ASG-SmartEdit™ Installation Guide

Version 7.0
Publication Number: SET0300-70
Publication Date: February 2003

ASG Worldwide Headquarters Naples Florida USA | asg.com | info@asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 941.435.2200 Fax: 941.263.3692 Toll Free: 800.932.5536 (USA only)

Contents

Preface

This *ASG-SmartEdit Installation Guide* provides a guide for installing and maintaining ASG-SmartEdit (herein called SmartEdit). SmartEdit is a COBOL intelligent editor that increases the speed and accuracy of modifying COBOL code using the ISPF Editor. SmartEdit assists COBOL programmers in understanding:

- Program structure
- How related data fields affect the logic of the program
- The impact of syntax modifications on program logic
- Execution paths of a COBOL program

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

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

About this Publication

This publication consists of these chapters:

- Chapter 1, "Introducing SmartEdit," contains an overview of ASG-ESW and SmartEdit.
- Chapter 2, "Customizing SmartEdit," contains a description of how to customize specific SmartEdit components, how to invoke SmartEdit, and how to validate the installation.

Related Publications

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

- The *ASG-Center Installation Guide* (CNX0300-*nn*) contains installation and maintenance information for ASG-Center, the common set of libraries shared by the ASG-Existing Systems Workbench (ASG-ESW) suite of products. ASG-Center must be installed before installing SmartEdit.
- *ASG-ESW Enhancement Summary* (ESW1000-*nn*) highlights the new functionality for this release.
- The *ASG-SmartEdit Installation Guide* (SET0300-*nn*) provides information about the installation and maintenance of ASG-SmartEdit.
- *ASG-SmartEdit Quick Reference Guide* (SET0600-*nn*) summarizes the syntax and usage of ASG-SmartEdit commands.
- The *ASG-SmartEdit User's Guide* (SET0200-*nn*) provides instructions and commands for using ASG-SmartEdit.

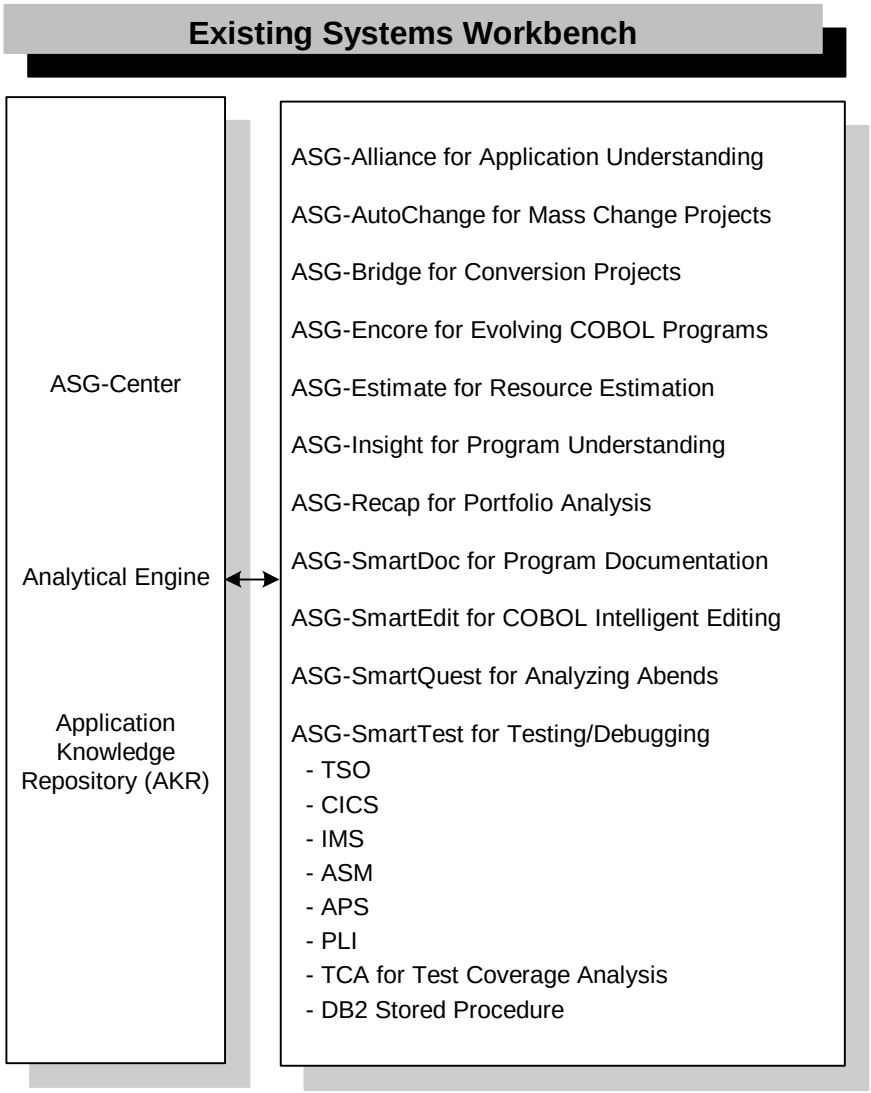
Note:

To obtain a specific version of a publication, contact ASG Customer Support.

ASG-Existing Systems Workbench (ASG-ESW)

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

Figure 1 • ASG Existing Systems Workbench



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

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

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

ESW Product	Herein Called	Description
ASG-SmartEdit	SmartEdit	The COBOL editing component that can be activated automatically when the ISPF/PDF Editor is invoked. SmartEdit provides comprehensive searching, inline copybook display, and syntax checking. SmartEdit allows you to include an additional preprocessor (for example, the APS generator) during syntax checking. SmartEdit supports all versions of IBM COBOL, CICS, SQL, and CA-IDMS.
ASG-SmartQuest	SmartQuest	The diagnostic tool for analyzing batch and CICS transaction abends. SmartQuest has been designed to make the maximum use of simple point-and-shoot techniques to enable fast and easy navigation through any data dump.
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 xi 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 (ESW Name)	Shortcut	ESW Pull-down Options
Alliance (Application Understanding)	AL	Understand ▸ Application
AutoChange (Conversion Set)	CC	Change ▸ Conversion Set
Bridge	BR	Change ▸ ASG-Bridge
Encore (Program Re-engineering)	EN	Re-engineer ▸ Program
Estimate	ES	Measure ▸ ASG-Estimate
Insight (Program Understanding)	IN	Understand ▸ Program
Recap (Portfolio Analysis)	RC	Measure ▸ Portfolio
SmartDoc (Program Documentation)	DC	Document ▸ Program
SmartEdit	SE	Change ▸ Program **Or** Change ▸ Program with Options
SmartQuest	SQV	Understand ▸ Abend/Dump
SmartTest (Testing/Debugging)	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, different fields, values, action bar options, and pull-down options display 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 1

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

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

Figure 2 • Encore Primary Screen

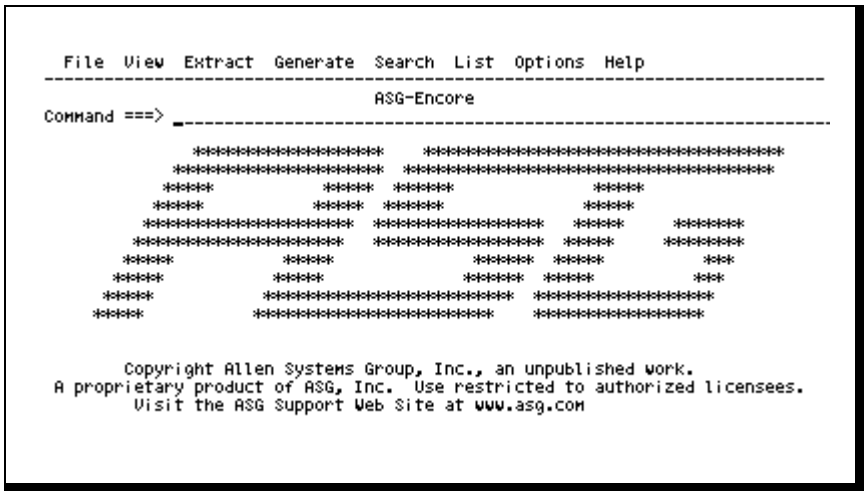
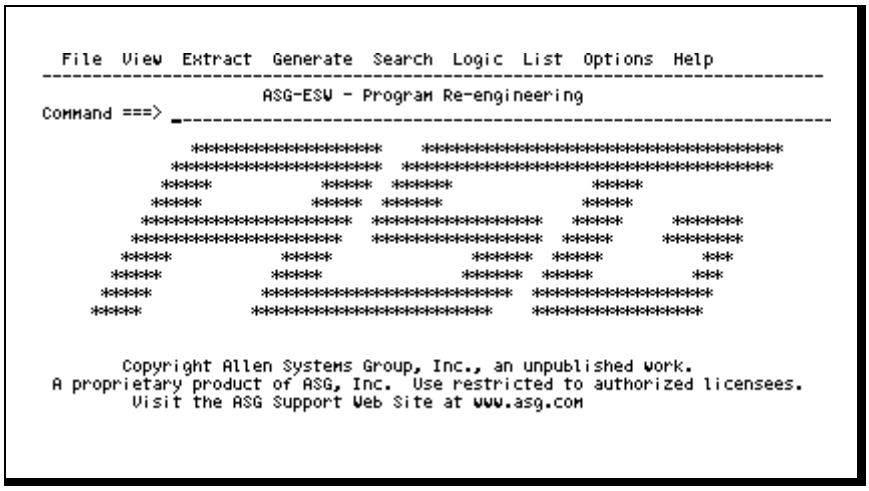


Figure 3 shows the Encore Primary screen that displays when you access Encore through ESW by selecting Re-engineer ▸ Program from the ESW action bar menu. Notice that the Primary screen name changes to ASG-ESW - Program Re-engineering when you enter Encore through ESW. Also, the Logic menu item displays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

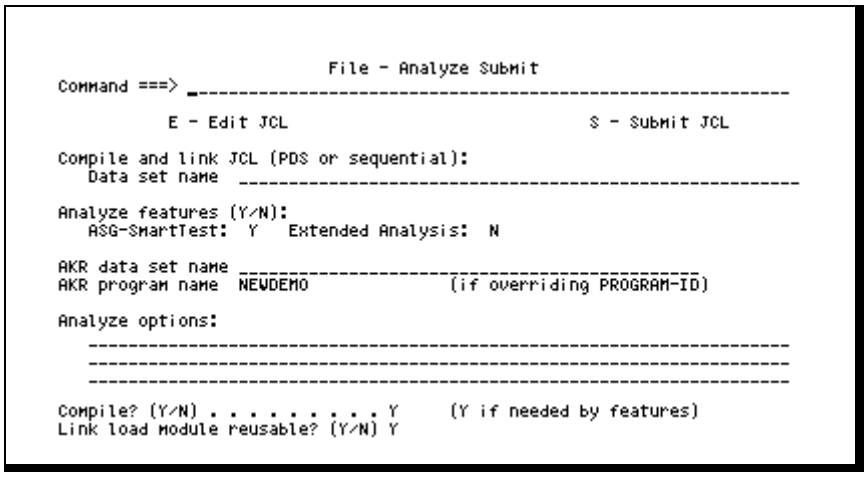
Figure 3 • ESW Encore Primary Screen



Example 2

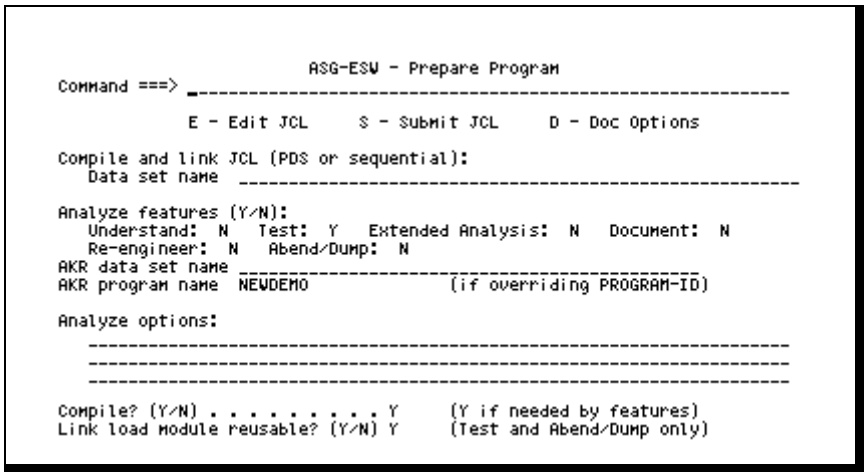
Figure 4 shows the File - Analyze Submit pop-up that displays when you access SmartTest directly. Figure 5 on page xiv shows the File - Analyze Submit pop-up that displays when you access SmartTest through ESW.

Figure 4 • File - Analyze Submit Screen



The actions shown on these screens can also vary. For example, the D - Doc Options action is only available on the File Prepare Program screen (or File - Analyze Submit screen) if SmartDoc is installed on your system. In Figure 4 on page xiii, the Doc Options action is not displayed.

Figure 5 • ASG-ESW - Prepare Program Screen (accessed through ESW)



Notice that the Analyze features field in Figure 5 lists additional ESW products than shown on Figure 4 on page xiii. This field is automatically customized to contain the ESW products you have installed on your system. These are the names of the analyze types:

Analyze Type	Analyze Type (ESW)
ASG-Encore	Re-engineer
ASG-Insight	Understand
ASG-SmartDoc	Document
ASG-SmartQuest	Abend/Dump
ASG-SmartTest	Test
Extended Analysis (ASG-SmartTest with Insight installed)	Extended Analysis

Publication Conventions

ASG uses these conventions in technical publications:

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

ASG Customer Support

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.

ASG Third-party Support. 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.

Intelligent Support Portal (ISP)

Online product support is available at: http://www.asg.com/support/support.asp via the ASG Intelligent Support Portal (ISP). Your logon information for ISP online support is:

Customer ID = *NNNNNNNNN*
Password = *XXXXXXXXXX*

where:

NNNNNNNNN is your customer ID supplied by ASG Product Distribution.
XXXXXXXXXX is your unique password supplied by ASG Product Distribution.

The *ASG-Intelligent Support Portal User's Guide* provides instructions on how to use the ISP and is located on the ASG Support web page.

Telephone Support

To expedite response time, 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 as displayed
- A description of the specific steps that immediately preceded the problem
- Verify whether you received an ASG Service Pack or cumulative service tape 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. You can access the latest software corrections and Service Packs via the ISP.
- The severity code (ASG Customer Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)

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

The Americas

	Phone	Fax	E-mail
United States and Canada	800.354.3578	1.703.464.4901	support@asg.com

Europe, Middle East, and Africa (EMEA)

During normal business hours, we recommend that you call the Central Support number first (except in South Africa).

	Phone	Fax	E-mail
Central Support	00.800.3544.3578	44.1727.812018	support.emea@asg.com
English	44.1727.736305	44.1727.812018	support.uk@asg.com
French	33.141.028590	33.141.028589	support.fr@asg.com
German	49.89.45716.200	49.89.45716.400	support.de@asg.com
Italian	39.0290450025		support.it@asg.com
Dutch	31.30.241.6133		support.nl@asg.com
Spanish	34.913.523.800	34.917.156.961	support.es@asg.com
South Africa	800.201.423		support.sa@asg.com

Asia Pacific (APAC)

	Phone	Fax	E-mail
Central Support	61.3.9645.8500	61.3.9645.8077	support.au@asg.com
Australia	800.637.947	61.3.9645.8077	support.au@asg.com
Hong Kong	800.96.2800		support.hk@asg.com
Japan	81.3.5326.3684	81.3.5326.3001	support.au@asg.com
Singapore	65.224.3080	65.224.8516	support.sg@asg.com

All Other Countries (Also for any non-working numbers)

	Phone	Fax	E-mail
All other countries	1.239.435.2201		support@asg.com

If you receive a voice mail message, follow the instructions to report a production-down or critical problem. Leave a detailed message including your name and phone number. An ASG Customer Support representative will be paged and will return your call as soon as possible. Please have available the information described previously when the ASG Customer Support representative contacts you.

ASG Documentation/Product Enhancements

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 (239) 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.

1 Introducing SmartEdit

This chapter contains an overview of ESW and SmartEdit and contains these sections:

Note:

Installation and customization of Center must be performed before installing SmartEdit. If Center has not been installed, see the *ASG-Center Installation Guide* at this time.

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 ASG Customer Support.

Overview

SmartEdit extends the power of the ISPF environment to enhance and modify COBOL programs efficiently by saving time and improving the logic quality. All command and function facilities are 100% compatible with the ISPF editor, requiring little or no learning curve.

SmartEdit Features

SmartEdit provides these command extensions within ISPF:

- Convenient COBOL sensitive facilities reduce the amount of time it takes to complete basic tasks such as organizing and viewing the source code. These facilities are included:
  - A cursor sensitive BRANCH command that automatically locates code for all COBOL transfer of control verbs such as PERFORMs and GO TOs.
  - A CHECK command that provides online syntax checking of COBOL code. Errors are highlighted and explanatory messages are provided below where the error is encountered.
  - A ZOOMIN command that displays excluded lines of code and expands copybooks in-line and in-context.
  - A ZOOMOUT command that excludes levels of code and reverses the effects of the ZOOMIN command.
  - A ZOOM DEF command that extracts logical database entity definitions from the DB2 system catalog or the IDMS data dictionary.
- The FINDXTND command is an automated COBOL sensitive search that performs these functions:
  - Displays all references by tagging and highlighting the references (REF), uses (USE), modifications (MOD), and definitions (DEF) of all related data fields.
  - Finds all related data such as aliases, renames, redefines, etc., for a specified data field.
  - Locates COBOL verbs and subsets such as COBOLII, CICS, IO, and Exits.
  - Provides an overview of the program structure such as divisions, structures, and perform ranges.
- COBOL understanding features that identify execution patterns between modules, paragraphs, and statements. These features are included:
  - A Tree View facility that exposes the logical structure of a program, such as perform ranges, calls, conditionals, and GO TOs.
  - A PREF (Paragraph Cross Reference) command that shows the program logic at the paragraph level displayed on the View - Paragraph Cross Reference pop-up.

Operating Environment

SmartEdit requires:

- MVS/XA, MVS/ESA, or OS/390
- Storage above the 16M line is used for load modules and GETMAINs
- Direct access storage

See the *ASG-Center Installation Guide* for the quantities.

- 3270 type terminals; Models 2, 3, 4, or 5
- MVS ISPF Version 3.1 or later
- TSO logon region size of 2048 KB or larger

COBOL Support

SmartEdit supports these COBOL compilers:

- COBOL II (including Release 3 and 4)
- ANSI COBOL
- CASE Generated COBOL
- COBOL/370
- COBOL D, E, and F
- COBOL for MVS and VM
- COBOL for OS/390
- Enterprise COBOL Release 3.1

2 Customizing SmartEdit

This chapter describes how to customize SmartEdit and contains these sections:

Center installation and customization must be performed before customizing SmartEdit. If Center has not been installed, see the *ASG-Center Installation Guide* at this time.

Step 1 - Modifying VIA$PRME to Override Default Installation Options

To override the SmartEdit default installation options, follow this step:

- Edit the VIA$PRME (SmartEdit installation options) CNTL member and change the appropriate option.

This is a list of the SmartEdit-specific installation options:

- Change-Man
- ChangeMan-Subsysid
- ChangeMan-Version
- Check-CICS-Options
- Check-Error-Level
- Check-Language
- COBOL-Edit-Profiles
- CUA-Profile-Types
- Editor-Exclude-Command
- Include-Libraries
- Librarian-Expand-COPY
- Panvalet-Expand-COPY
- Smartbrs-Edit-Opts-Panel
- User-Preprocessor

See the Installation Options appendix of the *ASG-Center Installation Guide* for more information about changing these installation option default values.

Step 2 - Adding SmartEdit Modules to MLPA/PLPA

The SmartEdit load module VIAFMAIN is re-entrant and is eligible for location in Extended LPA. These are some advantages of adding modules to MLPA/PLPA:

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

Note:

Copying the module VIAFMAIN to MLPA/PLPA is optional.

It is recommended that you keep the original ESW load library (from the installation tape) as a staging library so that any required PTFs can easily be applied. You can copy the re-entrant modules to Extended LPA and non-reentrant modules to a separate user library.

These steps also require changes to the user's logon or product allocations. Use the CNTL library members VIASLPAJ and VIASLPXJ to perform these copy steps.

Step 3 - Customizing Source Managers and Databases

Panvalet

To customize Panvalet

1. Edit the VIASPAMJ CNTL member by specifying a valid JOB card and the correct values for the VIASOFT, CENTER, ASMBLR, PANLIB, SYSOUT, and SYSDA parameters.

2. Submit this job 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.

3 Set the value of PANVALET-IPNEXIT to YES to specify that the Panvalet/ISPF exit is to be used by SmartEdit. The default is NO. The Center CNTL member VIA$PRMS contains the parameter PANVALET-IPNEXIT.

   The COBOL Editor uses the Panvalet PAM module to read source from the master file. For each member to be accessed, an authorization call is made to the standard Panvalet/ISPF exit named IPNEXIT, if it is available in the system. The authorization request type is for Browse access only.

4 Customize the Edit screen. If Panvalet requires changes to its customized edit screen, these changes may need to be included in the ESW screen VSPEDPAN. If this situation occurs, contact ASG Customer Support for instructions.

If the Panvalet load library is allocated under ISPLLIB and the LIBDEF facility is used, `MEMBER NOT FOUND` warning messages display each time you select a Panvalet member. These are the possible alternatives:

- Add SmartEdit allocations to LOGON and bypass LIBDEF processing.
- Add the Panvalet load library allocation to VIALLIB in LIBDEF processing.
- Set WTPMSG OFF (not recommended).

Librarian

To customize Librarian

1 Edit the VIASFAIJ CNTL member by specifying a valid JOB card and the correct values for the VIASOFT, CENTER, ASMBLR, LIBRLIB, LIBRMAC, SYSOUT, and SYSDA parameters.

   Submit this job to assemble and link-edit the Librarian VIASFAIJ module.

2 If you are running on Librarian R3.9, edit CNTL member VIASFAIR. Comment out these delete statements:

```
DELETE EPLOC=XFAIROPN
DELETE EPLOC=XFAIRMOD
DELETE EPLOC=XFAIRREC
DELETE EPLOC=XFAIRCLS
```

   Submit this job to assemble and link-edit the Librarian VIASFAIR module.

3 Customize the Edit screen. If Librarian requires changes to its customized edit screen, you may need to include these changes in the VSPEDLIB screen. If this situation occurs, contact ASG Customer Support for instructions.

If the Librarian load library is allocated under ISPLLIB and the LIBDEF facility is used, `MEMBER NOT FOUND` warning messages display each time a Librarian member is selected. These are the possible alternatives:

- Add SmartEdit allocations to LOGON and bypass LIBDEF processing.
- Add the Librarian load library allocation to VIALLIB in LIBDEF processing.
- Set WTPMSG OFF (not recommended).

DATACOM/DD

To have SmartEdit resolve the directives from your data dictionary, if your site uses the DATACOM/DD COPYDD directive

1. Edit the VIASDDCJ CNTL member by specifying a valid JOB card and the correct values for the VIASOFT, CENTER, ASMBLR, SYSOUT, SYSDA, and DDCLIB parameters.

2. Submit this job to assemble and link-edit the VIASDDC module.

IDMS

SmartEdit is distributed with the IDMS option set to OFF. When you are editing an IDMS program, select Options ▸ Processing Modes and set the IDMS option to ON. If you want to have the IDMS option set on for all users, see Creating a Site-specific Default Profile in the *ASG-Center Installation Guide*.

DB2

The VIASBIND member of the ESW CNTL library must be modified and executed. If you did not modify this option during Center installation, you can do so now. See the ESW Installation and Customization chapter in the *ASG-Center Installation Guide*.

SmartEdit is distributed with the DB2 option set to OFF. When you are editing a DB2 program, select Options ▸ Processing Modes and set the DB2 option to ON. If you want to have the DB2 option set on for all users, see Creating a Site-Specific Default Profile in the *ASG-Center Installation Guide.*

ChangeMan Support

To enable ChangeMan support

1. Set the value of ChangeMan to YES in the VIA$PRME CNTL member. This allows the ChangeMan processing mode to be set to ON by the user in a SmartEdit session.

2 Select Options ▸ Processing Modes and set the CHANGEMAN option ON when you are editing source members within ChangeMan. SmartEdit is distributed with the CHANGEMAN option set to OFF.

Note:

When the CHANGEMAN option is ON, you cannot modify the copy libraries displayed in the Copy/Include DataSet Name(s) field on the Options - COPY/ Include Libraries screen.

3 Modify the CMNSTG01 and CMNSTG04 panels in ChangeMan by adding these lines after the INIT statement:

```
&ASGPKGNM = &PKGNAME
VPUT ASGPKGNM PROFILE
```

ChangeMan support is only available when these criteria are met:

- You enter the SmartEdit session through the execution of the ChangeMan product.
- SmartEdit is available when ISPF Edit is invoked.
- ChangeMan EXIT 36 is modified to invoke SmartEdit as the primary editor under ChangeMan. SmartEdit uses TSO profile variables to determine if the ChangeMan environment is available. The STGLIB and PKGLVL variables must be present in the shared profile pool prior to invoking SmartEdit. You must insert code within EXIT 36 to create/populate these variables, then use VPUT to place the variables into the shared pool. Contact ASG Customer Support for additional information.

For support of ChangeMan Version 4.1.6 and above, modify the ChangeMan-Subsysid and ChangeMan-Version installation options.

To modify the ChangeMan-Subsysid, follow this step:

▸ Append the ChangeMan subsystem ID (which should be a one character identifier) to SE, i.e., -SERA (where A is the subsystem ID).

To modify the ChangeMan-Version parameter, follow this step:

▸ Add the 3 digit version of ChangeMan you are using, for example, 416 for Version 4.1.6.

See the Installation Options appendix in the *ASG-Center Installation Guide* for more information about the ChangeMan installation options.

Other Source Managers

To support user-developed or other vendor-developed source managers

1. Modify the distributed CLIST VIAEDUSR according to the instructions in the CLIST. VIAEDUSR is invoked when you select SourceManager ▸ Other (User) from the Options - COPY/Include Libraries screen.

2. Construct the user exit VIASEXT1. VIASEXT1 is a user exit program written in Assembler and is distributed in the CNTL installation library. After you modify the exit, use VIASEXTJ to assemble and link the modified exit code.

3. Edit options, if necessary. Changes to the Options - COPY/Include Libraries screen (VEPEOPTS) can be made to reflect the alternate source manager and dataset or member names used.

Step 4 - Customizing User Preprocessors

Note:

If your site uses only the standard CICS/DLI, IDMS, and DB2 translators, you can skip this customization step and proceed to "Step 5 - Adding SmartEdit to ISPF" on page 14 because SmartEdit has built-in support for these translators. However, if your site includes a user preprocessor as part of the compile process, you need to define the user preprocessor program to SmartEdit.

SmartEdit user preprocessor support provides for the execution of site-specific programs that modify source code prior to executing to the COBOL compiler. The support allows users to continue editing the original source while having the ability to display COBOL intelligent search results based on the expanded source from the user preprocessor(s).

A user preprocessor is any program executed during the compile process that alters the source program before the COBOL compile step. For example, this COBOL compile procedure contains two user preprocessors to locate and replace COBOL source:

1. Extract source from Panvalet.

2. Execute a preprocessor to locate and replace COBOL source.

3. Execute a CICS translator.

4. Execute a preprocessor to locate and replace COBOL source.

5. Execute the COBOL compiler.

The preprocessors in this example compile procedure must be defined to SmartEdit using the User-Preprocessor installation option parameter in VIA$PRME. In addition, each user preprocessor must have a CLIST to invoke the preprocessor program.

Within SmartEdit, user preprocessor groups are used to activate the set of steps for the program being edited. The User-Preprocessor installation option defines these user preprocessor groups. Each group is assigned a name. When a user preprocessor is activated in a SmartEdit session, the steps defined for the group are invoked in the order specified. The User-Preprocessor installation option allows you to define a maximum of seven user preprocessor groups. Each group can contain a maximum of seven steps that represent the compile procedure.

Each user preprocessor requires a CLIST for its execution within SmartEdit. The CLIST allocates the files needed for the user preprocessor program, invokes the program, and returns the DDNAME of a file containing the output of the preprocessor. Depending on the user preprocessor group definition, this file either represents the input to the COBOL compiler or the source to be passed to subsequent precompiler steps. When all user preprocessor steps are complete, SmartEdit frees this file.

Definition Steps

To define the User-Preprocessor installation option and the user preprocessor invocation CLIST

1. Begin with a copy of the JCL used to compile the programs that require user preprocessors.

2. Select a Group name to identify this compile procedure.

3. Define a step in the user preprocessor group for each STEP of the compile procedure. For each user preprocessor, use the JCL to determine the file (DDNAME and dataset name) allocations. Use the CLIST model, VIAFUSRS (provided with SmartEdit), as a guide to create the CLIST that invokes the user preprocessor.

To define a user preprocessor to SmartEdit

1. Define the user preprocessor groups by editing the User-Preprocessor installation option located in the VIA$PRME CNTL member.

   Note:

   For information about editing VIA$PRME and the format of the User-Preprocessor installation option, see the Installation Options appendix in the *ASG-Center Installation Guide*.

2. Construct a CLIST for executing each user preprocessor using the procedure described in the next section.

Constructing User Preprocessor CLIST

Center includes the example CLIST VIAFUSRS in the ASG.VIACEN*xx*.CLIST library as a model for constructing your preprocessor CLIST.

To construct a user preprocessor CLIST using VIAFUSRS

Note:

These steps correspond to the steps included in the model CLIST.

1. FREE all DDNAMEs used by the preprocessor as determined from the preprocessor step in the compile JCL.

2. Allocate each DDNAME required for the preprocessor using these considerations:

   — The primary COBOL input DDNAME must be allocated to &INDS.

   — SYSPRINT and SYSOUT files must be allocated with a disposition NEW

   — All other files should be allocated with the disposition SHR

3. Set up Abnormal Exit Error Handling by coding the appropriate return code checks in the CLIST.

   The example CLIST is set up to display the error file using ISPF Browse. Normally, ISPF Browse is invoked based on the value or values of the Preprocessor Return Code.

4. Call the user preprocessor, which involves inserting the CALL to the user preprocessor.

5. FREE all DDNAMEs allocated at the start of the CLIST, except the output DDNAME.

   After a normal exit from the preprocessor, all DDNAMEs allocated to the preprocessor must be FREEd—except the output DDNAME that is produced by the preprocessor and contains the COBOL source code. This DDNAME will be FREEd once SmartEdit has used it.

Step 5 - Adding SmartEdit to ISPF

Depending on your site needs, you can invoke SmartEdit using one of six methods. This section describes each method. Starting SmartEdit when ISPF Edit is invoked is the preferred method because of performance, user access, and installation considerations.

Note:

Review Appendix A, "SmartEdit Operational Considerations," on page 27 before making these changes.

To make SmartEdit available when ISPF Edit is invoked

1 Find the location in the LOGON or other CLIST where ISPF is normally invoked. This is a typical example of one site's initial method of invoking ISPF:

```
ISPF        &STR(&OPT)
```

2 Modify the command using one of these two options:

```
ISPSTART    PGM(VIAFEDIT) +
     PARM(PANEL(ISR@PRIM) OPT(&OPT)) +
     NEWAPPL(ISR)
```

3 Add the initial option, if any, into the OPT() string in the VIAFEDIT PARM() string. To pass parameters to VIAFMAIN proper, insert a colon (:) followed by the desired VIAFMAIN parms after the last ISPF parm. Typically, it is not necessary to pass parameters to VIAFMAIN.

Or

Comment out the ISPF command and replace it with a call to the VIAFISPF CLIST provided on the product tape (located in the ASG.VIACEN*xx*.CLIST dataset). For example:

```
IF &OPT EQ THEN DO
        %VIAFISPF
        END
        ELSE DO
        %VIAFISPF OPT(&OPT)
        END
```

4 Pass the initial option, if any, in an OPT() string.

To make SmartEdit available under ISPF Option 2

Note:

This step is not necessary if you have previously assigned SmartEdit as your default editor.

1 If not using LIBDEF, change the line that handles Option 2 on ISR@PRIM from:

```
2,'PGM(ISREDIT) PARM(P,ISREDM01)'
```

to:

```
2,'PGM(VIAFEDIT) PARM(PGM(ISREDIT) PARM(P,ISREDM01))'
```

2 If using LIBDEF, change the line that handles Option 2 on ISR@PRIM from:

```
2,'PGM(ISREDIT) PARM(P,ISREDM01)'
```

to:

```
2,'CMD(%VIAFOPT2)'
```

To make SmartEdit available when Panvalet or Librarian editors are invoked

Note:

This step is not necessary if you have previously assigned SmartEdit as your default editor.

1 If not using LIBDEF, locate the screen that is used to invoke Panvalet or Librarian and find the line that does the actual invocation. For example:

```
P,'PGM(PSPILINI) NOCHECK'
```

Or

```
L,'PANEL(LIBRPDF) NEWAPPL(LIB@)'
```

2 Modify the existing invocation to use VIAFEDIT and invoke VIAFEDIT with the existing dialog string as a PARM to VIAFEDIT. For example:

```
P,'PGM(VIAFEDIT)                      +
 PARM(PGM(PSPILINI)) NOCHECK'
```

Or

```
L,'PGM(VIAFEDIT)                      +
   PARM(PANEL(LIBRPDF) NEWAPPL(LIB@))'
```

3 If using LIBDEF, locate the screen that is used to invoke Panvalet or Librarian and find the line that does the actual invocation. For example:

```
L,'PANEL(LIBRPDF) NEWAPPL(LIB@)'
```

4 Make a copy of the VIAFOPT2 CLIST (located in ASG.VIACEN*xx*.CLIST). Locate the line in the new CLIST that invokes VIAFEDIT and modify it so that the dialog string used to invoke Panvalet or Librarian is passed to VIAFEDIT as a PARM. For example:

```
ISPEXEC SELECT PGM(VIAFEDIT)          +
        PARM(PGM(PSPILINI)) NOCHECK
```

Or

```
ISPEXEC SELECT PGM(VIAFEDIT)          +
         PARM(PANEL(LIBRPDF) NEWAPPL(LIB@))
```

5 Save the new CLIST. This CLIST may require additional modifications, depending on your site.

6 Modify the line used to invoke Panvalet or Librarian to use the new CLIST you created. For example:

```
P,'CMD(%VIAFPAN)'
```

Or

```
L,'CMD(%VIAFLIB)'
```

where *VIAFPAN* or *VIAFLIB* is the name of the CLIST you created from VIAFOPT2.

To make SmartEdit available under ISPF Option 3.4, E line command

Note:

This step is not necessary if you have previously assigned SmartEdit as your default editor.

1 If not using LIBDEF, change the line that handles Option 4 on screen ISRUTIL from:

```
4,'PGM(ISRUDL) PARM(ISRUDLP)'
```

to:

```
4,'PGM(VIAFEDIT) PARM(PGM(ISRUDL) PARM(ISRUDLP))'
```

2 If using LIBDEF, change the line that handles Option 4 on screen ISRUTIL from:

```
4,'PGM(ISRUDL) PARM(ISRUDLP)'
```

to:

```
4,'CMD(%VIAFOP34)'
```

To make SmartEdit available as a menu selection on an ISPF screen

Note:

This step is not necessary if the ESW product menu was installed as described in the *ASG-Center Installation Guide*.

1 Add this line to the screen that will be used to invoke SmartEdit.

```
%SE+-ASG-SMARTEDIT,COBOL SENSITIVE EDITOR %OPTIONS===>_Z  +
```

2 Add these lines to the)INIT section of the screen:

```
.ZVARS='(VEVOPTS)'
IF(&VEVOPTS=N)
   &VEVOPTS=NO
IF(&VEVOPTS¬=NO)
   &VEVOPTS=YES
```

3 Add these lines to the)PROC section of the screen:

```
IF(&VEVOPTS=N
   &VEVOPTS=NO
IF(&VEVOPTS¬=NO)
   &VEVOPTS=YES
VPUT (VEVOPTS) PROFILE
```

4 Add this line to the statement that processes the screen options:

```
SE,'CMD(SMARTEDT EPARM(&VEVOPTS))'
```

5 If the current ISPF environment is using LIBDEF, use the VIASMEDT CLIST rather than the SMARTEDT CLIST to invoke SmartEdit to make SmartEdit available only through a CLIST. The VIASMEDT CLIST invokes the SMARTEDT CLIST while specifying the NEWAPPL(VIAF) parameter to save and restore the LIBDEF environment.

Step 6 - Adding the SmartEdit-Browse Option to ISPF

If your site has purchased the SmartEdit-Browse option, it can be made available when the SmartEdit-Browse option is selected from an ISPF screen.

To make the SmartEdit-Browse option available as a menu selection on an ISPF screen

1 Add this line to the screen that will be used to invoke the Browse option:

```
% BR +-ASG-SMARTEDIT COBOL SENSITIVE BROWSE OPTION   %OPTIONS===>_Z +
```

2 Add these lines to the)INIT section of the screen:

Note:
This step may have been already done if SmartEdit was installed on this same screen.

```
.ZVARS = '(VEVOPTS)'
IF (&VEVOPTS = N)
    &VEVOPTS = NO
IF (&VEVOPTS ¬= NO)
    &VEVOPTS = YES
```

3 Add these lines to the)PROC section of the screen:

```
IF (&VEVOPTS = N)
    &VEVOPTS = NO
IF (&VEVOPTS ¬= NO
    &VEVOPTS = YES
VPUT (VEVOPTS) PROFILE
```

4 Add this line to the statement that processes the screen transactions:

```
BR, 'CMD(SMARTBRS EPARM(&VEVOPTS))'
```

If the current ISPF environment is using LIBDEF, use the VIASMBRS CLIST rather than the SMARTBRS CLIST to make the SmartEdit-Browse option available through a CLIST. The VIASMBRS CLIST invokes the SMARTBRS CLIST while specifying the NEWAPPL(VIAF) parameters to save and restore the LIBDEF environment.

Step 7 - Validating SmartEdit

Use the form in Appendix B, "Installation Checkout," on page 33 to assist in verifying the installation.

Note:

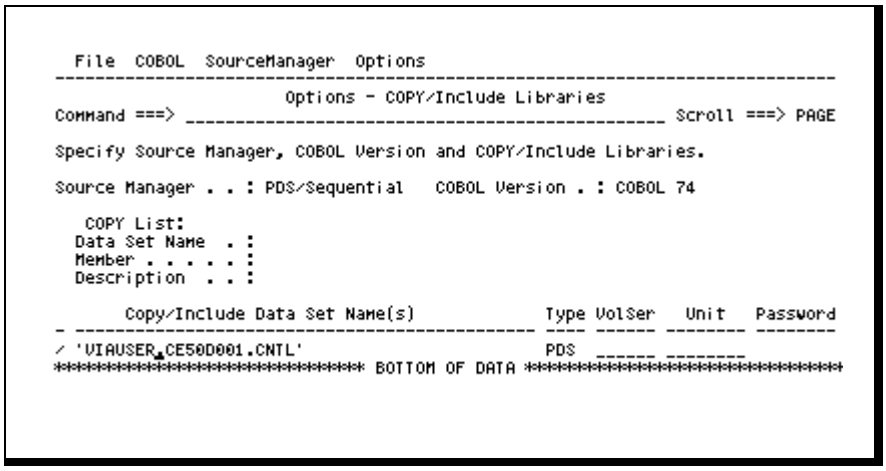
The dataset names used in these validation steps are the default installed names. If you have changed them, use the changed names in place of the default names.

To verify your SmartEdit installation

1 Access SmartEdit by selecting the correct option from the appropriate menu at your site. The Options - COPY/Include Libraries screen displays.

SmartEdit can be started as described in previous steps. If you start SmartEdit using the ESW product menu, select Change ▸ Program with options to display the Options - COPY/Include Libraries screen (see Figure 6). When you start SmartEdit using the ESW product menu, the product name displays as ESW - COBOL Editing.

Figure 6 • Options - COPY/Include Libraries Screen



2 Complete these steps:

a Verify that the Source Manager field specifies PDS/Sequential. If it does not, select SourceManager ▸ PDS/Sequential to change the value.

b Verify that the COBOL Version field specifies COBOL 74. If it does not, select COBOL ▸ COBOL 74 to change the value.

c Type `ASG.VIACEN`*`xx`*`.CNTL` in the Copy/Include Data Set Name(s) field, and `PDS` in the Type field. Type a `/` (slash) in the line command area beside the Copy/Include Data Set Name.

Press Enter to display the ISPF Edit - Entry screen.

3 Type `ASG.VIACENxx.CNTL` for the PDS name and `VIAFDEMO` for the Program name.

4 Press Enter to display the Editor screen for VIAFDEMO.

Note:

If the CUA action bar does not appear on the editor screen, type `CUA ON` in the command input area.

To verify product release information

1 Select Help ▸ About to display the Help - About pop-up. Verify the product release and level of SmartEdit. The product name, product release number, product maintenance level, and the operating system are indicated.

2 Press PF3 to exit and return to the ISPF Editor screen.

To validate the product options

Note:

This step is recommended before proceeding to verify valid product options for your logon user ID. See the online help or the *ASG-SmartEdit User's Guide* for more information about these options.

1 Select Options ▸ Product Parameters. Review and/or modify the parameter definitions, and press PF3/15.

2 Select Options ▸ Product Allocations. Review and/or modify the allocation information, and press PF3/15.

3 Select Options ▸ Log/List/Punch. Review and/or modify the log, list, and punch file defaults. Enter the Job statement information, and press PF3/15.

4 Select Check ▸ Compiler Options. Review and/or modify the check parameters, and press PF3/15.

To verify that SmartEdit terminates normally, follow this step:

- Exit SmartEdit by pressing PF3/15. Depending on how you accessed SmartEdit, you may have to press PF3/15 multiple times.

To validate each type of Source Manager

1. Enter SmartEdit for each type of Source Manager installed by using the invocation method installed for your site. The Options - COPY/Include Libraries screen displays. Select the desired Source Manager to be verified.

2. Press Enter. The Edit - Entry screen displays.

3. Type `ASG.VIACEN`*`xx`*`.CNTL` as the PDS name and `VIAFDEMO` as the member. Press Enter to display the Editor screen for VIAFDEMO.

4. Select Check ▸ Perform Syntax Check. The message `0 ERRORS FOUND` displays in the short message area at the upper right of the screen.

5. Type `COBEDIT OFF` and press Enter. The CUA action bar disappears and the top line of the screen changes to show EDIT in the top left corner (or PVEDIT for Panvalet and ELIPS EDIT for Librarian).

   Type `COBEDIT ON` and press Enter. The CUA action bar redisplays and the screen changes to show SmartEdit Rx.x (or ESW - COBOL Editing) at the beginning of line 3.

6. Select Search ▸ String to display the Search - Pattern String pop-up. Type `ZIP-CODE` in the String field, and press Enter. The short message `6 PATTERNS FOUND` displays in the short message area.

7. Select Search ▸ Data to display the Search - Data Name pop-up. Type `ZIP-CODE` in the Data Name field, and press Enter. The long message `ASG0639I 10 DATA REFS, 4 DEFS, 4 USES, 2 MODS, FOUND FOR ZIP-CODE` displays.

8. Type `KEYS` on the command line and press Enter. The Editor PF Key Definitions and Labels screen displays. Ensure that PF6/18 is set to RCHANGE, and PF10/22 is set to BRANCH.

9. Type `END` and press Enter to return to the Editor screen.

10. Select View ▸ Paragraph X-Ref to display the View - Paragraph Cross Reference Request pop-up and complete these steps:

    a. Type `PROGRAM-INIT` in the Label field.

    b. Select Previous in the Direction field.

    c. Press Enter. The View - Paragraph Cross Reference pop-up displays, with the TARGET field set to PROGRAM-INIT.

11. Type `END` and press Enter. The Editor screen displays.

12 Select Search ▸ Branch to display the Search - Branch Request pop-up and complete these steps:

a Type `P000-NEXT` in the Target field.

b Select Label in the Target Type field.

c Press Enter. The screen is scrolled to the paragraph label P000-NEXT.

13 Select Options ▸ Refresh. The Options - COPY/Include Libraries screen displays.

14 Press Enter. The Editor screen displays with the message `REFRESH COMPLETED`.

15 Select Search ▸ String to display the Search - Pattern String pop-up and complete these steps:

a Type `COPY` in the String field and press Enter.

b Type `++INCLUDE` in the String field for Panvalet validation. Type `-INC` in the String field for Librarian validation and press Enter. The message `1 PATTERN FOUND` displays.

16 Press PF5 (RFIND command). The message `BOTTOM OF DATA REACHED` displays.

17 Type `ZOOMIN` in the primary command area, move the cursor to the COPY, ++INCLUDE, or -INC statement, and press Enter. The message `25 LINE(S) COPIED IN` displays.

18 Select View ▸ Tree to display the View - Tree View Request pop-up and complete these steps:

a Type `MAX` in the LEVELS field and `1` in the Range field.

b Type `VIAFDEMO` in the Name field.

c Press Enter. The message `ASG0909I 5 OF 5 LEVELS DISPLAYED IN TREEVIEW` displays.

19 Type `END` and press Enter. The Editor screen displays.

20 Select Search ▸ Any to display the Search - Any/Unknown Type pop-up and complete these steps:

a Enter * (asterisk) for the Target.

b Select Print in the Action field.

c Press Enter.

The message `497 LINES PRINTED` displays.

21 Type `RECALL` in the primary command area and press Enter. LPRINT * ALL displays in the command input area. Clear the command line.

22 Select Search ▸ Data to display the Search - Data Name pop-up and complete these steps:

a Type `ZIP-CODE` in the Data Name field.

b Select All references in the Data References field.

c Select Of Size Change in the Indirect Impact field.

d Press Enter.

The message `ASG0639I 113 DATA REFS, 65 DEFS, 24 USES, 24 MODS, 31 LEVELS FOUND FOR ZIP-CODE` displays.

23 Type `HELP FINDXTND` in the primary command input area and press Enter. The long message `ASG4710I FINDXTND HIGHLIGHTS THE REQUESTED TARGETS` displays.

a Press PF1. A set of =NOTE= lines is inserted into the display expanding on the definition of the FINDXTND command.

b Press PF1 again. The help tutorial for FINDXTND displays.

c Press PF3/15 to return to the Editor screen. Type `RESET` in the primary command area and press Enter to remove the message and the `=NOTE=` lines from the display.

24 Select File ▸ Cancel. The Edit Entry screen displays. Press PF3/15 until the ISPF screen used to invoke SmartEdit displays.

To validate SmartEdit for Panvalet

1. Copy VIAFDEMO, VIAFMAST, and VIAFDEM1 from the CNTL library to your Panvalet library. Change the COPY VIAFMAST statement to ++INCLUDE VIAFMAST in VIAFDEMO and VIAFDEM1.

2. Enter SmartEdit by selecting the correct option on the appropriate menu determined by the method of invocation used at the site. The Options - COPY/Include Libraries screen displays.

3. Select Source Manager ▸ Panvalet to change the Source Manager field. If necessary, change the COBOL version field by selecting COBOL ▸ COBOL 74. Type the name of your Panvalet library in the COPY/Include Data Set Name(s) field. Type `PAN` in the Type field.

4. Press Enter. The PVEDIT Entry screen displays. Enter your Panvalet library name as the Panvalet library name, and VIAFDEMO as the program name.

5. Press Enter again. Continue the Panvalet validation using step 1 on page 21, beginning with step 4 on page 21, the Check facility.

To validate SmartEdit for Librarian

1. Copy VIAFDEMO, VIAFMAST, and VIAFDEM1 from the CNTL library to your Librarian library. Change the COPY VIAFMAST statement to -INC VIAFMAST in VIAFDEMO and VIAFDEM1.

2. Enter SmartEdit by selecting the correct option on the appropriate menu determined by the method of invocation at the site. The Options - COPY/Include Libraries screen displays.

3. Change the Source Manager field by selecting Source Manager ▸ Librarian. If necessary, change the COBOL version field by selecting COBOL ▸ COBOL 74. Type the name of your Librarian library in the COPY/Include Data Set Name(s) field. Type `LIB` in the Type field.

4. Press Enter. The ELIPS screen displays. Enter your Librarian library name as the Librarian LIBRARY, and VIAFDEMO as the program name.

5. Press Enter. Continue Librarian validation using step 1 on page 21, beginning with step 4 on page 21, the Check facility.

To validate IDMS by editing an IDMS program

1. Enter SmartEdit and select an IDMS program for editing.

2. Select Options ▸ Processing Modes and set IDMS ON.

3 Type `FX DATA` with an IDMS record or map as the target of the command.

4 Verify the command completed successfully and did not generate any error messages or log files.

5 Select Check ▸ Perform Syntax Check to verify that the preprocessors are specified correctly.

To validate DB2 by editing a DB2 program

Note:

If you have DB2, verify that ASG.VIACEN*xx*.CNTL(VIASBIND) has been run (see the *ASG-Center Installation Guide*). This installation step is required for sites having DB2. Failure to complete the DB2 installation step results in erroneous behavior by ESW products.

1 Enter SmartEdit and select a DB2 program for editing.

2 Select Options ▸ Processing Modes and set DB2 ON.

3 Issue the FX DATA command on a variable that is not locally declared.

4 Verify the command completed successfully and did not generate any error messages or log files.

5 Select Check ▸ Perform Syntax Check to verify that the preprocessors are specified correctly.

To validate user preprocessors

Note:

See "Step 4 - Customizing User Preprocessors" on page 11 for more information.

1 Enter SmartEdit and select a program that requires preprocessors.

2 Select Options ▸ Preprocessors. The Options - Preprocessors pop-up displays.

3 Select the desired preprocessor group by entering a non-blank in the selection field beside the group.

4 Select Check ▸ Perform Syntax Check to execute the preprocessor group. Verify that the user preprocessor(s) executed correctly. If errors were found, the error messages should be displayed with the original source statement.

The customization and checkout of SmartEdit is complete.

Appendix A
SmartEdit Operational Considerations

Overview

To make SmartEdit available automatically, the ESW module must get control whenever ISPF Edit is invoked. To accomplish this, use the VIAFEDIT load module to install the Edit Monitor into ISPF Edit. VIAFEDIT installs and activates the Edit Monitor and passes control to a specified ISPF/PDF dialog. The Edit Monitor works with ISPF to manage invoking ISPF Edit. The Edit Monitor also monitors ISPF Edit commands to determine whether the SmartEdit module (VIAFMAIN) will be loaded, called, or deleted.

VIAFEDIT Load Module

VIAFEDIT invokes the Edit Monitor, specifying a dialog string and a SmartEdit configuration.

The parm string, or the portion of the parm string before the first colon (outside of quotes and parentheses), is treated as the dialog string. The dialog string is invoked using the ISPF SELECT service. SmartEdit is available throughout the execution of the dialog. During any given Editor session, SmartEdit is initially active or inactive, according to the COBEDIT parameter in the default CSECT (VIASPRMS).

When the first command is entered, the load module VIAFMAIN is invoked. The portion of the parm string following the first colon (other than quotes and parentheses) is used as the VIAFMAIN parm string. VIAFMAIN processes the command just entered and any subsequent commands. When the dialog terminates, VIAFMAIN terminates. Before the dialog terminates, the VIAFSTOP CLIST can be used to terminate VIAFMAIN.

Examples

Note:

These examples only apply if you are not using LIBDEF.

To make SmartEdit available only during the execution of ISPF Option 2, follow this step:

- Change the Option 2 invocation on ISR@PRIM from:

```
2,'PGM(ISREDIT) PARM(P,ISREDM01)'
```

to:

```
2,'PGM(VIAFEDIT) PARM(PGM(ISREDIT) PARM(P,ISREDM01))'
```

To make SmartEdit available throughout an ISPF session, the user may start ISPF with this command:

```
ISPSTART     PGM(VIAFEDIT) +
             PARM(PANEL(ISR@PRIM):STM=200) +
             NEWAPPL(ISR)
```

In this example, VIAFMAIN would get the parm string 'STM=200'.

To make SmartEdit available only during execution of Panvalet, follow this step:

- Change the Panvalet invocation from:

```
P,'PGM(PSPILINI) NOCHECK'
```

to:

```
P,'PGM(VIAFEDIT) PARM(PGM(PSPILINI)) NOCHECK'
```

To make SmartEdit available only during execution of Librarian, follow this step:

- Change the Librarian invocation from:

```
L,'PGM(ELIPS) NOCHECK'
```

to:

```
L,'PGM(VIAFEDIT) PARM(PGM(ELIPS)) NOCHECK'
```

Use of the VIAS Application ID under VIAFEDIT causes problems with PF key usage (i.e., if the screen is split and another ESW product is running simultaneously), and is therefore not recommended. To keep the PF keys under VIAFEDIT separate from those under the standard ISPF/PDF editor, use an application ID such as VIAF.

For example, this line provides the screen option SE. The option behaves like the primary option 2, except that SmartEdit is available and the PF keys are kept separate:

```
SE, 'PGM(VIAFEDIT) PARM(PGM(ISREDIT) PARM(P,ISREDM01)
NEWAPPL(VIAF))'
```

VIAFPDF Load Module

VIAFPDF executes the VIAFISPF CLIST that invokes the Edit Monitor. You can copy and rename VIAFPDF to ISPF or PDF to make SmartEdit available when users access ISPF using the ISPF or PDF commands. The library containing the ESW load module must be allocated to STEPLIB in the logon PROC ahead of any libraries containing ISPF system modules. Additionally, the remainder of the allocations needed by SmartEdit must be done either in the logon PROC or logon CLIST.

Note:

Renamed versions of VIAFPDF should not be copied from the ESW libraries to prevent possible confusion with system load modules.

VIAFISPF CLIST

The VIAFISPF CLIST invokes ISPF with SmartEdit available. VIAFISPF requires preallocation (e.g., at logon) of ESW libraries.

To have VIAFISPF executed automatically at logon, find the place in the logon CLIST where ISPF is normally invoked. This is an example of an initial method of invoking ISPF:

```
ISPF &STR(&OPT)
```

You can make these changes to the statement:

```
IF &OPT EQ THEN DO
        %VIAFISPF
END
ELSE DO
        %VIAFISPF OPT(&OPT)
END
```

VIAFOPT2 CLIST

The VIAFOPT2 CLIST invokes ISPF Option 2 with SmartEdit available.

Using VIAFOPT2 is not recommended, since the LIBDEF allocations significantly slow the process of invoking Option 2. If the ESW libraries can be preallocated (e.g., at logon), then you can avoid this effect by changing the line that handles Option 2 from:

```
2,'PGM(ISREDIT) PARM(P,ISREDM01)'
```

to:

```
2,'PGM(VIAFEDIT) PARM(PGM(ISREDIT) PARM(P,ISREDM01))'
```

To use VIAFOPT2, follow this step:

- Modify the primary screen (ISR@PRIM), changing the line that handles Option 2 from:

```
2,'PGM(ISREDIT) PARM(P,ISREDM01)'
```

to:

```
2,'CMD(%VIAFOPT2)'
```

VIAFOP34 CLIST

The VIAFOP34 CLIST invokes ISPF Option 3.4 with SmartEdit available.

Using VIAFOP34 is not recommended, since the LIBDEF allocations significantly slow the process of invoking Option 3.4. If the ESW libraries can be preallocated (e.g., at logon), then you can avoid this effect by changing the line that handles Option 3.4 from:

```
4,'PGM(ISRUDL) PARM(ISRUDLP)'
```

to:

```
4,'PGM(VIAFEDIT) PARM(PGM(ISRUDL) PARM(ISRUDLP))'
```

To use VIAFOP34, follow this step:

- Modify the screen ISRUTIL, changing the line that handles Option 3.4 from:

```
4,'PGM(ISRUDL) PARM(ISRUDLP)'
```

to:

```
4,'CMD(%VIAFOP34)'
```

VIAFSTOP CLIST

The VIAFSTOP CLIST stops the main SmartEdit load module (VIAFMAIN).

The VIAFSTOP CLIST causes VIAFMAIN to terminate, freeing about 1.8MB of memory, while SmartEdit remains available. VIAFMAIN restarts when you enter a command in the Editor.

This CLIST requires preallocation of the ESW libraries.

SMARTEDT CLIST

The SMARTEDT CLIST is used to invoke SmartEdit from its own selection on an ISPF screen in a manner similar to that which is used to invoke other ESW products.

VIASMEDT CLIST

The VIASMEDT CLIST is used to invoke SmartEdit from any location under ISPF (except while under Panvalet). This CLIST is valuable for installation checkouts if delays occur while changes are made to the LOGON PROC or CLIST and ISPF screens.

VIAFUSRS CLIST

The VIAFUSRS CLIST is a skeleton CLIST used to create a CLIST to execute a user preprocessor with the SmartEdit CHECK facility.

VIAFUSRC CLIST

The VIAFUSRC CLIST is a sample CLIST that can be used to execute a user preprocessor within SmartEdit. VIAFUSRC uses IEBGENER as if it were a preprocessor.

VIAFUSRI CLIST

The VIAFUSRI CLIST is a sample CLIST that can be used to execute a user preprocessor within SmartEdit. VIAFUSRI specifies the IDMS preprocessor.

Appendix B
Installation Checkout

Company: _____

Installer: _____

Date: _____ / _____ / _____

Product Test

EDIT a member from each type of Source Manager that will be used (PDS/Sequential, Panvalet, and Librarian). During the EDIT session, execute these commands:

OK	Pending	Checklist
____	____	• CHECK. Issue the CHECK command using as many of the preprocessor support options as apply for this site (CICS, DLI, CICS/DLI, DB2, CICS/DB2, IDMS).
____	____	• COBEDIT OFF
____	____	• COBEDIT ON
____	____	• FX PATTERN <*text string*>
____	____	• FX <*data name*>
____	____	• KEYS
____	____	• PREF <*label name*>
____	____	• BRANCH. Issue the BRANCH command to a label not currently displayed on the screen.
____	____	• REFRESH. Issue the REFRESH command to ensure access to fresh copy libraries.

OK	Pending	Checklist
____	____	• ZOOMIN. Issue the ZOOMIN command to ensure access to Copy and Include members from their libraries.
____	____	• TREEVIEW
____	____	• LPRINT *
____	____	• RECALL.
____	____	• FINDXTND <*data name*> INDIRECT. Issue this command to ensure that tags appear.
____	____	• HELP FINDXTND to ensure three levels of help.
____	____	• SPF commands (FIND or CHANGE)
____	____	• END or PF 3/15 to quit.
____	____	• Test the SmartEdit functions using all available Source Managers (PDS/Sequential, Panvalet, and Librarian).

Appendix C
SmartEdit CNTL and CLIST Members

SmartEdit CNTL Members

Member	Description
VIA$PRME	Contains the SmartEdit installation option parameters.
VIAFCALC	Contains the SmartEdit COBOL training laboratory problem program.
VIAFDEMO	Contains a SmartEdit COBOL test program.
VIAFDEM1	Contains a SmartEdit COBOL test program.
VIAFDEM3	Contains the SmartEdit COBOL II Release 3 test program.
VIAFMAST	Contains the copy member for VIAFDEMO.
VIAFPRIM	Contains the SmartEdit COBOL training laboratory problem program.
VIAFPRT	Contains the SmartEdit COBOL training laboratory problem report program.

SmartEdit-Browse Option CLIST Members

Member	Description
SMARTBRS	The CLIST used by VIASMBRS to invoke the SmartEdit-Browse option.

Member	Description
VIAFBRWS	The CLIST used to invoke the SmartEdit-Browse option from the ESW screen.
VIASMBRS	The CLIST used to invoke the SmartEdit-Browse option with NEWAPPL(VIAF) to preserve and restore the existing LIBDEF environments.

SmartEdit CLIST Members

Member	Description
VIAEDUSR	The user CLIST to support source managers other than ISPF options of Librarian and Panvalet.
SMARTEDM	The CLIST to invoke SmartEdit with an explicit member name. This CLIST is intended as a replacement for the ISPEXEC EDIT service.
SMARTEDT	Used by the VIASMEDT CLIST to invoke the SmartEdit product from native TSO.
VIAFEDIM	The CLIST used internally by SmartEdit.
VIAFISPF	The CLIST that invokes the Edit Monitor to make SmartEdit available for all ISPF Edit functions.
VIAFOPT2	The CLIST that invokes the Edit Monitor to make SmartEdit available under ISPF Option 2.
VIAFIMPJ	The CLIST used internally by SmartEdit to import copy library definitions.
VIAFOP34	The CLIST that invokes the Edit Monitor to make SmartEdit available under ISPF Option 3.4.
VIAFSTOP	Is the CLIST that terminates the VIAFMAIN load module.
VIAFTEST	Invokes the SmartEdit program under TSO test for diagnostic purposes only.
VIAFUSRC	The sample user preprocessor CLIST that invokes IEBGENER as the user preprocessor.

Member	Description
VIAFUSRI	The sample user preprocessor CLIST that invokes the IDMS precompiler as the user preprocessor.
VIAFUSRS	The skeleton CLIST for invoking User Preprocessors.
VIASMEDM	The CLIST to invoke SmartEdit using the SMARTEDM CLIST with NEWAPPL(VIAF) for preserving and restoring the LIBDEF environment.
VIASMEDT	The CLIST to invoke SMARTEDT CLIST with NEWAPPL(VIAF) to preserve and restore existing LIBDEF environments.

Index