPL	COBOL compile (forces a COBOL compile and an analysis to be executed by the JCL)
NOCMPL	Suppress the COBOL compile (JCL bypasses the compile and executes only an analyze job)
(NO)SYSPRINT	Create separate compiler output file
(NO)VIADCOMP	Create SmartDoc intermediate compiler output file. The intermediate compiler output file is used to produce the SmartDoc Compiler Output
DPARM	SmartDoc run-time parameters

**Note:** \_\_\_\_\_

If no ESW analysis feature is specified (i.e., INS, ST, STX, SD, SDX, SDR, or EN), all processing is suppressed and the procedure executes a compile as it did before.

Source managers that do not use JCL for compiles, such as SCLM, Endeavor, and PROTEUS, require that procedures written contain the ESW analyze components mentioned above. A document on SCLM ESW enhancement suggestions may be obtained from the ASG Service Desk.

\_\_\_\_\_

These examples illustrate compile/analyze JCL before and after submitting the analyze job for Panvalet, CICS, CA-IDMS, and DB2.

## CA-Panvalet

This is an example of compile JCL with CA-Panvalet as it might appear in a dataset at your site.

```
// ASG JOB (ASG), 'CA-PANVALET COMPILE'
/*ROUTE PRINT DEST
/* PANVALET EXTRACT
/*
//PANEXT EXEC PGM=PAN#1, REGION=256K
//PANDD1 DD DSN=ASG.COBOL.PANLIB, DISP=SHR
//PANDD2 DD DSN=&&COBIN, UNIT=SYSDA, SPACE=(CYL, (1,1)),
// DISP=(NEW, PASS), DCB=(RECFM=FB, LRECL=80, BLKSIZE=3120)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
++WRITE WORK, VIADDDMO
/*
/*
/* COBOL COMPILE
/*
//COBCOMP EXEC PGM=IKFCBL00, REGION=1024K, COND=(8, LT, PANEXT),
// PARM='SIZE=512K, BUF=128K, LANGLVL(2), LIB, DYNAM'
//STEPLIB DD DSN=SYS1.COBOLII.COMPIER, DISP=SHR
//SYSIN DD DSN=&&COBIN, DISP=(OLD, DELETE)
//SYSLIB DD DSN=ASG.COBOL.COPYLIB, DISP=SHR
//SYSLIN DD DSN=&&LINKIN, UNIT=SYSDA, SPACE=(CYL, (1,1)),
// DISP=(NEW, PASS), DCB=(RECFM=FB, LRECL=80, BLKSIZE=3120)
//SYSPRINT DD SYSOUT=*, DCB=(RECFM=FBA, LRECL=121, BLKSIZE=1573)
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL, (1,1))
//SYSUT2 DD UNIT=SYSDA, SPACE=(CYL, (1,1))
//SYSUT3 DD UNIT=SYSDA, SPACE=(CYL, (1,1))
//SYSUT4 DD UNIT=SYSDA, SPACE=(CYL, (1,1))
//SYSUT5 DD UNIT=SYSDA, SPACE=(CYL, (1,1))
/*
/* LINK EDIT
/*
//LINKED EXEC PGM=IEWL, REGION=1024K, COND=(8, LT, COBCOMP)
//SYSLIB DD DSN=SYS1.COBOLII.COBLIB, DISP=SHR
//SYSLMOD DD DSN=ASG.BAR.LOAD, DISP=OLD
//SYSPRINT DD SYSOUT=*, DCB=(RECFM=FBA, LRECL=121, BLKSIZE=1573)
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL, (1,1))
//SYSLIN DD DSN=&LINKIN, DISP=(OLD, DELETE)
// DD *
NAME VIADDDMO(R)
/*
```

This is an example of the compile JCL as it would appear after the CA-Panvalet/compile/analyze JCL has been generated according to the rules in this section. Statements that have been added or modified are tagged to the right with ASG NEW and ASG MOD.

```

//ASG JOB ( ),'VIASANLZ'
//*   INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
//*****
//* THIS JCL HAS BEEN MODIFIED BY THE ASG ANALYZE *
//* SUBMIT FACILITY, WHICH CONVERTS COMPILE JCL INTO *
//* COMPILE AND ANALYZE JCL. NEW OR MODIFIED LINES *
//* CONTAIN 'VIA' IN COLUMNS 74 THROUGH 76. *
//*****
//VIAIN EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT2 DD DSN=&&VIAIN,DISP=(,PASS),UNIT=SYSDA,
//          SPACE=(TRK,(1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440)
//SYSUT1 DD *
* ANALYZE FEATURES:
SD,SDX,SDR
* ANALYZE OPTIONS:
NORET=(ABENDPGM),SEQ
* CMPL TO PERFORM THE COMPILE STEP:
CMPL
* OPTIONS FOR TREATMENT OF COMPILER LISTING:
VIADCOMP,NOSYSPRINT
/*
//* CA-PANVALET EXTRACT
//*
//PANEXT EXEC PGM=VIASPAN1,REGION=4096K
//PANDD1 DD DSN=ASG.COBOL.PANLIB,DISP=SHR
//PANDD2 DD DSN=&&COBIN,UNIT=SYSDA,SPACE=(CYL,(1,1)),
//          DISP=(NEW,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
++WRITE WORK,VIADDDMO
/*
//STEPLIB DD DSN=ASG.VIACENxx.LOADLIB,
//          DISP=SHR,DCB=BLKSIZE=23476
//          DD DSN=DB2TEST.DSNLOAD,DISP=SHR
//VIAINCLS DD DSN=&&VIAINCLS,DISP=(MOD,PASS),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//VIALOG DD SYSOUT=*
//VIAMRPT DD SYSOUT=*
//VIAPRINT DD SYSOUT=*
//*
//* COBOL COMPILE
//*
//COBCOMP EXEC PGM=VIACOBVS,REGION=4096K,COND=(8,LT,PANEXT),
//          PARM='SIZE=512K,BUF=128K,LANGLVL(2),LIB,DYNAM'
//STEPLIB DD DSN=SYS1.COBOLII.COMPIILER,DISP=SHR
//          DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR
//          DD DSN=DB2TEST.DSNLOAD,DISP=SHR
//SYSIN DD DSN=&&COBIN,DISP=(OLD,DELETE)
//SYSLIB DD DSN=ASG.COBOL.COPYLIB,DISP=SHR
//SYSLIN DD DSN=&&LINKIN,UNIT=SYSDA,SPACE=(CYL,(1,1)),
//          DISP=(NEW,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)

```

---

## ASG-Center Installation Guide

```
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1573)
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT5 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//VIAINCLS DD DSN=&&VIAINCLS,DISP=(MOD,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(1,1))
//VIASDTC DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7476)
//VIASDRPT DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7476)
//VIALOG DD SYSOUT=*
//VIAMRPT DD SYSOUT=*
//VIAPRINT DD SYSOUT=*
//VIAAKR DD DSN=ASG.VIACENxx.AKR,DISP=SHR
//VIAIN DD DSN=&&VIAIN,DISP=(OLD,PASS)
//*
//* LINK EDIT
//*
//LINKED EXEC PGM=IEWL,REGION=1024K,COND=(8,LT,COBCOMP)
//SYSLIB DD DSN=SYS1.COBOLII.COBLIB,DISP=SHR
//SYSLMOD DD DSN=ASG.UTIL.LOAD,DISP=OLD
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1573)
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIN DD DSN=&LINKIN,DISP=(OLD,DELETE)
// DD *
NAME VIADDDMO(R)
/*
```

## CICS

This is an example of compile JCL with CICS as it might appear in a dataset at your site.

```
//ASG JOB (ASG),'CICS COBOL '
//*
/* *****
/* * DFHEITCL PROC INVOCATION *
/* * *
/* * STANDARD CICS COBOL COMMAND LEVEL PROCEDURE FOR *
/* * TRANSLATING, COMPILING AND LINK EDITING SOURCE. *
/* * *****
/*
//COBOLC EXEC DFHEITCL,
// PARM.LKED='LET'
//TRN.SYSIN DD DSN=ASG.VIACENxx.CNTL(VIACDEMO),DISP=SHR
//LKED.SYSLMOD DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR
//LKED.SYSIN DD *
NAME VIACDEMO(R)
/*
```

These are examples of the compile JCL above, as it would appear after the CICS/compile/analyze JCL has been generated according to the rules in this section. Statements that have been added or modified are tagged to the right with ASG NEW and ASG MOD. Similar additions and modifications are made when DB2 and IDMS precompilers are used.

```

//ASG JOB ( ),'VIASANLZ'
//*   INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
//*****
//* THIS JCL HAS BEEN MODIFIED BY THE ASG ANALYZE *
//* SUBMIT FACILITY, WHICH CONVERTS COMPILE JCL INTO *
//* COMPILE AND ANALYZE JCL. NEW OR MODIFIED LINES *
//* CONTAIN 'VIA' IN COLUMNS 74 THROUGH 76. *
//*****
//VIAIN EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT2 DD DSN=&&VIAIN,DISP=(,PASS),UNIT=SYSDA,
//          SPACE=(TRK,(1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440)
//SYSUT1 DD *
* ANALYZE FEATURES:
SD,SDX,SDR
* Cmpl TO PERFORM THE COMPILE STEP:
Cmpl
* OPTIONS FOR TREATMENT OF COMPILER LISTING:
VIADCOMP,NOSYSPRINT
/*
/*
/*          *****
/*          * DFHEITCL PROC INVOCATION *
/*          * * *
/*          * STANDARD CICS COBOL COMMAND LEVEL PROCEDURE FOR *
/*          * TRANSLATING, COMPILING AND LINK EDITING SOURCE. *
/*          *****
/*
//DFHEITCL PROC SUFFIX=1$,
// INDEX='CICS311',
// INDEX2='CICS311',
//      OUTC=A,
//      REG=4096K,
//      LNKPARM='XREF',
//      WORK=SYSDA,VIAPGMA=VIACICS
/*
//TRN EXEC PGM=&VIAPGMA,
//      REGION=&REG
//STEPLIB DD DSN=&INDEX2..LOADLIB,DISP=SHR,DCB=BLKSIZE=32760
//      DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR
//      DD DSN=DB2TEST.DSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=&OUTC
//SYSPUNCH DD DSN=&&SYSCIN,
//      DISP=(,PASS),UNIT=&WORK,
//      DCB=BLKSIZE=400,
//      SPACE=(400,(400,100))
//VIATIN DD DSN=&&VIATIN,DISP=(MOD,PASS),SPACE=(CYL,(1,1)),
//      UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440)
//VIAACS DD DSN=&&VIAACS,DISP=(MOD,PASS),UNIT=SYSDA,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440),
//      SPACE=(TRK,(1,1))

```

## ASG-Center Installation Guide

```

//VIALOG DD SYSOUT=* ASG NEW
//VIAMRPT DD SYSOUT=* ASG NEW
//VIAPRINT DD SYSOUT=* ASG NEW
//VIAPGM DD DSN=&&DFHECP&SUFFIX, DISP=(NEW,DELETE), ASG NEW
// UNIT=SYSDA,SPACE=(TRK,(1,1)) ASG NEW
//ASG JOB ( ),'VIASANLZ'
//* INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
//***** ASG NEW
//* THIS JCL HAS BEEN MODIFIED BY THE ASG ANALYZE * ASG NEW
//* SUBMIT FACILITY, WHICH CONVERTS COMPILE JCL INTO * ASG NEW
//* COMPILE AND ANALYZE JCL. NEW OR MODIFIED LINES * ASG NEW
//* CONTAIN 'VIA' IN COLUMNS 74 THROUGH 76. * ASG NEW
//***** ASG NEW
//VIAIN EXEC PGM=IEBGENER ASG NEW
//SYSIN DD DUMMY ASG NEW
//SYSPRINT DD DUMMY ASG NEW
//SYSUT2 DD DSN=&&VIAIN,DISP=(,PASS),UNIT=SYSDA, ASG NEW
// SPACE=(TRK,(1,1)), ASG NEW
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440) ASG NEW
//SYSUT1 DD * ASG NEW
* ANALYZE FEATURES: ASG NEW
SD,SDX,SDR ASG NEW
* Cmpl TO PERFORM THE COMPILE STEP: ASG NEW
Cmpl ASG NEW
* OPTIONS FOR TREATMENT OF COMPILER LISTING: ASG NEW
VIADCOMP,NOSYSPRINT ASG NEW
/* ASG NEW
//*
//* *****
//* * DFHEITCL PROC INVOCATION *
//* * *
//* * STANDARD CICS COBOL COMMAND LEVEL PROCEDURE FOR *
//* * TRANSLATING, COMPILING AND LINK EDITING SOURCE. *
//* *****
//*
//DFHEITCL PROC SUFFIX=1$,
// INDEX='CICS311',
// INDEX2='CICS311',
// OUTC=A,
// REG=4096K, ASG MOD
// LNKPARM='XREF',
// WORK=SYSDA,VIAPGMA=VIACICS ASG MOD
//*
//TRN EXEC PGM=&VIAPGMA, ASG MOD
// REGION=&REG
//STEPLIB DD DSN=&INDEX2..LOADLIB,DISP=SHR,DCB=BLKSIZE=32760 ASG MOD
// DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR ASG NEW
// DD DSN=DB2TEST.DSNLOAD,DISP=SHR ASG NEW
//SYSPRINT DD SYSOUT=&OUTC
//SYSPUNCH DD DSN=&&SYSCIN,
// DISP=(,PASS),UNIT=&WORK,
// DCB=BLKSIZE=400,
// SPACE=(400,(400,100))
//VIATIN DD DSN=&&VIATIN,DISP=(MOD,PASS),SPACE=(CYL,(1,1)), ASG NEW
// UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440) ASG NEW
//VIAACS DD DSN=&&VIAACS,DISP=(MOD,PASS),UNIT=SYSDA, ASG NEW
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440), ASG NEW
// SPACE=(TRK,(1,1)) ASG NEW

```

```

//VIALOG DD SYSOUT=* ASG NEW
//VIAMRPT DD SYSOUT=* ASG NEW
//VIAPRINT DD SYSOUT=* ASG NEW
//VIAPGM DD DSN=&&DFHECP&SUFFIX, DISP=(NEW, DELETE), ASG NEW
// UNIT=SYSDA, SPACE=(TRK, (1, 1)) ASG NEW

```

**Note:**

The COB (COBOL Compiler) Step parameters were specified in the original DFHEITCL PROC. For SmartTest, use of the OPTIMIZE parameter is not recommended. Consult the *ASG-SmartTest Installation Guide* for more information.

**IDMS**

This is an example of compile JCL with IDMS as it might appear in a dataset at your site.

```

//ASG JOB ( ), 'ASG INSTALL'
//* INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
//IDMS PROC MEMBER=XXXXX,
// SRCLIB='IDMS.PROD10.SRCLIB',
// LOADLIB='ASG.UTIL.LOAD'
//*
//PREP EXEC PGM=IDMSDMLC, REGION=4096K
//STEPLIB DD DSN=IDMS.PROD10.LOADLIB, DISP=SHR
//SYSLST DD SYSOUT=*
//SYSPCH DD DSN=&&DMLC, UNIT=SYSDA, DISP=(NEW, PASS),
// SPACE=(TRK, (5, 5)), DCB=(RECFM=FB, LRECL=80, BLKSIZE=3120)
//SYSCTL DD DSN=IDMS.PROD10.SYSCTL, DISP=SHR
//SYSIPT DD DSN=&SRCLIB (&MEMBER), DISP=SHR
//J1JRNL DD DUMMY
//J2JRNL DD DUMMY
//*
//COB EXEC PGM=IKFCBL00,
// REGION=2048K,
// PARM='SIZE=512K, BUF=128K, LANGLVL(2), LIB, SOURCE, APOST'
//STEPLIB DD DSN=SYS1.COBOLII.COMPIILER, DISP=SHR
//SYSPRINT DD SYSOUT=*, DCB=(RECFM=FBA, LRECL=121, BLKSIZE=1573)
//SYSLIN DD DSN=&&SYSLIN, UNIT=SYSDA,
// DISP=(MOD, PASS), SPACE=(CYL, (1, 1)),
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=3120)
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
//SYSUT2 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
//SYSUT3 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
//SYSUT4 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
//SYSUT5 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
//SYSIN DD DSN=&&DMLC, DISP=(OLD, DELETE)
//SYSLIB DD DSN=IDMS.PROD10.SRCLIB, DISP=SHR
//*
//LINK EXEC PGM=IEWL,
// COND=(5, LT, COB),
// REGION=512K,
// PARM='LIST, MAP, CALL'
//SYSLIN DD DSN=&&SYSLIN, DISP=(OLD, DELETE)
// DD DDNAME=SYSIN
//SYSLIB DD DSN=IDMS.PROD10.LOADLIB, DISP=SHR
// DD DSN=SYS1.COBOLII.COBLIB, DISP=SHR
//SYSLMOD DD DSN=&LOADLIB (&MEMBER), DISP=SHR
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))

```

## ASG-Center Installation Guide

```
//SYSPRINT DD SYSOUT=*
//*
//          PEND
//*
//EMPINQ EXEC IDMS, MEMBER=EMPINQ
//*
//LINK.SYSIN DD *
//          INCLUDE SYSLIB (IDMS, IDMSCANC)
//          NAME EMPINQ (R)
```

This is an example of the compile JCL above as it appears after the IDMS/compile/analyze JCL has been generated according to the rules in this section. Statements that have been added or modified are tagged to the right with ASG NEW and ASG MOD.

```
//ASG JOB (ASG), 'ASG '
//* INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//***** ASG NEW
//* THIS JCL HAS BEEN MODIFIED BY THE ASG ANALYZE * ASG NEW
//* SUBMIT FACILITY, WHICH CONVERTS COMPILE JCL INTO * ASG NEW
//* COMPILE AND ANALYZE JCL. NEW OR MODIFIED LINES * ASG NEW
//* CONTAIN 'ASG' IN COLUMNS 74 THROUGH 76. * ASG NEW
//***** ASG NEW
//*
//VIAIN EXEC PGM=IEBGENER ASG NEW
//SYSIN DD DUMMY ASG NEW
//SYSPRINT DD DUMMY ASG NEW
//SYSUT2 DD DSN=&&VIAIN, DISP=(, PASS), UNIT=SYSDA, ASG NEW
//          SPACE=(TRK, (1, 1)), ASG NEW
//          DCB=(RECFM=FB, LRECL=80, BLKSIZE=7440) ASG NEW
//SYSUT1 DD * ASG NEW
* ANALYZE FEATURES: ASG NEW
INS, ST ASG NEW
* ANALYZE OPTIONS: ASG NEW
NORET=(ABENDPGM) ASG NEW
/* ASG NEW
//IDMS PROC MEMBER=XXXXX,
//          SRCLIB='IDMS.PROD10.SRCLIB',
//          LOADLIB='ASG.UTIL.LOAD'
//*
//PREP EXEC PGM=VIASIDMS, REGION=4096K ASG MOD
//STEPLIB DD DSN=IDMS.PROD10.LOADLIB, DISP=SHR
//          DD DSN=ASG.VIACENxx.LOADLIB, DISP=SHR ASG NEW
//SYSLST DD SYSOUT=*
//SYSPCH DD DSN=&&DMLC, UNIT=SYSDA, DISP=(NEW, PASS),
//          SPACE=(TRK, (5, 5)), DCB=(RECFM=FB, LRECL=80, BLKSIZE=3120)
//SYSCTL DD DSN=IDMS.PROD10.SYSCTL, DISP=SHR
//SYSIPT DD DSN=&SRCLIB (&MEMBER), DISP=SHR
//J1JRNL DD DUMMY
//J2JRNL DD DUMMY
//VIATIN DD DSN=&&VIATIN, DISP=(MOD, PASS), UNIT=SYSDA, ASG NEW
//          SPACE=(CYL, (1, 1)), DCB=(RECFM=FB, LRECL=80, BLKSIZE=7440) ASG NEW
//VIAACS DD DSN=&&VIAACS, DISP=(MOD, PASS), UNIT=SYSDA, ASG NEW
//          DCB=(RECFM=FB, LRECL=80, BLKSIZE=7440), ASG NEW
//          SPACE=(TRK, (1, 1)) ASG NEW
//VIALOG DD SYSOUT=* ASG NEW
//VIAMRPT DD SYSOUT=* ASG NEW
//VIAPRINT DD SYSOUT=* ASG NEW
//*
```

```

//COB      EXEC  PGM=VIACOBVS,                               ASG MOD
//          REGION=4096K,                                   ASG MOD
//          PARM='SIZE=512K,BUF=128K,LANGLVL(2),LIB,SOURCE,APOST'
//STEPLIB  DD   DSN=SYS1.COBOLII.COMPIILER,DISP=SHR
//          DD   DSN=ASG.VIACENxx.LOADLIB,DISP=SHR          ASG NEW
//SYSPRINT DD   SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1573)
//SYSLIN   DD   DSN=&&SYSLIN,UNIT=SYSDA,
//          DISP=(MOD,PASS),SPACE=(CYL,(1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT5   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN    DD   DSN=&&DMLC,DISP=(OLD,DELETE)
//SYSLIB   DD   DSN=IDMS.PROD10.SRCLIB,DISP=SHR
//VIAACS   DD   DSN=&&VIAACS,DISP=(MOD,DELETE),UNIT=SYSDA,  ASG NEW
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440),          ASG NEW
//          SPACE=(TRK,(1,1))                                ASG NEW
//VIALOG    DD   SYSOUT=*                                   ASG NEW
//VIAMRPT   DD   SYSOUT=*                                   ASG NEW
//VIAPRINT  DD   SYSOUT=*                                   ASG NEW
//VIAAKR    DD   DSN=ASG.VIACENxx.AKR,DISP=SHR             ASG NEW
//VIAIN     DD   DSN=&&VIAIN,DISP=(OLD,PASS)                 ASG NEW
//*
//LINK     EXEC  PGM=IEWL,
//          COND=(5,LT,COB),
//          REGION=512K,
//          PARM='LIST,MAP,CALL'
//SYSLIN   DD   DSN=&&SYSLIN,DISP=(OLD,DELETE)
//          DD   DDNAME=SYSIN
//SYSLIB   DD   DSN=IDMS.PROD10.LOADLIB,DISP=SHR
//          DD   DSN=SYS1.COBOLII.COBLIB,DISP=SHR
//SYSLMOD  DD   DSN=&LOADLIB(&MEMBER),DISP=SHR
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT DD   SYSOUT=*
//*
//          PEND
//*
//EMPINQ   EXEC  IDMS,MEMBER=EMPINQ
//*
//COB.VIATIN DD   DSN=&&VIATIN,DISP=(MOD,DELETE),UNIT=SYSDA,  ASG NEW
//          SPACE=(CYL,(1,1)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440)  ASG NEW
//*
//LINK.SYSIN DD *
//          INCLUDE SYSLIB(IDMS,IDMSCANC)
//          NAME EMPINQ(R)

```

**DB2**

This example illustrates the compile JCL with DB2 as it might appear at your site.

```

//ASG JOB ( ),'ASG INSTALL'
//*   INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
//VIAPCII PROC SYSOUT='*',
//          SYSDA='SYSDA',           WORK FILE UNIT NAME
//          COMPILR='IGYCRCTL',     COBOL COMPILER NAME
//          COBLIB='SYS1.COB2LIB',  COBOL II RUN TIME SUBROUTINES
//          DYNAM=DYNAM             DYNAM OR NODYNAM
//*

```

---

## ASG-Center Installation Guide

```
//DB2      EXEC PGM=DSNHPC,REGION=2048K,COND=(4,LT),
//          PARM='HOST(COB2),APOST,STDSQL(NO)'
//STEPLIB  DD DSN=DB2TEST.DSNLOAD,DISP=SHR
//SYSJIN   DD DSN=&SOURCE,UNIT=SYSDA,DISP=(MOD,PASS),
//          SPACE=(CYL,(1,1)),DCB=LRECL=80
//*
//*****
//DBRMLIB  DD DSN=ASG.SQL.CNTL(DB2DBRM),DISP=SHR
//*****
//*
//SYSLIB   DD DSN=DSN220.SRCLIB.DATA,DISP=SHR
//          DD DSN=DSN220.DSNSAMP,DISP=SHR
//          DD DSN=ASG.SQL.COBOL,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,DISP=(,PASS),SPACE=(CYL,(1,1))
//SYSUT2   DD UNIT=SYSDA,DISP=(,PASS),SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=*
//SYSTEM   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSJIN   DD DSN=SRENJLC.REL.COBOL(DBDELETE),DISP=SHR
//*
//COB2     EXEC PGM=&COMPILR,TIME=800,
//          PARM='LIB,APOST,SIZE=512K,BUF=128K'
//STEPLIB  DD DSN=COB2.V320.COB2COMP,DISP=SHR
//SYSJIN   DD DSN=&SOURCE,DISP=(OLD,DELETE)
//SYSLIB   DD DSN=ASG.SQL.COBOL,DISP=SHR
//SYSPRINT DD SYSOUT=&SYSOUT
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3458)
//SYSJIN   DD DSN=&&SYSJIN,DISP=(MOD,PASS),
//          UNIT=&SYSDA,SPACE=(CYL,(1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUT1   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT2   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT3   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT4   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT5   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT6   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT7   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT8   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT9   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//*
//LINK     EXEC PGM=IEWL,
//          PARM='LIST,MAP,CALL,AMODE(31),RMODE(ANY)',
//          COND=(5,LT,COB2),
//          REGION=512K
//SYSJIN   DD DSN=&&SYSJIN,DISP=(OLD,DELETE)
//          DD DDNAME=SYSJIN
//SYSJMOD  DD DSN=ASG.UTIL.LOAD,DISP=SHR
//SYSLIB   DD DSN=DB2TEST.DSNLOAD,DISP=SHR
//          DD DSN=&COBLIB,DISP=SHR
//SYSUT1   DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=&SYSOUT
//*
//          PEND
//*
//CTXP01   EXEC VIAPCII
//          NAME CTXP01(R)
```

This is an example of the compile JCL above as it appears after the DB2/compile/analyze JCL has been generated according to the rules in this section. Statements that have been added or modified are tagged to the right with ASG NEW and ASG MOD.

```
//ASG JOB ( ),'ASG INSTALL'
//* INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
/*****
/* THIS JCL HAS BEEN MODIFIED BY THE ASG ANALYZE * ASG NEW
/* SUBMIT FACILITY, WHICH CONVERTS COMPILE JCL INTO * ASG NEW
/* COMPILE AND ANALYZE JCL. NEW OR MODIFIED LINES * ASG NEW
/* CONTAIN 'ASG' IN COLUMNS 74 THROUGH 76. * ASG NEW
/***** ASG NEW
//*
//VIAIN EXEC PGM=IEBGENER ASG NEW
//SYSIN DD DUMMY ASG NEW
//SYSPRINT DD DUMMY ASG NEW
//SYSUT2 DD DSN=&&VIAIN,DISP=(,PASS),UNIT=SYSDA, ASG NEW
// SPACE=(TRK,(1,1)), ASG NEW
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440) ASG NEW
//SYSUT1 DD * ASG NEW
* ANALYZE FEATURES: ASG NEW
INS,ST ASG NEW
* ANALYZE OPTIONS: ASG NEW
NORET=(ABENDPGM) ASG NEW
/* ASG NEW
//VIAPCII PROC SYSOUT='*',
// SYSDA='SYSDA', WORK FILE UNIT NAME
// COMPILE='IGYCRCTL', COBOL COMPILER NAME
// COBLIB='SYS1.COB2LIB', COBOL II RUN TIME SUBROUTINES
// DYNAM=DYNAM,VIAPGMA=VIACOBII M OR NODYNAM ASG MOD
/*
//DB2 EXEC PGM=VIASQL,REGION=4096K,COND=(4,LT), ASG MOD
// PARM='HOST(COB2),APOST,STDSQL(NO)'
//STEPLIB DD DSN=DB2TEST.DSNLOAD,DISP=SHR
// DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR ASG NEW
//SYSCIN DD DSN=&SOURCE,UNIT=SYSDA,DISP=(MOD,PASS),
// SPACE=(CYL,(1,1)),DCB=LRECL=80
/*
/*****
//DBRMLIB DD DSN=ASG.SQL.CNTL(DB2DBRM),DISP=SHR
/*****
/*
//SYSLIB DD DSN=DSN220.SRCLIB.DATA,DISP=SHR
// DD DSN=DSN220.DSNSAMP,DISP=SHR
// DD DSN=ASG.SQL.COBOL,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(CYL,(1,1))
//SYSUT2 DD UNIT=SYSDA,DISP=(,PASS),SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD DSN=ASG.VIACENxx.COBOL(DBDELETE),DISP=SHR
//VIATIN DD DSN=&&VIATIN,DISP=(MOD,PASS),UNIT=SYSDA, ASG NEW
// SPACE=(CYL,(1,1)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440) ASG NEW
//VIAACS DD DSN=&&VIAACS,DISP=(MOD,PASS),UNIT=SYSDA, ASG NEW
// SPACE=(RECFM=FB,LRECL=80,BLKSIZE=7440), ASG NEW
// SPACE=(TRK,(1,1)) ASG NEW
//VIAINCLS DD DSN=&&VIAINCLS,DISP=(MOD,PASS), ASG NEW
// UNIT=SYSDA,SPACE=(CYL,(1,1)) ASG NEW
//VIALOG DD SYSOUT=* ASG NEW
//VIAMRPT DD SYSOUT=* ASG NEW
```

## ASG-Center Installation Guide

```

//VIAPRINT DD SYSOUT=*                                ASG NEW
//*
//COB2 EXEC PGM=&VIAPGMA,TIME=800,                    ASG MOD
// PARM='LIB,APOST,SIZE=512K,BUF=128K,REGION=4096K'  ASG MOD
//STEPLIB DD DSN=COB2.V320.COB2COMP,DISP=SHR,DCB=BLKSIZE=23476 ASG MOD
// DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR              ASG NEW
//SYSIN DD DSN=&SOURCE,DISP=(OLD,DELETE)
//SYSLIB DD DSN=ASG.SQL.COBOL,DISP=SHR
//SYSPRINT DD SYSOUT=&SYSOUT
//* DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3458)
//SYSLIN DD DSN=&&SYSLIN,DISP=(MOD,PASS),
// UNIT=&SYSDA,SPACE=(CYL,(1,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUT1 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT2 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT3 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT5 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT6 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT7 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT8 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSUT9 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//VIAACS DD DSN=&VIAACS,DISP=(MOD,DELETE),UNIT=SYSDA, ASG NEW
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440), ASG NEW
// SPACE=(TRK,(1,1)) ASG NEW
//VIALOG DD SYSOUT=* ASG NEW
//VIAMRPT DD SYSOUT=* ASG NEW
//VIAPRINT DD SYSOUT=* ASG NEW
//VIAAKR DD DSN=ASG.VIACENxx.AKR,DISP=SHR ASG NEW
//VIAIN DD DSN=&&VIAIN,DISP=(OLD,PASS) ASG NEW
//VIAPGM DD DSN=&&&COMPILR,DISP=(NEW,DELETE), ASG NEW
// UNIT=SYSDA,SPACE=(TRK,(1,1)) ASG NEW
//*
//LINK EXEC PGM=IEWL,
// PARM='LIST,MAP,CALL,AMODE(31),RMODE(ANY)',
// COND=(5,LT,COB2),
// REGION=512K
//SYSLIN DD DSN=&&SYSLIN,DISP=(OLD,DELETE)
// DD DDNAME=SYSIN
//SYSLMOD DD DSN=ASG.UTIL.LOAD,DISP=SHR
//SYSLIB DD DSN=DB2TEST.DSNLOAD,DISP=SHR
// DD DSN=&COBLIB,DISP=SHR
//SYSUT1 DD UNIT=&SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=&SYSOUT
//*
// PEND
//*
//*****
//*
//CTXP01 EXEC VIAPCII
//COB2.VIATIN DD DSN=&&VIATIN,DISP=(MOD,DELETE),UNIT=SYSDA, ASG NEW
// SPACE=(CYL,(1,1)),DCB=(RECFM=FB,LRECL=80,BLKSIZE=7440) ASG NEW
//COB2.VIAINCLS DD DSN=&&VIAINCLS,DISP=(MOD,DELETE), ASG NEW
// UNIT=SYSDA,SPACE=(CYL,(1,1)) ASG NEW
//*
//LINK.SYSIN DD *
NAME CTXP01(R)

```

## Analyze Summary Report

Information about the analyzed program is placed in the AKR when the analyze job completes. A summary report of the run-time statistics and diagnostic messages is also produced. This report varies depending on whether you specified the SOURCE or NOSOURCE option when you submitted the analyze job.

[Figure 19](#) is an example of the Analyze Summary report with the SOURCE option used. The information shown on this summary is described below.

**Figure 19 • Analyze Summary Report**

```

(A)
00001 000100 IDENTIFICATION DIVISION.
00002 000200 PROGRAM-ID. VIADDDMO.
00003 000300 AUTHOR. WRITTEN BY ASG IN LANG LEVEL 2.
00004 000400*
00005 000500 ENVIRONMENT DIVISION.
00006 000600 INPUT-OUTPUT SECTION.
00007 000700 FILE-CONTROL.
00008 000800 SELECT INFILE1 ASSIGN TO UT-S-INFILE1.
00009 000900 SELECT INFILE2 ASSIGN TO UT-S-INFILE2.
00010 001000 SELECT INFILE3 ASSIGN TO UT-S-INFILE3.

*STATISTICS* SOURCE RECORDS = 466 DATA DIVISION STATEMENTS = 120
PROCEDURE
DIVISION STATEMENTS = 220
*OPTIONS IN EFFECT* SIZE = 1048576, BUF = 262144, LINECNT = 54, SPACE1,
FLAGW, SEQ
*OPTIONS IN EFFECT* SOURCE, DMAP, PMAP, NOCLIST, SUPMAP, NOXREF, NOSXREF,
LOAD, NODECK
*OPTIONS IN EFFECT* APOST, NOTRUNC, NOFLOW, NOTERM, NONUM, NOBATCH, NONAME,
COMPILE=0
*OPTIONS IN EFFECT* NOSTATE, RESIDENT, DYNAM, LIB, NOSYNTAX, NOOPTIMIZE,
NOSYMDMP
*OPTIONS IN EFFECT* NOTEST, VERB, ZWB, SYST, NOENDJOB, NOMIGR, NOLVL, DUMP,
NOADV
*OPTIONS IN EFFECT* NOLST, NOFDECK, NOCDECK, LCOL1, L120, NOFDECK, NOCDECK,
LCOL1
*OPTIONS IN EFFECT* L120, DUMP, NOADV, NOPRINT, NOCOUNT, NOVBSUM, NOVBREF,
LANGLVL(1)

(B)
ASG1534I PROGRAM VIASCOPR STARTED
ASG1519I PROGRAM VIASCOPR COMPLETED WITH RETURN CODE 0000
ASG1534I PROGRAM IKFCBL00 STARTED
ASG1519I PROGRAM IKFCBL00 COMPLETED WITH RETURN CODE 0000
ASG1534I PROGRAM VIASSYMB STARTED
ASG1519I PROGRAM VIASSYMB COMPLETED WITH RETURN CODE 0000
ASG1534I PROGRAM VIASANLZ STARTED
ASG1519I PROGRAM VIASANLZ COMPLETED WITH RETURN CODE 0000
ASG1534I PROGRAM VIADBTCH STARTED
ASG1519I PROGRAM VIADBTCH COMPLETED WITH RETURN CODE 0000
ASG1025I THE PRODUCT LEVEL FOR ASG-CENTER-OS(XA) Rx.x IS 000.
ASG1435I ASG-CENTER-OS(XA) Rx.x LVLxxx -SUMMARY REPORT- PROGRAM=VIADDDMO
ASG1399I OPTIONS IN EFFECT ARE: SOURCE, NODMAP, NOPMAP.
ASG1394I SUMMARY OF OS/VIS COBOL SYMBOLS EXTRACTED FROM VIADDDMO.
ASG1395I 98 DATA NAME SYMBOLS PROCESSED.
ASG1396I 33 PROCEDURE SYMBOLS PROCESSED.
ASG1397I 131 TOTAL SYMBOLS.

```

## ASG-Center Installation Guide

```

ASG1398I      199 VERBS PROCESSED.
ASG1436I DIAGNOSTICS: 0 TOTAL - 0 WARNING, 0 ERROR, 0 SEVERE, 0 CATASTROPHE
ASG1437I ASG-CENTER-OS (XA) Rx.x LVLxxx - END OF SYMBOL EXTRACTION FOR VIADDDMO
(C)
ASG-CENTER-OS (XA) Rx.x LVLxxx      PROGRAM: VIADDDMO      DD-MMM-YY  HH:MM:SS
PAGE 1
(D)
LINE  ERROR MESSAGE

      ASG0237I      131 SYMBOLS PROCESSED.
      ASG0238I      131 SYMBOLS MATCHED.
      ASG0240I      199 VERBS PROCESSED.
(E)
DIAGNOSTICS LINES: 0 TOTAL - 0 WARNINGS, 0 CONDITIONALS, 0 ERRORS, 0 DISASTERS
(F)
SOURCE LINES:      466 TOTAL - 120 DATA DIVISION STATEMENTS, 220 PROCEDURE
                   DIVISION STATEMENTS
(G)
PARAMETERS PASSED: NOCOBOLII,LANGLVL(1),FEATURES=(I,S,X)
(H)
OPTIONS IN EFFECT: BUFMAXK=2000K, FEATURES=(INSIGHT,SMARTTEST,EXTENDED),
FLAG(W),
                   LINECNT=60, NORECUR, NOSEQ, NOSOURCE, SPACE1, LANGLVL(1),
(I)
ENTRY POINTS:      VIADDDMO
(J)
EXTERNAL CALLS:     VIASUB
(K)
END OF PROCESSING: DD-MMM-YY  HH:MM:SS

```

This table describes the sections labeled on this Analyze Summary report:

Label	Description
(A)	<p>A complete listing of the program is produced and shows statement numbers generated by the analyze job in the first six columns. These numbers are referenced in diagnostic messages. These notations can also appear on the source listing:</p> <p style="margin-left: 40px;">C Statement was inserted with a COPY statement.</p> <p style="margin-left: 40px;">** Original source statement number is out of sequence.</p> <p style="margin-left: 40px;">I Statement was inserted with an INSERT statement.</p>
(B)	<p>This portion of the Analyze Summary is the report from the ESW monitor facility. The job steps that were executed by the monitor facility are listed along with the return codes produced.</p>
(C)	<p>The Center (Analyze) release and product level is shown along with the date and time the analysis was performed.</p>
(D)	<p>LINE and ERROR MESSAGE - This information is shown only if there were error conditions encountered. If so, this area lists the line number on which the error occurred and the error message.</p>

Label	Description
(E)	DIAGNOSTICS LINES - Indicates the total number of messages issued with subtotals for each type.
(F)	SOURCE LINES - Indicates the number of source lines in the program. The number of statements within the DATA DIVISION and PROCEDURE DIVISION are also shown. Each level number is counted as one statement in the DATA DIVISION. Each verb is counted as one statement in the PROCEDURE DIVISION.
(G)	PARAMETERS PASSED - Lists all of the analyze options specified for this analyze.
(H)	OPTIONS IN EFFECT - Lists all options in effect, including default and user-specified options.
(I)	ENTRY POINTS - Lists the entry points in this program.
(J)	EXTERNAL CALLS - Lists the programs that this program CALLs.
(K)	END OF PROCESSING - Lists the day, month, year, and time the analyze job completed. This date and time is also reflected in the online AKR statistics.

## Adding Analyze Facilities to a Standard Compile Mechanism

### CLIST Compile Mechanism

You can incorporate the Analyze Submit Facility into many existing in-house compile mechanisms. This provides users with access to ESW analysis without the complications of learning a new or different compile sequence. Most ISPF dialogs that use combinations of screens, CLISTs, and the TSO SUBMIT command are easily adapted to invoke the Analyze Submit facility, as shown in [Figure 20](#).

**Figure 20 • Adapting an ISPF Dialog to Invoke the Analyze Submit Facility**

The first step is to add a line in a panel to query whether a ASG Analysis should be performed. The compile panel can be used for this purpose. The new line in the panel definition may look like:

```
+   ASG ANALYZE      ==>_Z+      (%Y+OR%N+)
```

The following line should be added to the )INIT section of the panel definition:

```
.ZVARS = '(VSVANLYZ)'
```

If there is already an assignment to .ZVARS, VSVANLYZ should be added to the list in the appropriate place. This is assuming the VSVANLYZ variable will be used to record the response to this question. If another variable name is to be used, the description below would need to change so it corresponds.

It may be appropriate to perform a 'VGET (VSVANLYZ) PROFILE' before the panel is displayed, and a 'VPUT (VSVANLYZ) PROFILE' after it is displayed.

The CLIST should then be changed to invoke VIASUBDS instead of SUBMIT, if the VSVANLYZ variable has a Y value. Assume the CLIST contains the following code:

```
IF &E = Y THEN DO
  WRITE EDITED JCL WILL NOT BE SUBMITTED BY CLIST
  ISPEXEC EDIT DATASET(&ZUSER..CNTL(MEMBERX))
END
IF &E = Y THEN DO
  SUBMIT &ZUSER..CNTL(MEMBERX)
END
```

This should be changed to read as follows:

```
IF &VSVANLYZ = Y THEN DO
  IF &E = Y THEN DO
    WRITE EDITED JCL WILL NOT BE SUBMITTED BY CLIST
    %VIASUBDS &ZUSER..CNTL(MEMBERX) EDIT
  END
  ELSE DO
    %VIASUBDS &ZUSER..CNTL(MEMBERX)
  END
END
ELSE DO
  IF &E = Y THEN DO
    WRITE EDITED JCL WILL NOT BE SUBMITTED BY CLIST
    ISPEXEC EDIT DATASET(&ZUSER..CNTL(MEMBERX))
  END
  IF &E = Y THEN DO
    SUBMIT &ZUSER..CNTL(MEMBERX)
  END
END
```

Many JCL skeleton generators can have the required Analyze features imbedded easily. The JCL modifications typically done by the ESW JCL converter can be added to existing skeletons to create JCL that executes an ESW analysis. Use the examples in [Figure 21 on page 125](#) and [Figure 22 on page 126](#) as a guide to make the required modifications.

### **ISPF Compile Mechanism**

You can also install the Analyze Submit facility to emulate the standard ISPF Compile Option (Option 5), by adding this line to the appropriate ISPF menu screen:

```
V5, 'CMD(%VIASISP5 ISRJPA) NOCHECK'
```

This option functions exactly like the ISPF Compile Option, except that the resulting JCL is not submitted directly. Instead, it is passed to the Analyze Submit facility. Review the VIASISP5 CLIST for other required modifications.

If emulation of the ISPF Compile Option is not sufficient for your site, you can configure the Analyze Submit facility to replace the ISPF option directly. Contact the ASG Service Desk for details.

### **Assembler Analyzer (SmartTest Only)**

The Assembler Analyzer gathers and stores Assembler source code and data information in the AKR. Assembler source code displays on the Program View screen as output by the Assembler, and you can step through at the Assembler instruction level. Data fields can be displayed and modified.

### **Assembler Analyzer Input**

These are the inputs to the Assembler analyzer:

- JCL to assemble and link the program.  
This JCL should be the complete JCL used to assemble the program. Specifically, the JCL should contain steps to fetch the source from the source manager (i.e., CA-Librarian or CA-Panvalet), invoke Assembler with the appropriate options, and invoke the linkage editor.
- Assembler Analyzer options.  
Analyze options are not used by the Assembler analyzer.
- Assembler Analyzer features.  
The SmartTest analysis is the only applicable Assembler feature.

Run an analyze job using any of these methods:

- Select File ► Prepare Program on the ESW Primary screen to use the ESW - Prepare Program pop-up.

See the complete description for submitting this job described in ["Using the ESW - Prepare Program Pop-up" on page 97](#).

- Execute the VIASUBDS CLIST from any ISPF screen by typing:

```
TSO VIASUBDS dsn parms
```

where *dsn* is a PDS member or sequential dataset containing the Assemble JCL and *parms* represents any of the available execution parameters. See ["Using ISPF" on page 100](#) for more information.

- Execute the VIASUB PDF edit macro on an ISPF/PDF Edit screen when editing the Assemble JCL from any source manager. This edit macro is executed by typing VIASUB *parms*, where *parms* represents any of the available execution parameters. See ["Using ISPF/PDF Edit" on page 100](#) for more information.

**Note:** \_\_\_\_\_

Using the VIASUBDS CLIST and the VIASUB edit macro requires the ESW CLIST library to be available through the standard SYSPROC allocations.

---

## Automatic JCL Modifications for Assembler Procedures

The analysis process automatically modifies the JCL based on the specified parameters. If problems arise, use this procedure as a checklist to perform the analyze process manually until the problem can be determined and resolved. You can make changes to the JCL, the assemble procedure, or a copy of the assemble procedure.

- 1 Replace the PGM=IEV90 parameter with VIAASMH in the assemble step(s).

- 2 Add these DD statements to the assemble step(s) for these ESW datasets:

```
//VIAAKR DD DSN=[specified AKR name],DISP=SHR  
//VIALOG DD SYSOUT=*  
//VIAMRPT DD SYSOUT=*  
//VIAPRINT DD SYSOUT=*
```

- 3 If the SYSIN DD statement contains FREE=CLOSE, change it to FREE=END.

- 4 Ensure that the ESW load libraries are available to the modified step by adding a STEPLIB DD statement specifying the ESW load libraries, or by concatenating these libraries to an existing STEPLIB DD statement.

- 5 Ensure that the JOB and modified STEP EXEC statements have a minimum of REGION=4096K.
- 6 Add a VIAIN DD statement that designates the features and options to be used during analysis. This is the format of the DD statement:

```
//VIAIN DD *
      ST
/*
```

## Assembler Analyze JCL

[Figure 21](#) and [Figure 22 on page 126](#) illustrate Assembler/analyze JCL before and after submitting the Assembler analyze job.

**Figure 21 • Assembler/Analyze JCL Before Modifications**

```
//ASG JOB (ASG), 'ASG ASSEMBLER'
/*ROUTE PRINT DEST
/*
//ASM EXEC PGM=IEV90, REGION=512K, PARM='OBJ, XREF (SHORT), TEST'
//SYSIN DD DSN=ASG.VIACENXX.CNTL (VIAPASM), DISP=SHR
//SYSLIB DD DSN=SYS1.MACLIB, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DUMMY
//SYSLIN DD DSN=&&SYSLIN, UNIT=SYSDA, SPACE=(CYL, (1, 1)),
// DISP=(MOD, PASS), DCB=(RECFM=FB, LRECL=80, BLKSIZE=2480)
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
//SYSUT2 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
//SYSUT3 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
/*
//LINK EXEC PGM=IEWL, PARM='LIST, MAP, CALL, LET', COND=(5, LT, ASM),
// REGION=300K
//SYSLIN DD DSN=&&SYSLIN, DISP=(OLD, DELETE)
// DD DDNAME=SYSIN
//SYSLMOD DD DSN=USER.TEST.LOADLIB, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA, SPACE=(CYL, (1, 1))
//SYSIN DD *
      NAME VIAPASM(R)
/*
```

Figure 22 • Assembler/Analyze JCL After Modifications

```

//ASG JOB (ASG),'ASG ASSEMBLE'
/*ROUTE PRINT DEST
/*****
/** THIS JCL HAS BEEN MODIFIED BY THE ASG ANALYZE      *
/** SUBMIT FACILITY, WHICH CONVERTS COMPILE JCL INTO  *
/** COMPILE AND ANALYZE JCL.  NEW OR MODIFIED LINES  *
/** CONTAIN 'ASG' IN COLUMNS 74 THROUGH 76.          *
/*****
/**
//ASM EXEC PGM=VIAASMH,REGION=4096K,PARM='OBJ,XREF(SHORT),TEST'
//STEPLIB DD DSN=ASG.VIACENXX.LOADLIB,DISP=SHR
//SYSIN DD DSN=ASG.VIACENXX.CNTL(VIAPASM),DISP=SHR
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DUMMY
//SYSLIN DD DSN=&&SYSLIN,UNIT=SYSDA,SPACE=(CYL,(1,1)),
// DISP=(MOD,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480)
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//VIALOG DD SYSOUT=*
//VIAMRPT DD SYSOUT=*
//VIAPRINT DD SYSOUT=*
//VIAAKR DD DSN=ASG.VIACENXX.AKR,DISP=SHR
//VIAIN DD *
* ANALYZE FEATURES (ST):
ST
/*
/**
//LINK EXEC PGM=IEWL,PARM='LIST,MAP,CALL,LET',COND=(5,LT,ASM),
// REGION=300K
//SYSLIN DD DSN=&&SYSLIN,DISP=(OLD,DELETE)
// DD DDNAME=SYSIN
//SYSLMOD DD DSN=USER.TEST.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN DD *
NAME VIAPASM(R)
/*

```

---

# 7

## Processing Considerations

---

This chapter describes the processing considerations, and includes these sections:

Topic	Page
<a href="#">Security Software</a>	<a href="#">127</a>
<a href="#">Creating a Site-specific Default Profile</a>	<a href="#">129</a>
<a href="#">ISPF/PDF Edit Environment</a>	<a href="#">129</a>
<a href="#">Analysis Space Requirements</a>	<a href="#">131</a>
<a href="#">Batch Execution</a>	<a href="#">133</a>
<a href="#">Reblocking and/or Copying the ESW Load Library</a>	<a href="#">133</a>

### Security Software

ESW products are designed to run as applications under IBM's ISPF dialog manager and have no features of their own to provide security or circumvent existing security systems. Such features were intentionally omitted from ESW products so that there would be no conflicts with existing or future technology in this area.

Access to external resources under ESW products is accomplished using standard IBM documented facilities, with no ESW provided SVCs or authorized code used anywhere in the system (except for service manager Endeavor support). References to datasets are monitored completely by such generally available software security products as IBM's RACF and Computer Associates' ACF2 and Top Secret.

In one isolated case, however, we have provided an interface to another vendor's software so that it can provide additional security. When running with CA-Panvalet, in the COBOL Editor feature of ESW products, a call is made to CA-Panvalet's member level security module IPNEXIT when resolving ++INCLUDE members. This facility can optionally be selected during the installation of ESW products and requires access to the CA-Panvalet supplied IPNEXIT module during ISPF processing. This exit allows security at the CA-Panvalet dataset level to include member level security.

Dataset	Read	Write	Execute
LOADLIB	YES	NO	YES
CNTL	YES	NO	NO
ISPPLIB	YES	NO	NO
ISPSLIB	YES	NO	NO
ISPTLIB	YES	NO	NO
CLIST	YES	NO	NO
VIAMSGS	YES	NO	NO
AKR	YES	YES*	NO
QUEUE	YES	YES**	NO
MLCNTL	YES	NO	NO
MLSYNRWD	YES	NO	NO
MLSYNTBL	YES	NO	NO
ALFTBLS	YES	YES	NO
CONTAINR	YES	YES	NO

\* If SmartEdit is the only product installed, there will not be an AKR.

\*\* This is the SmartTest QUEUE file.

## Creating a Site-specific Default Profile

### *To update the VIASPROF member in the ESW ISPTLIB*

- 1 From within the ESW product, modify all the desired fields on any product screen.
- 2 Exit cleanly from the product so the new field values are saved in the profile for this user ID.
- 3 Rename the VIASPROF member in the ESW ISPTLIB to serve as a backup copy of the default profile.
- 4 Copy the VIASPROF member from the ISPF profile (of the user ID where the new values have been saved) to the ESW ISPTLIB with the name VIASPROF. This becomes the default for all new users.

You can verify the new profile by renaming or deleting the VIASPROF (for the user ID used above) and then entering the product, or by entering the product from a user ID that has never accessed the product.

## ISPF/PDF Edit Environment

Feature	Description															
Macro Environment	The Editor feature operates in an ISPF/PDF program edit macro environment. Unpredictable results can occur if there are outstanding problems in the IBM ISPF/PDF product.															
TSO CALL	The ESW product load module is an ISPF application and should not be invoked with a TSO CALL statement. If this is done, unpredictable results can occur.															
Customized Edit Panel	<p>The Editor feature of ESW products uses customized ISPF/PDF edit panels with these names:</p> <table border="0"> <tr> <td>VSPEDPDS</td> <td>VSPE2BRO</td> <td>VSPENUSR</td> </tr> <tr> <td>VSPENPDS</td> <td>VSPEDPAN</td> <td>VSPE2USR</td> </tr> <tr> <td>VSPE2PDS</td> <td>VSPENPAN</td> <td>VSPEDLIB</td> </tr> <tr> <td>VSPEDBRO</td> <td>VSPE2PAN</td> <td>VSPENLIB</td> </tr> <tr> <td>VSPENBRO</td> <td>VSPEDUSR</td> <td>VSPE2LIB</td> </tr> </table> <p>The source manager determines which screen is used.</p>	VSPEDPDS	VSPE2BRO	VSPENUSR	VSPENPDS	VSPEDPAN	VSPE2USR	VSPE2PDS	VSPENPAN	VSPEDLIB	VSPEDBRO	VSPE2PAN	VSPENLIB	VSPENBRO	VSPEDUSR	VSPE2LIB
VSPEDPDS	VSPE2BRO	VSPENUSR														
VSPENPDS	VSPEDPAN	VSPE2USR														
VSPE2PDS	VSPENPAN	VSPEDLIB														
VSPEDBRO	VSPE2PAN	VSPENLIB														
VSPENBRO	VSPEDUSR	VSPE2LIB														

Feature	Description
BUILTIN Commands	<p>These PDF commands are intercepted by the COBOL editor:</p> <ul style="list-style-type: none"><li>• EXCLUDE (other than EXCLUDE ALL)</li><li>• RESET (if an ESW operand such as TAG given)</li><li>• RFIND (if last FX was more recent than last FIND)</li></ul> <p>If user macros invoke these commands, you can avoid our interception by preceding the command with the BUILTIN keyword.</p>
Recursive Edit	Recursive edit sessions are fully supported.
Profiles	<p>Online processing uses the ISPF Profile dataset to retain user variables and Editor profile information. As with any application using this dataset, the possibility exists that the dataset will overflow while saving user or Editor profile information. Although the result of the failure is somewhat unpredictable, this is a list of potential results:</p> <ul style="list-style-type: none"><li>• Profile dataset IO errors.</li><li>• Profile dataset file full abend (e.g., B37, D37, etc.).</li><li>• SPF screen abends.</li><li>• Variable information not being retained, with no apparent failure messages.</li></ul> <p>In the event that the dataset overflows using the ESW product (which generally would only occur during the initial use), the normal procedures for expanding a dataset should be followed.</p>
Edit Profile	The COBOL Editor operates under its own edit profile, named VIASEDIT. Because of this, a recovery dataset created while in the COBOL Editor must be restored from the COBOL Editor, since the recovery files are associated with the application ID.
User Abends	User abends are listed in <a href="#">Appendix E, "Abend Codes," on page 245</a> . If an abend occurs that is not listed, check the abend codes for the source manager that was running at the time of the abend. The COBOL Editor attempts to recover from errors generated during its processing.
ASG-SmartEdit Performance	Although the COBOL Editor is completely functional for large source members, resource utilization is significantly decreased in Insight's Source View facility. Therefore, it is recommended that source programs exceeding 10,000 lines be analyzed and then viewed in Insight's Source View facility.

## Analysis Space Requirements

These tables contain the Analyzer resource estimates used to process COBOL programs and applications of various sizes. Information in these tables is the result of running Analyze under these criteria:

Version: Center R6.0  
 CPU Type: 3090-600 running MVS/ESA  
 Disk Type: 3380  
 Analyze Parms: BUFMAXK=4096K  
 Compiler Parms: BUF=256K,SIZ=1024K

**Note:** \_\_\_\_\_

For the AKR estimates in these tables, a record contains 4096 characters.

Analyze Resource Estimates for ASG-SmartTest Short Analyze

Source Lines	Virtual Memory Size	XA Memory Size	CPU Time MM:SS	AKR Records	VIAUT2 Cyls
1000	1060K	12100K	0:03	120	1
2000	1060K	12200K	0:07	250	2
5000	1060K	12300K	0:12	500	4
10000	1060K	12500K	0:25	1000	8
20000	1060K	12800K	1:00	1800	14
50000	1060K	14000K	2:30	3600	28

Analyze Resource Estimates for ASG-Insight,  
 ASG-SmartTest Extended, ASG-SmartDoc Short Analyze

Source Lines	Virtual Memory Size	XA Memory Size	CPU Time MM:SS	AKR Records	VIAUT2 Cyls
1000	1060K	13600K	0:04	150	2
2000	1060K	13700K	0:10	300	3
5000	1060K	13800K	0:30	750	6
10000	1060K	14000K	1:00	1500	12
20000	1060K	16000K	2:00	3000	24
50000	1060K	18000K	5:00	6000	50

Analyze Resource Estimates for ASG-Encore and ASG-SmartDoc Extended Analyze					
Source Lines	Virtual Memory Size	XA Memory Size	CPU Time MM:SS	AKR Records	VIAUT2 Cyls
1000	256K	13600K	0:03	250	2
2000	256K	14000K	0:15	600	5
5000	256K	16000K	1:00	2000	15
10000	256K	20000K	5:00	5000	40
20000	256K	28000K	15:00	10000	80
50000	4096K	48000K	60:00	30000	240

Analyze Resource Estimates for ASG-Recap Analyze						
Pgms	Virtual Memory Size	XA Memory Size	CPU Time MM:SS	AKR Records	VIAUT2 Cyls	VIAUT3 Cyls
75	600K	20000K	18:00	4250	53	32
150	600K	25000K	31:00	8500	53	63
250	600K	30000K	44:00	11100	53	67
1000	600K	36000K	176:00	42000	53	278

Analyze Resource Estimate for ASG-Alliance or Estimate Analyze						
Pgms	Virtual Memory Size	XA Memory Size	CPU Time MM:SS	AKR Records	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

**Notes**

Programs with recursions or out-of-perform GO TO statements may take significantly more resources than the tables indicate.

SmartDoc Extended analysis for programs containing more than 10,000 lines is not recommended due to the large amount of output and resource consumption.

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

## Batch Execution

You can use ISPF BATCH to process SCRIPT files containing commands. When running under ISPF BATCH, use the RUNBATCH and INITCMD execution parameters.

The RUNBATCH parameter indicates that the product should not use the terminal for input or output. The INITCMD parameter indicates the first command to be executed. For Batch execution, the first command should be an EXECUTE *script* file command.

When running using the RUNBATCH execution parameter, the product halts execution when the script command file contains no more commands to be executed. The product writes the message END OF SCRIPT FILE ENCOUNTERED, PRODUCT HALTED to the VIALOG file, but does not exit normally. Error messages written to VIALOG after the this message, should be ignored.

## Reblocking and/or Copying the ESW Load Library

If you need to reblock and/or copy the ESW load library, use the VIASCOPY CNTL dataset member. Specify the correct values for these parameters:

Parameter	Description
VIASOFT	Specifies the high-level node where the ESW products are installed (node length is <=8 and the default is ASG).
CENTER	Specifies the second-level node where ESW products are installed (the node length is <=8 and the default is VIACEN <sub>xx</sub> ). If the ESW dataset names contain more than three nodes, then specify all nodes except the first and the last as CENTER. For example, the dataset name of SYS3.CEN <sub>xx</sub> .NEW.LOADLIB should have VIASOFT=SYS3 and CENTER=CEN <sub>xx</sub> .NEW.
SYSOUT	Specifies the correct SYSOUT character.
SYSDA	Specifies the appropriate UNIT for temporary datasets.
LBLKSIZ	Specifies the block size to be used for the load library.
LBLKCNT	Optionally, adjusts the primary space allocation block count to account for the change in LBLKSIZ. That is, set LBLKCNT to approximately (2000*23476)/LBLKSIZ.
MAXBLK	Specifies the block size to be used for the load library.



---

# 8

## Command Usage Facility

---

This chapter discusses the command usage facility and contains these sections:

Topic	Page
<a href="#">Introduction</a>	<a href="#">135</a>
<a href="#">Command Usage Information</a>	<a href="#">136</a>
<a href="#">CUF Log File Record Format</a>	<a href="#">136</a>
<a href="#">Enabling the Command Usage Facility</a>	<a href="#">138</a>
<a href="#">Command Usage Facility Limitations</a>	<a href="#">141</a>

### Introduction

The ESW Command Usage Facility (CUF) provides a mechanism for analyzing command usage in ESW products. When CUF is active, it records each instance of primary command execution within an ESW product. Each instance is recorded in a CUF log that you can use to create reports for managing ESW products.

**Note:** \_\_\_\_\_

If you do not want to implement the CUF, you can skip this chapter.

---

You can activate the CUF facility at scheduled intervals to gather usage statistics. This can help ensure that ESW is being used effectively during development and maintenance projects. There are two options for storing information captured by CUF:

- Sequential file
- SMF records

If a sequential file is chosen, each record of the file represents one command usage entry. Command usage information is stored in the format shown in "[CUF Log File Record Format](#)" on page 136. An installation option specifies the high-level qualifiers for the CUF log file.

If you choose SMF, user exit VIASEXT4 is provided to illustrate how to process CUF data. It is the responsibility of each installation to maintain the user exit.

## Command Usage Information

The Command Usage Facility log includes these fields:

Field	Description
Userid	Specifies the user ID of the person executing the command.
Date	Specifies the date the command was executed, in YYYY/DDD format.
Time	Specifies the time the command was executed, in HH:MM:SS format.
Product	Specifies the product in use.
Command	Specifies the full name of the command.
Command Operands	Identifies information specific to the logged command.

**Note:**

When the same command is typed differently (e.g., FX and FNDXTND), all variations are translated and logged as the command's full name.

## CUF Log File Record Format

The first record in the CUF log file contains a value that identifies it as a CUF log file. The flag consists of the userid, product, and date. The Command field contains the text CUF ENTER. A corresponding record containing the value CUF EXIT is logged at the end of the session to designate the last line of the log file.

This is the CUF log file naming convention:

```
USERID.CUF.ASG.CUF####
```

where:

*USERID.CUF* is the system-generated CUF file name.

*ASG.CUF* is the name given under the CUF-Hi-Level-Nodes parameter in the VIA\$PRMS CNTL member.

#### is a running log used each time a user goes into an ESW product.

This is an example of a CUF log file name:

```
VIAXXX.CUF.VIAXXX.CUF00001
```

[Figure 23](#) and [Figure 24 on page 138](#) display the standard and extended record layouts for information tracked by the CUF.

#### Figure 23 • Command Usage Facility - Standard Record Layout

Userid	:	8 characters
Date	:	8 characters
Time	:	8 characters
Product	:	16 characters
Command	:	16 characters
-----		
Total		56 characters
-----		
Example:		
***** Top of Data *****		
VIAXXX	2000/16410:07:44ALLIANCE	CUF ENTER
VIAXXX	2000/16410:07:44ALLIANCE	POPFILY
VIAXXX	2000/16410:07:48ALLIANCE	CUF EXIT
***** Bottom of Data *****		

**Figure 24 • Command Usage Facility - Extended Record Layout**

```

Userid           : 8 characters
Date            : 8 characters
Time           : 8 characters
Product        : 16 characters
Command       : 16 characters
Command operands : 64 characters
-----
Total           120 characters
-----

Example:

***** ***** Top of Data *****
000001 VIAXXX 2000/16413:04:24ALLIANCE      CUF ENTER CUF ENTER
000002 VIAXXX 2000/16410:07:24ALLIANCE      POPFILEY OPEN ROW(0) COLUMN(-4)
000003 VIAXXX 2000/16410:07:28ALLIANCE      CUF EXIT CUF EXIT
***** ***** Bottom of Data *****

```

## Enabling the Command Usage Facility

### Step 1 - Modifying the CUF Installation Options

The installation options for CUF are located in the VIA\$PRMS member in the CNTL library. To make changes to any of the default values, edit the VIA\$PRMS member.

This table lists each installation option. The default value of each installation option is underlined. A vertical bar separates valid values that may be entered.

Option	Description
CUF-Enabled=YES   <u>NO</u>	Specifies whether CUF is enabled or disabled. The default is NO.
CUF-Hi-Level-Nodes=ASG.CUF	Specifies the high-level qualifier(s) to be used when creating CUF log files. Multiple levels of qualification are supported.
	<p><b>Note:</b></p> <p>Do not set the CUF-Hi-Level-Modes value to blanks. The VIASCUFU utility is not able to accumulate statistics from the log files without a valid high-level qualifier.</p>

Option	Description
CUF-Log-Type=SMF   <u>SEQ</u>	<p>Specifies SEQuential or SMF records as the type of log. The default is SEQ.</p> <p><b>Note:</b> If you specify SMF, you must code and include the VIASEXT4 user exit.</p>
CUF-Log-Extended=YES   <u>NO</u>	<p>Specifies whether additional command information should be included. This additional information includes the command operands. The default is NO.</p>
CUF-Log-Primary= <u>5</u>	<p>Specifies the amount of primary space for the allocation of a CUF log file. This option is used only when CUF-Log-Type=SEQ is specified. The default is 5.</p>
CUF-Log-Secondary= <u>10</u>	<p>Specifies the amount of secondary space for the allocation of a CUF log file. This option is used only when CUF-Log-Type=SEQ is specified. The default is 10.</p>
CUF-Log-Unit= <u>SYSDA</u>	<p>Specifies the unit type used to allocate the CUF log file. This option is used only when CUF-Log-Type=SEQ is specified. The default is SYSDA.</p>
CUF-Log-Volume=	<p>Specifies the VOLUME SERIAL name used to allocate the CUF log file. This option is used only when CUF-Log-Type=SEQ is specified. There is no default.</p>
CUF-Sample-Rate= <u>1</u>	<p>Specifies an integer value that determines how many commands must be executed for each command logged. For example, a value of 10 means that every tenth command is logged. A value of 1 means that every command is logged. The default is 1.</p>

## Step 2 - Compiling the Sample CUF Report Program - VIASCUF<sub>R</sub>

The sample report, produced by program VIASCUF<sub>R</sub>, is provided for users who have specified CUF-Log-Type=SEQ. This report illustrates the CUF log file record format and generates a sample report showing a count of command usage by product.

You can customize the sample report. For example, the report might sort CUF data based on product, userid, and command; and then list the command usage for each product.

You must compile and link the program that produces the sample report, VIASCUF<sub>R</sub>, before it can be executed. Sample JCL to compile and link the sample report program is provided in member VIASCUF<sub>C</sub> in the CNTL library. You must edit this JCL to provide a JOB card and to specify the correct library information.

## Step 3 - Running the Sequential File Utility (Optional for CUF Log Files)

**Note:** \_\_\_\_\_

This utility is not available to extract data stored as SMF records.

---

A Batch file utility, VIASCUF<sub>U</sub>, is provided to concatenate the sequential CUF log files. Runtime options include high-level qualifier specification and log file type for a given utility action.

To execute VIASCUF<sub>U</sub> and produce the sample report program, VIASCUF<sub>R</sub>.

- 1 Edit the CNTL library member, VIASCUF<sub>J</sub>, specifying the correct value for these parameters:

Parameter	Description
High-level-file-qualification=	Specifies the high-level qualification for the CUF log files to be included in the concatenation list.
Log-file-type=STANDARD EXTENDED  <u>BOTH</u>	Specify the types of log files to concatenate. If BOTH is chosen, then the log files are concatenated into a file with the same LRECL as an EXTENDED file (currently 120 characters). The default is BOTH.
Concatenated Files	The VIACONCT DD of the VIASCUF <sub>J</sub> job contains the concatenated CUF log files to be used as input to the sample CUF report (VIASCUF <sub>R</sub> ).
Concatenated List	The VIACATLS DD of the VIASCUF <sub>U</sub> job contains a list of the names of the log files that were concatenated.

- 2 Run VIASCUF<sub>J</sub>.

Member VIASCUFJ in the CNTL library provides an example of how to run the utility to concatenate the sequential CUF log files and run the sample report. It is possible that CUF generates sizable enough data to preclude concatenation of files. In that case, the utility would generate a list of related file names rather than actually concatenating the files. For example, it is possible to specify which files to concatenate to feed into any utilities a site develops to work with CUF data. The JCL member VIASCUFJ provides an example of this. ESW does not provide JCL or procedures to determine how often the CUF log files should be processed. It is the responsibility of each installation to determine the appropriate schedule for processing these files.

In the situation that there are too many CUF log files to concatenate as input to the sample report, a VIACATLE DD is provided to specify a list of files. This list can be used as input to your utility that processes the CUF log files.

#### **Step 4 - Modifying the User Exit - VIASEXT4 (Optional for SMF Records)**

The VIASEXT4 user exit is provided for storing CUF log information in SMF records. The VIASEXT4 member in the CNTL library sets up an SMF record and has comments explaining what is required to make it fully operational. Each site must provide an SMF record number and an SVC number that performs the actual writing of the SMF record. ESW does not provide a sample report or an extract utility for CUF log information stored in SMF records.

## **Command Usage Facility Limitations**

CUF is based on VIASEXT3, the command tracking user exit, which has these limitations:

- Some screen actions are not logged as a command. In SmartEdit, when a paragraph is selected by typing `S` after executing a `PREF` command, the command is not logged because the screen handler controls the line command.
- Commands generated off of screens are not logged. For example, the `FX` command generated from the Search pull-down is not logged. It is tracked as an `ENTER` command.
- The action of changing option fields on a screen is tracked as an `ENTER` command.
- Some ESW internal commands may be logged.
- Numeric commands are not logged.
- A filter prevents commands such as `ENTER` and `END` from being logged.

If you are using SmartEdit, not all commands entered are handled by the product. PDF EDIT commands are passed to ISPF for handling; these commands are not logged by the CUF.



---

# 9

## Problem Diagnosis

---

This chapter describes how to diagnose and solve problems encountered using Center, and contains these sections:

Topic	Page
<a href="#">General Problems</a>	<a href="#">143</a>
<a href="#">Service Desk Commands</a>	<a href="#">144</a>
<a href="#">Reporting Internal Errors</a>	<a href="#">145</a>
<a href="#">Printing AKR Data for Debugging</a>	<a href="#">148</a>

### General Problems

These are some general problems you might encounter with Center:

- System abends of S106 and S80A may indicate insufficient region space. If this occurs, increase your region size and try running the product again.
- Inability to dynamically load ESW load modules (S806 abend) is probably caused by the ESW load library not being allocated to STEPLIB or ISPLLIB, or modules are missing from the ESW load library.
- Unable to analyze any of the ESW demo programs. Check for misapplied product maintenance or improper installation.

**Note:** \_\_\_\_\_

Type `TSO PROFILE WTPMSG` on the command line and press Enter to see additional system messages during your ISPF session.

---

## **DBCS Terminals Displaying in Lowercase Characters**

If the DBCS terminals display screens with lowercase characters, check ISPF terminal parameters (ISPF 0.1) to make certain the terminal is specified as a DBCS terminal type. For Japanese, the terminal type should be 3278KN.

## **Program Analyzer Problems**

- Ensure that the COBOL source being analyzed can be compiled without E or S level messages.
- For Assembler programs, ensure that the program can be assembled with a condition code of 0 or 4.
- In general, if problems occur when analyzing different types of program applications, verify that all source code assembles/compiles correctly on your host system.

## **Program Query Problems**

- For dialog management screen errors, check for recent environment changes such as insufficient region size, panel library missing, misapplied, or incorrect maintenance.
- For VSAM AKR initialization failures, ensure the AKR is defined as VSAM with options NUMBERED, CISIZE(4096), and RECORDSIZE(4089 4089).

**Note:** \_\_\_\_\_

For help on any ESW error message, such as ASG1234, type `HELP 1234` or `HELP ASG1234`.

---

## **Service Desk Commands**

### **PRODLVL Command**

#### **Function**

The PRODLVL command is used to display the current ESW product level.

#### **Operands**

None.

## Usage Notes

This information is requested when you contact the ASG Service Desk for assistance. You can display this information by selecting Help ► About, which displays the Help - About pop-up.

The PRODLVL command displays the product name, operating system, product release number, and release level on the message line, in this format:

```
ASG1554I ASG INSIGHT-OS (XA) Rn.n AT Lnnn, ASG-CENTER Rn.n AT Lnnn
```

where *n.n* is the release number and *nnn* is the release level.

## Reporting Internal Errors

The ESW products catch most system and internal errors and generate entries in a user log dataset. Under normal situations, online processing recovers from the errors and displays the general purpose error message ASG0999. Batch processing issues the ASG1999 message and then halts. If this condition occurs, save the log listing, gather all supporting information (such as the description of the problem, screen prints, and analyze output), and call the ASG Service Desk.

## Log File Contents

The log provides information necessary to isolate the source of a failure. The log printout should be made available to the ASG Service Desk when reporting problems. To print the log information, use the Options - Log/List/Punch Definition pop-up. Be sure the Log Dataset process option is PK (print/keep) or PD (print/delete) before entering 1 on the command line. See the product user's guide or the online help for a description of the Options - Log/List/Punch Definition pop-up.

The log file contains one or more of these sections, depending on the type and severity of the error:

- Date and time that the log file was initialized.
- Product level that is currently being executed.
- Run-time error message number 1056I, which may indicate the cause of the failure. The message includes this information:
  - Date and time of error.
  - Run-time error number.
  - Name of the module and procedure where the error occurred.

- Source statement number where the routine was called.
- A trace back showing a history of the execution process that led to the error.
- Trace back of Called Routines report.
- System abend data, including the PSW and registers at the time of abend.
- Program entry point.
- Small snapshot of key areas of storage.
- Various messages that support the error reporting process.

## **Batch and Online Error Logs**

[Figure 25](#) illustrates a log file listing after a run-time error has occurred.

**Figure 25 • Log Listing After Run-Time Error**

```
HH:MM:SS ASG0406E RLSVIEW HAD RUN TIME ERROR.
      TRACE BACK OF CALLED ROUTINES
ROUTINE          STMT AT ADDRESS IN  MODULE
RSERLOG          27      1572C8   RSERLOG
RLSVIEW          115     1FDC76   RLSVIEW
RLSVENTR         12      1F46D2   RLSVENTR
RLMAIN           35      14EBCC   RLMAIN
<MAIN-PROGRAM>  2       14EF72   RLMAIN
PASCAL/VS                1D5A1E

HH:MM:SS ASG1056I RUN-TIME ERROR 14 IN RLCPSPEC/RLCPPROG/34. SEE TRACEBACK.
HH:MM:SS ASG0380E UNRECOVERABLE ERROR HAS OCCURRED.
      TRACE BACK OF CALLED ROUTINES
ROUTINE          STMT AT ADDRESS IN  MODULE
RSERLOG          27      1572C8   RSERLOG
<MAIN-PROGRAM>  9       14F034   RLMAIN
PASCAL/VS                1D5A1E

HH:MM:SS ASG1056I RUN-TIME ERROR 14 IN RLCPSPEC/RLCPPROG/34. SEE TRACEBACK.
HH:MM:SS ASG0999T INTERNAL ERROR IN RLNSIGHT PRINT LOG AND CALL TECH. SUPPORT.
      TRACE BACK OF CALLED ROUTINES
ROUTINE          STMT AT ADDRESS IN  MODULE
RSERLOG          35      157334   RSERLOG
<MAIN-PROGRAM>  11      14F058   RLMAIN
PASCAL/VS                1D5A1E
```

[Figure 26](#) illustrates a log listing after a system abend type error has occurred.

**Figure 26 • Log Listing After System Abend**

```

DD MMM YY HH:MM:SS ASG1055I LOG FILE HAS BEEN INITIALIZED.
HH:MM:SS ASG1025I ASG Encore Rx.x AT Lxxx, ASG CENTER Rx.x AT Lxxx
HH:MM:SS ASG0414I OPENING MEMBER ACME0003 OF AKR USER.TEST.AKR.
HH:MM:SS ASG1009I SYSTEM ABEND 000C4000 HAS OCCURRED.
HH:MM:SS ASG1010I ABEND PSW: 078D0004 2E1D1D3E.
HH:MM:SS ASG1011I REGISTERS AT TIME OF ABEND:
HH:MM:SS ASG1012I R0 : 0005ADEC R1 : 0005AFE8 R2 : 001D1AD8 R3 : 00001042
HH:MM:SS ASG1012I R4 : 0005AF00 R5 : 0005A6B4 R6 : 0005ACE2 R7 : 00000006
HH:MM:SS ASG1012I R8 : 0015D56A R9 : 40000001 R10: 0014CD58 R11: 00059010
HH:MM:SS ASG1012I R12: 00057FD0 R13: 0005AB30 R14: AE1D1D3A R15: 00000000
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (2E1D1D2E) PSW-16:
HH:MM:SS ASG1020I 000000 23405830 D0A81233 45E0C860 4A30285A '. . . . Y . . . . H - [ . . . ! '
HH:MM:SS ASG1020I 000010 910133C8 47102290 41302880 50301000 'J . . H . . . . . & . . . '
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (001D1AC8) PROCEDURE:
HH:MM:SS ASG1020I 000000 000009D8 8080406C D9D3C3D7 D7D9D6C7 ' . . . Q . . RLCPPROG . . '
HH:MM:SS ASG1020I 000010 90E7D00C 47F0CE90 60080004 04B800B8 ' . X . . . 0 . . - . . . . . '
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (0005ABC0) LOCAL STORAGE:
HH:MM:SS ASG1020I 000000 0005A925 00064000 00061000 0005A1A0 ' . . Z . . . . . '
HH:MM:SS ASG1020I 000010 0005A6B4 0005AAAC 00000042 01010140 ' . . W . . . . . '
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (0005AFE8) OUTPUT PARAMETERS:
HH:MM:SS ASG1020I 000000 00061000 0005ACE2 000080B4 0005ABDC ' . . . . . S . . . . . '
HH:MM:SS ASG1020I 000010 0005AF00 0004E2C1 ' . . . . . SA '
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (0005AB30) DSA:
HH:MM:SS ASG1020I 000000 0005A108 0005A718 00000002 6E15B04C ' . . . . . X . . . . . > . . < '
HH:MM:SS ASG1020I 000010 0015A6D2 0005ADEC 0005B090 0015B098 ' . . WK . . . . . Q '
HH:MM:SS ASG1020I 000020 00001042 0005AF00 0005A6B4 0005ACE2 ' . . . . . W . . . . . S '
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (00057FD0) PCWA:
HH:MM:SS ASG1020I 000000 0005C000 00059000 D7C3E6C1 00000000 ' . . . . . PCWA . . . . '
HH:MM:SS ASG1020I 000010 00000000 00059154 0003ECB0 001608DC ' . . . . . J . . . . . '
HH:MM:SS ASG1020I 000020 0003EAF8 00059010 0005B0E0 0005A718 ' . . . 8 . . . . . X . . '
HH:MM:SS ASG1020I 000030 00000000 00000000 00000000 00000000 ' . . . . . '
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (00058480) PCWA+1200:
HH:MM:SS ASG1020I 0004B0 00000000 00000000 00000000 00000000 ' . . . . . '
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (00059010) GLOBAL AREA:
HH:MM:SS ASG1020I 000000 00000000 00058F70 00000000 5E14EF74 ' . . . . . '
HH:MM:SS ASG1020I 000010 0014E8F0 00000C40 00059C30 0014EF1E ' . . Y 0 . . . . . '
HH:MM:SS ASG1020I
HH:MM:SS ASG1020I (AE1D1D2A) R14-16:
HH:MM:SS ASG1020I 000000 D3D14770 23405830 D0A81233 45E0C860 'LJ . . . . . Y . . . . . H - '
HH:MM:SS ASG1020I 000010 4A30285A 910133C8 47102290 41302880 '[ . . ! J . . H . . . . . '

TRACE BACK OF CALLED ROUTINES
ROUTINE STMT AT ADDRESS IN MODULE
RSERLOG 11 1571A8 RSERLOG
ONERROR 30 158796 RSERR
RLCPPROG 34 1D1D3E RLCPSPEC
CP_CALL_CMDS 136 14CAAC RLCP
RLCP 47 14DCFE RLCP
RLSMAN 21 15CE6C RLSMAN
RLSVIEW 20 1FD588 RLSVIEW
RLSVENTR 12 1F46D2 RLSVENTR
RLMAIN 35 14EBCC RLMAIN
<MAIN-PROGRAM> 2 14EF72 RLMAIN
PASCAL/VS 1D5A1E

```

## **Printing AKR Data for Debugging**

Use the VIASPRTI program to print detailed information from the AKR. This information is used to help diagnose processing errors. VIASPRTI prints all details for a specified program, or selectively prints components of the AKR program. An ASG technical representative provides instructions on using the program should the need arise. The JCL to execute VIASPRTI is included in the ESW CNTL library. The VIASPRTI CLIST is also provided to display the AKR information online.

---

# 10

## Application Analyzer Execution Monitor Support

This chapter describes the application analyzer and execution monitor support, and contains these sections:

Topic	Page
<a href="#">Execution Monitor Support</a>	149
<a href="#">Execution Monitor Description File</a>	150

### Execution Monitor Support

**Note:**

This chapter applies only if you are installing the application-level products, Alliance, Estimate, or Recap.

The Analytical Engine's JCL Analyzer recognizes values supplied to PGM parameters in JCL EXEC statements as executable programs. However, some PGM parameter values contain names of programs whose purpose is to execute other programs. Such programs are called Execution Monitors or Indirect Program Invocation programs.

To analyze JCL that invokes programs through an Execution Monitor, you must supply the JCL Analyzer with the names of the Execution Monitors and the parameters passed to them. Supply the necessary information in a file called the Execution Monitor Description file and inform the JCL Analyzer of the file's name through the Execution-Monitor-Defs parameter in VIA\$PRMS.

The JCL Analyzer automatically recognizes standard Execution Monitors such as DFSRRC00 and DSNMTV01 for IMS, and IKJEFT01 for DB2.

## Execution Monitor Description File

You must create the Execution Monitor Description file yourself. ASG recommends that you create it as a member of the ESW CNTL library, but you can create it anywhere as long as the value for the VIA\$PRMS Execution-Monitor-Defs parameter contains the fully-qualified name.

Create the Execution Monitor Description file with a line length of 80. However, when editing the file, contain each line within the first 72 columns.

The Execution Monitor Description file contains a separate section for each Execution Monitor in use at your site. Each section contains the information necessary to identify an Execution Monitor and its input format to the JCL Analyzer. Sections are separated by a blank line.

This is the format for each section of the Execution Monitor Description file:

```
Monitor monitor-name:  
  [Options: option-name (option-value), ...;]  
  syntax-rule...;  
End monitor-name;
```

A clause enclosed by brackets ([ ]) indicates the clause is optional. A clause followed by an ellipsis (...) indicates one or more occurrences of the preceding clause could follow.

Each Execution Monitor Description file section is made up of a Monitor statement, optionally followed by an Options statement, followed by one or more syntax-rules, and followed by an End statement.

### Monitor *monitor-name*

These are the rules for using the Monitor keyword and the *monitor-name* values:

- Begin each section of the Execution Monitor Description file with the keyword Monitor.
- Type the name of the Execution Monitor after the keyword Monitor. The Execution *monitor-name* is the name used in the PGM= field of the EXEC JCL being analyzed.
- Separate the keyword Monitor and the *monitor-name* by a single space.
- Make sure the *monitor-name* is followed by a colon. For example:

```
Monitor monitor-name:
```

**Note:** \_\_\_\_\_

The Monitor keyword and *monitor-name* are required for each section.

---

**Options: option-name(option-value),...;**

Values supplied to the Options keyword describe where the Execution Monitor gets its input and how the Execution Monitor should interpret that input. Input can be obtained from an execution PARM string or from a DD statement whose ddname appears in an *input (ddname)* option. The Options statement is not required.

These are the valid *option-names* and *option-values*:

<i>input(ddname)</i>	Specifies the name of the execution JCL DD statement that contains the input data for the Execution Monitor. If the <i>input (ddname)</i> value is not provided, the Execution Monitor's input is assumed to be supplied in the EXEC statement PARM string.
<i>margins(column1,column2)</i>	Sets the left and right margins of the input file. The default is <i>margins(1, 72)</i> . When the input is taken from PARM string, the column numbers refer to character positions in the PARM string, regardless of the PARM string's position in the JCL.
<i>case(asis ignore)</i>	Specifies whether the input is case sensitive. The case option-value applies to an EXEC statement PARM string, as well as to the input ddname. The default is <i>ignore</i> .
<i>format(fixed free)</i>	Specifies whether the input to the Execution Monitor must be in fixed columns. The default is <i>free</i> .

**syntax-rule...;**

The *syntax-rule* parameter defines the syntax the Execution Monitor uses to determine:

- Programs to monitor
- Input passed to monitored programs

Each *syntax-rule* is made up of a *rule-name*, followed by a colon, followed by a list of syntax-expressions, for example:

```
rule-name: syntax-expression ...;
```

## rule-name

The *rule-name* parameter identifies a sentence, a phrase, a clause, or a word used in the input. *Rule-names* can be anything you choose; however, each *rule-name* is reserved and has a special meaning. For example:

<i>syntax</i>	Identifies the rule used for the entire PARM string or input(ddname). Each section in the Execution Monitor file must contain a rule for syntax. The syntax <i>rule-name</i> should appear only on the left-hand side of a rule.
<i>program-name</i>	Identifies the monitored program.
<i>parm-string</i>	Identifies the parameter string passed to the monitored program.
<i>comment</i>	Defines a sequence of characters to be ignored by the Execution Monitor.
<i>end-of-line</i>	Identifies the end of a line or the end of a record.
<i>any</i>	Matches any alphanumeric string, number, quotes, or punctuation.
<i>name</i>	Matches any alphanumeric string.

**Note:** \_\_\_\_\_

Except for *syntax*, reserved rule-names should appear only on the right-hand side of a rule.

\_\_\_\_\_

## syntax-expression . . . ;

Use the syntax description from your Execution Monitor documentation to write the *syntax-expression* portion of the syntax-rules. The *syntax-expression* can be a list of other syntax-expressions or rule-names. You can group syntax-expressions by parentheses or by single quotes.

You can use special characters in syntax-expressions to indicate the syntax-expression is optional, repeated multiple times, one of multiple choices, or a concatenation of other syntax-expressions.

**Details Applicable to Both rule-names and syntax-expressions**

These are the special characters that you can use with *rule-name* and *syntax-expression* parameters:

- ? A question mark (?) indicates the preceding clause is optional.
- \* An asterisk (\*) represents zero or more occurrences of the preceding clause.
- + A plus sign (+) represents one or more occurrences of the preceding clause.
- | A vertical bar (|) indicates two or more clauses represent alternatives.
- & An ampersand (&) indicates two clauses are concatenated without intervening blanks.

A *rule-name* or *syntax-expression* can be followed by a column specification, for example:

```
@column or @(start-column, end-column)
```

where *column*, *start-column*, and *end-column* are column numbers. The column specification limits the *rule-name* or *syntax-expression* to a fixed column or range of columns. Column specifications apply to Execution Monitor input, whether it comes from an input file or the PARM value in an EXEC statement.

You can follow a column specification with a length specification (written as *length*), where *length* is an integer representing the length of the *rule-name* or *syntax-expression*. For example, a program name starting in column 9 and extending a length of 8 columns would be written as *program-name@9:8*.

**End monitor-name;**

These are the rules for using the End keyword and *monitor-name* values:

- End each section of the Execution Monitor Description file with the keyword End.
- Type the name of the Execution Monitor after the keyword End. This name must match the *monitor-name* specified in the Monitor statement at the start of the section.
- Separate the keyword End and the name of the Execution Monitor by a single space.
- Make sure the name of the Execution Monitor is followed by a semicolon. For example:

```
End monitor-name;
```

**Note:**

The End keyword and monitor-name is required for each section.

## Example

This example assumes a site uses two Execution Monitors: LOADNRUN and PASCINTR.

## LOADNRUN

The LOADNRUN Execution Monitor is executed in this JCL:

```
// . . .
//STEP010 EXEC PGM=LOADNRUN, PARM='LUNAR 0,0,255'
//STEPLIB DD DSN=TEST.UTILITY.LOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
// . . .
```

If LOADNRUN loads and runs a program whose name appears in character positions 1 through 8 of the PARM string (with the rest of the PARM string being parameters to the program), the Execution Monitor description file would contain:

```
Monitor LOADNRUN:
Syntax: program-name@1:8 parm-string@9:255;
End LOADNRUN;
```

In the example above, the first 8 characters of the EXEC statement's PARM value represent the name of the monitored program. The rest of the string represents the parameter passed to the program.

## PASCINTR

The PASCINTR Execution Monitor is executed in this JCL:

```
// . . .
//STEP020 EXEC PGM=PASCINTR, PARM='NOCHECK'
//STEPLIB DD DSN=SYS1.PASCV20.LOADLIB, DISP=SHR
//SYSIN DD *
        display('Running TREK...');
        execute(TREK, 'users=3');
        end(0);
/*
//SYSLIB DD DSN=MYPASC.TEST.SRCLIB, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
// . . .
```

In the preceding example, PASCINTR is an interpreter whose job is to execute the statements in the SYSIN DD. The interpreter supports three types of statements: DISPLAY, EXECUTE, and END. The EXECUTE statement causes the interpreter to execute the program whose name is the first parameter in the statement, using the second parameter as the parameters passed to the executed program.

The Execution Monitor Description file's PASCINTR section would contain:

```

Monitor PASCINTR:
Option:  input(SYSIN), margins(1, 72);
Syntax:  statement*;
statement:
            execute-statement | other-statement;
execute-statement:
            'EXECUTE' '(' program-name
            (',' parm-string (',' name)*)? ')' ';' ;
other-statement:
            action '(' name (',' name)* ')' ';' ;
action:
            'DISPLAY | 'END';
End PASCINTR;

```

When the action is EXECUTE, PASCINTR executes the program TREK, whose name is the first parameter. PASCINTR passes the second parameter to TREK. PASCINTR is not sensitive to the case (uppercase/lowercase) used to pass values to it, nor is it column sensitive, as long as its input is coded between columns 1 and 72.

The above PASCINTR example section specifies this information to the JCL Analyzer:

- PASCINTR is an Execution Monitor.
- PASCINTR takes its input from the SYSIN DD.
- The syntax statement indicates PASCINTR's input is a sequence of zero or more statements.
- The statement rule says each statement is either an execute-statement or an other-statement.
- The execute-statement rule says an execute statement starts with the word EXECUTE, followed by a left parenthesis, followed by the program name, followed by an optional list of additional parameters, followed by a right parenthesis, followed by a semicolon. The optional list of parameters should start with a comma, followed by the parm string, followed by a list of zero or more names separated by commas.
- The other-statement rule says that an other-statement is an action followed by a list of names, separated by commas, enclosed by parentheses, and terminated with a semicolon.
- The action rule says an action can be either DISPLAY or END.



---

## Appendix A

---

# Installation Options

Installation option settings for individual ESW products are stored in members contained in the ESW CNTL library. For each product you are installing, edit the installation options or verify that the defaults are appropriate for your site. These ESW CNTL library members contain default installation options:

CNTL Member	Product
VIA\$PRMA	Encore
VIA\$PRMB	Alliance
VIA\$PRMD	SmartDoc
VIA\$PRME	SmartEdit
VIA\$PRMG	Bridge
VIA\$PRMJ	Estimate
VIA\$PRMM	AutoChange
VIA\$PRMP	SmartTest
VIA\$PRMS	Center
VIA\$PRMU	AutoChange, Bridge
VIA\$PRMX	Recap

For details about installation options contained in these files, see the appropriate product's installation guide.

## File Format

Each installation option member consists of records containing a keyword, an equal sign (=), and a value or list of values. For example:

```
Work-Unit-Type=CYLS
```

## Syntax Specification

- The keywords cannot be changed and you cannot add additional keywords to the files. No distinction is made between lowercase and uppercase letters within a keyword.
- Blanks on either side of the equals sign are optional.
- Multiple values can be entered within the same record (where space permits) by enclosing the entire list in parentheses and separating each value by a comma, with or without blanks. For example:

```
Analyze-Pgm-No-Return=(ABEND*, ILBOABN*, ABORT*)
```

- Multiple values that do not fit in a single record may be entered in consecutive records by ending the continued record with a comma that immediately follows a value specification. Values may not be split across records. The end of the value list is denoted by a closing parenthesis with no comma after it. For example:

```
Inputs=(DFH*, ABC*, XYZ*,  
        *JLK, MNO*, *XYZ)
```

```
Load-Library=(VIASHxx.PROD.XALOAD,  
              VIASHxx.DEVL.XALOAD,  
              VIAPRxx.PROD.XALOAD,  
              VIAPRxx.DEVL.XALOAD)
```

- A semicolon (outside of quotes) logically ends the record, thereby allowing comments to follow it. For example:

```
Load-Library=(VIASHxx.PROD.XALOAD, ;This is a comment  
              VIAPRxx.PROD.XALOAD);This is another comment
```

The semicolon does not need to be separated from the value by a blank.

- Any record with an asterisk (\*) in column 1 is treated as a comment record.

Records that fail to meet these criteria are treated as errors and ignored. Required keywords that are not specified (either through omission or error) are assigned a predefined default value. Errors are listed along with the Job output.

## CNTL Member Parameters

### VIA\$PRMB – Alliance

Parameter	Page
AKR-Buffer-MaxK=8192	<a href="#">172</a>

### VIA\$PRMM – AutoChange

Parameter	Page
Disable-Foreground-Execution=	<a href="#">196</a>
Undoc-AKR-Pgm-Wait-Timeout=	<a href="#">219</a>

### VIA\$PRMG – Bridge

Parameter	Page
Bridge-Generate-Max-Per-JOB=10	<a href="#">188</a>

### VIA\$PRMU – AutoChange and Bridge

Parameter	Page
Autoch-Base-AKR-DSN=	<a href="#">187</a>
Bridge-Base-AKR-DSN=	<a href="#">187</a>
Bridge-Log-DSN=	<a href="#">188</a>

### VIA\$PRMS – Center

Parameter	Page
AKR-Buffer-MaxK=	<a href="#">172</a>
AKR-Compression=	<a href="#">173</a>
AKR-DSORG-VSAM=	<a href="#">173</a>

Parameter	Page
ASG-Pkey-Message=	<a href="#">185</a>
ASG-Pkey-Warning-Days=	<a href="#">186</a>
Alternate-Lang-Defs=	<a href="#">173</a>
Alternate-Lang-Entities=	<a href="#">175</a>
Alternate-Lang-Types=	<a href="#">175</a>
Alt-High-Level-Node=	<a href="#">175</a>
AMF-Statistics-Required-Node=	<a href="#">176</a>
Analyze-COBOL-LangLvl=	<a href="#">176</a>
Analyze-COBOL-Live-Exits=	<a href="#">176</a>
Analyze-Compiler-Alias-Names=	<a href="#">177</a>
Analyze-Compile-Required=	<a href="#">177</a>
Analyze-Compiler-Names=	<a href="#">177</a>
Analyze-Dynamic-Call=	<a href="#">177</a>
Analyze-Extraction-Alias=	<a href="#">178</a>
Analyze-Extraction-Pgm=	<a href="#">178</a>
Analyze-Extra-Memory=	<a href="#">178</a>
Analyze-Flag=	<a href="#">178</a>
Analyze-In-Out-Programs=	<a href="#">178</a>
Analyze-Input-Programs=	<a href="#">179</a>
Analyze-Lines-Per-Page=	<a href="#">179</a>
Analyze-Link-Edit-Module=	<a href="#">179</a>
Analyze-Output-Programs=	<a href="#">179</a>
Analyze-Pgm-No-Return=	<a href="#">179</a>

Parameter	Page
Analyze-Region-Size=	<a href="#">180</a>
Analyze-Return-Code=	<a href="#">180</a>
Analyze-Sequence-Check=	<a href="#">180</a>
Analyze-Source=	<a href="#">180</a>
Analyze-Source-Spacing=	<a href="#">180</a>
Analyze-STEPLIB-DCB-Info=	<a href="#">181</a>
Analyze-STEPLIB-Libraries=	<a href="#">181</a>
Analyze-Submit-JCL-Node=	<a href="#">181</a>
Analyze-Work-Primary=	<a href="#">181</a>
Analyze-Work-Secondary=	<a href="#">182</a>
Analyze-Work-Unit=	<a href="#">182</a>
APPC-Multi-User=	<a href="#">182</a>
ASG-Hi-Level-Nodes=	<a href="#">185</a>
ASM-Steplib-Groups=	<a href="#">186</a>
ASM-Version=	<a href="#">187</a>
Character-Colon=	<a href="#">189</a>
Character-Vertical-Bar=	<a href="#">189</a>
CICS-ASM-Preprocessor=	<a href="#">189</a>
CICS-COBOL-Preprocessor=	<a href="#">190</a>
CICS-CSD-Library=	<a href="#">190</a>
CICS-DLI-library=	<a href="#">190</a>
CICS-PLI-Preprocessor=	<a href="#">190</a>
COBOL2-Compiler=	<a href="#">192</a>

Parameter	Page
COBOL2-Library=	<a href="#">192</a>
COBOL2-SYSTEM-LRECL=	<a href="#">192</a>
Cobol-370-library=	<a href="#">190</a>
Cobol-370-Module=	<a href="#">190</a>
COBOLII=	<a href="#">192</a>
Cobol-MVSVM-library=	<a href="#">191</a>
Cobol-MVSVM-Module=	<a href="#">191</a>
Cobol-OS390-Library=	<a href="#">191</a>
Cobol-OS390-Module=	<a href="#">191</a>
Cobol-Version=	<a href="#">191</a>
CobolVS-Library=	<a href="#">192</a>
CobolVS-Module=	<a href="#">192</a>
DB2-Library=	<a href="#">193</a>
DB2-Plan=	<a href="#">193</a>
DB2-Preprocessor=	<a href="#">193</a>
DB2-Sort-Lib=	<a href="#">194</a>
DB2-STDSQL=	<a href="#">194</a>
DB2-Subsystem=	<a href="#">194</a>
Dead-CICS-File-Name=	<a href="#">194</a>
Dead-IMS-DD=	<a href="#">194</a>
Dead-JCL-DD=	<a href="#">194</a>
Dead-PSB=	<a href="#">195</a>
Discovered-Called-Programs=	<a href="#">196</a>

Parameter	Page
Discovered-CICS-Transid=	<a href="#">196</a>
Discovered-IMS-Format=	<a href="#">196</a>
Discovered-Load-Module=	<a href="#">196</a>
Discovered-Program-Csect=	<a href="#">196</a>
Discovered-Stage1-Transactions=	<a href="#">197</a>
DSN-Qualifier-Node=	<a href="#">197</a>
ENDEVOR-CONLIB=	<a href="#">197</a>
ENDEVOR-Steplib=	<a href="#">197</a>
ENDEVOR-TypeLength-NE80=	<a href="#">197</a>
ENDEVOR-User-SVC-Number=	<a href="#">198</a>
Execution-Monitor-Defs=	<a href="#">198</a>
Format-Date=	<a href="#">198</a>
Format-Time=	<a href="#">198</a>
IDMS-ASM-Preprocessor=	<a href="#">198</a>
IDMS-COBOL-Preprocessor=	<a href="#">199</a>
IDMS-Data-Dictionary-Name=	<a href="#">199</a>
IDMS-IDMSDMLC-library=	<a href="#">199</a>
IDMS-Load-Libraries=	<a href="#">199</a>
IDMS-Local-Mode-Libraries=	<a href="#">199</a>
IDMS-Node-Name	<a href="#">199</a>
IDMS-PLI-Preprocessor	<a href="#">199</a>
IDMS-SYS-Control-Library=	<a href="#">200</a>
IMSVS-RESLIB-library=	<a href="#">200</a>

Parameter	Page
JCL-SUBSYS=	<a href="#">201</a>
JES-Proc=	<a href="#">201</a>
JOB-Proc=	<a href="#">202</a>
Language=	<a href="#">202</a>
Librarian-ISPF-Module=	<a href="#">203</a>
Librarian-Release=	<a href="#">203</a>
Librarian-STEPLIB=	<a href="#">203</a>
Librarian-Subsystem-name=	<a href="#">203</a>
Member-Error-Count=	<a href="#">204</a>
Memory-Threshold-RMODE24=	<a href="#">204</a>
Memory-Threshold-RMODE31=	<a href="#">204</a>
Online-List-Unit=	<a href="#">205</a>
Online-List-Volume=	<a href="#">205</a>
Online-Log-Unit=	<a href="#">205</a>
Online-Log-Volume=	<a href="#">205</a>
Online-Max-Function-Time=	<a href="#">205</a>
Online-Perm-Unit=	<a href="#">205</a>
Online-Punch-Unit=	<a href="#">206</a>
Online-Punch-Volume=	<a href="#">206</a>
Online-Work-Block=	<a href="#">206</a>
Online-Work-Volume=	<a href="#">206</a>
Panvalet-IPNEXIT=	<a href="#">206</a>
Panvalet-ISPF-Module=	<a href="#">207</a>

Parameter	Page
Panvalet-Release=	<a href="#">207</a>
Panvalet-STEPLIB=	<a href="#">207</a>
Panvalet-Subsystem-name=	<a href="#">207</a>
PLI-Steplib-Group=	<a href="#">208</a>
PLI-Version=	<a href="#">209</a>
PROCLIB-JES-Alternates=	<a href="#">209</a>
PROCLIBs=	<a href="#">209</a>
Report-Banner-Page=	<a href="#">210</a>
Report-Master-Index=	<a href="#">212</a>
Report-Recursion=	<a href="#">214</a>
Report-Writer-Libraries	<a href="#">214</a>
Reserved-Subpools=	<a href="#">214</a>
Reserve-MACRO=	<a href="#">214</a>
SCLM=	<a href="#">215</a>
Script-Libraries=	<a href="#">215</a>
Separator-Date=	<a href="#">215</a>
Separator-Decimal-Point=	<a href="#">215</a>
Separator-Thousands=	<a href="#">215</a>
Separator-Time=	<a href="#">215</a>
SMS=	<a href="#">216</a>
SMS-Data-Class=	<a href="#">216</a>
SMS-Mgmt-Class=	<a href="#">216</a>
SMS-Storage-Class=	<a href="#">217</a>

Parameter	Page
Source-Manager=	<a href="#">217</a>
SYSOUT-Class=	<a href="#">218</a>
Translator-Monitor-Alias-Names=	<a href="#">218</a>
Translator-Program-Names=	<a href="#">218</a>
Translator-User-Preprocessor=	<a href="#">219</a>
Translator-Work-Primary=	<a href="#">219</a>
Translator-Work-Secondary=	<a href="#">219</a>
Translator-Work-Unit=	<a href="#">219</a>
User-Exit-Copy-DD=	<a href="#">220</a>
Work-SYSDA=	<a href="#">221</a>

### *Center Parameters for Alliance, Estimate, and Recap*

Parameter	Page
Appl-Anlz-AKR-Buffer-Maxk=	<a href="#">182</a>
Appl-Anlz-Excl-CSECT=	<a href="#">183</a>
Appl-Anlz-Excl-DD=	<a href="#">183</a>
Appl-Anlz-JCL-Add-TR-To-JOB-Card=	<a href="#">183</a>
Appl-Anlz-JCL-Fragment-DSN=	<a href="#">183</a>
Appl-Anlz-JCL-Fragment-Password=	<a href="#">184</a>
Appl-Anlz-JCL-Fragment-Volser=	<a href="#">184</a>
Appl-Anlz-JCL-Time=	<a href="#">184</a>
Appl-Work-Primary=	<a href="#">184</a>
Appl-Work-Secondary=	<a href="#">184</a>
Appl-Work-SYSDA=	<a href="#">184</a>

Parameter	Page
Appl-Work-Unit=	<a href="#">185</a>
Appl-Work-Volume=	<a href="#">185</a>
CICS-CSD-Library=	<a href="#">190</a>
CICS-DLI-library=	<a href="#">190</a>
Librarian-Extract=	<a href="#">202</a>
Library-Percent=	<a href="#">204</a>
Report-Lines-Per-Page=	<a href="#">212</a>

**VIA\$PRMA – Encore**

Parameter	Page
Gen-COBOL-Last-Node=	<a href="#">198</a>
NETRON=	<a href="#">204</a>

**VIA\$PRMJ – Estimate**

Parameter	Page
AKR-Buffer-MaxK=	<a href="#">172</a>
Default-language-criteria=	<a href="#">195</a>
Disable-Foreground-Execution=	<a href="#">196</a>
Online-Max-Function-Time=	<a href="#">205</a>
Report-Lines-Per-Page	<a href="#">212</a>
Undoc-AKR-Pgm-Wait-Timeout=	<a href="#">219</a>

**Note:**

See ["Center Parameters for Alliance, Estimate, and Recap"](#) on page 166 for additional Estimate parameters.

## VIA\$PRMD – SmartDoc

Parameter	Page
Character-Back-Slash=	<a href="#">189</a>
Perform-Hierarchy-Chart-GOTO=	<a href="#">207</a>
Perform-Hier-Chart-Conditionals=	<a href="#">207</a>
Perform-Hier-Struct-Duplicates=	<a href="#">207</a>
Report-Advanced-Source=	<a href="#">209</a>
Report-Call=	<a href="#">211</a>
Report-Compiler-Output=	<a href="#">211</a>
Report-Condensed-Source=	<a href="#">211</a>
Report-Copy=	<a href="#">211</a>
Report-Data-Division=	<a href="#">211</a>
Report-Enhanced-Data-Xref=	<a href="#">211</a>
Report-Metrics=	<a href="#">212</a>
Report-Paragraph-Xref=	<a href="#">212</a>
Report-Perform-Hierarchy-Chart=	<a href="#">212</a>
Report-Perform-Interface=	<a href="#">213</a>
Report-Program-Exception=	<a href="#">213</a>
Report-Structure-Chart=	<a href="#">213</a>
Report-Subset=	<a href="#">213</a>
Report-Verb-Context=	<a href="#">214</a>
Report-Verb-Frequency=	<a href="#">214</a>
SmartDoc-COBOL-List-To-SYSPRINT=	<a href="#">216</a>
SmartDoc-Help=	<a href="#">216</a>

Parameter	Page
SmartDoc-Minimum-Reports=	<a href="#">216</a>
Structure-Chart-Birds-Eye=	<a href="#">217</a>
Structure-Chart-Conditionals=	<a href="#">217</a>
Structure-Chart-GOTOs=	<a href="#">217</a>
Structure-Chart-Horizontal-Size=	<a href="#">217</a>
Structure-Chart-Max-Pages=	<a href="#">218</a>
Structure-Chart-Mode=	<a href="#">218</a>
Structure-Chart-Vertical-Size=	<a href="#">218</a>

**VIA\$PRME – SmartEdit**

Parameter	Page
Change-Man=	<a href="#">188</a>
ChangeMan-SUBSYSID=	<a href="#">188</a>
ChangeMan-Version=	<a href="#">188</a>
Check-Error-Level=	<a href="#">189</a>
Check-Language=	<a href="#">189</a>
CICS-Preproc-Opt	<a href="#">190</a>
COBOL-Edit-Profiles=	<a href="#">191</a>
CUA-Profile-Types=	<a href="#">193</a>
Editor-Exclude-Command=	<a href="#">197</a>
Include-Libraries=	<a href="#">200</a>
Librarian-Expand-COPY=	<a href="#">202</a>
Panvalet-Expand-COPY=	<a href="#">206</a>
User-Preprocessor=	<a href="#">220</a>

### **VIA\$PRMP – SmartTest**

Parameter	Page
ASMH-Deck=	<a href="#">186</a>
ASMH-Object=	<a href="#">186</a>
DB2-Procedure-Plan=	<a href="#">193</a>
IMS-TSR=	<a href="#">200</a>

### **VIA\$PRMX – Recap**

Parameter	Page
Report-Appl-Comparison=	<a href="#">210</a>
Report-Appl-Definition=	<a href="#">210</a>
Report-Appl-Exception=	<a href="#">210</a>
Report-Appl-FunctionPoint=	<a href="#">210</a>
Report-Appl-Metrics=	<a href="#">210</a>
Report-Appl-Progress=	<a href="#">210</a>
Report-Enterprise-Exception=	<a href="#">211</a>
Report-Enterprise-Metrics=	<a href="#">212</a>
Report-Executive-Summary=	<a href="#">212</a>
Report-Pgm-Comparison=	<a href="#">213</a>
Report-Pgm-Metric-History=	<a href="#">213</a>
Report-Program-Progress=	<a href="#">213</a>

**Note:**

See "[Center Parameters for Alliance, Estimate, and Recap](#)" on page 166 for additional Recap parameters.

---

## Changing an Installation Option Default Value

If it is necessary to change any of the Center installation option default values to conform to your site specifications, this should be done when the Center files are installed. See ["ESW Installation and Customization" on page 9](#) for additional information about how to do this.

If a specific product requires an installation option value different than the default value in the VIA\$PRMS installation options member, the default override should be added to the product's installation options member. The product members (such as VIA\$PRMB) are used only by online sessions, not by analyze Batch jobs. The override to the VIA\$PRMS value is applied whenever an online session for that product is initiated.

It is recommended that Alliance and Estimate online sessions always run with an AKR-Buffer-MaxK value of 8192 or higher, but the VIA\$PRMS default is 4096 (which is optimum for other products). Using Alliance as an example, the AKR-Buffer-MaxK option and its changed value of 8192 should be added to VIA\$PRMB, the Alliance installation options member. This avoids situations where the VIA\$PRMS default value is larger than necessary because of one product's special requirements. Whenever a Alliance online session is initiated, it receives an override AKR-Buffer-MaxK value of 8192.

In addition to the product specific files, ESW allows each user to include a file containing overrides to the default options for all products. This file must be a preallocated PDS or sequential file with the DDNAME VIAUPARM and can contain any of the installation options described in this TSM.

After the standard option files have been loaded, ESW attempts to open VIAUPARM. If it exists, the options in that file replace any matching options. This allows the user file to contain only required overrides.

Installation options that use program names accept special characters to signify generic names. An asterisk (\*) represents any number of characters. A question mark (?) represents one character. For example:

Example	Description
DBA*	All programs that begin with DBA and end with any characters.
D?A*	All programs that begin with D followed by any single character, followed by A, then followed by any characters.
DBA???	All programs that begin with DBA and end with any three characters.

**Note:**

Each installation option is valid only if the relevant product is being installed.

## Installation Options

The installation options are described in this section. The default value, if applicable, is underlined. A vertical bar separates the valid values that you can enter.

### **AKR-Buffer-MaxK=4096**

Center (VIA\$PRMS), Alliance (VIA\$PRMB), Estimate (VIA\$PRMJ)

Specifies the maximum amount of main storage that can be allocated by ESW products for input/output buffers to the AKR. This value is expressed in thousands. Large values can substantially decrease CPU and clock times in both attached Batch and online processing.

The default value for the AKR-Buffer-MaxK parameter in VIA\$PRMS is 4096. The default value on this parameter in VIA\$PRMB and VIA\$PRMJ is 8192, which overrides the Center value.

The maximum value for AKR-Buffer-MaxK that can be specified as a VIAUPARM override is 32767. The maximum value that can be specified in VIA\$PRMS is 16384.

### Online Sessions

This option may be specified in the individual product's installation member rather than in the VIA\$PRMS member. Including this option in the product member allows the desired default to be in effect for that product's online session. Batch jobs are not affected by adding AKR-Buffer-MaxK to VIAUPARM if VIAUPARM is allocated to the online TSO session. For Alliance and Estimate online sessions, AKR-Buffer-MaxK should be set to 8192 or larger. For SmartEdit, a value as low as 200 (200KB) is acceptable.

**Caution!** Allocating large AKR-Buffer-MaxK values to TSO regions without increasing the TSO region size results in a S878 abend.

### Batch Analyze Jobs

Program analyze jobs, such as Insight, allocate AKR buffer space to DD VIAAKR and temporary AKRs. Application analyze jobs, such as Alliance, allocate AKR buffer space to the DDs VIAUT2 and VIAUT3. The AKR-Buffer-MaxK parameter specifies the program level analyze and VIAUT2 buffer sizes. Another parameter, Appl-Anlz-AKR-Buffer-MaxK, can be used to allocate the VIAUT3 buffer space. Use the Appl-Anlz-AKR-Buffer-MaxK parameter as the primary buffer allocation value for application analyzes. If the Appl-Anlz-AKR-Buffer-MaxK parameter is not used, the VIAUT3 allocation defaults to AKR-Buffer-MaxK value. The application parameter makes it possible to allocate larger buffers for Batch application analyzes. Both program and application analyze jobs use only the parameter values in the VIA\$PRMS member, not those in the individual product members. See "[Appl-Anlz-AKR-Buffer-Maxk=8192](#)" [on page 182](#) for more information about Appl-Anlz-AKR-Buffer-MaxK.

To override AKR-Buffer-MaxK for Batch processing, specify the fully-qualified dataset and member name of the VIAUPARM dataset on the SYSUT1 DD of the VIAUPARM IEBGENER step. For example:

```
//SYSUT1 DD DSN=ASG.CExx.VIAUPARM(UPARM1) , DISP=SHR
```

You can also override the AKR-Buffer-MaxK parm by entering the fully-qualified dataset and member name on the Analyze Submit screen. Region size on both the JOB and STEPS should be increased by an equivalent amount to optimize performance and minimize the possibility of S878 ABENDs.

### **AKR-Compression=YES | NO**

Center (VIA\$PRMS)

Disables AKR data compression when you specify NO. If you specify YES, each ESW product determines whether its data is compressed. AKR compression becomes effective only if the AKR is allocated using Center Release 4.0 L005 or above. The default is YES.

### **AKR-DSORG-VSAM=YES | NO**

Center (VIA\$PRMS)

Specifies whether all AKRs allocated or expanded are to be VSAM RRDS or BDAM file organization. The default is NO (BDAM file organization). To allocate or expand AKRs as VSAM datasets, specify YES.

**Note:** \_\_\_\_\_

A BDAM AKR is allocated as a physical sequential (PS) dataset. However, to represent the direct accessing technique used against the dataset, the term BDAM AKR is used.

### **Alternate-Lang-Defs=(n,external\_name,'description',def\_dsn,def\_mem,load\_dsn,load\_mem,parms\_panel,parms\_var, model)**

**Note:** \_\_\_\_\_

This option has been discontinued and is only provided for customers currently on maintenance.

Center (VIA\$PRMS)

Specifies the Alternate Languages to be included on the Edit - Alternate Language List pop-up in application-level products. For example:

```
Alternate-Lang-Defs=(1,NAT1.2,'Natural V1.2',
    <none>,VIASDLN1,
    <none>,VIASNP12,
    <none>,<none>,alf_2_akr_mod,
    2,NAT2.1S,'Natural V2.1S',
    <none>,VIASDLNS,
    <none>,VIASNP2S,
    <none>,<none>,alf_2_akr_mod,
    3,NAT2.1R,'Natural V2.1R',
    <none>,VIASDLNR,
    <none>,VIASNP2R,
    <none>,<none>,alf_2_akr_mod,
    4,EASYTRIV,'Easytrieve',
    <none>,VIASDLET,
    <none>,VIASNPWB,
    <none>,<none>,alf_2_akr_mod)
```

where:

Parameter Value	Description
<i>n</i>	Is an integer indicating the number of the entry in the Edit - Alternate Languages List pop-up. Entry numbers must be in ascending order starting with 1.
<i>external_name</i>	Is a 1- to 8-character name that displays in the Name column of the Edit - Alternate Language List pop-up.
<i>description</i>	Is the long description that displays in the description column of the Edit - Alternate Language List pop-up.
<i>def_dsn</i>	Is reserved for future use. The default value is none.
<i>def_mem</i>	Is the Alternate Language description member.
<i>load_dsn</i>	Is reserved for future use. The default value is none.
<i>load_mem</i>	Is the Alternate Language processing load module.
<i>parms_panel</i>	Is reserved for future use. The default value is none.
<i>parms_var</i>	Is reserved for future use. The default value is none.
<i>model</i>	Is the name of the Alternate Language Facility import engine model used to load the AKR. Currently, the only allowed value is <i>alf_2_akr_mod</i> .

**Alternate-Lang-Entities=(x1,y1,x2,y2,...)**

Center (VIA\$PRMS)

Specifies the collection entities for the Alternate Language Facility, where  $x$  is the entity name constant and  $y$  is the internal index used to define the entity.

**Note:** \_\_\_\_\_

This Alternate Language Entities installation option contains constants that are required for correct functioning of Alliance and Estimate for Alternate Languages. This option must not be modified. If these values are changed, Alliance and Estimate do not function correctly for languages supported by the Alternate Language Facility.

---

**Alternate-Lang-Types=(x1,y1,'NNNNNNNN',x2,y2,'NNNNNNNN',...)**

Center (VIA\$PRMS)

Specifies the mapping of the language element type to the generic language type for the Alternate Language Facility, where  $x$  is the Alternate Language Facility primary language element type (e.g., NATURAL\_STATEMENT) and  $y$  is the Alternate Language Facility generic element type (CONTAINER, STATEMENT, SYMBOL, or RELATION).

**Note:** \_\_\_\_\_

This Alternate Language Entities installation option contains constants that are required for correct functioning of Alliance and Estimate for Alternate Languages. This option must not be modified. If these values are changed, Alliance and Estimate do not function correctly for languages supported by the Alternate Language Facility.

---

**Alt-High-Level-Node=**

Center (VIA\$PRMS)

Specifies the alternate high-level node for the work/restart files. This parameter is used when your site does not permit programmers to allocate files with their USERID as the high-level node. There is no default value.

### ***AMF-Statistics-Required-Node=No | Yes***

Center (VIA\$PRMS)

Specifies whether the CHECKSUM logic is turned off. The Application Maintenance Facility (AMF) uses ISPF statistics to detect if a member has been modified and needs to be reanalyzed. If the ISPF statistics do not exist for a member, AMF uses a special CHECKSUM calculation to compare the member's current CHECKSUM value to the last analyze CHECKSUM value. However, this calculation can be resource intensive because AMF reads each line in the member to calculate the CHECKSUM value.

The default value, YES, disables the CHECKSUM logic. If you set AMF-Statistics-Required-Node to YES and there are no ISPF statistics for a member, AMF does not determine if the member is modified and assumes that it is not.

### ***Analyze-COBOL-LangLvl=1 | 2***

Center (VIA\$PRMS)

Specifies whether to use the 1968 or 1974 American National Standard COBOL definitions when compiling source elements with meanings that have changed. LANGLVL(1) indicates the 1968 standard is to be used; LANGLVL(2) indicates the 1974 standard (X3.23-1974) is to be used. The default is 2. COBOL II, COBOL for MVS and VM, and COBOL for OS/390 and VM programs ignore the LANGLVL value.

### ***Analyze-COBOL-Live-Exits=YES | NO***

Center (VIA\$PRMS)

Specifies how the COBOL Analyzer handles LIVE exits. Live exits refer to exits from PERFORM ranges that are left dangling by imbedded PERFORMs or GO TOs in the original PERFORMed paragraphs. The compiler allows such coding techniques, but they are not recommended due to their complexity. The recommended and default value is NO for this option.

**Note:** \_\_\_\_\_

To avoid significant resource requirements resulting from Analyze-COBOL-Live-Exits=YES on all analyze jobs, the Analyze-COBOL-Live-Exits override option should be used only for programs known to have such coding technique, and as directed by the ASG Service Desk.

---

**Analyze-Compile-Required=**YES | NO

Center (VIA\$PRMS)

Specifies the default value of the compile flag when a Batch Analyze job runs. Specifying YES causes the compile and analysis to be executed. A value of NO can be used for an Insight-only analysis that bypasses the compile step. Compiles are not done for Application Product analysis. The default is YES.

**Analyze-Compiler-Alias-Names=**(VIACOBVS,VIAOPT3,VIACOBII,VIAOPTII,VIAASMH,VIAHLASM,VIAPLI,VIAOPT3,VIAPLI)

Center (VIA\$PRMS)

Specifies the compiler alias name(s), up to eight characters each, that Analyze uses instead of the compiler names specified in the Analyze-Compiler-Names option. If there is more than one name, separate them with a comma and enclose the list in parentheses. The sequence of these names must match the sequence given in the Analyze-Compiler-Names option.

**Analyze-Compiler-Names=**(IKFCBL00,CPXUPTSM,IGYCRCTL,CAOTSMON,IEV90,IFOX00,ASMA90,IEL0AA,CAOMSMON,IEL1AA)

Center (VIA\$PRMS)

Specifies the compiler name(s) supported by Analyze. These are OS/VS COBOL, CA-OPTIMIZER OS/VS COBOL, COBOL II, CA-OPTIMIZER COBOL II, Assembler H, High-level Assembler, OS PL/I Optimizing Compiler R1.5 or 2.x, CA-OPTIMIZER OS/VSCOBOL R6.0, and PLI for MVS and VM compiler. The sequence of these names must match the sequence given in the Analyze-Compiler-Alias-Names option. If there is more than one name, separate them with a comma and enclose the list in parentheses.

**Analyze-Dynamic-Call=**YES | NO

Center (VIA\$PRMS)

Specifies whether product functions look up called programs by using the variable name in dynamic calls as the name of the called program. If NO is specified, dynamic calls are not processed by the analyze and information for them is not available to ESW product functions. The default is YES. This option only applies to program-level products (SmartDoc, Insight, SmartTest, and Encore).

### **Analyze-Extra-Memory=YES | NO**

Center (VIA\$PRMS)

Specifies whether Analyze attempts to conserve memory by swapping some of the data between memory and disk. This feature is intended for use with extremely large source programs where the available region size is insufficient. This feature results in more disk I/O and additional CPU usage, but less memory consumption. The default is NO.

### **Analyze-Extraction-Alias=(VIASLIBR,VIASPAN1)**

Center (VIA\$PRMS)

Specifies the list of ESW monitor alias names that correspond to extraction program names given in the Analyze-Extraction-Pgm option. The name VIASLIBR corresponds to the CA-Librarian extraction program (normally AFOLIBR), and VIASPAN1 corresponds to the CA-Panvalet extraction program (normally PAN#1).

### **Analyze-Extraction-Pgm=(AFOLIBR,PAN#1)**

Center (VIA\$PRMS)

Specifies the list of extraction program names to be recognized by the Analyze Submit facility. When one of these names is found in the compile JCL, it is replaced with the corresponding name from the list given in the Analyze-Extraction-Alias option, so that information about INCLUDE statements can be gathered. The default names are AFOLIBR for CA-Librarian and PAN#1 for CA-Panvalet.

### **Analyze-Flag=W | E | X**

Center (VIA\$PRMS)

Specifies the message types to be listed for the compilation and analysis. A value of W indicates all warning and diagnostic messages are listed. An E indicates that all error messages are listed; warning messages are not. A value of X indicates all messages of the specified level or above are listed. The FLAG parameter does not apply for application-level products. The default is W.

### **Analyze-In-Out-Programs=(DBAOPEN\*,DBAREAD\*,DBACLOSE\*,DLI\*,DB2\*)**

Center (VIA\$PRMS)

Specifies the CALLED programs that contain Input and Output statements. Commands that search for Input and Output show the CALL statement for the programs listed. This is the default:

DBAOPEN\*, DBAREAD\*, DBACLOSE\*, DLI\*, DB2\*

where asterisk (\*) is a wildcard character representing zero or more characters.

**Analyze-Input-Programs=(DBAREAD\*)**

Center (VIA\$PRMS)

Specifies the CALLED programs that contain Input statements. Commands that search for Input shows the CALL statement for the programs listed. The default is DBAREAD\*, where asterisk (\*) is a wildcard character representing zero or more characters.

**Analyze-Lines-Per-Page=1..60..99**

Center (VIA\$PRMS)

Indicates the number of lines to be printed on an analyzed source listing page. The value must be 1 through 99. The default is 60.

**Analyze-Link-Edit-Module=IEWL**

Center (VIA\$PRMS)

Specifies the program name or names of the link editor. The default is IEWL and should rarely need to be changed. The Analyze Submit facility searches for a job step that invokes this program or one of these programs, and adds REUS to its PARM string if the SmartTest feature was selected and LINK LOAD MODULE REUSABLE? was answered Y.

**Analyze-Output-Programs=(xxxxxxxx,xxxxxxxx,...xxxxxxxx)**

Center (VIA\$PRMS)

Specifies the CALLED programs that contain Output statements. Commands that search for Output shows the CALL statement for the programs listed.

**Analyze-Pgm-No-Return=(ABEND\*,ILBOABN\*,ABORT)**

Center (VIA\$PRMS)

Specifies the CALLED programs (up to eight characters) that do not return. The Program Analyzer treats them as non-returning CALLs. The default is ABEND\*,ILBOABN\*,ABORT, where asterisk (\*) is a wildcard character that represents zero or more characters.

### **Analyze-Region-Size=4096**

Center (VIA\$PRMS)

Specifies the minimum region size (in 1KB=1024 byte units) required for the Analyze execution. The value specified is added to the JOB statement and the analyze step if a region size is not specified on the default JOB statement. If the user includes a region parameter on the default JOB statement, then that value is used as the region parameter on the JOB statement and for the analyze step.

### **Analyze-Return-Code=MAX | COMPILE**

Center (VIA\$PRMS)

Specifies the return code level for an Analyze execution. MAX indicates the highest return code from all substeps should be returned. COMPILE indicates only the return code from the compile step should be returned during an Analyze job. The default is MAX.

### **Analyze-Sequence-Check=YES | NO**

Center (VIA\$PRMS)

Specifies whether the Program Analyzer is to check the sequence of source module statements. If the statements are out of sequence, a warning message is printed. If the YES option of the Analyze-Source parameter is specified, a flag (\*\*) is printed between the Program Analyzer sequence and the source sequence numbers. The default is NO.

### **Analyze-Source=YES | NO**

Center (VIA\$PRMS)

Specifies whether the analyzed source program is to be listed. The default is NO. Unless a program listing is needed, the NO default should be used to reduce the system resources needed to produce a listing. You can override this parameter by typing SOURCE as an analyze option on the Analyze Submit screen.

### **Analyze-Source-Spacing=1 | 2 | 3**

Center (VIA\$PRMS)

Specifies the line spacing for the generated source listing when the YES option of the Analyze-Source parameter is specified. The default is single spaced.

### **Analyze-STEPLIB-DCB-Info=YES | NO**

Center (VIA\$PRMS)

This option suppresses DCB information in STEPLIB statements in JCL streams. Set this option to NO if your site standards require that no DCB information appear in STEPLIB statements.

### **Analyze-STEPLIB-Libraries=ASG.VIACENXX.LOADLIB**

Center (VIA\$PRMS)

Specifies the name(s) of the libraries that Analyze should include as STEPLIBs when a Batch analysis is executed. If there is more than one name, separate them with a comma and enclose the list in parentheses. Usually, the Center LOADLIB is the only one required. If you are using an application analyze user exit, you may need to include the runtime library in this parm, such as the COBOL runtime library.

**Note:** \_\_\_\_\_

This parameter is shipped with the default shown. It needs to be modified to reflect the exact high-level and middle-level nodes specified during installation. If no value is specified, the ESW libraries must be in the MVS Link List.

---

### **Analyze-Submit-JCL-Node=VIAJCL**

Center (VIA\$PRMS)

Specifies the last node of the dataset name generated by the Analyze Submit facility for the output JCL. This controls the edit profile that is used when you use the E (Edit) command. The VIAJCL parameter here should normally match the VIAJCL parameter in the VIASGBL CLIST (which is used for AKR allocation/ expansion). The default is VIAJCL.

### **Analyze-Work-Primary=Z**

Center (VIA\$PRMS)

Specifies the primary disk space value used for work space allocation during Analyze execution in units specified by the Analyze-Work-Unit option. If Alliance or Encore is being installed, a value of 60 cylinders is recommended.

### **Analyze-Work-Secondary=5**

Center (VIA\$PRMS)

Specifies the secondary disk space value used for work space allocation during Analyze execution in units specified by the Analyze-Work-Unit option. If Alliance or Encore is being installed, a value of 20 cylinders is recommended.

### **Analyze-Work-Unit=CYLS | *TRKS* | *BLKS***

Center (VIA\$PRMS)

Specifies the disk space allocation type used for work space allocation during Analyze execution. The default is CYLS.

**Note:** \_\_\_\_\_

If this option is changed, you should also modify the Analyze-Work-Primary and Encore install, Analyze-Work-Primary of 60 cylinders, and Analyze-Work-Secondary of 20 cylinders is recommended.

\_\_\_\_\_

### **APPC-Multi-User=YES | *NO***

Center (VIA\$PRMS)

Specifies whether the ESW APPC server is to operate in multi-user or single-user mode. In multi-user mode, the APPC server can support many conversations simultaneously. In single-user mode, each PC user starts a dedicated task to service only that PC. The default is YES.

### **Appl-Anlz-AKR-Buffer-Maxk=8192**

Center (VIA\$PRMS) for Application-level Products

Specifies the maximum amount of main storage that can be allocated to the AKR input/output buffers for application analyzes. This storage space is used by the temporary AKR allocated to DD VIAUT3. Large values can substantially decrease CPU and clock times in Batch. You should set the Appl-Anlz-Buffer-Maxk parameter in VIA\$PRMS rather than in the individual product members because analyze Batch jobs do not use the individual product members. You must specify this value in kilobytes (KB). The default value is 8192 KB and the maximum recommended value is 98304 KB.

Use the Appl-Anlz-AKR-Buffer-Maxk parameter as your primary buffer allocation statement for application analyzes. If the Appl-Anlz-AKR-Buffer-Maxk parameter is not used, the installation defaults to the AKR-Buffer-Maxk parameter. The Appl-Anlz-AKR-Buffer-Maxk parameter makes it possible to allocate larger buffers for Batch application analyze jobs. See "[AKR-Buffer-MaxK=4096](#)" on page 172 for more information about the AKR-Buffer-Maxk parameter and the differences between the two parameters.

**Note:**

JOB and STEP region sizes should be increased accordingly to account for the increased buffer size. Failing to do so results in S878 ABENDs.

**Appl-Anlz-Excl-CSECT=(DFH\*,IHE\*,ILBO\*,IHDF\*,AMPX\*,IGZ\*)**

Center (VIA\$PRMS) for Application-level Products

Specifies the CSECT names that are not included in the result of the load module analyze phase. Wild cards are valid in Appl-Excl-CSECT with a question mark (?) denoting exactly one character and an asterisk (\*) denoting any number of characters.

**Appl-Anlz-Excl-DD=(SYSABEND,SYSDUMP,SYSUDUMP,SYCHK,  
SYSCKEOV,IMSACB,IMS,DFSRESLB,DFSVSAMP,IEFRDER,IMSMON,PROC  
LIB,RECON1,RECON2,RECON3)**

Center (VIA\$PRMS) for Application-level Products

Specifies the DD names that are not included in the result of the JCL analyze phase.

**Appl-Anlz-JCL-Add-TR-To-JOB-Card=YES | NO**

Center (VIA\$PRMS) for Application-level Products

Specifies that the TIME and REGION parameters are added to the JOB statement for the analyze job if not already present. The value for the TIME parameter is taken from the installation option Appl-Anlz-JCL-Time. The value for REGION is taken from the installation option Analyze-Region-Size. The default is YES. A value of NO indicates that these parameters should not be added to the JOB statement.

**Appl-Anlz-JCL-Fragment-DSN=**

Center (VIA\$PRMS) for Application-level Products

Specifies the DSN of a dataset containing JCL that is included in the generated JCL for the analyze step. The JCL fragment is appended to the end of the job stream.

### ***Appl-Anlz-JCL-Fragment-Password=***

Center (VIA\$PRMS) for Application-level Products

Specifies the password for a password protected dataset containing JCL that are included in the generated JCL for the analyze step.

### ***Appl-Anlz-JCL-Fragment-Volser=***

Center (VIA\$PRMS) for Application-level Products

Specifies the volume serial number for a dataset containing JCL that will be included in the generated JCL for the analyze step.

### ***Appl-Anlz-JCL-Time=***

Center (VIA\$PRMS) for Application-level Products

Specifies the time parameter for the JCL for the analyze step. The value specified is added to the JOB statement and the analyze step if a TIME parameter was not included on the default JOB statement. If the user includes a TIME parameter on the default JOB statement, then that value is used as the TIME parameter on the JOB statement and for the analyze step.

### ***Appl-Work-Primary=50***

Center (VIA\$PRMS) for Application-level Products

Specifies the primary disk space value used for work file allocation during an online session during the analyze (VIAUT3). The type of units is specified by the Application-Work-Unit installation option. The user may override this option using the Option-Product Allocations pop-up. The default is 50.

### ***Appl-Work-Secondary=10***

Center (VIA\$PRMS) for Application-level Products

Specifies the secondary disk space value used for work file allocation during an online session and during the analyze (VIAUT3). The type of units is specified by the Appl-Work-Unit option. The user may override this option using the Option-Product Allocations pop-up. The default is 10.

### ***Appl-Work-SYSDA=SYSDA***

Center (VIA\$PRMS) for Application-level Products

Specifies a group name or unit address for the application restart/work file (VIAUT3), or default to the generic Group name or unit address SYSDA. The default value is SYSDA.

### ***Appl-Work-Unit=CYLS|TRKS|BLKS***

Center (VIA\$PRMS) for Application-level Products

Specifies the disk space allocation type used for work space allocation for the restart/work file (VIAUT3). The user may override this option using the Option-Product Allocations pop-up. The default is CYLS.

**Note:** \_\_\_\_\_

If this option is changed, you should also modify the Appl-Work-Primary and Appl-Work-Secondary options.

---

### ***Appl-Work-Volume=xxxxxx***

Center (VIA\$PRMS) for Application-level Products

Specifies the default VOLUME SERIAL name used online for allocating temporary work file datasets. When the Appl-Work-Volume option is blank, the volume is based on the unit type and TSO UADS specifications for the current user ID. The user may override this option using the Option-Product Allocations pop-up.

### ***ASG-Hi-Level-Nodes=ASG.VIACENxx***

Center (VIA\$PRMS)

Specifies the high-level qualifiers for the Center message file and the Alliance query, DB2, export, and training files. The default is ASG.VIACENxx. This is a required option for proper product execution. It is used to allocate the product message facilities and the indicated Alliance files. Ensure that the qualifiers shown here are consistent with the qualifiers used for the ESW product libraries.

**Note:** \_\_\_\_\_

An incorrect value for this parameter causes a User Abend 975 at startup.

---

### ***ASG-Pkey-Message=YES|NO***

Center (VIA\$PRMS)

Controls whether ASG APM error messages are displayed when a product key is not found or is invalid. If this option is set to YES, messages display for all uninstalled products. If set to NO, these warning messages are suppressed.

**ASG-Pkey-Warning-Days=[5...45|45]**

Center (VIA\$PRMS)

Controls the display of product expiration warning messages. Allowable values are 5 through 45 days. The expiration message begins appearing on the *nnth* day prior to expiration.

**ASMH-Deck=YES|NO**

SmartTest-ASM (VIA\$PRMP)

Specifies the default value of the DECK option for Analyze Assembler H (IEV90) support. This should duplicate the value in the actual Assembler H installation for your shop. The default is YES.

**ASMH-Object=YES|NO**

SmartTest-ASM (VIA\$PRMP)

Specifies the default value of the OBJECT option for Analyze Assembler H (IEV90) support. This should duplicate the value in the actual Assembler H installation for your shop. The default is NO.

**ASM-Steplib-Groups=(t,xxxxxx,vvvv,l,xxxxxxxx.xxxxxxxx.xxxxxxxx,xxxxxx,vvvv,l,xxxxxxxx.xxxxxxxx.xxxxxxxx,...)**

Center (VIA\$PRMS)

Specifies the Assembler load module name, Assembler version, and STEPLIB library(s) to use when analyzing Assembler programs as described in this table:

Variable	Description
<i>t</i>	Represents the number of Assembler information groupings to follow.
<i>xxxxxx</i>	Represents the load module name of the Assembler.
<i>vvvv</i>	Represents the Assembler version associated with the named load module, e.g., ASMH, HLASM.
<i>l</i>	Represents the number of STEPLIB libraries to follow.
<i>xxxxxxxx.xxxxxxxx.xxxxxxxx</i>	Represents the dataset name of the STEPLIB library to be used when analyzing Assembler programs.

The default values specify:

- Assembler load module names (IEV90 and ASMA90)
- Assembler versions (ASMH and HLASM)
- No STEPLIB libraries (0)

Typing 0 for the number of STEPLIB libraries for Assembler indicates that there are no STEPLIB libraries named. In this example, the STEPLIB library defaults to LINKLIB:

```
ASM-Steplib-Groups=(IEV90,ASMH,0,  
                   ASMA90,HLASM,0)
```

This example shows the use of different STEPLIB libraries depending on the version of Assembler.

```
ASM-Steplib-Groups=(IEV90,ASMH,0,  
                   ASMA90,HLASM,1,HLASM.PROD9410.LOADLIB)
```

### ***ASM-Version=ASMH|HLASM***

Center (VIA\$PRMS)

Specifies the version of the Assembler to be used when analyzing Assembler programs. The default is HLASM.

### ***Autoch-Base-AKR-DSN=ASG.VIACENxx.AUTOCH.AKR***

Autochange and Bridge (VIA\$PRMU)

Specifies the DSN of the Center Base AKR. This AKR is used to maintain conversion set information such as Center Strategies and Inline Strategies. AKR information is stored in the AKR member \$\$AUTOCH. This member can be deleted only by using the Batch AKR utility.

### ***Bridge-Base-AKR-DSN=ASG.VIACENxx.BRIDGE.AKR***

Autochange and Bridge (VIA\$PRMU)

Specifies the DSN of the Bridge Management AKR. This AKR is used to maintain information such as the names of the generate libraries and Bridge Definition relationships. AKR information is stored in the AKR member \$\$BRIDGE. This member can be deleted only by using the Batch AKR utility.

### **Bridge-Log-DSN=ASG.VIACENxx.BRIDGE.LOG**

Autochange and Bridge (VIA\$PRMU)

Specifies the name of the dataset for the Bridge Event Log. The Event Log is a dataset where significant events such as creating, renaming, deleting, importing, exporting, and generating Bridge Definitions are recorded.

### **Bridge-Generate-Max-Per-JOB=10**

Bridge (VIA\$PRMG)

Specifies maximum number of Bridge Definitions allowed per Bridge Definition Generate job. The default is 10.

### **Change-Man=NO|YES**

SmartEdit (VIA\$PRME)

Specifies whether the site uses the ChangeMan product and whether SmartEdit allows the Change Man processing mode to be set ON. When the ChangeMan processing mode is set ON, SmartEdit can extract copy libraries associated with a source member from the ChangeMan member definition and automatically populate the SmartEdit Options - COPY/Include Libraries screen. ChangeMan support is available only when SmartEdit is entered through execution of the ChangeMan product. The ChangeMan processing mode is set ON in SmartEdit using the SET command or the Options - Processing Modes pop-up. The default is NO.

### **ChangeMan-SUBSYSID=SER**

SmartEdit (VIA\$PRME)

Specifies the ChangeMan subsystem identifier. For support of ChangeMan Version 4.1.6 and above, modify ChangeMan-Subsysid= by appending the ChangeMan subsystem ID. This should be a one character identifier to SE, for example, -SERA (where A is the subsystem ID).

### **ChangeMan-Version=xxx**

SmartEdit (VIA\$PRME)

Specifies the version of ChangeMan you are using. Modify the ChangeMan-Version= parameter to the 3 digit version of ChangeMan you are using, for example, use 416 for Version 4.1.6.

**Character-Back-Slash=\**

SmartDoc (VIA\$PRMD)

Specifies the back slash character for SmartDoc Structure Charts. Modify this character when the printer to be used for SmartDoc reports does not have a \ (back slash).

**Character-Colon=:**

Center (VIA\$PRMS)

Specifies the time field separator for SmartDoc and Recap reports. Modify this option if the printer to be used for SmartDoc and Recap reports does not supply a colon.

**Character-Vertical-Bar=|**

Center (VIA\$PRMS)

Specifies the vertical bar character for SmartDoc and Recap reports, and for Structure View in Insight and Encore. Modify this character when the printer to be used for SmartDoc reports and Recap does not have a vertical bar (|), and/or to produce box figures in Structure View.

**Check-Error-Level=W|E|S|U|I**

SmartEdit (VIA\$PRME)

Specifies default minimum error reporting level to the SmartEdit CHECK feature.

**Check-Language=EN|UE|JP|JA**

SmartEdit (VIA\$PRME)

Specifies the default source language format option to the SmartEdit CHECK feature. Selections are English, uppercase English, mixed case English within Japanese, and uppercase English within Japanese.

**CICS-ASM-Preprocessor=DFHEAP1\$**

Center (VIA\$PRMS)

Specifies the correct load module name of the CICS preprocessor for Assembler programs at your site.

### **CICS-COBOL-Preprocessor=DFHECP1\$**

Center (VIA\$PRMS)

Specifies the correct load module name for the CICS and DL/I preprocessor for COBOL at your site.

### **CICS-CSD-Library=**

Center (VIA\$PRMS) for Application-level Products

Specifies the CICS load library that contains the CICS utility program DFHCSDUP. This utility is used during the Alliance and Recap analyze to extract information from the CICS CSD file. There is no default.

### **CICS-DLI-library=xxxxxxxx.xxxxxxxxx.xxxxxxxxx**

Center (VIA\$PRMS) for Application-level Products and SmartEdit

Specifies the correct CICS and DL/I preprocessor load library name for your site.

### **CICS-PLI-Preprocessor=DFHEPP1\$**

Center (VIA\$PRMS)

Specifies the correct CICS PLI preprocessor load module name for your site.

### **CICS-Preproc-Opt=**

SmartEdit (VIA\$PRME)

Overrides the default preprocessor option. ASG does not recommend the use of this parameter. Consult the ASG Service Desk before making any changes.

### **Cobol-370-library=xxxxxxxx . xxxxxxxxxxx . xxxxxxxxxxx**

Center (VIA\$PRMS)

Specifies the library containing the COBOL 370 compiler for the SmartEdit Check Facility. Leave this parm blank if the compiler is in the Link List Library or LPA.

### **Cobol-370-Module=IGYCTRL**

Center (VIA\$PRMS)

Specifies the member name of the COBOL 370 compiler load module (up to eight characters).

**COBOL-Edit-Profiles=(P1,P2,...)|ALL**

SmartEdit (VIA\$PRME)

Specifies a list of edit profiles for which SmartEdit will be initially active. COBEDIT applies only if SmartEdit is installed as a replacement for the normal ISPF editor. In this case, when the user enters the editor, SmartEdit will become active (changing the behavior of commands such as HELP and EXCLUDE) only if the edit profile is one of P1, P2, ..., or ALL. If the edit profile is not one of the ones in the list, SmartEdit will initially be inactive, but the user can explicitly activate SmartEdit using the COBEDIT command. See the ISPF/PDF documentation for the definition of Edit Profile; typically, it is the last node of the dataset name being edited.

**Cobol-MVSVM-library=xxxxxxxx.xxxxxxxxx**

Center (VIA\$PRMS)

Specifies the library containing the COBOL MVS/VM compiler for the SmartEdit Check Facility. Leave this parm blank if the compiler is in the Link List Library or LPA.

**Cobol-MVSVM-Module=IGYCRCTL**

Center (VIA\$PRMS)

Specifies the member name of the COBOL MVS/VM compiler load module (up to eight characters).

**Cobol-OS390-Library=xxxxxxxx.xxxxxxxxx**

Center (VIA\$PRMS)

Specifies the library containing the COBOL OS/390 compiler for the SmartEdit Check Facility. Leave this parm blank if the compiler is in the Link List Library or LPA.

**Cobol-OS390-Module=IGYCRCTL**

Center (VIA\$PRMS)

Specifies the member name of the COBOL OS/390 compiler load module (up to eight characters).

**Cobol-Version=COBOL68|COBOL74|COBOLII|COBOLIIR3|COBOL370|COB  
MVSVM|COBOS390**

Center (VIA\$PRMS)

Specifies the default COBOL version. The default is COBMVSVM.

***COBOL2-Compiler=IGYCRCTL***

Center (VIA\$PRMS)

Specifies the member name (up to eight characters) of the COBOLII compiler load module.

***COBOL2-Library=xxxxxxxx.xxxxxxxx***

Center (VIA\$PRMS)

Specifies the library containing the COBOLII compiler for the SmartEdit Check Facility. Leave this blank if the compiler is in the Link List Library or LPA.

***COBOL2-SYSTEM-LRECL=81***

Center (VIA\$PRMS)

Specifies the LRECL to be used for the COBOLII SYSTEM dataset. The default is 81.

***COBOLII=YES|NO***

Center (VIA\$PRMS)

Specifying YES overrides the Analyze-COBOL-LangLvl option and processes the input program as COBOL II. The default is NO.

***Cobol/SF=YES|NO***

Encore (VIA\$PRMA)

Specifying YES enables you to interface between Encore and COBOL/SF, if COBOL/SF is installed at your site. The default is NO.

***CobolVS-Library=xxxxxxxx.xxxxxxxx***

Center (VIA\$PRMS)

Specifies the library containing the OS/VS COBOL compiler for the SmartEdit Check facility. Leave COBOLVS-Library blank if the compiler is in the Link List Library or LPA.

***CobolVS-Module=IKFCBL00***

Center (VIA\$PRMS)

Specifies the load module name for the OS/VS COBOL compiler.

### **Complex-Process-Threshold=1...100...10000**

ASG-Recap (VIA\$PRMX)

Specifies the Recap complex processing threshold used for function point reports. General Systems Characteristics that have a complex processing number higher than the threshold will be flagged with an asterisk (\*). The default is 100. The value accepted is in the range 1 - 10,000, inclusive.

### **CUA-Profile-Types=ALL|(P1,P2,...)**

SmartEdit (VIA\$PRME)

Specifies default dataset profile types for which SmartEdit initially displays the CUA action bar. P1, P2, etc., are any acceptable dataset profile types (such as COB, COBOL, and CNTL). If you access products through the Center CLIST, this option is not used and SmartEdit is invoked when editing any dataset.

### **DB2-Library=xxxxxxxx.xxxxxxxxx**

Center (VIA\$PRMS)

Specifies the correct name for the DB2 load library for the call attach facility for your site. This is the load library from which DSNALI and DSNHLI2 will be loaded. This is also the load library used to invoke the preprocessor for the SmartEdit CHECK feature.

### **DB2-Plan=VIAPLAN**

Center (VIA\$PRMS)

Specifies the DB2 plan name used by ESW products to access DB2 table definitions. This must be the same name as specified for the PLAN parameter of the VIASBIND step of Common Customization.

### **DB2-Preprocessor=DSNHPC**

Center (VIA\$PRMS)

Specifies the correct load module name for the DB2 preprocessor for your site.

### **DB2-Procedure-Plan=VIAPPLAN**

SmartTest-DB2-SP (VIA\$PRMP)

Specifies the DB2 plan name to be used by the SmartTest-DB2-SP option to access SYSIBM.SYSPROCEDURES Table definitions. This must be the same name as that specified for the PLAN parameter of the VIAPBIND step during SmartTest customization. The default is VIAPPLAN.

### ***DB2-Sort-Lib=SYS1.SORTLIB***

Center (VIA\$PRMS)

Specifies the sort library to be used during the Alliance export process for application analysis information. This library is used to sort the exported application analysis information in preparation for loading into DB2 tables. The default is SYS1.SORTLIB.

### ***DB2-STDSQL=YES|NO***

Center (VIA\$PRMS)

For Alliance, Estimate, and Recap, specify YES if you are using the ANSI 86 SQL standard.

### ***DB2-Subsystem=DB2T***

Center (VIA\$PRMS)

Specifies the name of the DB2 subsystem for your site. If you have multiple subsystems, set SUBSYS to the most frequently used DB2 subsystem. This is the default used by analyze. For programs requiring a different subsystem, code the SUBSYS=xxxx parameter in the Analyze Options area of the Analyze Submit screen to override this default.

### ***Dead-CICS-File-Name=(CICS FCT name1, CICS FCT name2, ...)***

Center (VIA\$PRMS)

Specifies CICS file names that you do not want to be marked as Dead by the Application Analyzer. There is no default. This parameter supports both ? and \* formats.

### ***Dead-IMS-DD=(IMS DD1, IMS DD2, ...)***

Center (VIA\$PRMS)

Specifies IMS DD names (DATASET, AREA, or DDNAME) that you do not want to be marked as Dead by the Application Analyzer. There is no default. This parameter supports both ? and \* formats.

### ***Dead-JCL-DD=(JCL DD1, JCL DD2, ...)***

Center (VIA\$PRMS)

Specifies JCL DD names that you do not want to be marked as Dead by the Application Analyzer. There is no default. This parameter supports both ? and \* formats.

**Dead-PSB=(PSB name1, PSB name2, ...)**

Center (VIA\$PRMS)

Specifies PSB names that you do not want to be marked as Dead by the Application Analyzer. There is no default. This parameter supports both ? and \* formats.

**Default-language-criteria=(<base>,<name>,<displayid>,<search>,<generic lang>,<text>,...)**

Estimate (VIA\$PRMJ)

Specifies the default language criteria, for example:

```
Default-language-criteria=(BASE,
    ASM--MVS,ASM,VIAJASM,MVS-ASSEMBLER,
    'MVS ASSEMBLER Language Versions',
    BASE,COB--MVS,COB,VIAJSCOB,MVS-COBOL,
    'MVS COBOL Language Versions',
    ALTN,NAT1.2,NAT,VIAJSNAT,NATURAL,
    'NATURAL Language Versions',
    ALTN,NAT2.1S,NAT,VIAJSNAT,NATURAL,
    'NATURAL Language Versions')
```

where:

Parameter Value	Description
<i>base</i>	Specifies BASE if the language is supported natively and ALTN for all other languages.
<i>name</i>	Specifies the exact language name string included in the installation option Alternate-Lang-Def.
<i>displayid</i>	Specifies the three letters to be used for the language on displays and reports.
<i>search</i>	Specifies the default search criteria member name that contains the seeding search criteria for the language. The member name must be in the library indicated in the ASG-Hi-Level-Nodes installation option.
<i>generic lang</i>	Specifies the generic name to be used when there are multiple language versions. For example, NATURAL 1.2 and NATURAL 2.1 should be specified as NATURAL.
<i>text</i>	Specifies the descriptive text to be used for the language in displays and reports.

### ***Disable-Foreground-Execution=NO|YES***

Estimate (VIA\$PRMJ), AutoChange (VIA\$PRMM)

Specifies whether the foreground Job submission is disabled through Estimate. Because the foreground execution is considerably more expensive in TSO resource usage than Batch, it may be important to use Batch. If foreground Job execution is disabled, then the generation of candidate lists and components can only be completed in Batch. The default is NO (foreground Job submission is enabled).

### ***Discovered-Called-Programs=(name1, name2, ...)***

Center (VIA\$PRMS)

Specifies called program names that you do not want to be marked as Discovered by the Application Analyzer. There is no default.

### ***Discovered-CICS-Transid=(name1, name2, ...)***

Center (VIA\$PRMS)

Specifies CICS transaction names that you do not want to be marked as Discovered by the Application Analyzer. There is no default. This parameter supports both standard ? and wildcard \* formats.

### ***Discovered-IMS-Format=(name1, name2, ...)***

Center (VIA\$PRMS)

Specifies IMS MFS names that you do not want to be marked as Discovered by the Application Analyzer. There is no default.

### ***Discovered-Load-Module=(name1, name2, ...)***

Center (VIA\$PRMS)

Specifies load module names that you do not want to be marked as Discovered by the Application Analyzer. There is no default.

### ***Discovered-Program-Csect=(name1, name2, ...)***

Center (VIA\$PRMS)

Specifies program CSECT names that you do not want to be marked as Discovered by the Application Analyzer. There is no default.

***Discovered-Stage1-Transactions=(name1, name2, ...)***

Center (VIA\$PRMS)

Specifies IMS transaction names that you do not want to be marked as Discovered by the Application Analyzer. There is no default.

***DSN-Qualifier-Node=***

Center (VIA\$PRMS)

Allows a special user qualifier name to be appended after the USERID and/or prefix for all product generated datasets (i.e., Log, List, and Punch files). The DSN-Qualifier-Node parameter should match the &USERDSQ parameter in the VIASGBL CLIST. There is no default value.

***Editor-Exclude-Command=YES|NO***

SmartEdit (VIA\$PRME)

Specifies whether to use the ISPF/PDF EXCLUDE command. If this option is set to NO, the EXCLUDE command will be used. The default is NO.

***ENDEVOR-CONLIB=***

Center (VIA\$PRMS)

Specifies the Endeavor CONLIB library DSN for the Center Endeavor interface. There is no default value.

***ENDEVOR-Steplib=***

Center (VIA\$PRMS)

Specifies the name of the User authorized library containing the Endeavor load modules. There is no default value.

***ENDEVOR-TypeLength-NE80=(DOC=496,EXE=496)***

Center (VIA\$PRMS)

Specifies the TYPE and record length for each Endeavor TYPE that is other than 80 bytes. Specify only the TYPEs used by ESW products. ESW products automatically handle TYPEs defined as 80 bytes.

### ***ENDEVOR-User-SVC-Number=214***

Center (VIA\$PRMS)

Specifies the user SVC number to be used by ESW products to access Endeavor resources. This option also indicates the intent to support Endeavor and initiates additional processing to take place in ESW products. The default is 214.

### ***Execution-Monitor-Defs=(dataset-name, member-name)***

Center (VIA\$PRMS)

Specifies the dataset and member name of the Execution Monitor Description file. This file is used to describe to the Analytical Engine how programs are invoked by an Execution Monitor. If this parameter is omitted or if the dataset name and member do not exist, the Analytical Engine assumes the program name coded in a JCL step is the program being executed. For details, ["Application Analyzer Execution Monitor Support" on page 149](#). There is no default.

### ***Format-Date=MDY|DMY|YMD***

Center (VIA\$PRMS)

Specifies the Recap date format that is to be used commonly in the installation sites. The default is MDY (Month Day Year). The other formats are DMY (Day Month Year) or YMD (Year Month Day).

### ***Format-Time=12HOUR|24HOUR***

Center (VIA\$PRMS)

Specifies the Recap time format. The default is 12HOUR.

### ***Gen-COBOL-Last-Node=COBOL***

Encore (VIA\$PRMA)

Specifies the last node of the dataset name generated by Encore for the output COBOL modules.

### ***IDMS-ASM-Preprocessor=IDMSDMLA***

Center (VIA\$PRMS)

Specifies the correct load module name for the IDMS preprocessor for Assembler at your site.

**IDMS-COBOL-Preprocessor=IDMSDMLC**

Center (VIA\$PRMS)

Specifies the correct load module name for the IDMS preprocessor for COBOL at your site.

**IDMS-Data-Dictionary-Name=xxxxxxxx**

Center (VIA\$PRMS)

Specifies the default Data Dictionary name where Center and/or SmartEdit look for IDMS data definitions.

**IDMS-IDMSDMLC-library=xxxxxxxx . xxxxxxxx . xxxxxxxx**

Center (VIA\$PRMS)

Specifies the correct name for the IDMS load library containing the IDMSDMLC preprocessor load module.

**IDMS-Load-Libraries=(dsn1,dsn2,dsn3,..)**

Center (VIA\$PRMS)

Specifies the library name that contains the IDMS load modules including the Schema, Subschema, and Data Definition Language compilers.

**IDMS-Local-Mode-Libraries=(ddname1,dsn1,ddname2,dsn2,...)**

Center (VIA\$PRMS)

Specifies the correct ddname and dataset name for the IDMS datasets when running on local mode (e.g., DICTDB, DLOddb).

**IDMS-Node-Name=xxxxxxxx**

Center (VIA\$PRMS)

Specifies the correct IDMS node name. If the default node will be used, this option may be left blank.

**IDMS-PLI-Preprocessor=IDMSDMLP**

Center (VIA\$PRMS)

Specifies the correct IDMS PLI preprocessor load module name for your site.

**IDMS-SYS-Control-Library=xxxxxxxx . xxxxxxxxxxx . xxxxxxxxxxx**

Center (VIA\$PRMS)

Specifies the default IDMS system control library (SYSCTL) for use in running IDMS under Central Version control.

**IMS-TSR=U|L**

SmartTest (VIA\$PRMP)

Specifies the time stamp extension to the I/O PCB. The control region parameter, TSR=(U/L), specified in DFSPBxxx PROCLIB member, controls the representation of the time stamp with respect to local time versus Coordinated Universal Time (UTC). The default value is L.

**IMSVS-RESLIB-library=xxxxxxxx . xxxxxxxxxxx . xxxxxxxxxxx**

Center (VIA\$PRMS)

Specifies the correct name for the IMSVS.RESLIB for your site.

**Include-Libraries=((x,y,PDS,PAN,LIB,USR,DDC),(x,y,PDS,PAN,LIB,USR,DDC),. . .,(x,y,PDS,PAN,LIB,USR,DDC))**

SmartEdit (VIA\$PRME)

Specifies the non-standard keywords (other than ++INCLUDE, -INC, and COPY) used by an installation to include source from libraries other than the typical source library. The variables can have these values:

Variable	Description
x	Indicates the column in the source where the keyword can be found.  <b>Note:</b> _____ 0 indicates the keyword can be found anywhere from columns 1 through 80.
y	Specifies a keyword (up to 32 characters).
PDS	Specifies whether the keyword includes source from partitioned datasets.
PAN	Specifies whether the keyword includes source from CA-Panvalet datasets.
LIB	Specifies whether the keyword includes source from CA-Librarian datasets.

Variable	Description
<i>USR</i>	Specifies whether the keyword includes source from any other dataset type.
<i>DDC</i>	Specifies whether the keyword includes source from a DATACOM/DD data dictionary.

All Include-Libraries parameters are positional. To exclude a parameter, enter a comma in the appropriate position, for example:

```
Include-Libraries = ((0,++INC,,PAN,,))
```

If your site uses the DATACOM/DD COPYDD directive and you want to resolve the directives from your Data Dictionary, enter the COPYDD keyword. For example:

```
Include-Libraries=((0,COPYDD,,,,DDC))
```

***JCL-SUBSYS=(*subsystem name,ddname parameter number,...*)***

Center (VIA\$PRMS)

Indicates to the JCL analysis which parameters of a home-grown subsystem name are required ddnames. The IBM-supplied BLSR subsystem is handled internally, but other subsystems that can refer to other DDnames must be supplied in the JCL-SUBSYS parameter. For example:

If the JCL contains:

```
//dd1 dd dsn=a.b.c
//dd2 dd subsys (FJAM, parm1, parm2, DD1)
```

then the JCL-SUBSYS parameter would be:

```
JCL-SUBSYS (FJAM, 3)
```

***JES-Proc=xxxxxxxx . xxxxxxxxxxx . xxxxxxxxxxx(xxxxxxxx)***

Center (VIA\$PRMS)

Specifies a fully-qualified dataset and member name that contains the JES startup procedure (or a copy of it). The value can be up to 54 characters. For example:

```
JES-PROC=SYS1.PROCLIB (JES2)
```

The Analyze Submit facility uses the information in this member to search for cataloged procedures when submitting JCL for analyze jobs. If specified, this parameter overrides any values assigned to the PROCLIBs and PROCLIB-JES-Alternates installation parameters. For ease of use, this option is the recommended method to establish a search for cataloged procedures.

***JOB-Proc=xxxxxxx***

Center (VIA\$PRMS)

Specifies the DD name that can be used in a special DD statement to identify a job procedure library. For example, some products allow JCL to contain a statement such as //PROCLIB DD DSN=job.lib, DISP=SHR. This causes job.lib to be considered a procedure library for the current job. If this type of statement is used, specify JOB-Proc=PROCLIB. The JOB-Proc option should be blank if this type of facility is not installed.

***Language=ENGLISH|JAPANESE***

Center (VIA\$PRMS)

Specifies the language type for the Center message and panel libraries. The default is ENGLISH.

***Librarian-Expand-COPY=YES|NO***

SmartEdit (VIA\$PRME)

Specifies whether COPY statements will be expanded from Librarian datasets. The default value is YES. This parameter is used only if SmartEdit is installed.

***Librarian-Extract=YES|NO***

Center (VIA\$PRMS) for Application-level Products

**Note:** \_\_\_\_\_

\_\_\_\_\_ This parameter is applicable only if you are using Librarian version 3.9 or higher. \_\_\_\_\_

Specifies the keyword to be used by the analyze for application-level products to retrieve source from a CA-Librarian file. Specify YES if you are using -EXTRACT, and NO if you are using -SEL. This parameter is used only for application-level products.

These considerations apply when deciding the correct specification of the Librarian-Extract parameter: security authorization and COPY statements.

**1 Security authorization**

The -SEL keyword opens the CA-Librarian file for READ/WRITE access. The -EXTRACT keyword opens the file for READ access only. ESW products only access the file for READ. If Librarian-Extract=NO is specified, indicating that the analyze job is to use -SEL, and if the CA-Librarian file is security protected for WRITE access, then the proper access must be granted to the analyze job.

**2 COPY statements**

For COBOL programs that contain COPY statements whose source resides in a CA-Librarian file, the -EXTRACT keyword requires the use of the CA-Librarian provided subsystem interface to resolve these COPY statements.

***Librarian-ISPFLIB=ELIPS***

Center (VIA\$PRMS)

Specifies the Librarian ISPF EDIT module name (up to eight characters). The default is ELIPS. Librarian users running version 3.5 or earlier may need to specify the Librarian-Release parameter.

***Librarian-Release=00.0***

Center (VIA\$PRMS)

Specifies the Librarian release level. The default is 00.0 (none).

***Librarian-STEPLIB=xxxxxxxx . xxxxxxxxxxx . xxxxxxxxxxx***

Center (VIA\$PRMS)

The CA-Librarian STEPLIB dataset name. Leave blank if the dataset resides in a link list.

***Librarian-Subsystem-name=xxxxxxxx***

Center (VIA\$PRMS)

The CA-Librarian access method subsystem name.

### ***Library-Percent=100***

Center (VIA\$PRMS) for Application-level Products

Specifies the percentage of members in a library that can fail analysis during the analyze job. This percentage is applied to each type of member in the library (COBOL, CICS, IMS, JCL, LOAD Module). When this percentage is reached, the analyze will stop processing the library for the specific member type. A list of members that were not analyzed is printed in the file named VIAARPT in the analyze job. The user may override this option when submitting the analyze. The default percentage is 100%.

### ***Member-Error-Count=4000***

Center (VIA\$PRMS)

Specifies the maximum number of analysis errors and related messages allowed for a member during the analyze job. If this number of errors is exceeded, the analyze terminates processing for that member. The number specified must be between 1 and 4000. Users may override this option when submitting an analyze. The default is 4000. The count for each member includes INFORMATIONAL, WARNING, ERROR, and SEVERE messages, not just ERROR level messages.

### ***Memory-Threshold-RMODE24=512***

Center (VIA\$PRMS)

Specifies the memory threshold (in kilobytes) for load modules that reside below the 24-bit addressing limit (16 megabytes). This parameter should only be modified on instructions from the ASG Service Desk.

### ***Memory-Threshold-RMODE31=6144***

Center (VIA\$PRMS)

Specifies the memory threshold (in kilobytes) for load modules that reside above 16 megabytes. This parameter should only be modified on instructions from the ASG Service Desk.

### ***NETRON=YES | NO***

Encore (VIA\$PRMA)

Specifies if NETRON/CAP is to be supported by Encore. The default is NO.

### **Online-List-Unit=SYSDA**

Center (VIA\$PRMS)

Specifies the default unit device name (up to eight characters) used online for allocating the VIALIST dataset. The default value is SYSDA.

### **Online-List-Volume=xxxxxx**

Center (VIA\$PRMS)

Specifies the default VOLUME SERIAL name used online for allocating the VIALIST dataset. When the Online-List-Volume option is blank, the volume is based on the unit type and TSO UADS specifications for the current user ID.

### **Online-Log-Unit=SYSDA**

Center (VIA\$PRMS)

Specifies the default unit device name (up to eight characters) used online for allocating the VIALOG dataset. The default value is SYSDA.

### **Online-Log-Volume=xxxxxx**

Center (VIA\$PRMS)

Specifies the default VOLUME SERIAL name used online for allocating the VIALOG dataset. When the Online-Log-Volume option is blank, the volume is based on the unit type and TSO UADS specifications for the current user ID.

### **Online-Max-Function-Time=200**

Center (VIA\$PRMS), Estimate (VIA\$PRMJ)

Specifies the maximum number of CPU seconds that can be used for each online function. The function is canceled and an error message is generated if this time expires. Processing continues at some point prior to the canceled function. The default value is 200. Type `Debug STIMER nnnn` on the command line to temporarily change the Online-Max-Function-Time value.

### **Online-Perm-Unit=CYLS|TRKS|BLKS**

Center (VIA\$PRMS)

Specifies the disk space allocation units in CYLS, TRKS, or BLKS that are used by the Log, List, and Punch files. The default is TRKS.

### **Online-Punch-Unit=SYSDA**

Center (VIA\$PRMS)

Specifies the default unit device name (up to eight characters) used online for allocating the VIAPUNCH dataset. The default value is SYSDA.

### **Online-Punch-Volume=xxxxxx**

Center (VIA\$PRMS)

Specifies the default VOLUME SERIAL name used online for allocating the VIAPUNCH dataset. When the PUNVOL option is blank, the volume is based on the unit type and TSO UADS specifications for the current user ID.

### **Online-Work-Block=7476**

Center (VIA\$PRMS)

Specifies the default block size for all work file allocations done with the SYSDA option. If the allocated file is blocked with fixed length records, the block size is rounded down to a multiple of the record length.

### **Online-Work-Volume=xxxxxx**

Center (VIA\$PRMS)

Specifies the default VOLUME SERIAL name used online for allocating temporary workspace datasets. When the Online-Work-Volume option is blank, the volume is based on the unit type and TSO UADS specifications for the current user ID.

### **Panvalet-Expand-COPY=YES|NO**

SmartEdit (VIA\$PRME)

Specifies whether COPY members will be expanded from Panvalet datasets. The default value is NO. If the value is set to YES, ++INCLUDE members will not be expanded.

### **Panvalet-IPNEXIT=YES|NO**

Center (VIA\$PRMS)

Specifies whether the Panvalet IPNEXIT user exit routine is to be called during COBOL Editor processing. The default is NO.

***Panvalet-ISPF-Module=PSPILINI***

Center (VIA\$PRMS)

Specifies the online Panvalet ISPF load module name (up to eight characters). The default is PSPILINI. Panvalet R12.0 or earlier users may need to specify IPNINIT as the load module name.

***Panvalet-Release=00.0***

Center (VIA\$PRMS)

Specifies the Panvalet release number. The default is 00.0 (none). Do not include alphabetic suffixes.

***Panvalet-STEPLIB=xxxxxxxx.xxxxxxxxx.xxxxxxxxx***

Center (VIA\$PRMS)

The CA-Panvalet STEPLIB dataset name. Leave blank if the dataset resides in a link list. The default is blanks.

***Panvalet-Subsystem-name=xxxxxxxx***

Center (VIA\$PRMS)

If installed, specifies the CA-Panvalet access method subsystem name.

***Perform-Hier-Chart-Conditionals=YES|NO***

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc will include conditionals in the Perform Range Hierarchy Chart. The default is NO.

***Perform-Hier-Struct-Duplicates=YES|NO***

SmartDoc (VIA\$PRMD)

Specifies whether the SmartDoc Perform Range Hierarchy Chart is to show duplicate Perform ranges. Usually this option should be NO.

***Perform-Hierarchy-Chart-GOTO=YES|NO***

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc will include GO TOs in the Perform Range Hierarchy Chart. The default is YES.

**PLI-Steplib-Groups=(t,xxxxxx,vvv,l,xxxxxxxxxxxxxxx,xxxxxx,  
vvv,l,xxxxxxxxxxxxxxx,...)**

Center (VIA\$PRMS)

Specifies the compiler load module name, compiler version, and STEPLIB library(s) to use when analyzing PL/I programs.

t            Represents the number of compiler information groupings to follow.

xxxxxx     Represents the load module name of the compiler.

vvv         Represents the compiler version associated with the named load module, e.g., 1.5, 2.X, and MVS&VM.

l            Represents the number of STEPLIB libraries to follow.

xxxxxxxxxxxxxxx

             Represents the dataset name of the STEPLIB library to be used when analyzing PL/I programs.

For example:

```
PLI-Steplib-Groups=(3, IEL0AA, 1.5, 0,  
                  IEL0AA, 2.X, 0,  
                  IEL1AA, MVS&VM, 0)
```

This example specifies these default values:

- Three compiler load module names (IEL0AA, IEL0AA, and IEL1AA)
- Compiler versions (1.5, 2.X, and MVS&VM)
- No STEPLIB libraries (0)

**Note:** \_\_\_\_\_

Typing 0 for the number of STEPLIB libraries for the compiler indicates that there are no STEPLIB libraries named. In this case, the STEPLIB library defaults to LINKLIB.

---

This example illustrates the use of different STEPLIB libraries depending on the version of the compiler.

```
PLI-Steplib-Groups=(3, IEL0AA, 1.5, 1, PLIOPT.V151.PLICOMP,  
                  IEL0AA, 2.X, 3, SYS1.PLIBASE, PLIOPT.V230.PLICOMP, USER.TEST.LIB,  
                  IEL1AA, MVS&VM, 2, USER.TEST.LIB, USER.PROD.LIB)
```

**PLI-Version=2.X | 1.5 | MVS&VM**

Center (VIA\$PRMS)

Specifies the version of the PL/I compiler to be used as the default PL/I compiler during the analyze.

**PROCLIB-JES-Alternates=(nnnnnnnn,d,xxx.xxx.xxxx,nnnnnnnn,d,xxx.xxx.xxxx,nnnnnnnn,d,xxx.xxx.xxxx)**

Center (VIA\$PRMS)

Specifies the alternate procedure library lists available to the JES2 directive /\*JOBPARM or the JES3 directive /\*MAIN.

- |                         |  |
|-------------------------|--|
| <i>nnnnnnnn</i>         | Represents the tag number or name (1 through 8 alphanumeric characters). |
| <i>d</i>                | Represents the number of libraries associated with this tag.             |
| <i>xxx . xxx . xxxx</i> | Represents the names of the libraries associated with this tag.          |

If the JCL contains a directive mentioned above, the Analyze Submit facility uses the procedure libraries associated with this tag instead of the libraries specified in the PROCLIBs option. See ["JES-Proc=xxxxxxxx . xxxxxxxx . xxxxxxxx\(xxxxxxxx\)" on page 201.](#)

**PROCLIBs=SYS1.PROCLIB**

Center (VIA\$PRMS)

Specifies the name(s) of the qualified dataset(s) that will be searched for cataloged procedures. If there is more than one name, separate them with a comma and enclose the list in parentheses. The Analyze Submit facility requires this parameter for standard compile procedures. See ["JES-Proc=xxxxxxxx . xxxxxxxx . xxxxxxxx\(xxxxxxxx\)" on page 201.](#)

**Report-Advanced-Source=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the SmartDoc Advanced Source Listing is to be run.

**Report-Appl-Comparison=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Application Comparison report will be produced. The default is YES.

**Report-Appl-Definition=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Application Definition report will be produced. The default is YES.

**Report-Appl-Exception=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Application Exception report will be produced. The default is YES.

**Report-Appl-FunctionPoint=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Application Function Point Analysis report will be produced. The default is YES.

**Report-Appl-Metrics=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Application Metrics report will be produced. The default is YES.

**Report-Appl-Progress=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Application Progress report will be produced. The default is YES.

**Report-Banner-Page=YES|NO**

Center (VIA\$PRMS)

Specifies whether the banner page will be produced for SmartDoc and Recap reports. The default is YES.

**Report-Call=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the SmartDoc Call Statement report is to be run. The default is YES.

**Report-Compiler-Output=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the SmartDoc Compiler Output report is to be run. The default is YES.

**Report-Condensed-Source=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the SmartDoc Condensed Source Listing is to be run. The default is YES.

**Report-Copy=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the SmartDoc Copy report is to be run. The default is YES.

**Report-Data-Division=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the SmartDoc Data Division report is to be run. The default is YES.

**Report-Enhanced-Data-Xref=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the SmartDoc Enhanced Data Cross Reference report is to be run. The default is YES.

**Report-Enterprise-Exception=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Enterprise Exception report will be produced. The default is YES.

**Report-Enterprise-Metrics=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Enterprise Metrics report will be produced. The default is YES.

**Report-Executive-Summary=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Executive Summary report will be produced. The default is YES.

**Report-Lines-Per-Page=60**

Center (VIA\$PRMS) for Application Definition Products, Estimate (VIA\$PRMJ)

Specifies the number of lines per page that will be printed on reports generated by application-level products. The default is 60.

**Report-Master-Index=YES|NO**

Center (VIA\$PRMS)

Specifies whether the SmartDoc or Recap Master Index list will be produced. The default is YES.

**Report-Metrics=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to run the Metrics report. The default is YES.

**Report-Paragraph-Xref=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to run the Paragraph Cross-Reference report. The default is YES.

**Report-Perform-Hierarchy-Chart=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to run the Perform Range Hierarchy Chart. The default is YES.

**Report-Perform-Interface=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to run the Perform Range Usage and Interface report. The default is YES.

**Report-Pgm-Comparison=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Program Comparison report will be produced. The default is YES.

**Report-Pgm-Metric-History=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Program History report will be produced. The default is NO.

**Report-Program-Exception=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to run the Program Exception report. The default is YES.

**Report-Program-Progress=YES|NO**

Recap (VIA\$PRMX)

Specifies whether the Recap Program Progress report will be produced. The default is YES.

**Report-Structure-Chart=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to produce a Structure Chart. The default is YES.

**Report-Subset=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to run the Subsets report. The default is YES.

**Report-Recursion=YES|NO**

Center (VIA\$PRMS)

Specifies whether the recursion report from the Analyze job should be printed. The Report-Recursion parameter is not used by Alliance and Recap. The default is NO.

**Report-Verb-Context=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to run the Verb Summary report with the verbs shown in source context. The default is YES.

**Report-Verb-Frequency=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc is to run the Verb Frequency portion of the Verb Summary report. The default is YES.

**Report-Writer-Libraries:**

Center (VIA\$PRMS)

Specifies the default library containing the Report Writer load library used for the SmartEdit Check Facility.

**Reserve-MACRO=YES|NO**

Center (VIA\$PRMS)

Specifies that the OS Reserve macro can be used for the AKR when processing in a shared DASD environment. The default value of YES is recommended for data integrity.

**Reserved-Subpools=(0,1,2,6,10,13,14,15,78)**

Center (VIA\$PRMS)

Specifies the OS GETMAIN subpool areas that the product cannot FREEMAIN by subpool number. Subpools 0, 1, 2, 6, 10, 13, 14, 15, and 78 are defaulted and cannot be overridden.

### **SCLM=YES|NO**

Center (VIA\$PRMS)

Specify YES if IBM's Software Configuration and Library Manager (SCLM) is being used to store COBOL source. This will enable online product components to access source from SCLM. The default is NO.

### **Script-Libraries=(dsn1,dsn2,...)**

Center (VIA\$PRMS)

Specifies the dataset names of default script libraries to be used. Up to 4 library names can be specified. The datasets are concatenated in the order specified. The default is ASG.VIACEN<sub>xx</sub>.SCRIPT.

**Note:** \_\_\_\_\_

ASG.VIACEN<sub>xx</sub>.SCRIPT must be preallocated (LRECL=80). The defaults can also be defined as members of a PDS.

---

### **Separator-Date=-**

Center (VIA\$PRMS)

Specifies the Recap date separator. The default is a hyphen (-).

### **Separator-Decimal-Point=.**

Center (VIA\$PRMS)

Specifies the Recap decimal point character. The default is a period (.)

### **Separator-Thousands=','**

Center (VIA\$PRMS)

Specifies the Recap thousands separator. The default is a comma (,).

### **Separator-Time=:**

Center (VIA\$PRMS)

Specifies the Recap time separator. The default is a colon (:).

### **SmartDoc-COBOL-List-To-SYSPRINT=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the compiler listing should be written to SYSPRINT as usual when SmartDoc reports are generated. This is usually not necessary, since the SmartDoc Compiler Output report contains the same information. The default is NO.

### **SmartDoc-Help=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the descriptive help information about a report is to be produced on the first page of that report. The default is YES.

### **SmartDoc-Minimum-Reports=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the only SmartDoc reports to be run are the Advanced Source Listing and the Enhanced Data Cross-Reference. The default is NO.

### **SMS=YES|NO**

Center (VIA\$PRMS)

If your site uses SMS managed datasets, specify SMS=YES to cause ESW products to use the DATACLAS, MGMTCLAS, and STORCLAS parameters for allocating new datasets such as AKRs. The default value is NO.

### **SMS-Data-Class=**

Center (VIA\$PRMS)

If your site uses SMS managed datasets, specify the SMS DATA class to be used for new datasets to be allocated by ESW products. To take effect, the SMS option must also be specified. There is no default value.

### **SMS-Mgmt-Class=**

Center (VIA\$PRMS)

If your site uses SMS managed datasets, specify the SMS management class to be used for new datasets to be allocated by ESW products. To take effect, the SMS option must also be specified. There is no default value.

### **SMS-Storage-Class=**

Center (VIA\$PRMS)

If your site uses SMS managed datasets, specify the SMS storage class to be used for new datasets to be allocated by ESW products. To take effect, the SMS option must also be specified. There is no default value.

### **Source-Manager=YES|NO**

Center (VIA\$PRMS)

Specifies whether an auxiliary source manager is used. Source-Manager must be set to YES for DATACOM/DD COPYDD support. If Panvalet or Librarian is used at your site without the optional ISPF support, it may be desirable to use this option. The default is NO.

### **Structure-Chart-Birds-Eye=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether the Bird's Eye View representation of the Structure Chart is to be generated. This report is shown in Tile Mode, with each box condensed to one character. The default is YES.

### **Structure-Chart-Conditionals=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc will include conditionals in the structure chart.

### **Structure-Chart-GOTOs=YES|NO**

SmartDoc (VIA\$PRMD)

Specifies whether SmartDoc will include GO TOs in the Structure Chart.

### **Structure-Chart-Horizontal-Size=3...9...31**

SmartDoc (VIA\$PRMD)

Specifies the horizontal box size used on the SmartDoc Structure Chart. The minimum value is 3 (6 for DBCS); 31 is the maximum. The actual maximum horizontal box size may be constrained by physical limitations such as page size (Report-Lines-Per-Page). The default is 9.

### **Structure-Chart-Max-Pages=999999**

SmartDoc (VIA\$PRMD)

Specifies the maximum number of pages that are to be produced in a Structure Chart. The default is 999999.

### **Structure-Chart-Mode=TM|PM**

SmartDoc (VIA\$PRMD)

Specifies SmartDoc Structure Chart mode (Tile or Page Mode). Tile mode allows the user to paste the pages together to form a single diagram. Page mode prints the chart in a multiple-page format, which is useful if the report is to be stored in a binder. The default is Tile Mode (TM).

### **Structure-Chart-Vertical-Size=3...6...31**

SmartDoc (VIA\$PRMD)

Specifies the vertical box size SmartDoc is to use for the Structure Chart. The size must be a number from 3 to 31. The maximum may be constrained by physical limitations such as page size (Report-Lines-Per-Page). The default is 6.

### **SYSOUT-Class=\***

Center (VIA\$PRMS)

Specifies the SYSOUT class to be used for the VIALOG and VIAPRINT DD statements.

### **Translator-Monitor-Alias-Names=(VIACICS,VIASQL,VIASIDMS,VIASRW)**

Center (VIA\$PRMS)

Specifies the Translator alias name(s) used by Analyze during execution. If there is more than one name, separate them with a comma and enclose the list in parentheses.

### **Translator-Program-Names=(DFHECP1\$,DSNHPC,IDMSDMLC,SPCRWCOB)**

Center (VIA\$PRMS)

Specifies the Translator name(s) supported by Analyze. If there is more than one name, separate them with a comma and enclose the list in parentheses. The current default is for CICS, SQL, IDMS, and Report Writer translations.

***Translator-User-Preprocessor=(module name,input DDname,output DDname,·module name,input DDname,output DDname)***

Center (VIA\$PRMS)

Specifies the user preprocessor load module names, the input DDnames, and the output DDnames to be supported by the Analyze. This user preprocessor is executed after a preprocessor supported by the analyze, such as CICS, and before the compile step. It is intended to allow only minor modifications to the source, such as changing datanames. Use of this preprocessor for extensive changes, such as expanding macros and resolving copybooks, is not recommended.

***Translator-Work-Primary=1***

Center (VIA\$PRMS)

Specifies the primary disk space value used for translation work space allocation during Analyze execution in units specified by the Translator-Work-Unit option.

***Translator-Work-Secondary=1***

Center (VIA\$PRMS)

Specifies the secondary disk space value used for translation work space allocation during Analyze execution in units specified by the Translator-Work-Unit option.

***Translator-Work-Unit=CYLS|TRKS|BLKS***

Center (VIA\$PRMS)

Specifies the disk space allocation type used for translation work space allocation during Analyze execution. The default is CYLS.

**Note:** \_\_\_\_\_

If you change this option, you should also change the Translator-Work-Primary and Translator-Work-Secondary options.

---

***Undoc-AKR-Pgm-Wait-Timeout=xx***

Estimate (VIA\$PRMJ), AutoChange (VIA\$PRMM)

Specifies the wait time for AKR contention.

### **User-Exit-Copy-DD=**

Center (VIA\$PRMS)

Specifies the ddname that the application analyzer should allocate for the application user exit to access COPY libraries during the analyze process. There is no default.

### **User-Preprocessor=(cnt1,name,description,cnt2,type,clist,maxrc,**

.  
.

**type,clist,maxrc,  
name,description,cnt2  
type,clist,maxrc,**

.  
.

**type,clist,maxrc)**

SmartEdit (VIA\$PRME)

Specifies the information necessary for the execution of a user preprocessor during a SmartEdit session. This parameter includes specification of the order in which preprocessors are invoked.

*cnt1* - The count of the number of preprocessor Group entries included. The maximum number of group entries is 7.

*name* - The name of the preprocessor group.

*description* - A 30 character field that describes the preprocessor group. If embedded blanks are used, the description must be enclosed in quotes.

*cnt2* - The count of the number of preprocessor entries that follow for the preprocessor Group. The maximum is 7.

*type* - The type of preprocessor. These are the valid types:

CICS	Program contains only EXEC CICS statements
DLI	Program contains only EXEC DLI statements
CICSDLI	Program contains both EXEC CICS and EXEC DLI statements
DB2STDNO	DB2
DB2STD86	DB2 using SQL 86 option
IDMSLOC	IDMS Local Mode Dictionary

IDMSCEN	IDMS Central Mode Dictionary
USER	User preprocessor invoked using a CLIST

*clist* - The name of the CLIST to be executed to invoke the USER preprocessor. This parameter is valid only for the USER preprocessor type.

See the *ASG-SmartEdit Installation Guide* for more information about the CLIST used to execute the user preprocessor.

*maxrc* - The maximum valid return code returned by the CLIST. This parameter is valid only for the USER preprocessor type. A return code exceeding this value is interpreted as an error. For example:

- User-Preprocessor=(1,GROUP1,'DESCRIPTION',1,USER,USRPREP,0)  
This entry specifies one user preprocessor group named GROUP1 with its description DESCRIPTION. The group GROUP1 contains one USER type preprocessor step that invokes a CLIST named USRPREP. The maximum valid return code from the CLIST is 0.
- User-Preprocessor=(2,GROUP1,'DESCRIPTION1',3,DB2,CICS,USER,PREPDD1,8,GROUP2,'DESCRIPTION2',3,CICS,USER,PREPDD2,8,IDMSLOC)

This entry specifies two user preprocessor groups named GROUP1 and GROUP2.

GROUP1 has a description DESCRIPTION1 and consists of three steps: DB2 followed by CICS followed by a USER preprocessor step that invokes a CLIST named PREPDD1. The maximum valid return code from the CLIST is 8.

GROUP2 has a description DESCRIPTION2 and consists of three steps: CICS followed by a USER preprocessor step that invokes a CLIST named PREPDD2. The maximum valid return code from the CLIST is 8. The third step is an IDMS Local Mode preprocessor step, IDMSLOC.

### **Work-SYSDA=SYSDA**

Center (VIA\$PRMS)

Specifies the default unit device name (up to eight characters) used for allocating a temporary dataset. The default value is SYSDA.



---

## Appendix B

---

# CNTL, Skeleton, and CLIST Members

## Center CNTL Members

Member	Description
VIA\$PRMA	The default Encore installation options file. Use this file to specify Encore-specific default options.
VIA\$PRMB	The default Alliance installation options file. Use this file to specify Alliance-specific default options.
VIA\$PRMD	The default SmartDoc installation options file. Use this file to specify SmartDoc-specific default options.
VIA\$PRME	The default SmartEdit installation options file. Use this file to specify SmartEdit-specific default options.
VIA\$PRMG	The default Bridge installation options file. Use this file to specify Bridge-specific default options.
VIA\$PRMJ	The default Estimate installation options file. Use this file to specify Estimate-specific default options.
VIA\$PRMM	The default AutoChange installation options file. Use this file to specify AutoChange-specific default options.
VIA\$PRMP	The default SmartTest installation options file. Use this file to specify SmartTest-specific default options.
VIA\$PRMS	The default Center R4.3 and above installation options file. Use this file to specify Center-specific default options.
VIA\$PRMU	The default AutoChange and Bridge installation options file. Use this file to specify default options that are shared by AutoChange and Bridge.

Member	Description
VIASPRMX	The default Recap installation options file. Use this file to specify Recap specific default options.
VIABQCPS	The JCL to list the copybooks (COPY, ++INCLUDE, -INC, etc.) found in the application. Every program that references the copybook is displayed. The copybook library and member are also optionally displayed.
VIABQCPY	This JCL is the inverse of VIABQPS, listing programs by copybooks.
VIASAKRA	The JCL to allocate and initialize the AKR. Not applicable for SmartEdit-only installations.
VIASAKRU	The JCL to execute the AKR utilities. Not applicable for SmartEdit-only installations.
VIASAKRX	The JCL to expand the AKR. Not applicable for SmartEdit-only installations.
VIASALDF	The default parameters for the Alternate-Lang-Defs installation option that enables the Alternate Language Facility.
VIASALFJ	The JCL to compile the sample application analyzer user exit VIASALFU.
VIASALFU	The source code for sample application analyzer user exit for Alternate Language members.
VIASANJC	The JCL to convert compile JCL and execute analyze for SmartDoc sites that do not have ISPF.
VIASASMJ	The JCL to compile the sample application analyzer user exit VIASASMU.
VIASASMU	The source code for sample application analyzer user exit for Assembler members.
VIASBASE	The macro to define the location of installation options.
VIASBASJ	The JCL to assemble the macro VIASBASE.
VIASBDAM	The JCL to allocate and format the Batch execution queue file, VIAQUEUE.
VIASBIND	The JCL to bind the DBRM to a plan. This member is applicable only for DB2 installations.

Member	Description
VIASCCSJ	The JCL to compile the sample application analyzer user exit VIASCCSU.
VIASCCSU	The source code for sample application analyzer user exit for CICS members.
VIASCI	The JCL PROC to compile and link-edit a COBOL II program.
VIASCOBJ	The JCL to compile the sample application analyzer user exit VIASCOBU.
VIASCOBU	The source code for sample application analyzer user exit for COBOL members.
VIASCOPY	The JCL to copy and/or reblock the ESW load library.
VIASCPNL	The JCL to copy ISPPLIB and VIAMSGS libraries to ISPPALT and VIAMALT.
VIASCUFC	The JCL to compile and link the ESW command usage facility sample report program VIASCUFR.
VIASCUFJ	The JCL to run the CUF utility VIASCUFU to concatenate CUF log files.
VIASCUFR	The sample report program for the command usage facility.
VIASDBRM	The DBRM used in VIASBIND JCL to bind the DBRM to a plan. This member is applicable only for DB2 installations.
VIASDDC	The JCL to interface between Center and the data dictionary (DDCALL) module.
VIASDDCJ	The JCL to assemble and link edit the Center interface module VIASDDC.
VIASDIFF	The utility program used to print physical lines of code delta.
VIASDINS	The JCL to uninstall products and print a letter to be sent to ASG.
VIASDLAS	The Language Description File for AS400 COBOL.
VIASDLB0	The Language Description File for UNISYS B Series ANSI 68 and 74 COBOL.
VIASDLB1	The Language Description File for UNISYS A Series ANSI 68 COBOL.

Member	Description
VIASDLB2	The Language Description File for UNISYS A Series ANSI 74 COBOL.
VIASDLCB	The Language Description File for FLCOBOL.
VIASDLCD	The Language Description File for CDC 3600 COBOL.
VIASDLET	The Language Description File for Easytrieve and Easytrieve Plus.
VIASDLHP	The Language Description File for Hewlett Packard 3000 COBOL.
VIASDLH1	The Language Description File for Honeywell ANSI 68 COBOL.
VIASDLH2	The Language Description File for Honeywell ANSI 74 COBOL.
VIASDLMF	The Language Description File for Microfocus COBOL.
VIASDLM2	The Language Description File for Model 204 User Language.
VIASDLNR	The Language Description File for Natural V2.1R.
VIASDLNS	The Language Description File for Natural V2.1S.
VIASDLN1	The Language Description File for Natural V1.2.
VIASDLU1	The Language Description File for UNIVAC 9000 ANSI 68 COBOL.
VIASDLU2	The Language Description File for UNIVAC 1100 ANSI 74 COBOL.
VIASDLVS	The Language Description File for IBM DOS VSE COBOL.
VIASDLVX	The Language Description File for DEC VAX ANSI 74 COBOL.
VIASEXTJ	The JCL to assemble and link edit the VIASEXT1 module. REQUIRED FOR AUXILIARY SOURCE MANAGERS (other than PDS, CA-Panvalet, and CA-Librarian).
VIASEXT1	The user exit to support an auxiliary source manager.
VIASEXT3	The user exit used for tracking product and command usage.
VIASEXT4	The user exit used for tracking command usage.
VIASEX3J	The user exit used for tracking product and command usage.

Member	Description
VIASFAIJ	The JCL to assemble and link edit the CA-Librarian VIASFAIR module. REQUIRED FOR CA-Librarian SOURCE MANAGER.
VIASFAIR	The Assembler source program for CA-Librarian.
VIASFAI2	The Assembler program source to interface with the CA-Librarian FAIR release 2.0 modules.
VIASGENR	The JCL used by sites with DBCS terminals that display languages other than Japanese.
VIASIMSJ	The JCL to compile the sample application analyzer user exit VIASIMSU.
VIASIMSU	The source code for sample application analyzer user exit for IMS members.
VIASINSB	The JCL to perform a Batch Insight analyze from an application definition previously defined using Alliance, Recap, or Estimate.
VIASINST	The JCL to allocate ESW libraries and download products from tape.
VIASJCLJ	The JCL to compile the sample application analyzer user exit VIASJCLU.
VIASJCLU	The source code for sample application analyzer user exit for JCL members.
VIASLJB1	The JCL to relink and patch ISRJB1.
VIASLPAJ	The JCL to copy re-entrant modules to MLPA/PLPA.
VIASLPXJ	The JCL to copy non-reentrant modules to the ESW production library.
VIASMLDF	The MLS definition file.
VIASMLSI	The JCL to allocate the VSAM datasets needed for Alternate Language support in Alliance and Estimate.
VIASMLSM	The MLS message file.
VIASMPRT	The JCL to print the ESW message file.
VIASMSCL	The model Endeavor SCL member.
VIASNDVJ	The JCL to install the temporary SVC for Endeavor support.

Member	Description
VIASNDVR	The PROC executed by VIASNDVJ.
VIASNDVX	The sample user exit for Endeavor support.
VIASNDXJ	The JCL to compile sample user exit, VIASNDVX.
VIASPAM	The Assembler source program for CA-Panvalet.
VIASPAMJ	The JCL to assemble and link edit the VIASPAM module.
	<p><b>Note:</b></p> <p>Do not execute VIASPAMJ if you are using CA-Panvalet R12 or above and the module PAM is available in LINKLST or LPA.</p>
VIASPLIJ	The JCL to compile the sample Application Analyzer user exit VIASPLIU.
VIASPLIU	The source code for sample Application Analyzer user exit for PL/I members.
VIASPRTI	The JCL for printing data from the AKR.
VIASSCLA	The Endeavor SCL template for ADD.
VIASSCLL	The Endeavor SCL template for LIST.
VIASSCLR	The Endeavor SCL template for RETRIEVE with expansion of includes.
VIASSCLN	The Endeavor SCL template for RETRIEVE without expansion of includes.
VIASSQL	The ESW SQL Interface, compiled and linked when VIASBIND is run. Allows ESW products to execute SQL statements.
VIATRNC	The JCL to compile ESW training programs.
VIATRND	The data file for ESW training programs.
VIATR NJ	The JCL to execute ESW training programs.
VIATR N1	The COBOL ESW training program.
VIATR N2	The COBOL ESW training program.
VIATR N3	The COBOL ESW training copy member.

Member	Description
VIASSTRN4	The COBOL ESW training program.
VIASSTRN5	The COBOL ESW training program.
VIASSTRN6	The COBOL ESW training copy member.

## Center Skeleton (SLIB) Members

Member	Description
VIASAKAP	Initializes the AKR. This member is not applicable for a SmartEdit-only installation.
VIASAKRA	Generates JCL stream to allocate and initialize an ESW AKR. This member is not applicable for a SmartEdit-only installation.
VIASAKRX	Generates JCL stream to expand an ESW AKR. This member is not applicable for a SmartEdit-only installation.
VIASAKXP	Expands the AKR. This member is not applicable for a SmartEdit-only installation.
VIASJOB	Generates JOB statements for printing Log/List/Punch files.
VIASJRPT	Submits a Batch job to execute the Estimate reports.
VIASJSUM	Submits a Batch job to execute the roll up of summary information in Estimate.
VIASLLSP	Generates JCL stream to print Log/List/Punch files.
VIASLSSB	Generates a module from a logic segment.

## Center CLIST Members

Member	Description
CENTER	Used by the VIACEN CLIST to invoke the ASG-Existing Systems Workbench - ASG-ESW primary screen.
PRODINFO	Displays product release and maintenance level information for all installed ESW products.
VIAACSTM	Performs Encore customization.
VIABCSTM	Invoked by VIASCUST to perform Alliance customization.
VIACEN	Invokes the CENTER CLIST using the application ID (VIAC).
VIADCSTM	Invoked by VIASCUST to perform SmartDoc customization.
VIAEDUSR	Invokes a source manager other than those already handled by the edit options screens.
VIAEPANL	Modifies a copy of the ISPF/PDF 4.1 edit panel DTL(s) to include SmartEdit logic.
VIAGCSTM	Invoked by VIASCUST to perform Bridge customization.
VIAICSTM	Invoked by VIASCUST to perform Insight customization.
VIAJCSTM	Invoked by VIASCUST to perform Estimate customization.
VIAMCSTM	Invoked by VIASCUST to perform AutoChange customization.
VIAOCSTM	Invoked by VIASCUST to perform Client Server customization.
VIAPCSTM	Invoked by VIASCUST to perform SmartTest customization.
VIARCSTM	Invoked by VIASCUST to perform Encore customization.
VIAUCSTM	Invoked by VIASCUST to perform customization of both Bridge and AutoChange.
VIAXCSTM	Invoked by VIASCUST to perform Recap customization.
VIASAJCE	Edit macro used internally. This member is not applicable for a SmartEdit-only installation.
VIASAJCN	Used by VIASAJCS to deactivate the VIASUB command. This member is not applicable for a SmartEdit-only installation.

Member	Description
VIASAJCS	Submits or edits the JCL resulting from an ANALYZE JCL conversion. This member is not applicable for a SmartEdit-only installation.
VIASAJCX	Used internally by VIASUB and VIASUBDS. This member is not applicable for a SmartEdit-only installation.
VIASAJC5	Invokes VIASUBDS on JCL generated by ISPF primary option 5. This member is not applicable for a SmartEdit-only installation.
VIASALCL	Displays the datasets allocated to a TSO session.  <b>Note:</b> See <a href="#">Appendix F, "Listing Allocated Datasets," on page 251</a> for information about using VIASALCL.
VIASBAT	Invoked to generate JCL for the Program Analyzer, AKR allocation, or AKR expansion jobs. This member is not applicable for a SmartEdit-only installation.
VIASBRWS	Transfers control to ISPF browse with the specified dataset (DSN).
VIASCCOM	Used by VIASCUST to perform common customization.
VIASCMGB	Edit macro used by VIASCUST to update VIASGBL.
VIASCMPR	Edit macro used to update CNTL (VIA\$PRMS). This macro is used internally by VIASCUST.
VIASCM11	Edit macro used to perform common customization.
VIASCM12	Edit macro used to perform common customization.
VIASCPRM	Used by VIASCUST to provide the parameters for the common customization.
VIASCUST	Performs common customization and to call the various product-specific customization files to do their own customization.
VIASCSTM	Invoked by VIASCUST to perform Center customization.
VIASDEL	Invoked to delete a Log file.
VIASEDBR	Edits or browses the file specified.
VIASEDID	Invokes the editor for a specified DDNAME.

Member	Description
VIASEDIM	Used internally by the Analyze Submit facility.
VIASEDIT	Invoked to edit generated JCL for the Program Analyzer, AKR allocation or AKR expansion jobs.
VIASGBL	Contains the defaults for the Center dataset names and the ISPF LIBDEF Facility option.
VIASGVFP	Gets variables from the user's ISPF profile.
VIASISP5	Manages the ESW version of the ISPF primary option 5 (Batch - Submit Job for Language Processing).
VIASJEDT	Invokes an ISPF browse session for a specified dataset and member.
VIASLLSP	Invoked to print the Log file.
VIASNINS	Responds to selection of a product that is not installed.
VIASPI14	Used internally by VIASPROF.
VIASPARM	Displays the name of the ESW library containing the Center installation parameter member VIA\$PRMS.
VIASPROC	Adds or removes SYSPROC allocations.
VIASPROF	Converts prior release profiles to Center 6.0.
VIASPRO1	CLIST used internally by VIASPROF.
VIASPRO2	CLIST used internally by VIASPROF.
VIASPRTI	AKR dump utility used for debugging by the ASG Service Desk.
VIASPR14	CLIST used internally by VIASPROF.
VIASPR40	CLIST used internally by VIASPROF.
VIASPR40	CLIST used internally by VIASPROF.
VIASPTFR	Edit macro to facilitate backing out a PTF.
VIASREP	Renames a specified dataset (OLDDSN).
VIASUB	Edit macro for submitting Batch analyze jobs.

Member	Description
VIASUBDS	Submits Batch analyze jobs.
VIASVPTF	Displays the PTFs that have been applied.
VIASVPTX	CLIST used internally by VIASVPTF.
VIAYBRO	Invokes ISPF Browse for application-level products.
VIAYEDIT	Invokes ISPF to edit a query file for Alliance.
VIAYJCLE	Invokes ISPF Browse for application-level products upon a JCL error.



---

## Appendix C

---

# VIASCUST Customization CLIST

VIASCUST assists with customizing CNTL, CLIST, PTF, and ISPSLIB members during product installation. One of the most important things VIASCUST does is invoke these product-specific customization CLISTS:

- VIAACSTM — Encore
- VIABCSTM — Alliance
- VIADCSTM — SmartDoc
- VIAGCSTM — Bridge
- VIAICSTM — Insight
- VIAJCSTM — Estimate
- VIAMCSTM — AutoChange
- VIAOCSTM — Client Server
- VIAPCSTM — SmartTest
- VIASCSTM — Center
- VIAUCSTM — Bridge and AutoChange (shared)
- VIAXCSTM — Recap

Before you invoke VIASCUST, you must supply parameters for the customization.

### *To customize the parameters*

- 1 Edit VIASCPRM in the CLIST library. This member contains several parameter assignments with this format:

```
SET parameter = value
```

- 2 Modify the value portion as appropriate for your site. The parameter descriptions are listed in [VIASCPRM Parameters](#) in the order they appear within this member. Check your changes carefully because potential errors are propagated throughout the libraries. A roll-back feature does not exist; therefore, errors must be manually corrected.

**Note:**

If the value you specify for a parameter is longer than the default value, and if there is not enough room on a line for the specified value in a member being customized, then the default value remains in that member. No error messages are produced. Check the results of VIASCUST carefully.

---

## VIASCPRM Parameters

Parameter	Description
VIASOFT	Specifies the high-level node of the ESW libraries. This should be the same as the VIASOFT symbolic parameter used for the install JCL. The node length is $\leq 8$ and the default is ASG.
CENTER	Specifies the second level node of the ESW libraries. This should be the same as the CENTER symbolic parameter used for the install JCL. The node length is $\leq 8$ and the default is VIACEN $_{xx}$ , where $_{xx}$ is the Center release number.
LIBDEF	Specifies YES if LIBDEF is used to invoke the products, or NO if LIBDEF is not used. If LIBDEF is NO, then the ESW libraries have to be allocated during logon.
ALTLIB	Specifies YES if the TSO ALTLIB facility is used to allocate the ESW CLIST library. If ALTLIB is NO, then the ESW CLIST library must be allocated to the SYSPROC DD.
VIAWORK	Specifies the high-level node of the ESW VSAM work datasets. VIAWORK is the high-level qualifier for work datasets used during an Easytrieve analysis. Users must have update access to these datasets for MLS processing.

Parameter	Description
JOBC1	<p>Specifies the first line of a JOB statement to be inserted into each member that is a job stream, replacing the dummy JOB statement. If you do not want to have these replacements done, then leave the JOBC1 parameter unmodified.</p> <p>A plus sign (+) is used to indicate a continuation to the next line. Also, the value is delimited by &amp;STR (on the left and on the right). For example, to specify a first line of:</p> <pre>//JOBNAME JOB (12345), 'FRED',</pre> <p>the lines for JOBC1 in VIASCPRM should read:</p> <pre>SET JOBC1 = + &amp;STR(//JOBNAME JOB (12345), 'FRED', )</pre>
JOBC2	Specifies the second line of a JOB statement to be inserted into each JCL member.
JOBC3	Specifies the third line of a JOB statement to be inserted into each JCL member.
JOBC4	Specifies the fourth line of a JOB statement to be inserted into each JCL member.
CVSCLIB	Specifies the fully-qualified dataset name of the VS COBOL compiler library. Do not enclose the dataset name in quotes, even though it is fully qualified. If the VS COBOL compiler is fetched from the link list or from the LPA, then blank out the value.
CVSRLIB	Specifies the fully-qualified dataset name of the VS COBOL run time library. Do not enclose the dataset name in quotes, even though it is fully qualified. This parameter is only needed for SmartTest.
CIICLIB	Specifies the fully-qualified dataset name of the COBOL II compiler library. Do not enclose the dataset name in quotes, even though it is fully qualified. If the COBOL II compiler is fetched from the link list or from the LPA, then blank out the value.
CIIRLIB	Specifies the fully-qualified dataset name of the COBOL II run time library. Do not enclose the dataset name in quotes, even though it is fully qualified. This parameter is only needed for SmartTest.
CICS	Specifies the high-level qualifier for the CICS libraries.
USERLIB	Specifies the fully-qualified dataset name of the load library where demo programs are stored. Do not enclose the dataset name in quotes, even though it is fully qualified. This parameter is only needed for SmartTest.

Parameter	Description
LISTUNT	Specifies the default unit device name used online for allocating the VIALIST dataset. The default value is SYSDA.
LOGUNT	Specifies the default unit device name used online for allocating the VIALOG dataset. The default value is SYSDA.
PUNUNT	Specifies the default unit device name used online for allocating the VIAPUNCH dataset. The default value is SYSDA AKRDSN. If there is a common AKR where most users store their analyzed programs, then specify the fully-qualified dataset name of this AKR here. Do not enclose the dataset name in quotes, even though it is fully qualified. If there is no common AKR, or if the dataset name is unknown, then leave the AKRDSN parameter unmodified. In this case, the CNTL member VIASAKRU can be customized as needed for each execution.
ASMBLR	Specifies the load module name of the assembler. The default value of IEV90 is for Assembler H. For High-level Assembler H, the value should be ASMA90.
CMDLIST	Specifies YES if each command executed in a ESW CLIST should be echoed to the screen, or NO if commands should not be echoed. Ordinarily, CMDLIST should be set to YES only for diagnostic purposes.
CIIMOD	Specifies the load module name of the COBOL II compiler. The default value of IGYCRCTL rarely needs to be modified.
CVSMOD	Specifies the load module name of the VS COBOL compiler. The default value of IKFCBL00 rarely needs to be modified.
SYSDA	Specifies the unit name to be used for allocating temporary datasets. This is normally the same as the SYSDA symbolic parameter used for the install JCL.
SYSOUT	Specifies the SYSOUT class to be used for report datasets such as SYSPRINT. This is normally the same as the SYSOUT symbolic parameter used for the install JCL.  The value is delimited by &STR (on the left and) on the right. For example, to specify SYSOUT class X, the line should read:  <code>SET SYSOUT = &amp;STR(X)</code>

Parameter	Description
USERDSQ	Allows a special qualifier name to be appended after the USERID and/or prefix for all product-generated datasets (i.e., Log, List, and Punch files). There is no default. The USERDSQ parm is overridden by the DSN-Qualifier-Node parameter (see " <a href="#">DSN-Qualifier-Node=" on page 197</a> ) located in the VIA\$PRMS CNTL member.
VIAJCL	Specifies the last node to be used for generated JCL dataset names.

## CLIST Invocation

**Note:** \_\_\_\_\_

If you are upgrading an existing Center installation, you must first ensure that SYSPROC is pointing to the correct CLIST library.

**To invoke the customization CLIST**

- 1 Allocate the ESW CLIST library as part of SYSPROC. The VIASPROC job provided in the CLIST library adds the ESW CLIST library to SYSPROC.
- 2 Type TSO VIASCUST on any ISPF screen.

The CLIST edits CNTL and ISPSLIB library members, making the modifications you indicated in VIASCPRM. It also modifies the VIASGBL member in the CLIST library, and the SPZAPJCL member in the PTF library.

Some members being customized belong to Center, while others belong to individual products. When the CLIST tries to customize a member belonging to a product that is not installed, it ignores that member, without reporting any errors. Therefore, there is no need to customize the CLIST to indicate which products are installed.

VIASCUST may take several minutes to execute (perhaps in excess of 30 minutes). For each member processed, VIASCUST sends generated messages to a log file.

**Note:** \_\_\_\_\_

A return code of 4 for VIASCUST indicates the member was not found or no changes were made to the member.



---

## Appendix D

---

# Maintenance

### Problems or Questions

If you have questions or problems:

- 1 Gather all applicable JCL listings, output listings, online log file listings, screen messages or screen prints, and diagnostic information.
- 2 Contact the ASG Service Desk.

### Applying Interim Modifications

Periodic product updates are provided between major releases. Modifications may also be received from the ASG Service Desk as needed. These modifications will most likely be in AMASPZAP format or as object module replacements.

Modifications are to be placed in the ASG.VIACEN<sub>xx</sub>.PTF dataset that was included on the installation tape. This PDS contains two members when installed: SPZAPJCL, which contains the JCL to apply AMASPZAP modifications, and SAMPLPTF, which contains a dummy PTF. You can add PTFs to the ASG.VIACEN<sub>xx</sub>.PTF dataset as needed.

## PTF Display Facility

The PTF Display facility allows you to verify which PTFs have been applied.

### *To display this information on any ISPF screen*

- 1 Execute the VIASVPTF CLIST by typing:

```
TSO VIASVPTF PROD(pprr)
```

where *pp* is the product code and *rr* is the version number. These are the valid product codes and version numbers:

Product Code Release Number	Description
EN60	Encore Version 6.0
AL60	Alliance Version 6.0
BR60	Bridge Version 6.0
CE60	Center Version 6.0
CC60	AutoChange Version 6.0
IN60	Insight Version 6.0
RC60	Recap Version 6.0
DC60	SmartDoc Version 6.0
SE60	SmartEdit Version 6.0
ST60	SmartTest Version 6.0
SM60	SmartTest-IMS Version 6.0
SC60	SmartTest-CICS Version 6.0
TM60	Estimate Version 6.0

**Note:**

Using the VIASVPTF CLIST requires the ESW CLIST library to be available through the standard SYSPROC allocations.

---

- 2 Press Enter. The PTF Display screen, shown in [Figure 27](#), displays.

**Figure 27 • PTF Display Screen**

```
ASG-CENTER Rx.x ----- ASG PTF Display ----- ROW 1 TO 2 OF 2
===>                                           SCROLL ===> CSR

  Userid: ASG                               Date: YY/MM/DD
  Prefix: ASG                               Time: HH:MM

STO30B08 STO30B39 STO30S01 STO30S50 STO30005 STO30250 STO30400 STO30T12
STO30T99 STO30U19 STO30U70 STO30V09 STO30V98 STO30W03 STO30W90
***** BOTTOM OF DATA *****
```



---

## Appendix E

---

### Abend Codes

Abend codes in the range 900 through 999 (X'384' through X'3E7') bypass ESW error recovery, causing the abend to be handled by ISPF or by the system.

These descriptions can be accessed online by typing `HELP ABENDS` on the command line.

Abend Code	Hexadecimal	Descriptions
965	X'3C5'	Unable to intercept program.
967	X'3C7'	The ASG CENTER-AUTHORIZE password was not specified during installation.
968	X'3C8'	An internal error occurred during initialization. Call the ASG Service Desk.
970	X'3CA'	An internal error occurred in package load module processing. Call the ASG Service Desk.
972	X'3CC'	The ESW Edit Monitor encountered a severe error. Call the ASG Service Desk.
975	X'3CF'	Unable to allocate the message file. Additional information is in the Log file. Process the Log file using the Options - Log/List/Punch Definition screen. This abend may be caused by an installation error for the option ASG-Hi-Level-Nodes in the CNTL file VIA\$PRMS or by changing the blocksize for the message file (VIAMSGS) to other than 4096.

**Note:**

DBCS Installations - If you are using a DBCS terminal, ensure that a VIAMALT message file exists on your system.

---

Abend Code	Hexadecimal	Descriptions
976	X'3D0'	Program abend while running under the ESW analyze monitor. Check the job output to determine which step ended with an error. The error is listed in the VIAMRPT DD.
977	X'3D1'	Interrupt active test abend. During a SmartTest session, the attention key was pressed multiple times to interrupt the test. The ATTENTION key needs only to be pressed once to interrupt the test.
978	X'3D2'	User has not connected or communicated with the Batch test session in <i>nn</i> minutes. A time-out has occurred where the time is determined by the SMF job wait time limit.
979	X'3D3'	Internal error occurred when trying to submit a SmartTest Batch connect job for testing. Call the ASG Service Desk.
980	X'3D4'	SmartTest execution after task ended. Call the ASG Service Desk.
984	X'3D8'	Unable to allocate log file for message to be displayed. Verify the Log file Generic unit and Volume serial parameters are valid for your site. These can be found on the Options - Product Allocations pop-up.
985	X'3D9'	Load module VIASISPF not found in the ESW load library. Call the ASG Service Desk.
988	X'3DC'	Unable to determine PCWA address. Call the ASG Service Desk.
994	X'3E2'	VIASPCHS load module not found in ESW load library. Call the ASG Service Desk.
995	X'3E3'	Unable to allocate the VIASBASE parameter library. Check for WTO messages that identify the specific problem. This abend is normally caused by a failure to execute the VIASBASJ job during installation.

Abend Code	Hexadecimal	Descriptions
996	X'3E4'	VIA $\times$ PTCH load module not found in ESW load library, where $\times$ can be one of these: A - Encore B - Alliance D - SmartDoc F - SmartEdit G - Bridge J - Estimate I - Insight M - AutoChange P - SmartTest R - Encore X - Recap  Call the ASG Service Desk.
998	X'3E6'	Abend was specifically requested in ONERROR routine. Give the dump to the ASG Service Desk.

These abend codes in the range 2304 through 2457 (X'900' through X'999') are processed by ESW error recovery, causing diagnostic information to be written to the Log file.

Abend Code	Hexadecimal	Description
2403	X'963'	An error occurred in SmartTest-IMS. The Log file contains additional information.
2404	X'964'	Incorrect parameter given to R\$BS. Process the Log file using the Options - Log/List/Punch Definition pop-up and call the ASG Service Desk.
2406	X'966'	The Attention Key (PA1) was pressed.
2409	X'969'	An error occurred in the Package Manager (i.e., a load module could not be loaded). A WTP message with more information precedes this abend. Verify that the load module dataset contains the specified load module. If the error and solution are not apparent, call the ASG Service Desk.
2417	X'971'	Program ENQUEUE timed out in FML. Internal error. Call the ASG Service Desk.

Abend Code	Hexadecimal	Description
2419	X'973'	An ATTN/PA1 key was detected while SmartTest-CICS internal processing was being performed.
2433	X'981'	ISPF environment not found. Call the ASG Service Desk.
2434	X'982'	Memory management table is not empty. Call the ASG Service Desk.
2435	X'983'	Continuation stack is not empty. Call the ASG Service Desk.
2438	X'986'	VIASPAM (for CA-Panvalet site) or VIASFAIR (for CA-Librarian site) has not been assembled and linked. See <a href="#">"ESW Installation and Customization" on page 9</a> for instructions on customization.
2448	X'990'	PCWAUSER control block not found. Call the ASG Service Desk.
2457	X'999'	Maximum CPU time has expired. Check the value of Online-Max-Function-Time in the default options, and if too small, increase. Otherwise, if you suspect a loop, call the ASG Service Desk. To make a temporary change, type <code>Debug STIMER nnnn</code> on the command line to set the number of CPU seconds. Type <code>Debug STIMER</code> to check the current value.

These abends apply to SmartTest-CICS only. If you receive one of these abends, contact the ASG Service Desk.

Abend Code	Description
VEXE	The SmartTest Command Level requester module VIACEXEC has encountered a logic error. A message may have been written to the CSSL that provides basic information about the request failure. Obtain the transaction dump and contact the ASG Service Desk.
VIA1	Internal error processing a temporary storage control record.
VIA2	Internal error, unable to find the SmartTest control blocks on the user TCA chain.
VIA3	Internal error, unexpected module encountered in the Return processing logic. Expected DFHEIP.
VIA4	Internal error, unable to find SmartTest control blocks.

Abend Code	Description
VIA5	The SmartTest-CICS monitor VIACEMAN has been corrupted.
VIA6	Internal error found while processing the VIACTBLS file. Possible installation error or storage corruption by another task.
VIA7	Internal error processing a File Utility request.
VIA8	SmartTest modules could not be loaded, or they are incompatible with the TSO components. Ensure that the module VIACEMT1 reflects the current release for SmartTest and for the CICS region in which it is running.



---

## Appendix F

---

# Listing Allocated Datasets

During standard operation, ESW products allocate required datasets as needed. In debugging install-related problems, you may find it helpful to list allocated datasets using the VIASALCL CLIST.

**Note:**

ESW product datasets must be allocated before you can run VIASALCL.

VIASALCL lists these types of allocated datasets for the user executing it:

- TSO
- LINKLIST
- LPALIST

## Primary Commands

Command	Description
DD	Changes the current display argument. The default argument is an asterisk (*), which displays all lines. For example:  DD SYSPROC (shows only SYSPROC allocations) DD VIA* (shows all DDs beginning with VIA)
REFRESH	Rebuilds all allocation lists.
RESET	Resets row messages.
RESETX	Resets all the excluded lines.

Command	Description
VIEW	Changes the current view argument. Valid arguments are TSO, LINKLIST, and LPALIST. When displaying a LINKLIST or LPALIST view, IPL information displays. For example:  VIEW TSO VIEW LINKLIST VIEW LPALIST
WHEREIS	Searches within the current view for the <i>membername</i> specified in all datasets of the current DD display argument. For example:  WHEREIS < <i>membername</i> >

## Line Commands

Line Command	Description
B or S	Invokes ISPF Browse for the selected dataset.
E	Invokes ISPF Edit for the selected dataset.
F	Deallocates the selected DD.
I	Displays dataset information.
R	Resets a single excluded line.
X	Excludes a dataset from the display.

You can use line commands on multiple lines. For example, if you type E on several lines, ISPF Edit displays the selected datasets serially. PF3 takes you to the next selection.

**Note:** \_\_\_\_\_

To save changes to edited datasets, you must possess appropriate update authorization. ESW products do not extend or circumvent software security packages or standard security procedures.

---

---

## Appendix G

---

# User Exit Changes

**Note:** \_\_\_\_\_

This appendix applies only to sites using user exits and upgrading from a Center release prior to release 4.4.

---

The parameter layout for Application Analyze User Exits has been modified to coincide with IBM's Language Environment LE/370. The linkage section of each user exit includes a new two-byte PARM-LENGTH field. The affected user exits reside in the ESW CNTL library and are named:

- VIASASMU
- VIASALFU
- VIASCCSU
- VIASCOBU
- VIASIMSU
- VIASJCLU
- VIASNDVX
- VIASPLIU

If your site uses a user exit from a Center release prior to Release 4.4, ASG recommends you recompile and relink the user exit with the new PARM-LENGTH field in the linkage section.

***To continue using your current user exit***

- 1 Specify OLDUSEREXIT either in the application definition (as a compile exec parm/user exit parm) or as an analyze submit option.
- 2 Run the analyze with a run-time library for the user exit that is older than IBM's LE/370 run-time libraries for OS/390.

Running an analyze while using the new LE/370 run-time library and the OLDUSEREXIT option causes the system to issue CEE3606, CEE3608, and CEE3611 run-time messages. If you run an analyze with a user exit that uses the old parameter layout without specifying the OLDUSEREXIT option, the parms the Application Analyzer passes to the user exit are off by two bytes.

**Note:** \_\_\_\_\_

VIASNDVX is a user exit that can be invoked after extracting source code from Endeavor. With the VIASNDVX user exit, the analyze option OLDUSEREXIT does not apply. If you are using the VIASNDVX user exit, you must recompile VIASNDVX with the extra two-byte PARM-LENGTH field in the linkage section.

---

---

## Appendix H

---

# DBCS Addendum

**Note:** \_\_\_\_\_

This appendix applies only to sites using double-byte character sets (DBCS).

## Introduction

The ESW products are distributed with a set of Japanese translated screens and messages. DBCS environments other than Japan must simulate translated screens and messages using the existing English versions. This appendix describes the installation process for the Japanese and non-Japanese DBCS installations.

## Downloading the Product Tape

To download the Japanese translations of the messages and screens, follow this step:

- ▶ Uncomment the JCL line containing ALTPM='NE' in the installation JCL jobstream.

If you are installing at a non-Japanese DBCS site, there are no special changes required to the download JCL.

## Copying the Message File

**Note:**

If the Japanese translated messages have been downloaded, skip this procedure.

An alternate message library must exist to properly display informational messages in a DBCS environment. The English messages can be copied to an alternate message library using the JCL contained in the ASG.VIACENxx.CNTL(VIASGENR) member. Verify that the high-level node, second level node, dataset unit name, and dataset volume name are correct for your site. Then execute VIASGENR. VIASGENR copies VIAMSGS to VIAMALT.

## Modifying VIA\$PRMS

These parameters are contained in the ASG.VIACExx.CNTL(VIA\$PRMS) member. They should be reviewed and changed where needed for DBCS installations.

Parameter	Description
Character-Vertical-Bar=	Some DBCS printers do not have the vertical bar character ( ) available. If this is the case, this parameter should be changed to an uppercase I or to some other appropriate character.
Format-Date=MDY	This parameter determines the date format on reports and should be changed according to your site standards. Replace the MDY (Month Day Year) with either DMY or YMD.
Format-Time=12HOUR	Depending upon site standards, you may want to change this parameter to 24HOUR.
Language=ENGLISH	This parameter is used when generating messages in Batch analyze reports. If the Japanese translation of the products is being used, change this parameter to Language=JAPANESE. All other DBCS installations should leave this parameter set to Language=ENGLISH.

## **Adding ESW Products to the TSO/ISPF Environment**

This section describes the allocation of the ESW panel, Skeleton, Table, and Message file libraries as they relate to DBCS installations. If both the English and the DBCS procedures are performed as outlined, the product functions in either usage environment.

These are the three main methods of allocating the ISPF libraries:

- LIBDEF facility
- TSO logon procedure
- CLIST executed after logon

**Note:** \_\_\_\_\_

If you use the LIBDEF facility, the DBCS issues are automatically handled within the ESW product CLISTs. Follow the installation procedures outlined in ["ESW Installation and Customization" on page 9](#). There should be no other changes required.

---

See ["Step 1 - Adding the ESW CLIST Library to SYSPROC" on page 25](#) for procedures that describe adding the ESW CLIST library to your logon procedure.

These are the references to the procedures for allocating these libraries:

- Panel libraries is shown in [Figure 28 on page 258](#).
- Skeleton libraries is shown in [Figure 29 on page 259](#).
- Tables libraries is shown in [Figure 30 on page 259](#).

See ["Logon Procedure Samples" on page 260](#) for more information.

Figure 28 • Panel Library Allocations

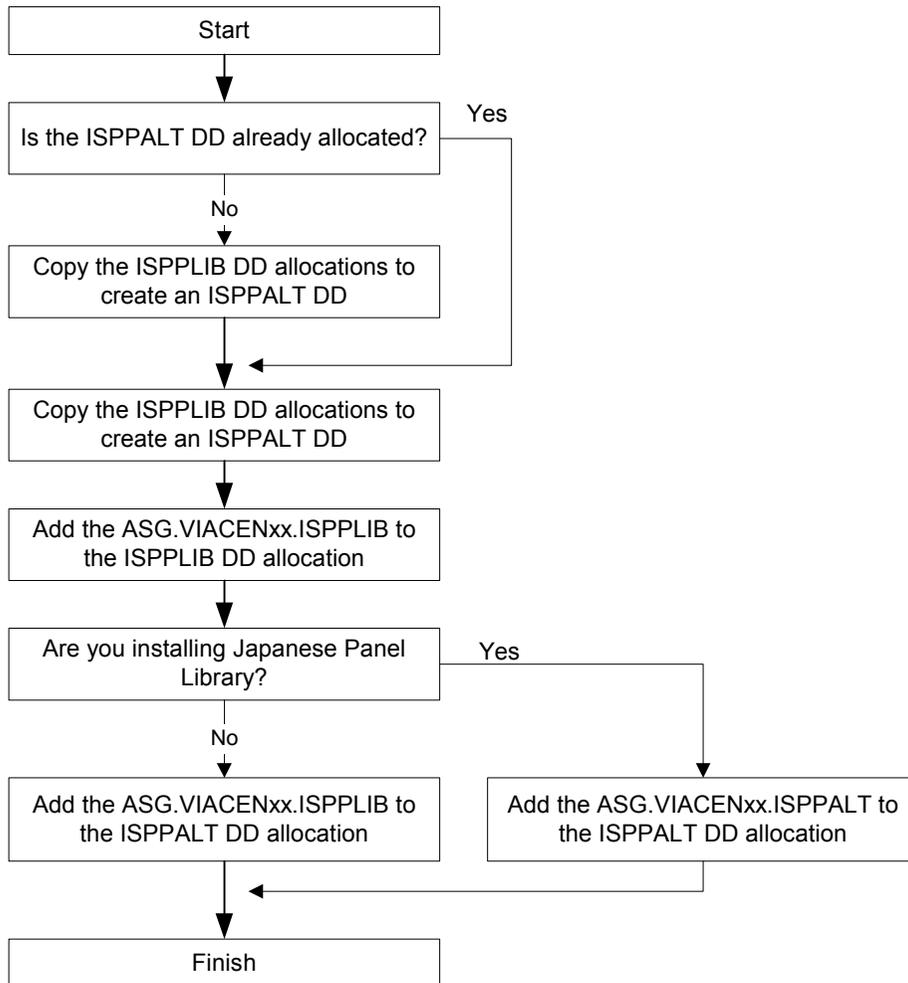


Figure 29 • Skeleton Library Allocations

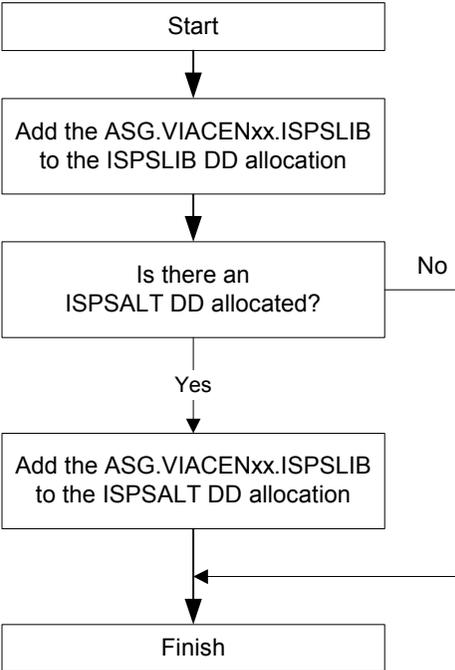
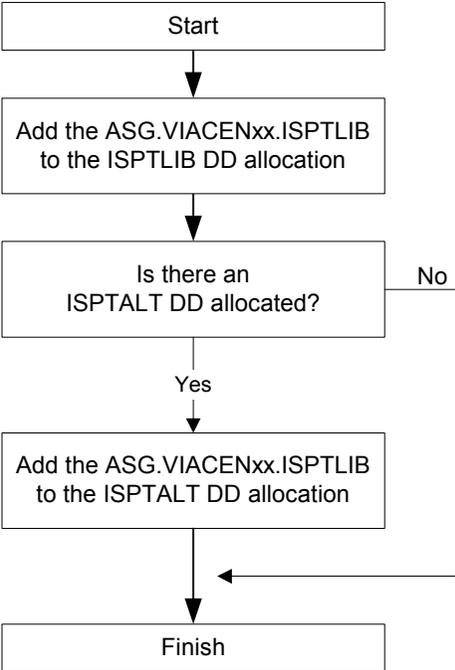


Figure 30 • Table Library Allocations



## Logon Procedure Samples

These samples show changes to a typical logon procedure. Users who allocate the ESW libraries using a CLIST should perform the same changes in their CLIST environments.

### Sample 1: Implementing DBCS (Non-Japanese)

If Japanese translation libraries are being installed, you can skip this sample and continue to Sample 2.

**Note:** \_\_\_\_\_

In the examples below, lines marked with <====Added have been added to the original procedure. This example shows only the necessary part of the logon procedure, there may be other portions that are not affected by the Center installation and that are not shown.

```
//SYSPROC DD DSN=SYS1.CMDPROC,DISP=SHR
//          DD DSN=SYS1.UTIL.CLIST,DISP=SHR
//          DD DSN=ASG.VIACENxx.CLIST,DISP=SHR      <====Added
//ISPLLIB  DD DSN=ISP.V3R3M0.ISPLLIB,DISP=SHR
//          DD DSN=ISR.V3R3M0.ISRLLIB,DISP=SHR
//          DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR    <====Added
//ISPLLIB  DD DSN=ISP.V3R3M0.ISPPLIB,DISP=SHR
//          DD DSN=ISR.V3R3M0.ISRPLIB,DISP=SHR
//          DD DSN=ASG.VIACENxx.ISPPLIB,DISP=SHR   <====Added
//ISPTLIB  DD DSN=ISP.V3R3M0.ISPTLIB,DISP=SHR
//          DD DSN=ISR.V3R3M0.ISRTLIB,DISP=SHR
//          DD DSN=ASG.VIACENxx.ISPTLIB,DISP=SHR   <====Added
//ISPSLIB  DD DSN=ISP.V3R3M0.ISPSLIB,DISP=SHR
//          DD DSN=ISR.V3R3M0.ISRSLIB,DISP=SHR
//          DD DSN=ASG.VIACENxx.ISPSLIB,DISP=SHR   <====Added
//ISPPALT  DD DSN=ISP.V3R3M0.ISPPCHU,DISP=SHR
//          DD DSN=ISR.V3R3M0.ISPPCHU,DISP=SHR
//          DD DSN=ASG.VIACENxx.ISPPLIB,DISP=SHR   <====Added
```

If an ISPPALT DD was not previously allocated in the logon procedure the new one would consist of:

```
//ISPPALT  DD DSN=ISP.V3R3M0.ISPPLIB,DISP=SHR      <====Added
//          DD DSN=ISR.V3R3M0.ISPPLIB,DISP=SHR      <====Added
//          DD DSN=ASG.VIACENxx.ISPPLIB,DISP=SHR      <====Added
```

Once an ISPPALT DD is allocated, ISPF uses it for any DBCS terminal. Therefore, when creating a new ISPPALT allocation, it is extremely important to copy the ISPF files required for normal operation from the ISPLLIB, and then add the new ESW alternate library to it.

If an alternate Table and Skeleton library was required, then it would be created using the ISPTLIB and ISPSLIB as a model. These lines would look like:

```
//ISPTALT DD DSN=ISP.V3R3M0.ISPTLIB,DISP=SHR <===Added
// DD DSN=ISR.V3R3M0.ISRTLIB,DISP=SHR <===Added
// DD DSN=ASG.VIACENxx.ISPTLIB,DISP=SHR <===Added
//ISPSALT DD DSN=ISP.V3R3M0.ISPSLIB,DISP=SHR <===Added
// DD DSN=ISR.V3R3M0.ISRSLIB,DISP=SHR <===Added
// DD DSN=ASG.VIACENxx.ISPSLIB,DISP=SHR <===Added
```

In these examples, the English panel library was used in the Alternate library definition because there was no alternate translation available. Therefore, the DBCS requirement for an alternate panel library was satisfied; however, the English panels are presented regardless of the terminal type used.

## Sample 2: Implementing DBCS (Japanese)

### Note:

In these examples, the lines marked with <===Added have been added to the original procedure. This example shows only the necessary part of the logon procedure. There may be other portions that are not affected by the Center installation and that are not shown.

```
//SYSPROC DD DSN=SYS1.CMDPROC,DISP=SHR
// DD DSN=SYS1.UTIL.CLIST,DISP=SHR
// DD DSN=ASG.VIACENxx.CLIST,DISP=SHR <===Added
//ISPLLIB DD DSN=ISP.V3R3M0.ISPLLIB,DISP=SHR
// DD DSN=ISR.V3R3M0.ISRLLIB,DISP=SHR
// DD DSN=ASG.VIACENxx.LOADLIB,DISP=SHR <===Added
//ISPPLIB DD DSN=ISP.V3R3M0.ISPPLIB,DISP=SHR
// DD DSN=ISR.V3R3M0.ISRPLIB,DISP=SHR
// DD DSN=ASG.VIACENxx.ISPPLIB,DISP=SHR <===Added
//ISPTLIB DD DSN=ISP.V3R3M0.ISPTLIB,DISP=SHR
// DD DSN=ISR.V3R3M0.ISRTLIB,DISP=SHR
// DD DSN=ASG.VIACENxx.ISPTLIB,DISP=SHR <===Added
//ISPSLIB DD DSN=ISP.V3R3M0.ISPSLIB,DISP=SHR
// DD DSN=ISR.V3R3M0.ISRSLIB,DISP=SHR
// DD DSN=ASG.VIACENxx.ISPSLIB,DISP=SHR <===Added
//ISPPALT DD DSN=ISP.V3R3M0.ISPPJPN,DISP=SHR
// DD DSN=ISR.V3R3M0.ISPPJPN,DISP=SHR
// DD DSN=ASG.VIACENxx.ISPPALT,DISP=SHR <===Added
```

If you did not previously allocate an ISPPALT DD in the logon procedure, the new one would consist of:

```
//ISPPALT DD DSN=ISP.V3R3M0.ISPPLIB,DISP=SHR <====Added
//          DD DSN=ISR.V3R3M0.ISPPLIB,DISP=SHR <====Added
//          DD DSN=ASG.VIACENxx.ISPPLIB,DISP=SHR <====Added
```

Once you allocate an ISPPALT DD, ISPF uses it for any DBCS terminal. Therefore, when creating a new ISPPALT allocation, it is extremely important to copy those ISPF files required for normal operation from the ISPPLIB, and add the new ESW alternate library to it.

If an alternate Table and Skeleton library was required, then it would be created using the ISPTLIB and ISPSLIB as a model. These lines would look like:

```
//ISPTALT DD DSN=ISP.V3R3M0.ISPTLIB,DISP=SHR <====Added
//          DD DSN=ISR.V3R3M0.ISRSLIB,DISP=SHR <====Added
//          DD DSN=ASG.VIACENxx.ISPTLIB,DISP=SHR <====Added
//ISPSALT DD DSN=ISP.V3R3M0.ISPSLIB,DISP=SHR <====Added
//          DD DSN=ISR.V3R3M0.ISRSLIB,DISP=SHR <====Added
//          DD DSN=ASG.VIACENxx.ISPSLIB,DISP=SHR <====Added
```

In these examples, the Japanese panel library was used in the Alternate library definition; however, because no translation is available for the Table and Skeleton libraries, the English libraries were used in those allocations.

## Message File Allocations

The ESW message file is not in the same format as the ISPF message files that you can allocate under ISPMLIB. Because of the format difference, never add the ASG.VIACENxx.VIAMSGS and ASG.VIACENxx.VIAMALT files to an ISPMLIB or ISPMALT allocation in a logon procedure.

The ESW message file is dynamically allocated from within the ESW products. This allocation is done using the high-level qualifiers specified in the ASG.VIACENxx.CNTL(VIA\$PRMS) member with the ASG-Hi-Level-Nodes= parameter. These high-level node names are concatenated with .VIAMSGS for the English message file and with .VIAMALT for the alternate message file.

**Note:** \_\_\_\_\_

If you installed the Japanese translation libraries, the translated message file should already exist under the name of ASG.VIACENxx.VIAMALT.

---

If you are not installing the Japanese translations, execute the Copy Message File discussed in ["Copying the Message File" on page 256](#) to create a VIAMALT file.

If ESW cannot locate the proper message file during ESW product startup, a U975 abend occurs.

## **Detecting and Handling DBCS Terminals**

These are the three triggers that affect the ESW products and create changes from a SBCS to DBCS environment:

- VTAM Queriable terminals
- ISPF Terminal type of 3278KN
- Language=JAPANESE parameter in VIA\$PRMS

### ***VTAM Queriable Terminals***

During startup, ISPF queries the terminal capabilities through VTAM. If a terminal is Queriable, the DBCS attributes are passed back through VTAM to ISPF. This causes ISPF to use alternate language support libraries if they are allocated. This also causes ESW products to use the alternate screen and message libraries. If a terminal has DBCS capabilities and has signed onto TSO as non-queriable, the DBCS qualities of the terminal cannot be determined by ISPF and the terminal will therefore be used as a SBCS terminal.

### ***ISPF Terminal Type of 3278KN***

The terminal type can be specified to ISPF through one of the ISPF 0.x options. If a terminal type of 3278KN is used, the ESW products uppercase all screens and messages before they are sent to the terminal. Some languages, and in particular the DBCS languages, use the English lowercase EBCDIC character codes for other language alphabet characters. Using this terminal type ensures that all ESW product screens are readable.

### ***Language=JAPANESE Parameter in VIA\$PRMS***

This parameter causes alternate language messages to be used in Batch analyze reports.



---

# Appendix I

---

## Using Process Support

### Overview

ESW process support provides the ability to view AKRs, open AKR directories, and view members defined to an AKR. The AKR members are identified as enterprise, application, and program members. You can perform various actions (i.e., analysis or edit) on the members.

These are some exceptions to viewing AKR information:

- Internal members are not displayed.
- AutoChange program members are not displayed. Although AutoChange programs are analyzed by Insight and can be viewed in Insight, all phases in AutoChange have a hierarchical dependency. A simple analyze of an AutoChange member may cause the program to have to repeat the Plan and Apply phases. The presence of an AutoChange member within the AKR is identified using the name ACENTITY. Selecting this member invokes AutoChange.
- Bridge DCD members are not listed in the AKR directory. Bridge definitions contain copybooks, not programs, and there is no analyze involved. The presence of a Bridge member within the AKR is identified using the name BREntity. Selecting this member invokes Bridge.

**Note:** \_\_\_\_\_  
Process support is available through the ESW primary screen.  
\_\_\_\_\_

## Listing AKRs

An AKR list is available from the File pull-down through a wild card selection feature, such as list all AKRs that meet criteria VIA\*.\*.AKR. The list is sorted by AKR names. You can also list the most recently used AKRs by user. The list is sorted in most recently used order.

### To list AKRs

- 1 Select File ► List AKRs. The Display AKR Data Set List pop-up, shown in [Figure 31](#), displays. Enter the dataset name qualifier for the AKRs you want to list. For example, if you specify VIA123, all datasets that have VIA123 as a high-level qualifier, and AKR as the last node, will be displayed. You can also use an asterisk (\*) as a wildcard character.

**Figure 31 • Display AKR Data Set List Pop-up**

```

Display AKR List
Command ==> _____
Specify an AKR name level, * for wild card and press Enter
to continue.
AKR Name Level  VIA123
    
```

Press Enter. The AKR List screen, shown in [Figure 32](#), displays.

**Or**

Select File ► Recent AKRs. The AKR List screen displays the last five AKRs that you used.

**Figure 32 • AKR List Screen**

```

Options  Help
-----
Command ==> _____ AKR LIST Scroll ==> CSR
Select an AKR data set then press Enter. 1 of 5

AKR Name                               --Number of--   Auto   >
Ent Appl  Pgm  Brg  chg   >
-----
- VIA123.CE44T205.AKR                0   1   0 NO  NO
- VIA123.CE60CC60.AKR                0   0   1 YES YES
- VIA123.CE50CC51.AKR                1  13   1 YES YES
- VIA123.CE45L000.AKR                0   0   3 NO  YES
- VIA123.CE45L031.AKR                0   0   1 NO  YES
***** BOTTOM OF DATA *****
    
```

- 2 Scroll right to see additional information, as shown in [Figure 33](#).

**Figure 33 • AKR List Screen (Scrolled Right)**

```

Options  Help
-----
                                AKR LIST
Command ==> _____ Scroll ==> CSR
Select an AKR data set then press Enter.                                1 of 5
<
< AKR Name                                Last      --Records--
                                Update    Alloc  Used   Volume
-----
- VIA123.CE44T205.AKR                05JAN2000   180  45.0% SRT868
- VIA123.CE56CC60.AKR                26OCT2000   408  30.4% SRT802
- VIA123.CE50CC51.AKR                11JUN2001   408  40.2% SRT820
- VIA123.CE45L000.AKR                14OCT1999   408  76.2% SRT900
- VIA123.CE45L031.AKR                10JAN2000   408  30.6% SRT852
***** BOTTOM OF DATA *****

```

**Fields**

Field	Description
Ent	Specifies the number of enterprises defined in the AKR.
Appl	Specifies the number of applications defined in the AKR.
Pgm	Specifies the number of programs defined in the AKR.
Brg	Indicates whether the AKR includes Bridge definitions (Yes/No).
AutoChg	Indicates whether the AKR includes Conversion Sets (Yes/No).
Last Update	Specifies the date the AKR was last updated.
Records Alloc	Indicates the total records allocated for the AKR.
Records Used	Specifies the records used by the AKR, rounded to the nearest .1 percent.
Volume	Indicates the volume name for the AKR. If the AKR is migrated or not accessible, that status is shown in this field.

## Actions

Use these actions on the AKR List screen:

Action	Description
slash (/)	Displays the AKR List Actions pop-up, shown in <a href="#">Figure 34</a> . Use this screen to select the process to be performed on the AKR.
S	Displays the AKR Member List screen, shown in <a href="#">Figure 37 on page 269</a> . Use this screen to view the contents of an AKR.
M	Displays the ASG-ESW AKR Utility pop-up, shown in <a href="#">Figure 35</a> . Use this screen to perform utility actions on the AKR.

**Figure 34 • Data Set List Actions Pop-up**

```

                                AKR List Actions

AKR Name : VIA123.CE50CC51.AKR

AKR List Action
—  1.  Select AKR
   2.  Manage AKR

Select a choice and press ENTER to process AKR action.
    
```

**Figure 35 • ASG-ESW - AKR Utility**

```

                                ASG-ESW - AKR Utility

Command ==> _____

      Blank - Display member list      D - Delete member
      A     - Allocate/expand AKR      R - Rename member

Application Knowledge Repository (AKR):

Data set name . . 'VIA123.CE50CC51.AKR'
Member . . . . . _____ (if "R" or "D" selected)
New name . . . . . _____ (if "R" selected)

Volume serial . . _____ (if not cataloged)
Password . . . . . _____ (if password protected)
    
```

If you select a migrated AKR from the AKR List screen, you receive the confirmation pop-up shown in [Figure 36](#), which asks you to select an action.

**Figure 36 • AKR Action Confirmation Pop-up**

```

                                AKR Action Confirmation

The AKR you selected has been migrated, press ENTER to recall the data set
and continue, or press PF3 to cancel the action.
    
```

## Viewing an AKR

When you select an AKR to view from the AKR List screen, the AKR Member List screen displays. Because you can share AKRs across ESW products, information for different products can be stored in the same AKR. The AKR Member List screen enables you to filter this information. You can choose to list enterprise members, application members, and/or programs.

### To view an AKR

- 1 From the AKR List screen, type S next to an AKR name and press Enter. The AKR Member List screen, shown in [Figure 37](#), displays.

Figure 37 • AKR Member List Screen

```

Options  Help
-----
AKR Member List
Command ==> _____ Scroll ==> CSR
AKR Name : VIACPQ.CE50CC51.AKR                      1 of 16

  Member      Member      Analyze   - Num of -   -----   Last Update   -----
  Name        Level        Type      App/Pgm/Ln   Date        Time        Jobname
-----
- ACENTITY    AutoChange   N/A
- BRENTITY    Bridge       N/A
- CONNIE      Application   TM          0 11JUN2001  14:46:02  VIA1234
- CONNIETM    Enterprise   E,TM        3 11JUN2001  14:42:40  VIA1234
- CONNIE1     Application   TM          1 11JUN2001  14:34:53  VIA1234
- RICH        Application   RC          0 09JUN2001  23:42:14  VIA1234
- RICHARD     Application   AL          0 11JUN2001  14:29:51  VIA1234
- RICHARD1    Application   TM          0 14MAY2001  16:48:35  VIA1234
- RICHARD5    Application   RC,AL,TM    0 14MAY2001  16:48:35  VIA1234
- RICHRICH    Application   RC,AL,TM    0 24MAY2001  16:21:52  VIA1234
- RICH2       Application   RC          0 04JUN2001  15:37:56  VIA1234
- RICH3       Application   AL          0 04JUN2001  15:49:43  VIA1234
- RICH4       Application   AL          0 04JUN2001  15:49:46  VIA1234
- RICH5       Application   AL          0 04JUN2001  16:42:16  VIA1234
-

```

Field	Description
Member Name	Specifies the enterprise name, the application name, or the program name. The name BRENTITY indicates a Bridge entity, and the name ACENTITY indicates an AutoChange entity.
Member Level	Indicates that the member is an enterprise, an application, an entity, or individual program.
<b>Note:</b>	Program members defined to the AKR using AutoChange do not display because they are part of the Conversion set.

Field	Description
Analyze Type	Specifies the type of analyze that was performed on the member. These are the valid values: <ul style="list-style-type: none"><li>• AL (Alliance)</li><li>• T or TM (Estimate)</li><li>• RC (Recap)</li><li>• I or IN (Insight)</li><li>• ST (SmartTest)</li><li>• STX (SmartTest with extended analysis)</li><li>• DS (SmartDoc)</li><li>• DX (SmartDoc Extended)</li><li>• DS (SmartDoc with COBOL compile)</li><li>• DA (SmartDoc Extended with COBOL compile)</li><li>• EN (Encore)</li></ul>
Number of App/Pgm/Ln	Specifies the number of applications in the enterprise, the number of programs in the application, or the number of lines in the program.
Last Update Date	Specifies the date the application or program was last updated.
Last Update Time	Specifies the time the application or program was last updated.
Last Update Jobname	The TSO ID the application or program was last updated.

- 2** Type / or S next to a member name and press Enter. Depending on the member level you selected, the appropriate AKR Member Action pop-up, displays.

All actions are not available for all member types. If an action is not applicable to the member, it does not display. If the action displays with a status of Unavailable, either you do not have the ESW product(s) necessary to perform the action, or the member was not analyzed with that product. You receive a message if you attempt to select them.

Selecting an	Displays the...
Enterprise member	AKR Enterprise Member Actions pop-up, shown in <a href="#">Figure 38</a> .
Application member	AKR Application Member Actions pop-up, shown in <a href="#">Figure 39</a> .
Program member	AKR Program Member Actions pop-up, shown in <a href="#">Figure 40 on page 272</a> .
AutoChange or Bridge member	AKR Bridge/AutoChange Action pop-up, shown in <a href="#">Figure 41 on page 272</a> .

**Figure 38 • AKR Enterprise Member Actions Pop-up**

```

AKR Enterprise Member Actions

AKR Name : VIA123.CE50CC51.AKR
Member . : CONNIETM

Member Action
— 1. Analyze
   2. Estimate

Select a choice and press ENTER to process action.

```

**Figure 39 • AKR Application Member Actions Pop-up**

```

AKR Application Member Actions

AKR Name : VIA123.CE50CC51.AKR
Member . : CONNIE1

Member Action
— 1. Analyze
   2. Understand          (Unavailable)
   3. Measure             (Unavailable)

Select a choice and press ENTER to process action.

```

Figure 40 • AKR Program Member Actions Pop-up

```
AKR Program Member Actions

AKR Name : VIA123.CE50CC51.AKR
Member . : NEWDEMO

Member Action
— 1. Analyze
   2. Understand (Unavailable)
   3. Change (Unavailable)
   4. Test (Unavailable)
   5. Document (Unavailable)
   6. Re-engineer (Unavailable)

Select a choice and press ENTER to process action.
```

Figure 41 • AKR Bridge/AutoChange Action Pop-up

```
AKR Bridge/AutoChange Action

AKR Name : VIA123.CE50CC50.AKR
Member . : ACENTITY

Member Action
— 1. Change

Select a choice and press ENTER to process action.
```

### Member Actions

These sections describe the actions available for each type of member. These behaviors apply to the actions:

- If an action is marked Unavailable, it indicates that you do not have the necessary ESW product installed to perform the action, or that the member was not analyzed with that product.
- After you select an action that invokes an ESW product, you are in that product and backtracking (using PF3) displays the screens for that particular product. To view the process support screens again, you must exit from the product and return to the ESW primary screen.

### Enterprise Members

Action	Description
Analyze	Displays the Analyze - Enterprise Member List pop-up from which you can select an application or program to analyze. This option is available only when Estimate is installed.
Change	Not applicable to enterprise level members.
Document	Not applicable to enterprise level members.

Action	Description
Estimate	Invokes Estimate and displays the applications and programs on the Enterprise View screen. This option is available only when Estimate is installed.
Measure	Not applicable to enterprise level members.
Reengineer	Not applicable to enterprise level members.
Test	Not applicable to enterprise level members.
Understand	Not applicable to enterprise level members.

### Application Members

Action	Description
Analyze	Displays the Analyze - Select Analysis Type pop-up, shown in <a href="#">Figure 47 on page 276</a> , displays. If you have Recap installed, the Analyze - Select Analysis Type pop-up, shown in <a href="#">Figure 48 on page 276</a> , displays.
Change	Not applicable to application level members.
Document	Not applicable to application level members.
Measure	Invokes Recap and displays the Application Maintenance Facility screen.
Reengineer	Not applicable to application level members.
Test	Not applicable to application level members.
Understand	Invokes Alliance and displays the Application Maintenance Facility screen.

### Program Members

Action	Description
Analyze	Invokes the ESW - Prepare Program pop-up, shown in <a href="#">Figure 50 on page 277</a> .
Change	Invokes SmartEdit and displays the Copy/Include Libraries screen.
Document	Invokes SmartDoc and displays the primary product screen.

Action	Description
Measure	Not applicable to program level members.
Reengineer	Invokes Encore and displays the primary product screen, with the selected program opened. The message PROGRAM XXXXXXXX SUCCESSFULLY OPENED displays.
Test	Invokes SmartTest and displays the Environment Selection screen with the AKR name specified.
Understand	Invokes Insight and displays the Source View screen.

### AutoChange/Bridge Members

Action	Description
<u>C</u> hange	Invokes the AutoChange or Bridge primary screen. Change is the only action available for AutoChange (ACENTITY) and Bridge (BRENTITY) members.

## Analyzing an Enterprise Member

### To analyze an enterprise member

- 1 Select an enterprise member on the AKR Member List screen, by typing / or S next to a member name pressing Enter. The AKR Enterprise Member Actions pop-up, shown in [Figure 38 on page 271](#), displays.
- 2 Select Analyze to perform the Analyze action on an application within the enterprise member. Estimate is invoked and the Analyze - Enterprise Member List pop-up, shown in [Figure 42](#), displays.

**Figure 42 • Analyze - Enterprise Member List Pop-up**

```

Options  Help
-----
              Analyze - Enterprise Member List
Command ==>> _____ Scroll ==>> CSR

AKR Name : VIA123.CE60CC51.AKR
Ent Name : CONNIETM                               1 of 3

  App Name
  -----
  CONNIE
  CONNIE1
  RICHARD
***** BOTTOM OF DATA *****

```

- 3 Select an application from the Analyze - Enterprise Member List screen, by typing / or S next to a member name and pressing Enter. The Application Analyze pop-up, shown in [Figure 43](#), displays.

**Figure 43 • Application Analyze**

```

Application Analyze

_ 1. Select analysis type...
   2. Submit application analysis batch job...
   3. Exit

```

- 4 Choose the Select Analyze type action and press Enter. The Analyze - Select Analysis Type pop-up, shown in [Figure 44](#), displays.

**Figure 44 • Analyze - Select Analysis Type Pop-up (without Recap Analyze)**

```

Analyze - Select Analysis Type

Select the desired type of analysis
_ 1. Full analysis... all libraries/members
   2. Incremental... select libraries/members
   3. Auto analysis... auto-selected members
   4. Exit

```

- 5 Select the type of Analysis and press Enter. You receive a confirmation screen depending on the type of analysis you selected. Press Enter to confirm and return to the Application Analyze pop-up.
- 6 Select Submit application analysis batch job. The Analyze Submit Application pop-up, shown in [Figure 45](#), displays.

**Figure 45 • Analyze - Submit Application Screen for an Enterprise Member**

```

Analyze - Submit Application                                USER2

Specify option and jobcard information. Then press PF key for action.

AKR: 'VIAUSR.TEST.AKR'

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

Do Semantic Linking (Y/N)? N

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

PF4=Options  PF5=Edit JCL  PF6=Submit JCL

```

## Analyzing an Application Member

### To analyze an application member

- 1 Select an application on the AKR Member List screen by typing / or S next to a member name and pressing Enter. The AKR Application Member Actions pop-up, similar to the one shown in [Figure 39 on page 271](#), displays.
- 2 Select the Analyze action. The Application Analyze pop-up, shown in [Figure 46](#), displays.

**Figure 46 • Application Analyze Pop-up**

```
Application Analyze

1. Select analysis type...
2. Submit application analysis batch job...
3. Exit
```

- 3 Type 1 for the Select analysis type option and press Enter. Depending on whether you have Recap installed, the Analyze - Select Analysis Type pop-up, shown in [Figure 47](#) or [Figure 48](#), displays.

**Figure 47 • Analyze - Select Analysis Type Pop-up (without Recap Analyze)**

```
Analyze - Select Analysis Type

Select the desired type of analysis
_ 1. Full analysis... all libraries/members
   2. Incremental... select libraries/members
   3. Auto analysis... auto-selected members
   4. Exit
```

**Figure 48 • Analyze - Select Analysis Type Pop-up (with Recap Analyze)**

```
Analyze - Select Analysis Type

Select the desired type of analysis
_ 1. Full analysis... all libraries/members (create new version)
   2. Partial... select libraries/members (create new version)
   3. Incremental... select libraries/members (update old version)
   4. Auto analysis... auto-selected members (update old version)
   5. Exit
```

- 4 Select the type of Analysis and press Enter. Depending on the type of analysis you selected, you receive a confirmation screen. Press Enter to return to the Application Analyze pop-up.

- 5 Select Submit application analysis batch job. The Analyze Submit Application pop-up, shown in [Figure 49](#), displays.

**Figure 49 • Analyze - Submit Application Screen for an Application Member**

```

                                Analyze - Submit Application                                USER2
Specify option and jobcard information.  Then press PF key for action.
AKR: 'VIAUSR.TEST.AKR'
Run reports _
Analyze features (Y/N)
  ASG-Recap      : Y
  ASG-Alliance  : y
  ASG-Estimate   : y
Do Semantic Linking (Y/N)?  N
Job statement information
//NAME  JOB (ACCOUNT), 'NAME',
//      MSGCLASS=X
//*    INSERT '/*ROUTE PRINT NODE.USER' HERE IF NEEDED.
//*
PF4=Options  PF5=Edit JCL  PF6=Submit JCL  PF10=Report Selection

```

## Analyzing a Program Member

### To analyze a program member

- 1 Select a program on the AKR Member List screen by typing / or S next to a member name and pressing Enter.
- 2 Select the Analyze action and press Enter. The ASG-ESW Prepare Program screen, shown in [Figure 50](#), displays. The existing analysis types are displayed as default features.

**Figure 50 • ASG-ESW Prepare Program Screen.**

```

                                ASG-ESW - Prepare Program
Command ==> _____
                                E - Edit JCL      S - Submit JCL      D - Doc Options
Compile and link JCL (PDS or sequential):
  Data set name 'VIAUSR.UTIL.CNTL(CMPLNKVS)'
Analyze features (Y/N):
  Understand: Y  Test: N  Extended Analysis: N  Document: N
  Re-engineer: N
AKR data set name 'VIACPO.CE60TEST.AKR'
AKR program name _____ (if overriding PROGRAM-ID)
Analyze options:
  _____
  _____
  _____
Compile? (Y/N) . . . . . N      (Y if needed by features)
Link load module reusable? (Y/N) N      (Test only)

```

## Changing an AutoChange or Bridge Member

If you selected an AutoChange or Bridge member on the AKR Member List screen, the only member action that is available is Change, as shown on [Figure 41 on page 272](#). Selecting the Change action invokes the AutoChange or Bridge primary screen.

## Creating Application Definitions

The Define Application option enables you to define an application at the ESW primary screen without invoking Alliance, Estimate, or Recap. If you do not have Alliance or Recap installed on your system, this option is disabled.

### To create an application

- 1 Select File ► Define Application. The File - New Application pop-up, shown in [Figure 51](#), displays.

Figure 51 • File - New Application Pop-up

```
File - New Application
Command ==> -----
Type AKR information, application name and analyze features.
Then press enter.

Application Knowledge Repository (AKR):
  Data set name . . . 'VI&USER.GENERAL.AKR'
  Application name . . . TESTAPPL

  Volume serial . . . ----- (if not cataloged)
  Password . . . . . (if password protected)

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

- 2 Specify the AKR dataset name for the AKR that is to contain the information for the application.
- 3 Type a 1- to 10-character name for the application you want to create. The default is the name of the last application opened or created.

- 4 Complete these fields, if necessary:

Field	Description
Volume serial	Specify the volume serial number of the device where the AKR resides. This field is required if the dataset is not cataloged.
Password	Specify the password for the dataset, if it is password protected.
Analyze features	Specify the type of analysis you want to perform on the application.

- 5 Press Enter. The Application Maintenance Facility screen, shown in [Figure 52](#), displays.

Figure 52 • Application Maintenance Facility Screen

```

File Edit View Select Zoom Options Help
-----
Application Maintenance Facili  NO COMPONENTS DEFINED
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
0 lines hidden 1 of 1
Definition Entity Src Type Status >
----->
- TESTAPPL APPLICATION EMPTY DEF
***** BOTTOM OF DATA *****

```

In ESW, the Application Maintenance Facility is available with ASG-Alliance, ASG-Estimate, and ASG-Recap. The Application Maintenance Facility screen currently has different pull-down options for each product. Because this version of the Application Maintenance Facility screen is accessed from the primary ESW screen, it contains a combined set of pull-down options applicable to all three products.

This table shows the source of these pull-down options:

Action Bar Item	Contains options from...
File	Estimate and Alliance
Edit	Estimate and Alliance
View	Estimate and Recap

Action Bar Item	Contains options from...
Select	Estimate, Alliance, and Recap
Zoom	Estimate, Alliance, and Recap
Options	Estimate, Alliance, and Recap
Help	Estimate, Alliance, and Recap

See the *ASG-Application Definition and Analysis User's Guide* for more information about the Application Maintenance Facility pull-down options.

- 6 Press PF3 to return to the ESW primary screen.

## Symbols

\$\$AUTOCH member installation  
option 187

## A

abend 130, 143  
codes 245  
interrupt 246  
run-time 146  
S05C 7  
S106 143  
S806 143  
S80A 143  
system 147  
ACENTITY member name 269  
AKR  
allocate 72, 229, 231–232  
Batch utility 74  
BDAM 71  
BDAM considerations 3  
BDAM option 173  
convert members 80  
defining additional 72  
display information 148  
expand 72, 229, 232  
initialization failures 144  
initialize 7, 72, 229  
input/output buffers 172  
listing 266  
maintaining 71  
OS Reserve macro 214  
print for debugging 148  
printing data 228  
size 73  
space requirements 14  
update 7  
update members to current level 80  
viewing 269  
VSAM 71  
VSAM option 173  
AKR Catalog Information pop-up 73

AKR utility log  
contents 89  
example 90  
AKR-Buffer-MaxK installation option 172  
AKRCOMP  
see AKR-Compression  
AKR-Compression installation option 173  
AKRDSN parameter 238  
AKR-DSORG-VSAM installation  
option 173  
AKRVSAM  
see AKR-DSORG-VSAM  
Alliance  
accessing from ESW screen xv  
Analyze-Work-Primary installation  
option 181  
description xii  
linking xv  
Alternate-Lang-Defs installation option 173  
Alternate-Lang-Entities installation  
option 175  
Alternate-Lang-Types installation  
option 175  
Alt-High-Level-Node installation  
option 175  
ALTLIB parameters 236  
AMASPZAP modifications 241  
AMF-Statistics-Required-Node 176  
analyze  
a COBOL program 94  
adding to compile mechanism 122  
automatic JCL modifications for  
Assembler 124  
automatic JCL modifications for  
COBOL 106  
CICS JCL before modifications 111  
compile/link JCL 95  
compiler names 177  
DB2 JCL before modifications 117  
dynamically called programs 177  
features 95  
JCL conversion 231

- job submit 98
  - methods 96
  - options 95
  - overview 93
  - process 96
  - requirements 94
  - source program 95
  - Summary report 119
  - user preprocessors 219
  - using File - Analyze Submit
    - pop-up 97
  - using ISPF 100
  - using ISPF/PDF edit 100
  - VIAIN DD statement 106
  - VIASUB edit macro 98, 100, 105
  - VIASUBDS CLIST 96, 100, 105
- ANALYZE command 97
- Analyze Submit Facility 93
- Analyze Submit Parameters screen
  - example 104
  - field descriptions 104
- Analyze Submit screen
  - example 98
  - field descriptions 98
- Analyze Summary report 119
- analyze types 270
- Analyze-COBOL-LangLvl installation
  - option 176
- Analyze-COBOL-Live-Exits installation
  - option 176
- Analyze-Compiler-Alias-Names installation
  - option 177
- Analyze-Compile-Required installation
  - option 177
- Analyze-Compiler-Names installation
  - option 177
- Analyze-Dynamic-Call installation
  - option 177
- Analyze-Extraction-Alias installation
  - option 178
- Analyze-Extraction-Pgm installation
  - option 178
- Analyze-Extra-Memory installation
  - option 178
- Analyze-Flag installation option 178
- Analyze-In-Out-Programs installation
  - option 178
- Analyze-Input-Programs installation
  - option 179
- Analyze-Lines-Per-Page installation
  - option 179
- Analyze-Link-Edit-Module=IEWL 179
- Analyze-Output-Programs installation
  - option 179
- Analyze-Pgm-No-Return installation
  - option 179
- Analyze-Region-Size installation option 180
- Analyze-Return-Code installation
  - option 180
- Analyze-Sequence-Check installation
  - option 180
- Analyze-Source installation option 180
- Analyze-Source-Spacing installation
  - option 180
- Analyze-STEPLIB-DCB-Info installation
  - option 181
- Analyze-STEPLIB-Libraries installation
  - option 181
- Analyze-Submit-JCL-Node installation
  - option 181
- Analyze-Work-Primary installation
  - option 106, 181
- Analyze-Work-Secondary installation
  - option 182
- Analyze-Work-Unit installation option 182
- analyzing
  - application member 276
  - enterprise member 274
  - program member 277–278
- ANLZSTAT command 78
- APPC-Multi-User installation option 182
- Appl-Anlz-AKR-Buffer-Maxk installation
  - option 182
- Appl-Anlz-Excl-CSECT installation
  - option 183
- Appl-Anlz-Excl-DD installation option 183
- Appl-Anlz-JCL-Add-TR-To-JOB-Card installation option 183
- Appl-Anlz-JCL-Fragment-DSN installation
  - option 183
- Appl-Anlz-JCL-Fragment-Password installation option 184
- Appl-Anlz-JCL-Fragment-Volser installation option 184
- Appl-Anlz-JCL-Time installation
  - option 184
- application definitions, creating 278
- application-level products, defined 2
- Appl-Work-Primary installation option 184
- Appl-Work-Secondary installation
  - option 184
- Appl-Work-SYSDA installation option 184
- Appl-Work-Unit installation option 185
- Appl-Work-Volume installation option 185
- Apply interim modifications 241
- ASG-Hi-Level-Nodes installation
  - option 185
- ASMA90 compiler name 177

- ASMBLR parameter 238
  - ASMH-Deck installation option 186
  - ASMH-Object installation option 186
  - ASM-Steplib-Groups installation option 186
  - ASM-Version installation option 187
  - Assembler
    - compiler names 177
    - condition codes 144
    - support 123
  - Assembler H
    - compiler support 3
    - DECK analyze option 186
    - OBJECT analyze option 186
    - support 238
  - AutoChange
    - accessing from ESW screen xv
    - description xii
  - Autoch-Base-AKR-DSN installation option 187
  - Auxiliary source manager support 226
- B**
- Batch AKR utility
    - command format 76
    - command syntax 77
    - control cards 76
    - description 74
    - JCL 74
    - log 89
    - log example 90
    - print directory report 90
    - punch directory 91
    - punch directory example 93
    - utility log 89
  - Batch analyze, submit job 232
  - Batch error log 146
  - Batch execution of script files 133
  - BDAM AKR
    - considerations 3
    - conversion 4
    - maintaining 71
  - BLKSREQ
    - see Analyze-STEPLIB-DCB-Info
  - BRENTITY member name 269
  - Bridge
    - accessing from ESW screen xv
    - description xii
  - Bridge Event Log 188
  - Bridge-Base-AKR-DSN installation option 187
  - Bridge-Generate-Max-Per-JOB installation option 188
  - Bridge-Log-DSN installation option 188
  - BUFMAXK installation option 172
  - BUILTIN commands 130
- C**
- CALL installation option 211
  - CALL Statement report 211
  - CAOMSMON compiler name 177
  - CAOTSMON compiler name 177
  - catalog information 73
  - CBLKSIZ value 17
  - CENTER CLIST member 230
  - CENTER parameter 15, 41–42, 133, 236
  - Center, description xii
  - Change-Man installation option 188
  - ChangeMan-SUBSYSID installation option 188
  - ChangeMan-Version installation option 188
  - Character-Back-Slash installation option 189
  - Character-Colon installation option 189
  - Character-Vertical-Bar installation option 189
  - CHECK feature
    - minimum error reporting level 189
    - source language format option 189
  - Check-Error-Level installation option 189
  - Check-Language installation option 189
  - CICS
    - compile/analyze JCL 111
    - preprocessor 190
  - CICS parameter 237
  - CICS-ASM-Preprocessor installation option 189
  - CICS-COBOL-Preprocessor installation option 190
  - CICS-CSD-Library installation option 190
  - CICS-DLI-library installation option 190
  - CICS-PLI-Preprocessor installation option 190
  - CICS-Preproc-Opt installation option 190
  - CIICLIB parameter 237
  - CIIMOD parameter 238
  - CIIRLIB parameter 237
  - CKELANG installation option 189
  - CKERLVL installation option 189
  - CLIST
    - customizing members 235
    - library block size 17
    - logon procedure cards 47
    - VIASVPTF 242
  - CLIST library, create or modify 15

- CLIST member
  - CENTER 230
  - customizing with VIASCUST
    - CLIST 235
  - VIACEN 230
  - VIAEDUSR 230
  - VIAEPANL 230
  - VIASAJC 231
  - VIASAJCE 230
  - VIASAJCN 230
  - VIASAJCS 231
  - VIASAJCX 231
  - VIASALCL 231
  - VIASBAT 231
  - VIASBRWS 231
  - VIASCCOM 231
  - VIASCM11 231
  - VIASCM12 231
  - VIASCMGB 231
  - VIASCMPR 231
  - VIASCPRM 231
  - VIASCUST 231, 235
  - VIASDEL 231
  - VIASEDBR 231
  - VIASEDID 231
  - VIASEDIM 232
  - VIASEDIT 232
  - VIASGBL 42, 197, 232
  - VIASGVFP 232
  - VIASISP 232
  - VIASJEDT 232
  - VIASLLSP 232
  - VIASNINS 232
  - VIASPI14 232
  - VIASPR 232
  - VIASPR14 232
  - VIASPRO1 232
  - VIASPRO2 232
  - VIASPROC 232
  - VIASPROF 129, 232
  - VIASPRTI 148, 232
  - VIASPTFR 232
  - VIASREP 232
  - VIASUB 232
  - VIASUBDS 96, 100, 105, 231, 233
  - VIASVPTF 233, 242
  - VIASVPTX 233
  - VIAYBRO 233
  - VIAYEDIT 233
  - VIAYJCLE 233
- CMDLIST parameter 42, 238
- CMPOUT installation option 211
- CNDSRC installation option 211
- CNTL library
  - create or modify 15
  - customization of members 235
  - executing VIASPRTI 148
  - modifying members 35
- CNTL member
  - customizing with VIASCUST
    - CLIST 235
  - VIA\$PRMB 223
  - VIA\$PRMD 223
  - VIA\$PRME 223
  - VIA\$PRMG 223
  - VIA\$PRMJ 223
  - VIA\$PRMP 223
  - VIA\$PRMS 223
  - VIA\$PRMX 224
  - VIABCSTM 230
  - VIADCSTM 230
  - VIAGCSTM 230
  - VIAICSTM 230
  - VIAITRN 229
  - VIAITRN1 228
  - VIAITRN2 228
  - VIAITRN3 228
  - VIAITRNC 228
  - VIAITRND 228
  - VIAITRNJ 228
  - VIAJCSTM 230
  - VIAMCSTM 230
  - VIAOCSTM 230
  - VIARCSTM 230
  - VIASAKRA 224
  - VIASAKRU 224, 238
  - VIASAKRX 224
  - VIASALDF 224
  - VIASALF 224
  - VIASALFU 224
  - VIASANJC 224
  - VIASASMJ 224
  - VIASASMU 224
  - VIASBASE 224
  - VIASBASJ 224
  - VIASBDAM 224
  - VIASBIND 224
  - VIASCCSJ 225
  - VIASCSJ 225
  - VIASCOBJ 225
  - VIASCOBU 225
  - VIASCOPY 7–8, 225
  - VIASCPNL 225
  - VIASCTM 231
  - VIASCUFC 225
  - VIASCUFJ 140, 225
  - VIASCUFR 140, 225

- VIASCUFU 140
- VIASDBRM 225
- VIASDDC 225
- VIASDDCJ 225
- VIASDIFF 225
- VIASDINS 225
- VIASDLAS 225
- VIASDLB 225–226
- VIASDLCB 226
- VIASDLCD 226
- VIASDLET 226
- VIASDLH1 226
- VIASDLH2 226
- VIASDLHP 226
- VIASDLM2 226
- VIASDLMF 226
- VIASDLN1 226
- VIASDLNR 226
- VIASDLNS 226
- VIASDLU1 226
- VIASDLVS 226
- VIASDLVX 226
- VIASEX3J 226
- VIASEXT 226
- VIASEXT1 226
- VIASEXT4 141
- VIASEXTJ 226
- VIASFAI2 227
- VIASFAIJ 36, 227
- VIASFAIR 227, 248
- VIASGENR 227
- VIASIMSJ 227
- VIASIMSU 227
- VIASINST 227
- VIASJCLJ 227
- VIASJCLU 227
- VIASLJB 227
- VIASLPAJ 44, 227
- VIASLPXJ 44, 227
- VIASMLSI 61, 228
- VIASMPRT 227
- VIASMSCL 227
- VIASNDVJ 227
- VIASNDVR 228
- VIASNDVX 55, 228
- VIASNDXJ 55, 228
- VIASPAM 228, 248
- VIASPAMJ 36, 228
- VIASPLIJ 228
- VIASPLIU 228
- VIASPRTI 228
- VIASSCLA 228
- VIASSCLL 228
- VIASSCLN 54–55
- VIASSCLR 228
- VIAUCSTM 230
- VIAXCSTM 230
- COB2LIB
  - see COBOL2-Library
- COB2MOD
  - see COBOL2-Compiler
- COB2TERM
  - see COBOL2-SYSTEMM-LRECL
- COBEDIT installation option 191
- COBOL
  - compiler library 237
  - compiler names 177
  - compiler support 3
- COBOL E compiler support 3
- COBOL editor environment
  - considerations 130
- COBOL F compiler support 3
- COBOL for MVS and VM compiler
  - support 3
- COBOL for OS/390 and VM
  - compiler support 3
- COBOL II compiler support 3
- COBOL/370 compiler support 3
- COBOL2-Compiler installation option 192
- COBOL2-Library installation option 192
- COBOL2-SYSTEMM-LRECL installation
  - option 192
- Cobol-370-library
  - installation option 190
- Cobol-370-library installation option 190
- Cobol-370-Module=IGYCRCTL installation
  - option 190
- COBOL-Edit-Profiles installation
  - option 191
- COBOLII
  - compiler 192, 238
  - compiler names 177
- COBOLII installation option 192
- Cobol-MVSVM-library installation
  - option 191
- Cobol-MVSVM-Module=IGYCRCTL
  - installation option 191
- Cobol-OS390-Library installation
  - option 191
- Cobol-OS390-Module=IGYCRCTL
  - installation option 191
- CobolSF installation option 192
- Cobol-Version installation option 191
- CobolVS-Library installation option 192
- CobolVS-Module installation option 192
- COBVER
  - see Cobol-Version

- COLON character
    - Recap 189
    - SmartDoc 189
  - Command Usage Facility
    - concatenating files 140
    - enabling 138
    - installation options 138
    - installing 48
    - introduction 135
    - limitations 141
    - record layout 136
    - SMF user exit VIASEXT4 141
  - commands
    - ANALYZE 97
    - ANLZSTAT 78
    - BUILTIN 130
    - Comment 79
    - CONVERT Batch AKR 80
    - COPY Batch AKR 81
    - DELETE Batch AKR 82
    - EXCLUDE 197
    - EXPORT Batch AKR 83
    - HELP Batch AKR 84
    - INIT Batch AKR 85
    - INITCMD 133
    - MOVE Batch AKR 86
    - PRINT Batch AKR 87
    - PRODLVL 144
    - PUNCH Batch AKR 88
    - RUNBATCH 133
    - VIASUB 96, 230
  - Comments command 79
  - compiler
    - alias names 177
    - Assembler 3, 177
    - COBOL 3
    - COBOL for MVS and VM 3
    - COBOLII 177, 192
    - High-level Assembler 3
    - OS/VS COBOL 177, 192
    - PL/I 3
  - compiler name for analyze 177
  - Compiler Output report 211, 216
  - Complex-Process-Threshold installation option 193
  - COMPVAL
    - see Analyze-Compiler-Alias-Names
  - COMPVAL installation option 177
  - COMPPGM
    - see Analyze-Compiler-Names
  - COMPPGM installation option 177
  - concatenating files, Command Usage Facility 140
  - Condensed Source Listing 211
  - continuation stack 248
  - conventions page xviii
  - CONVERT Batch AKR command 80
  - converting AKR from VSAM to BDAM 4
  - COPY Batch AKR command 81
  - COPY DD statement in analyze 220
  - COPY installation option 211
  - copy report, SmartDoc 211
  - CPU
    - seconds 205
    - time 248
  - CPXUPTSM compiler name 177
  - create a site-specific default profile 129
  - creating application definitions 278
  - CUA action bar 193
  - CUA-Profile-Types installation option 193
  - CUF-enabled installation option 138
  - CUF-Hi-Level-Nodes installation option 138
  - CUF-Log-Extended installation option 139
  - CUF-Log-Primary installation option 139
  - CUF-Log-Secondary installation option 139
  - CUF-Log-Type installation option 139
  - CUF-Log-Unit installation option 139
  - CUF-Log-Volume installation option 139
  - CUF-Sample-Rate installation option 139
  - customized Edit panel 129
  - CVSCLIB parameter 237
  - CVSMOD parameter 238
  - CVSRLIB parameter 237
- D**
- Data Division report 211
  - DATACLAS
    - see SMS-Data-Class
  - DATACOM/DD
    - SRCMGR installation option 217
    - support 217
  - dataset, temporary 221
  - DATAX installation option 211
  - date format 198
  - date separator 215
  - DB2
    - compile/analyze JCL 117
    - customizing options 35
    - subsystem 35
  - DB2-Library installation option 193
  - DB2PLAN
    - see DB2-Plan
  - DB2-Plan installation option 193
  - DB2-Preprocessor installation option 193
  - DB2-Procedure-Plan installation option 193
  - DB2-Sort-Lib installation option 194
  - DB2-STDSQL installation option 194

- DB2-Subsystem installation option 194
  - DBCS 45, 144, 217, 227
  - DCB information, suppress in
    - STEPLIBs 181
  - DCB=BLKSIZE 45
  - Dead-CICS-File-Name installation
    - option 194
  - Dead-JCL-DD installation option 194
  - Dead-JCL-DD installation options 194
  - Dead-PSB installation option 194–195
  - decimal point character 215
  - DECK analyze option
    - Assembler H 186
  - default option parameter files
    - changing the default value 171
    - syntax specification 158
  - default profile, creating a site-specific 129
  - default script libraries 215
  - default-language-criteria installation
    - option 60, 195
  - defaults
    - CUA profile types 193
    - work file block size 206
    - work space allocation type 182
    - work volume 206
  - DELETE Batch AKR command 82
  - DFP maintenance 6
  - diagnostic messages 178
  - dialog management, panel errors 144
  - direct access space 14
  - Disable-Foreground-Execution installation
    - option 196
  - Discovered-Called-Programs installation
    - option 196
  - Discovered-CICS-Transid installation
    - option 196
  - Discovered-IMS-Format installation
    - option 196
  - Discovered-Load-Module installation
    - option 196
  - Discovered-Program-Csect installation
    - option 196
  - Discovered-Stage1-Transactions installation
    - option 197
  - disk space allocation
    - default block size 206
    - default work space allocation
      - type 182
    - default work volume 206
    - translator units 219
    - units 205
    - VIAUT3 for Alliance and Recap 185
  - DL/I preprocessor 190
  - DOCOMP
    - see Analyze-Compile-Required
  - downloading products 15
  - DSNACAB load module 193
  - DSNACAF load module 193
  - DSN-Qualifier-Node installation option 197
  - DUPPERF installation option 207
  - dynamically called programs
    - analyze 177
    - in analyze job 177
    - Insight 177
  - DYNCALL
    - see Analyze-Dynamic-Call
- E**
- Easytrieve
    - CNTL member VIASMLSI 227
    - language support 60
  - edit
    - macro environment 129
    - panels customized 129
  - edit profiles installation option 191
  - Editor-Exclude-Command installation
    - option 197
  - Encore
    - accessing from ESW screen xv
    - analysis option 95
    - Analyze-Work-Primary installation
      - option 181
    - description xiii
  - Endevor
    - customization and operation 57
    - default ASG SVC 49
    - model user exit compile JCL 55
    - model user exit VIASNDVX 55
  - Endevor SCL model template
    - VIASSCLA 53
    - VIASSCLL 53
    - VIASSCLN 54–55
    - VIASSCLR 53
  - Endevor support, VIASSVCE module 49
  - Endevor SVC
    - installing 52
    - removing 52
  - Endevor-CONLIB installation option 197
  - Endevor-Steplib installation option 197
  - Endevor-TypeLength-NE installation
    - option 197
  - Endevor-User-SVC-Number installation
    - option 198
  - Enhanced Data Cross-reference report 211
    - SmartDoc 211
  - ENQ macro 4
  - error messages, obtaining online help 144

- Estimate
  - accessing from ESW screen [xv](#)
  - description [xiii](#)
  - Easytrieve CNTL member [227](#)
- ESW
  - description [xi](#)
  - invoking products [xiv](#)
  - print message file [227](#)
  - product integration [xv](#)
- ESW load library
  - moving modules [44](#)
  - reblocking and/or copying [133](#)
- ESW parameter [42](#)
- EXCLUDE command [197](#)
- Execution Monitor Description file
  - creating [150](#)
  - example [154](#)
  - option-clause [151](#)
  - syntax-rule [151](#)
- Execution Monitor support, Indirect Program Invocation support [149](#)
- execution of ISPF Batch [133](#)
- Execution-Monitor-Defs installation option [198](#)
- exits, live [176](#)
- expanding AKRs [72](#)
- EXPORT Batch AKR command [83](#)
- EXTRMAL
  - see Analyze-Extraction-Alias
- EXTRPGM
  - see Analyze-Extraction-Pgm
- F**
- FAST installation option [216](#)
- File - AKR Allocate/Expand screen [72](#)
- File - AKR Utility screen [72](#)
- FLAG
  - see Analyze-Flag
- Format-Date installation option [198](#)
- Format-Time installation option [198](#)
- Fortran
  - analyzing Fortran applications [70](#)
  - language support [60](#)
- FREEMAIN [214](#)
- G**
- Gen-COBOL-Last-Node installation option [198](#)
- global resource manager
  - GRS [4](#)
  - Multi-Image Manager [5](#)
- Global Resource Serialization
  - Configuration [4](#)
- GOTOS
  - include in SmartDoc [207, 217](#)
  - include in SmartTest [207](#)
- GRS [4](#)
- H**
- HCOND installation option [207](#)
- help
  - error messages [144](#)
  - generated on SmartDoc reports [216](#)
  - see SmartDoc Help [216](#)
- HELP Batch AKR command [84](#)
- HGOTO installation option [207](#)
- HSIZE installation option [217](#)
- I**
- IBM GRS global resource manager [4](#)
- IDEAL
  - analyzing IDEAL applications [61](#)
  - language support [60](#)
- IDMS
  - default Data Dictionary [199](#)
  - default node name [199](#)
  - load library [199](#)
- IDMS-ASM-Preprocessor installation option [198](#)
- IDMS-COBOL-Preprocessor installation option [199](#)
- IDMSCTL
  - see IDMS-SYS-Control-Library
- IDMS-Data-Dictionary-Name installation option [199](#)
- IDMSDDB
  - see IDMS-Local-Mode-Libraries
- IDMSDNM
  - see IDMS-Data-Dictionary-Name
- IDMS-IDMSDMLC-library installation option [199](#)
- IDMSLD
  - see IDMS-Load-Libraries
- IDMS-Load-Libraries installation option [199](#)
- IDMS-Local-Mode-Libraries installation option [199](#)
- IDMSNNM installation option [199](#)
- IDMS-Node-Name installation option [199](#)
- IDMS-PLI-Preprocessor installation option [199](#)
- IDMS-SYS-Control-Library installation option [200](#)
- IEL0AA compiler name [177](#)
- IEV90 compiler name [177](#)
- IFOX00 compiler name [177](#)

- IGYCRCTL compiler name 177
- IKFCBL00 compiler name 177
- IMSVS-RESLIB-library installation
  - option 200
- INCL installation option 200
- Include-Libraries installation option 200
- Indirect program invocation
  - Execution Monitor support 149
- indirect program invocation
  - Execution-Monitor-Defs installation
    - option 198
- INIT Batch AKR command 85
- INITCMD command 133
- INPUTS
  - see Analyze-Input-Programs
- Insight
  - accessing from ESW screen xv
  - description xiii
  - dynamically called programs 177
  - using analysis functions xv
- installation
  - JCL 7-8, 13, 15
  - option definitions 157
  - overview 9
  - steps 11
- installation option
  - \$\$AUTOCH 187
  - AKR-Buffer-MaxK 172
  - AKR-Compression 173
  - AKR-DSORG-VSAM 173
  - Alternate-Lang-Defs 173
  - Alternate-Lang-Entities 175
  - Alternate-Lang-Types 175
  - Alt-High-Level-Node 175
  - AMF-Statistics-Required-Node 176
  - Analyze-COBOL-LangLvl 176
  - Analyze-COBOL-Live-Exits 176
  - Analyze-Compiler-Alias-Names 177
  - Analyze-Compile-Required 177
  - Analyze-Compiler-Names 177
  - Analyze-Dynamic-Call 177
  - Analyze-Extraction-Alias 178
  - Analyze-Extraction-Pgm 178
  - Analyze-Extra-Memory 178
  - Analyze-Flag 178
  - Analyze-In-Out-Programs 178
  - Analyze-Input-Programs 179
  - Analyze-Lines-Per-Page 179
  - Analyze-Output-Programs 179
  - Analyze-Pgm-No-Return 179
  - Analyze-Region-Size 180
  - Analyze-Return-Code 180
  - Analyze-Sequence-Check 180
  - Analyze-Source 180
  - Analyze-Source-Spacing 180
  - Analyze-STEPLIB-DCB-Info 181
  - Analyze-STEPLIB-Libraries 181
  - Analyze-Submit-JCL-Node 181
  - Analyze-Work-Primary 181
  - Analyze-Work-Secondary 182
  - Analyze-Work-Unit 182
  - APPC-Multi-User 182
  - Appl-Anlz-AKR-Buffer-Maxk 182
  - Appl-Anlz-Excl-CSECT 183
  - Appl-Anlz-Excl-DD 183
  - Appl-Anlz-JCL-Add-TR-To-JOB-Car
    - d 183
  - Appl-Anlz-JCL-Fragment-DSN 183
  - Appl-Anlz-JCL-Fragment-Password
    - 184
  - Appl-Anlz-JCL-Fragment-Volser
    - 184
  - Appl-Anlz-JCL-Time 184
  - Appl-Work-Primary 184
  - Appl-Work-Secondary 184
  - Appl-Work-SYSDA 184
  - Appl-Work-Unit 185
  - Appl-Work-Volume 185
  - ASG-Hi-Level\_Nodes 185
  - ASMHDCK 186
  - ASMH-Deck 186
  - ASMHOBJ 186
  - ASMH-Object 186
  - ASM-Steplib-Groups 186
  - ASM-Version 187
  - Autoch-Base-AKR-DSN 187
  - Bridge-Base-AKR-DSN 187
  - Bridge-Generate-Max-Per-JOB 188
  - Bridge-Log-DSN 188
  - CALL 211
  - Change-Man 188
  - ChangeMan-SUBSYSID 188
  - ChangeMan-Version 188
  - Character-Back-Slash 189
  - Character-Colon 189
  - Character-Vertical-Bar 189
  - Check-Error-Level 189
  - Check-Language 189
  - CICS-ASM-Preprocessor 189
  - CICS-COBOL-Preprocessor 190
  - CICS-CSD-Library 190
  - CICS-DL1-library 190
  - CICS-PLI-Preprocessor 190
  - CICS-Preproc-Opt 190
  - CKELANG 189
  - CKERLVL 189
  - CMPOUT 211
  - CNDSRC 211

COBEDIT 191  
COBOL2-Compiler 192  
COBOL2-Library 192  
COBOL2-SYSTEM-LRECL 192  
Cobol-370-library 190  
Cobol-370-Module=IGYCRCTL 190  
COBOL-Edit-Profiles 191  
COBOLII 192  
Cobol-MVSVM-library 191  
Cobol-MVSVM-Module=IGYCRCTL 191  
Cobol-OS390-Library 191  
Cobol-OS390-Module=IGYCRCTL 191  
Cobol-Version 191  
CobolVS-Library 192  
CobolVS-Module 192  
Command Usage Facility 138  
common parameters 33  
Complex-Process-Threshold 193  
COMPMAL 177  
COMPPGM 177  
COPY 211  
CUA-Profile-Types 193  
CUF-enabled 138  
CUF-Hi-Level-Nodes 138  
CUF-Log-Extended 139  
CUF-Log-Primary 139  
CUF-Log-Secondary 139  
CUF-Log-Type 139  
CUF-Log-Unit 139  
CUF-Log-Volume 139  
CUF-Sample-Rate 139  
DATAX 211  
DB2-Library 193  
DB2-Plan 193  
DB2-Preprocessor 193  
DB2-Procedure-Plan 193  
DB2-Sort-Lib 194  
DB2-STDSQL 194  
DB2-Subsystem 194  
Dead-CICS-File-Name 194  
Dead-JCL-DD 194  
Dead-PSB 194-195  
default-language-criteria 195  
Disable-Foreground-Execution 196  
Discovered-Called-Programs 196  
Discovered-CICS-Transid 196  
Discovered-IMS-Format 196  
Discovered-Load-Module 196  
Discovered-Program-Csect 196  
Discovered-Stage1-Transactions 197  
DSN-Qualifier-Node 197  
DUPPERF 207  
Editor-Exclude-Command 197  
Endevor-CONLIB 197  
Endevor-Steplib 197  
Endevor-TypeLength-NE 197  
Endevor-User-SVC-Number 198  
ENQ 4  
Execution-Monitor-Defs 198  
FAST 216  
Format-Date 198  
Format-Time 198  
Gen-COBOL-Last-Node 198  
HCOND 207  
HGOTO 207  
HSIZE 217  
IDMS-ASM-Preprocessor 198  
IDMS-COBOL-Preprocessor 199  
IDMS-Data-Dictionary-Name 199  
IDMS-IDMSDMLC-library 199  
IDMS-Load-Libraries 199  
IDMS-Local-Mode-Libraries 199  
IDMSNNM 199  
IDMS-Node-Name 199  
IDMS-PLI-Preprocessor 199  
IDMS-SYS-Control-Library 200  
IDMS-SYSCTL-Library 200  
IMSVS-RESLIB-library 200  
INCL 200  
Include-Libraries 200  
JES-Proc 201  
JOB-Proc 202  
LANGLVL 192  
language 202  
LIBCOPY 202  
Librarian-Expand-COPY 202  
Librarian-Extract 202  
Librarian-ISPF-Module 203  
Librarian-Release 203  
Librarian-STEPLIB 203  
Librarian-Subsystem-name 203  
Library-Percent 204  
Member-Count 204  
Memory-Threshold-RMODE 204  
METRICS 212  
MLPA 43  
NETRON 204  
NLDTFMT 198  
NLTMFMT 198  
Online-List-Unit 205  
Online-List-Volume 205  
Online-Log-Unit 205  
Online-Log-Volume 205  
Online-Max-Function-Time 205  
Online-Perm-Unit 205  
Online-Punch-Unit 164, 206

- Online-Punch-Volume 206
- Online-Work-Block 206
- Online-Work-Volume 206
- PANCOPY 206
- Panvalet-Expand-COPY 206
- Panvalet-IPNEXIT 206
- Panvalet-ISPF-Module 207
- Panvalet-Release 207
- Panvalet-STEPLIB 207
- Panvalet-Subsystem-name 207
- PARAX 212
- PDFXCMD 197
- PERF 213
- Perform-Hier-Chart-Conditionals 207
- Perform-Hier-Struct-Duplicates 207
- PHEIR 212
- PLI-Steplib-Groups 187, 208
- PLI-Version 209
- PLPA 43
- PRGEXC 213
- PROCLIB-JES-Alternates 209
- PROCLIBs 209
- QNAME 4
- Report-Advanced-Source 209
- Report-Appl-Comparison 210
- Report-Appl-Definition 210
- Report-Appl-Exception 210
- Report-Appl-FunctionPoint 210
- Report-Appl-Metrics 210
- Report-Appl-Progress 210
- Report-Banner-Page 210
- Report-Call 211
- Report-Compiler-Output 211
- Report-Condensed-Source 211
- Report-Copy 211
- Report-Data-Division 211
- Report-Enhanced-Data-Xref 211
- Report-Enterprise-Exception 211
- Report-Enterprise-Metrics 212
- Report-Executive-Summary 212
- Report-Lines-Per-Page 212
- Report-Master-Index 212
- Report-Metrics 212
- Report-Paragraph-Xref 212
- Report-Perform-Hierarchy 212
- Report-Perform-Interface 213
- Report-Pgm-Comparison 213
- Report-Pgm-Metric-History 213
- Report-Program-Exception 213
- Report-Program-Progress 213
- Report-Recursion 214
- Report-Structure-Chart 213
- Report-Subset 213
- Report-Verb-Context 214
- Report-Verb-Frequency 214
- RESERVE 4
- Reserved-Subpools 214
- Reserve-MACRO 214
- RNAME 4
- SCLM 215
- SCOND 217
- Script-Libraries 215
- SEDCUA 193
- Separator-Date 215
- Separator-Decimal-Point 215
- Separator-Thousands 215
- Separator-Time 215
- SGOTO 217
- site-specific profile 129
- SmartDoc-COBOL-List-To-SYSPRI  
NT 216
- SmartDoc-Help 216
- SmartDoc-Minimum-Reports 216
- SMS 216
- SMS-Data-Class 216
- SMS-Mgmt-Class 216
- SMS-Storage-Class 217
- SOURCE 180
- Source-Manager 217
- STMDE 218
- STRUC 213
- Structure-Chart-Birds-Eye 217
- Structure-Chart-Conditionals 217
- Structure-Chart-GOTOs 217
- Structure-Chart-Horizontal-Size 217
- Structure-Chart-Max-Pages 218
- Structure-Chart-Mode 218
- Structure-Chart-Vertical-Size 218
- SUBSET 213
- SYSOOT-Class 218
- SYSPRNT 216
- Translator-Monitor-Alias-Names 218
- Translator-Program-Names 218
- Translator-User-Preprocessor 219
- Translator-Work-Primary 219
- Translator-Work-Secondary 219
- Translator-Work-Unit 219
- TRANUSR 219
- TSPUNIT 219
- user override file 171
- User-Exit-Copy-DD 220
- USERPP 220
- User-Preprocessor 220
- VERBC 214
- VF 214
- VIAJCL 198
- VSIZE 218
- VSPPRIM 106

- VSPSECD 106
- VSPUNIT 106, 181
- Work-SYSDA 221
- insufficient region space 143
- interim modifications, applying 241
- internal error reporting 145
- interrupt abend 246
- Invoke ESW products
  - from ISPF 47
  - with CLIST 48
- IOS
  - see Analyze-In-Out-Programs
- IPNEXIT
  - see Panvalet-IPNEXIT
- ISPF
  - BATCH execution 133
  - dialogs 122
  - environment 248
  - IBM APARs 6
  - invoke ESW products 47
  - LIBDEF Facility option 232
  - maintenance 6
  - modify 44
  - profile dataset 130
  - V3R5 6
- ISPF/PDF environment considerations 129
- ISPLLIB load library
  - example 45
  - general problems 143
  - using 45
- ISPPALT 225
- ISPLLIB panel library
  - create or modify 15
  - example 45
  - JCL to copy 225
  - using 45
- ISPSLIB skeleton library
  - create or modify 15
  - customizing members with
    - VIASCUST CLIST 235
  - examples 46
  - using 45
- ISPTLIB table library
  - create or modify 15
  - examples 47
  - using 45
- J**
- JCL
  - installation 15
  - load installation 13
  - VIASBIND 35
- JES startup procedure 201
- JES-Proc installation option 201
- JOB1 parameter 237
- JOB2 parameter 237
- JOB3 parameter 237
- JOB4 parameter 237
- JOBPROC
  - see JOB-Proc
- JOB-Proc installation option 202
- K**
- keywords, non-standard 200
- L**
- LAM
  - see Librarian-Subsystem-name
- LANG
  - see language
- LANGLVL
  - see Analyze-COBOL-LangLvl
- LANGLVL installation option 192
- language format option 189
- language installation option 202
- LBLKCNT parameter 133
- LBLKSIZ parameter 133
- LBLKSIZE parameter 7–8
- LIBCOPY installation option 202
- LIBDEF facility 45
- LIBDEF parameter 42, 236
- LIBERPCT
  - see Library-Percent
- LIBEXTR
  - see Librarian-Extract
- LIBMOD
  - see Librarian-ISPF-Module
- Librarian
  - Assembler source program 227
  - datasets 200, 202
  - extraction program 178
  - FAIR modules 227
  - ISPF EDIT module 203
  - release level 203
  - requirement 227
  - VIASFAIJ 36
- Librarian-Expand-COPY installation option 202
- Librarian-Extract installation option 202
- Librarian-ISPF-Module installation option 203
- Librarian-Release installation option 203
- Librarian-STEPLIB installation option 203
- Librarian-Subsystem-name installation option 203
- libraries, staging 44
- library, CLIST 42

- Library-Percent installation option 204
  - LIBSTLB
    - see Librarian-STEPLIB
  - LIBVER
    - see Librarian-Release
  - LINECNT
    - see Analyze-Lines-Per-Page
  - link editor 179
  - list file 205–206
  - LISTUNT
    - see Online-List-Unit
  - LISTUNT parameter 238
  - LISTVOL
    - see Online-List-Volume
  - live exits 176
  - load library
    - block size 133
    - for demo programs 237
    - general problems 143
    - using 45
  - LOADJCL EXEC statement 13
  - LOADLIB
    - see Analyze-STEPLIB-Libraries
  - LOADLIB load library, create or modify 15
  - Log file
    - delete 231
    - printing 232
  - log file 145, 205–206, 246
  - log listing 145
  - Log/List/Punch files
    - printing 229
  - Log/List/Punch files, printing 229
  - LOGUNT
    - see Online-Log-Unit
  - LOGUNT parameter 238
  - LOGVOL
    - see Online-Log-Volume
- M**
- maintenance 241
  - MAXBLK parameter 133
  - MBRERCNT
    - see Member-Count
  - Member-Error-Count installation option 204
  - memory management table 248
  - memory requirement 44
  - Memory-Threshold-RMODE installation
    - option 204
  - Message file 185
  - message file 245
  - messages
    - diagnostic 178
    - warning 178
  - METRICS installation option 212
  - metrics report 212
  - MGMTCLAS
    - see SMS-Mgmt-Class
  - MIMS
    - see Multi-image Manager
  - minimum reports 216
  - MLPA load modules
    - copying 227
    - installing 43
  - Model 204
    - analyzing applications 67
    - splitter utility 68
    - User Language 60
  - monitor, analyze 246
  - MOVE Batch AKR command 86
  - Multi-image Manager 5
  - MVS maintenance 6
  - MVS/SP sites 172
- N**
- NATURAL
    - analyzing applications 62
    - language support 60
    - splitter utility 66
  - NDVRC Endeavor utility module 49
  - NETRON installation option 204
  - NLDTFMT installation option 198
  - NLTMFMT installation option 198
  - NORET
    - see Analyze-Pgm-No-Return
- O**
- OBJECT option, Assembler H 186
  - ONERROR routine 247
  - online error log 146
  - Online-List-Unit installation option 205
  - Online-List-Volume installation option 205
  - Online-Log-Unit installation option 205
  - Online-Log-Volume installation option 205
  - Online-Max-Function-Time installation
    - option 205
  - Online-Perm-Unit installation option 205
  - Online-Punch-Unit 164
  - Online-Punch-Unit installation option 164, 206
  - Online-Punch-Volume 206
  - Online-Punch-Volume installation
    - option 206
  - Online-Work-Block installation option 206
  - Online-Work-Volume installation
    - option 206
  - Options - Log/List/Punch Definition
    - pop-up 145

- options, definitions 157
- OS GETMAIN 214
- OS Reserve macro 214
- OS/VS COBOL compiler 192, 237
- OUTPUTS
  - see Analyze-Output-Programs
- override installation options, VIAUPARM DD 171
- overview, installation 9
- P**
- page mode, structure chart 218
- PANCOPY installation option 206
- panel library 45
- PANMOD
  - see Panvalet-ISPF-Module
- PANSTLB
  - see Panvalet-STEPLIB
- PANV
  - see Panvalet-Subsystem-name
- Panvalet
  - Assembler source program 228
  - datasets 200, 206
  - extraction program 178
  - IPNEXIT user exit 206
  - ISPF load module 207
  - VIASPAM module 228
  - VIASPAMJ 36
- Panvalet-Expand-COPY installation option 206
- Panvalet-IPNEXIT installation option 206
- Panvalet-ISPF-Module installation option 207
- Panvalet-Release installation option 207
- Panvalet-STEPLIB installation option 207
- Panvalet-Subsystem-name installation option 207
- PANVER
  - see Panvalet-Release
- Paragraph Cross-Reference report 212
- parameters, VIASCPRM 236
- PARAX installation option 212
- partitioned datasets 200
- PCWA address 246
- PCWAUSER control block 248
- PDFXCMD installation option 197
- PERF installation option 213
- Perform Range Hierarchy Chart
  - include conditionals 207
  - installation option 207
  - SmartDoc 207
- Perform Range Hierarchy chart 212
- Perform Range Usage and Interface Report 213
- performance
  - ASG-SmartEdit 130
  - improvements 44
- Perform-Hier-Chart-Conditionals
  - installation option 207
- Perform-Hier-Struct-Duplicates installation option 207
- PERMUNT parameter 17
- PERMVOL parameter 17
- PHEIR installation option 212
- PL/I compiler support 3
- PL/I support
  - MVS & VM 3
  - OS PL/I Version 2 compiler 3
- PLI-Steplib-Groups installation option 187, 208
- PLI-Version installation option 209
- PLPA load modules
  - copying 227
  - installing 43
- PMTHR24
  - see Memory-Threshold-RMODE
- PMTHR31
  - see Memory-Threshold-RMODE
- pop-up
  - Options - Log/List/Punch Definition
- PPLIB1
  - see CICS-DL1-library
- PPLIB2
  - see DB2-Library
- PPLIB4
  - see IDMS-IDMSDMLC-library
- PPLIB5
  - see IDMS-SYSCTL-Library
- PPMOD2
  - see DB2-Preprocessor
- PRGEXC installation option 213
- primary space allocation block count
  - adjust 133
- PRINT Batch AKR command 87
- process support
  - exceptions 265
  - listing AKRs 266
  - overview 265
- processing considerations 127
- PROCLIB-JES-Alternates installation option 209
- PROCLIBs installation option 209
- PRODLVL command 144
- product integration xv
- product level 144
- PROF variable 42
- profile 130

- program analyzer
  - installation option 180
  - problems 144
  - resource estimates 131
- program exception report 213
- program query problems 144
- PTF
  - applying 241
  - backing out 232
  - creating or modifying library 15
  - customizing installation with VIASCUST CLIST 235
- PTF Display facility 242
- PUNCH Batch AKR command 88
- punch file 205–206
- PUNUNT parameter 238
- PUNVOL
  - see Online-Punch-Volume
- Q**
- Q NAMES 4
- R**
- reblocking and/or copying load library 133
- Recap
  - accessing from ESW screen xv
  - COLON character 189
  - colon replacement option 215
  - date format 198
  - date separator 215
  - decimal point separator 215
  - description xiii
  - thousands separator 215
  - time format 198
  - time separator 215
  - vertical bar character 189
  - VIAUT3 space allocation 185
- RECUR
  - see Report-Recursion
- Recursion report 214
- recursive edit 130
- region size 143
- release level 145
- report
  - Analyze Summary 119
  - Recursion 214
  - Subsets 213
  - Verb Frequency 214
  - Verb Summary 214
- Report-Advanced-Source installation option 209
- Report-Appl-Comparison installation option 210
- Report-Appl-Definition installation option 210
- Report-Appl-Exception installation option 210
- Report-Appl-FunctionPoint installation option 210
- Report-Appl-Metrics installation option 210
- Report-Appl-Progress installation option 210
- Report-Banner-Page installation option 210
- Report-Call installation option 211
- Report-Compiler-Output installation option 211
- Report-Condensed-Source installation option 211
- Report-Copy installation option 211
- Report-Data-Division installation option 211
- Report-Enhanced-Data-Xref installation option 211
- Report-Enterprise-Exception installation option 211
- Report-Enterprise-Metrics installation option 212
- Report-Executive-Summary installation option 212
- Report-Lines-Per-Page installation option 212
- Report-Master-Index installation option 212
- Report-Metrics installation option 212
- Report-Paragraph-Xref installation option 212
- Report-Perform-Hierarchy-Chart installation option 212
- Report-Perform-Interface installation option 213
- Report-Pgm-Comparison installation option 213
- Report-Pgm-Metric-History installation option 213
- Report-Program-Exception installation option 213
- Report-Program-Progress installation option 213
- Report-Recursion installation option 214
- Report-Structure-Chart installation option 213
- Report-Subset installation option 213
- Report-Verb-Context installation option 214
- Report-Verb-Frequency installation option 214
- requirements, space 14

- RESERVE
  - see Reserved-Subpools
- RESERVE macro [4](#)
- Reserved-Subpools installation option [214](#)
- Reserve-MACRO installation option [214](#)
- RESVDA
  - see Reserve-MACRO
- RETCODE
  - see Analyze-Return-Code
- RNAMEs [4](#)
- RUNBATCH command [133](#)
  
- S**
- S05C abend [7](#)
- S106 abend [143](#)
- S806 abend [143](#)
- S80A abend [143](#)
- SAMPLPTF PTF member [241](#)
- SCLASS
  - see SYSOUT-Class
- SCLM [215](#)
  - see SCLM
- SCLM installation option [215](#)
- SCOND installation option [217](#)
- screen
  - AKR Catalog Information [73](#)
  - Analyze Submit [98](#)
  - Analyze Submit Parameters [104](#)
  - File - AKR Allocate/Expand [72](#)
  - File - AKR Utility [72](#)
- Script-Libraries installation option [215](#)
- SCRPTLIB
  - see Script-Libraries
- security software
  - ACF2 [127](#)
  - IPNEXIT [128](#)
  - Panvalet [128](#)
  - RACF [127](#)
  - Top Secret [127](#)
- SEDCUA installation option [193](#)
- see PROCLIB-JES-ALTERNATIVE [209](#)
- Separator-Date installation option [215](#)
- Separator-Decimal-Point installation option [215](#)
- Separator-Thousands installation option [215](#)
- Separator-Time installation option [215](#)
- SEQ
  - see Analyze-Sequence-Check
- serialization of AKRs [4](#)
- Service Packs, using [1](#)
- SETRNLIB [15](#)
- SGOTO installation option [217](#)
- shared DASD [214](#)
- site-specific default profile, creating [129](#)
  
- skeleton library [46](#)
- skeleton member
  - VIASAKAP [41, 229](#)
  - VIASAKRA [229](#)
  - VIASAKRX [229](#)
  - VIASAKXP [41, 229](#)
  - VIASJOBBC [229](#)
  - VIASJRPT [229](#)
  - VIASJSUM [229](#)
  - VIASLLSP [229](#)
- SmartDoc
  - accessing from ESW screen [xv](#)
  - CALL Statement Report [211](#)
  - COLON character [189](#)
  - Compiler Output report [211, 216](#)
  - Condensed Source Listing [211](#)
  - Copy report [211](#)
  - Data Division report [211](#)
  - description [xiii](#)
  - Enhanced Data Cross-reference report [211](#)
  - horizontal box size [217](#)
  - include conditionals [207, 217](#)
  - include GOTOs [207, 217](#)
  - Metrics report [212](#)
  - minimum reports [216](#)
  - Paragraph Cross-Reference report [212](#)
  - Perform Range Hierarchy Chart [207](#)
  - Perform Range Hierarchy chart [207, 212](#)
  - Perform Range Usage and Interface report [213](#)
  - Program Exception report [213](#)
  - reports [216](#)
  - Structure Chart [213](#)
  - structure chart [217–218](#)
  - Subsets report [213](#)
  - SYSPRINT option [216](#)
  - Verb Summary report [214](#)
  - vertical bar character [189](#)
- SmartDoc-COBOL-List-To-SYSPRINT installation option [216](#)
- SmartDoc-Help installation option [216](#)
- SmartDoc-Minimum-Reports installation option [216](#)
- SmartEdit
  - accessing from ESW screen [xv](#)
  - BUFMAXK option [172](#)
  - CUA default profile types [193](#)
  - description [xiv](#)
  - edit profiles [191](#)
  - IDMS libraries options [200](#)
  - Librarian-Expand.COPY [202](#)

- minimum error reporting level 189
    - performance 130
    - source language format option 189
    - SYSTEM LRECL 192
    - user preprocessor option 220
  - SmartTest
    - accessing from ESW screen xv
    - Batch connect job 246
    - description xiv
    - execution 246
    - internal error 246
  - SMS
    - data class option 216
    - installation option 216
    - management class option 216
    - see SMS 216
  - SMS-Data-Class installation option 216
  - SMS-Mgmt-Class installation option 216
  - SMS-Storage-Class installation option 217
  - SORTLIB
    - see DB2-Sort-Lib
  - SOURCE
    - see Analyze-Source
  - SOURCE installation option 180
  - source manager
    - compile JCL 107
    - Endevor 107
    - Librarian 217
    - Panvalet 108, 217
    - PROTEUS 107
    - SCLM 107
  - Source-Manager installation option 217
  - space requirements
    - AKR 14, 73
    - analysis 131
    - Analyze-Source-Spacing
      - parameter 180
  - SPUNITS
    - see Online-Perm-Unit
  - SPZAPJCL PTF member 241
  - SRCMGR
    - see Source-Manager 217
  - statistics, AMF-Statistics-Required-Node
    - parameter 176
  - STEPLIB load library 143
  - STIMER
    - see Online-Max-Function-Time
  - STMDE installation option 218
  - storage requirements, ESW products 14
  - STORCLAS
    - see SMS-Storage-Class
  - STRUC installation option 213
  - structure chart
    - birds-eye view 217
    - horizontal box size 217
    - include conditionals 217
    - include GOTOs 217
    - Max-Pages 218
    - specify tile/page mode 218
    - vertical box size 218
  - Structure-Chart-Birds-Eye installation
    - option 217
  - Structure-Chart-Conditionals installation
    - option 217
  - Structure-Chart-GOTOs installation
    - option 217
  - Structure-Chart-Horizontal-Size installation
    - option 217
  - Structure-Chart-Max-Pages installation
    - option 218
  - Structure-Chart-Mode installation
    - option 218
  - Structure-Chart-Vertical-Size installation
    - option 218
  - submit an analyze job 97
  - subpool areas 214
  - SUBSET installation option 213
  - Subsets report 213
  - SUBSYS
    - see DB2-Subsystem
  - suppress DCB information in STEPLIB
    - statements 181
  - SVC for Endevor
    - installing 52
    - removing 52
    - VIASNDVJ CNTL member 52
    - VIASNDVR CNTL member 52
  - swap space 44
  - SYSDA
    - see Work-SYSDA
  - SYSDA parameter 42, 133, 238
  - SYSOUT parameter 16, 133, 238
  - SYSOUT-Class installation option 218
  - SYSPRINT option 216
  - SYSPRINT SYSOUT class 16
  - SYSPRINT installation option 216
  - SYSPROC allocations 232
  - SYSTEM dataset LRECL for
    - SmartEdit 192
- T**
- table library 47
  - TAPEUNT parameter 16
  - temporary data set 16
  - temporary dataset 221
  - temporary datasets 133

- thousands separator [215](#)
- Tile mode structure chart [218](#)
- time format, Recap [198](#)
- training libraries [15](#)
- TRANMAL
  - see Translator-Monitor-Alias-Names
- TRANPGM
  - see Translator-Program-Names
- Translator-Monitor-Alias-Names installation option [218](#)
- Translator-Program-Names installation option [218](#)
- Translator-User-Preprocessor installation option [219](#)
- Translator-Work-Primary installation option [219](#)
- Translator-Work-Secondary installation option [219](#)
- Translator-Work-Unit installation option [219](#)
- TRANUSR installation option [219](#)
- TSO
  - CALL [129](#)
  - UADS specifications [205–206](#)
- TSPPRIM
  - see Translator-Work-Primary
- TSPSECD
  - see Translator-Work-Secondary
- TSPUNIT
  - see Translator-Work-Unit
- TSPUNIT installation option [219](#)
- U**
- UADS specification [205](#)
- Undoc-AKR-Pgm-Wait-Timeout installation option [219](#)
- units
  - disk space allocation [205](#)
  - translator work space [219](#)
- user abends [130](#)
- user exit
  - copybook file format [39](#)
  - VIASALFU [37](#)
  - VIASASMU [37](#)
  - VIASCCSU [37](#)
  - VIASCOBU [37](#)
  - VIASEXT3 [141](#)
  - VIASEXT4 [141](#)
  - VIASIMSU [37](#)
  - VIASJCLU [37](#)
  - VIASPLIU [37](#)
  - VIAYINCL file format [39](#)
- user preprocessors
  - analyze [219](#)
- USERDSQ
  - see DSN-Qualifier-Node
- USERDSQ parameter [42, 239](#)
- User-Exit-Copy-DD installation option [220](#)
- USERLIB parameter [237](#)
- USERPP installation option [220](#)
- User-Preprocessor installation option [220](#)
- V**
- VCHAR
  - see Character-Vertical-Bar
- VCOBLIB
  - see CobolVS-Library [192](#)
- VCOBMOD
  - see CobolVS-Module
- Verb Frequency report [214](#)
- Verb Summary report [214](#)
- VERBC installation option [214](#)
- vertical bar character
  - Recap [189](#)
  - SmartDoc [189](#)
- VF installation option [214](#)
- VIA\$PRMB CNTL member [223](#)
- VIA\$PRMD CNTL member [223](#)
- VIA\$PRME CNTL member [223](#)
- VIA\$PRMG CNTL member [223](#)
- VIA\$PRMJ CNTL member [223](#)
- VIA\$PRMP CNTL member [223](#)
- VIA\$PRMS CNTL member [223](#)
- VIA\$PRMX CNTL member [224](#)
- VIAASMH compiler alias name [177](#)
- VIABCSTM CNTL member [230](#)
- VIACEN CLIST member [230](#)
- VIACEXEC SmartTest module [248](#)
- VIACOBII compiler alias name [177](#)
- VIACOBVS compiler alias name [177](#)
- VIACTBLS file [249](#)
- VIADCSTM CNTL member [230](#)
- VIAEDUSR CLIST member [230](#)
- VIAEPANL CLIST member [230](#)
- VIAGCSTM CNTL member [230](#)
- VIAICSTM CNTL member [230](#)
- VIAIN DD statement [106](#)
- VIAITRN CNTL member [229](#)
- VIAITRN1 CNTL member [228](#)
- VIAITRN2 CNTL member [228](#)
- VIAITRN3 CNTL member [228](#)
- VIAITRNC CNTL member [228](#)
- VIAITRND CNTL member [228](#)
- VIAITRNJ CNTL member [228](#)
- VIAJCL
  - see Analyze-Submit-JCL-Node
- VIAJCL installation option [198](#)

- VIAJCL parameter 42, 239
- VIAJCSTM CNTL member 230
- VIALIST dataset 205, 238
- VIALOG dataset 205, 238
- VIALOG DD statement 218
- VIAMALT 225
- VIAMCSTM CNTL member 230
- VIAMSGS library
  - create or modify 15
  - member 225
- VIAOCSTM CNTL member 230
- VIAOPT3 compiler alias name 177
- VIAOPTII compiler alias name 177
- VIAPRINT DD statement 218
- VIAPUNCH dataset 206, 238
- VIAQUEUE Batch execution queue file 224
- VIARCSTM CNTL member 230
- VIASAJC CLIST member 231
- VIASAJCE CLIST member 230
- VIASAJCN CLIST member 230
- VIASAJCS CLIST member 231
- VIASAJCX CLIST member 231
- VIASAKAP skeleton member 41, 229
- VIASAKRA CNTL member 224
- VIASAKRA skeleton member 229
- VIASAKRU CNTL member 224, 238
- VIASAKRU JCL 74
- VIASAKRX CNTL member 224
- VIASAKRX skeleton member 229
- VIASAKXP skeleton member 41, 229
- VIASALCL CLIST member 231
- VIASALDF CNTL member 224
- VIASALF CNTL member 224
- VIASALFJ 37
- VIASALFU 37
- VIASALFU CNTL member 224
- VIASANJC CNTL member 224
- VIASASMJ 37
- VIASASMJ CNTL member 224
- VIASASMU 37
- VIASASMU CNTL member 224
- VIASBASE CNTL member 224
- VIASBASJ CNTL member 224
- VIASBAT CLIST member 231
- VIASBDAM CNTL member 224
- VIASBIND CNTL member 224
- VIASBIND JCL 35
- VIASBRWS CLIST member 231
- VIASCCOM CLIST member 231
- VIASCCSJ 37
- VIASCCSJ CNTL member 225
- VIASCCSU 37
- VIASCSII CNTL member 225
- VIASCM11 CLIST member 231
- VIASCM12 CLIST member 231
- VIASCMGB CLIST member 231
- VIASCMPR CLIST member 231
- VIASCOBJ 37
- VIASCOBJ CNTL member 225
- VIASCOBU CNTL member 37, 225
- VIASCOPY CNTL member 7–8, 133, 225
- VIASCPNL CNTL member 225
- VIASCPRM CLIST member 231
- VIASCPRM parameter
  - CICS 237
  - CHCLIB 237
  - CIIMOD 238
  - CIIRLIB 237
  - CVSCLIB 237
  - CVSMOD 238
  - CVSRLIB 237
- VIASCTM CNTL member 231
- VIASCUFC CNTL member 225
- VIASCUFJ CNTL member 140, 225
- VIASCUFR CNTL member 140, 225
- VIASCUFU CNTL member 140
- VIASCUST CLIST member 231, 235
- VIASDBRM CNTL member 225
- VIASDDC CNTL member 225
- VIASDDCJ CNTL member 225
- VIASDEL CLIST member 231
- VIASDIFF CNTL member 225
- VIASDINS CNTL member 225
- VIASDLAS CNTL member 225
- VIASDLB CNTL member 225–226
- VIASDLCB CNTL member 226
- VIASDLCD CNTL member 226
- VIASDLET CNTL member 226
- VIASDLH1 CNTL member 226
- VIASDLH2 CNTL member 226
- VIASDLHP CNTL member 226
- VIASDLM2 CNTL member 226
- VIASDLMF CNTL member 226
- VIASDLN1 CNTL member 226
- VIASDLNR CNTL member 226
- VIASDLNS CNTL member 226
- VIASDLU1 CNTL member 226
- VIASDLU2 CNTL member
  - VIASDLU2 226
- VIASDLVS CNTL member 226
- VIASDLVX CNTL member 226
- VIASEDBR CLIST member 231
- VIASEDID CLIST member 231
- VIASEDIM CLIST member 232
- VIASEDIT CLIST member 232
- VIASEDIT edit profile 130
- VIASEX3J CNTL member 226
- VIASEXT CNTL member 226

- VIASEXT1 CNTL member 226
- VIASEXT3 CNTL member 141
- VIASEXT4 CNTL member 141
- VIASEXTJ CNTL member 226
- VIASFAI2 CNTL member 227
- VIASFAIJ CNTL member 36, 227
- VIASFAIR CNTL member 227, 248
- VIASGBL 181
- VIASGBL CLIST member 42, 197, 232
- VIASGENR CNTL member 227
- VIASGVFP CLIST member 232
- VIASIMSJ 37
- VIASIMSJ CNTL member 227
- VIASIMSU 37
- VIASIMSU CNTL member 227
- VIASINST CNTL member 227
- VIASISP CLIST member 232
- VIASISPF 246
- VIASJCLJ 37
- VIASJCLJ CNTL member 227
- VIASJCLU 37
- VIASJCLU CNTL member 227
- VIASJEDT CLIST member 232
- VIASJOB skeleton member 229
- VIASJRPT skeleton member 229
- VIASJSUM skeleton member 229
- VIASLJB CNTL member 227
- VIASLLSP CLIST member 232
- VIASLLSP skeleton member 229
- VIASLPAJ CNTL member 44, 227
- VIASLPXJ CNTL member 44, 227
- VIASMLSI CNTL member 61, 227
- VIASMPRT CNTL member 227
- VIASMSCL CNTL member 227
- VIASNDVJ CNTL member 227
- VIASNDVJ, installing ASG SVC for Endeavor 52
- VIASNDVJ CNTL member 228
- VIASNDVJ Endeavor support JCL 52
- VIASNDVJ procedure to execute VIASSVCE 49
- VIASNDVJ, installing ASG SVC for Endeavor 52
- VIASNDVX CNTL member 55, 228
- VIASNDXJ CNTL member 55, 228
- VIASNINS CLIST member 232
- VIASOFT parameter 15, 41, 236
- VIASPAM CNTL member 228, 248
- VIASPAMJ CNTL member 36, 228
- VIASPCHS load module 246
- VIASPI14 CLIST member 232
- VIASPLIJ 37
- VIASPLIJ CNTL member 228
- VIASPLIU 37
- VIASPLIU CNTL member 228
- VIASPR CLIST member 232
- VIASPR14 CLIST member 232
- VIASPRO1 CLIST member 232
- VIASPRO2 CLIST member 232
- VIASPROC CLIST member 232
- VIASPROF CLIST member 129, 232
- VIASPRTI CLIST member 148, 232
- VIASPRTI CNTL member 228
- VIASPTFR CLIST member 232
- VIASREP CLIST member 232
- VIASSCLA CNTL member 228
- VIASSCLA, Endeavor model SCL template 53
- VIASSCLL CNTL member 228
- VIASSCLL, Endeavor model SCL template 53
- VIASSCLN Endeavor model SCL template 54-55
- VIASSCLR CNTL member 228
- VIASSCLR, Endeavor model SCL template 53
- VIASSVCE Endeavor support load module 52
- VIASSVCE, module to install ASG SVC 49
- VIASUB CLIST member 232
- VIASUB command 230
- VIASUB edit macro 98, 100, 105
- VIASUBDS CLIST member 96, 100, 105, 231, 233
- VIASVPTF CLIST 242
- VIASVPTF CLIST member 233
- VIASVPTX CLIST member 233
- VIAUCSTM CNTL member 230
- VIAUPARM, installation option override file 171
- VIAUT3, work file defaults 184
- VIWORK parameter 236
- VIAXCSTM CNTL member 230
- VIAYBRO CLIST member 233
- VIAYEDIT CLIST member 233
- VIAYINCL user exit file format 39
- VIAYJCLE CLIST member 233
- volume serial 206
- VREGION
  - see Analyze-Region-Size
- VSAM AKR 71
- VSIZE installation option 218
- VSPE2BRO edit panel 129
- VSPE2LIB edit panel 129
- VSPE2PAN edit panel 129
- VSPE2PDS edit panel 129
- VSPE2USR edit panel 129

VSPEDBRO edit panel [129](#)  
VSPEDLIB edit panel [129](#)  
VSPEDPAN edit panel [129](#)  
VSPEDPDS edit panel [129](#)  
VSPEDUSR edit panel [129](#)  
VSPENBRO edit panel [129](#)  
VSPENLIB edit panel [129](#)  
VSPENPAN edit panel [129](#)  
VSPENPDS edit panel [129](#)  
VSPENUSR edit panel [129](#)  
VSPPRIM  
    see Analyze-Work-Primary  
VSPPRIM installation option [106](#)  
VSPSECD  
    see Analyze-Work-Secondary  
VSPSECD installation option [106](#)  
VSPUNIT [182](#)  
    see Analyze-Work-Unit parameter  
VSPUNIT installation option [106](#), [181](#)

## W

warning messages [178](#)  
WORKBLK  
    see Online-Work-Block  
WORKDSN variable [42](#)  
Work-SYSDA installation option [221](#)  
WORKVOL  
    see Online-Work-Volume

## X

XLIVE  
    see Analyze-COBOL-Live-Exits  
XMEM  
    see Analyze-Extra-Memory





ASG Worldwide Headquarters Naples Florida USA | [asg.com](http://asg.com)