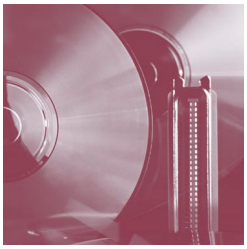
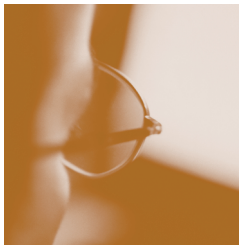
ASG-SmartDoc™ User's Guide

Version: 6.0

Publication Number: DCX0200-60

Publication Date: February 2002





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

ASG Documentation/Product Enhancement Fax Form

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

Company Name	Telephone Number	Site ID	Contact name

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

Enhancement Request:

ASG Support Numbers

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

Please have this information ready:

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

If You Receive a Voice Mail Message:

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

Severity Codes and Expected Support Response Times

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

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

Business Hours Support

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

Non-Business Hours - Emergency Support

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

ASG Web Site

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

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

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

Contents

Preface

This *ASG-SmartDoc User's Guide* tells you how to use ASG-SmartDoc (herein called SmartDoc). SmartDoc gives you comprehensive COBOL program knowledge through information produced from static analysis and generated documentation. All documentation SmartDoc generates presents information in an easy-to-read format. Additionally, program structure charts, software metrics, and a program summary report provide you with high level program information.

Allen Systems Group, Inc. (herein called 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, "Introduction," gives an overview of SmartDoc and describes the program documentation produced by SmartDoc, such as program structure charts, software metrics, and program summary reports.
- Chapter 2, "Concepts," contains a detailed description of the concepts and tools used by SmartDoc, including the Application Knowledge Repository (herein called AKR), Program Metrics, Control Flow Analysis, Data Flow Analysis, Subsets, Perform Ranges, Data Items, Input/Output Data Items, and Program Structure.
- Chapter 3, "Getting Started With SmartDoc," contains the information required to run and produce SmartDoc reports.
- Chapter 4, "Techniques," provides a basic understanding of the purpose and usage techniques incorporated in producing SmartDoc reports. Each report is described, followed by a description of its use in a programming task.
- Chapter 5, "Reports," contains descriptions and illustrations of each report produced by SmartDoc. Some reports are shown in the multiple formats produced by different analysis jobs.

- Chapter 6, "File," contains a description of the File pull-down, used to analyze a program, generate the SmartDoc reports, analyze programs prior to use by SmartDoc, and to exit SmartDoc.
- Chapter 7, "View," contains a description of the View pull-down, used to view metrics data for the programs that reside in the AKR.
- Chapter 8, "Options," contains a description of the Options pull-down, used to access the pop-ups that customize the SmartDoc environment. Customizing the SmartDoc environment includes defining and processing the Log file, and determining the values of the PF keys.
- Chapter 9, "Help," contains a description of the Help pull-down, used to access the Online Help facility. This chapter describes the actions available on the Help pull-down
- Chapter 10, "Metrics," contains a description of SmartDoc generated metrics used to manage the program maintenance life cycle by providing information about the relative complexity and quality of a program.
- Chapter 11, "Analyze," describes the analyze process used by SmartDoc. The analyze process gathers information about the program, including program relationships, logic, data and execution paths, and stores this information in the AKR. After the analyze information is placed in the AKR, it is available to ESW products in online and batch environments, where it is accessed to provide valuable information about the design and operation of user systems.
- Chapter 12, "SmartDoc Options," contains a description of SmartDoc options used to control report generation and to specify various report formats. Most of these options are specified by using the SmartDoc Options screen (when the online component is available). When ISPF is not installed, these options can be specified in the VIAIN DD statement of the analyze job by using the DPARM parameter.
- Chapter 13, "AKR Utilities," contains a description of the online and batch AKR utilities used by SmartDoc.
- Chapter 14, "Online Component Commands," contains a description of SmartDoc's online component commands. These commands are entered on SmartDoc screens in the same manner as ISPF commands, in the command input area on line four.
- Chapter 15, "Help Facility," contains a description of the comprehensive and context sensitive Help facilities provided to answer most questions online. The Help Tutorial contains help information on several subjects, such as screens, pop-ups, reports, commands, messages, and abends. The Help Tutorial also includes a Table of Contents that describes each major SmartDoc function, and a comprehensive Index for viewing specific information.

Related Publications

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

- The *ASG-Center Installation Guide* (CNX0300-*nn*) contains ASG-Center installation and customization procedures. ASG-Center must be installed before ASG-SmartDoc is installed.
- The *ASG-SmartDoc Installation Guide* (DCX0300-*nn*) provides instruction for installing and maintaining ASG-SmartDoc.
- The *ASG-SmartDoc User's Guide* (DCX0200-*nn*) describes ASG-SmartDoc instructions and report generation.

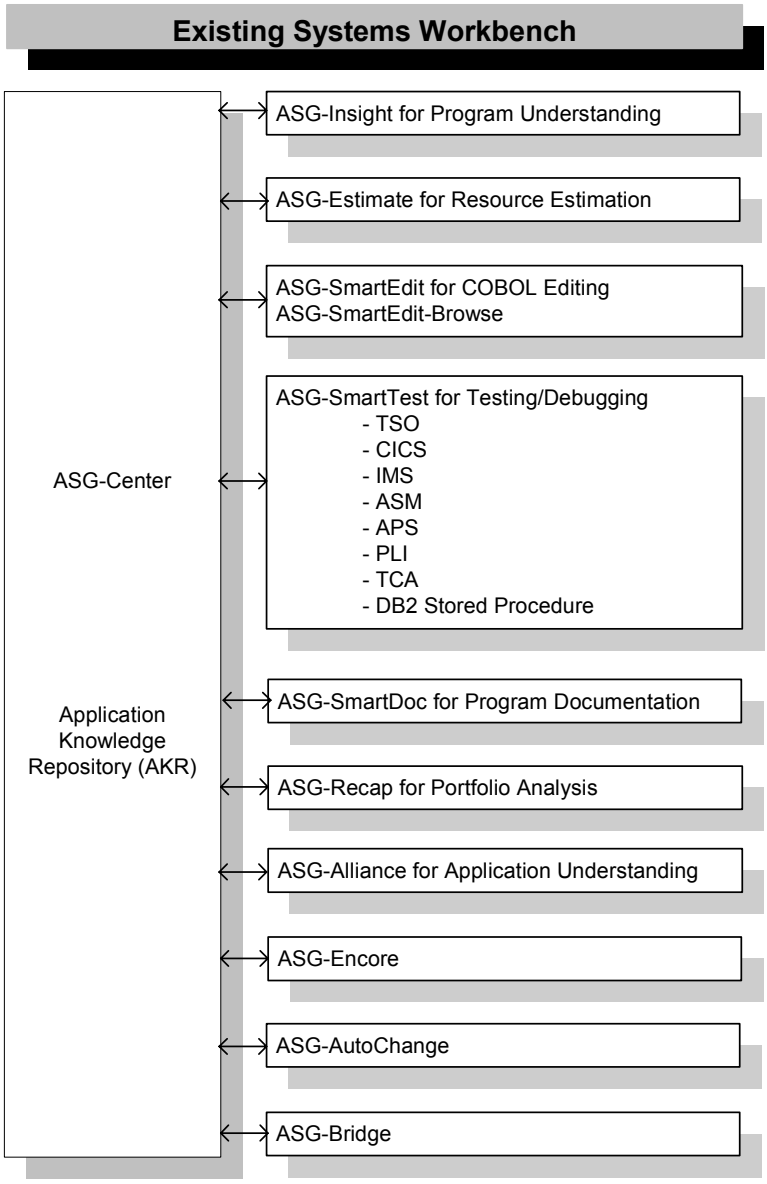
Note:

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

ASG-Existing Systems Workbench (ASG-ESW)

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

Figure 1 • ASG Existing Systems Workbench



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

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

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

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

Invoking ESW Products

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

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

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

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

ESW Product Integration

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

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

Examples

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

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

Figure 2 • Encore Primary Screen

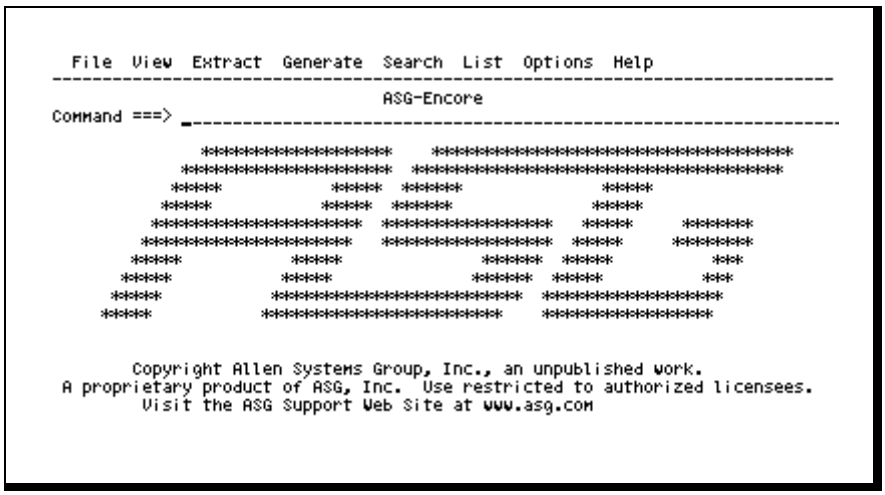
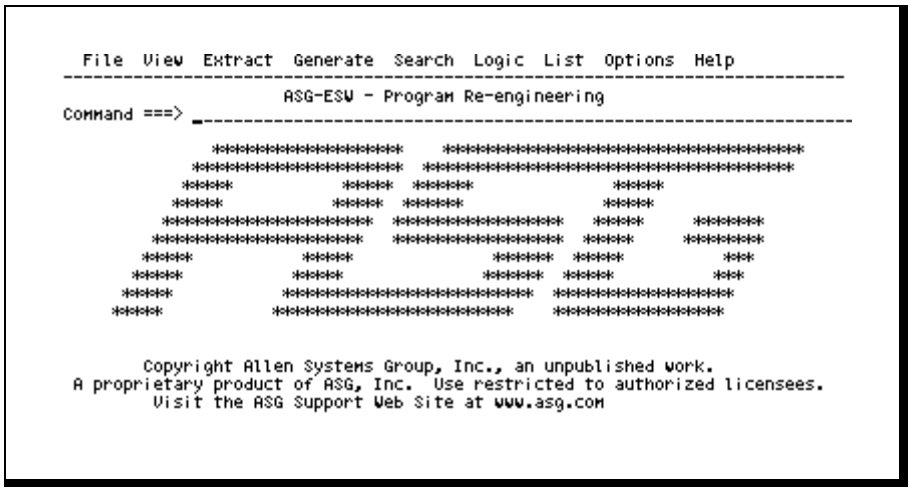


Figure 3 shows the Encore Primary screen that displays when you access Encore through ESW by selecting Re-engineer ▸ Program from the ESW action bar menu. Notice that the Primary screen name changes to ASG-ESW - Program Re-engineering when you enter Encore through ESW. Also, the Logic menu item displays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

Figure 3 • ESW Encore Primary Screen



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

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

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

Figure 4 • File - Analyze Submit Screen

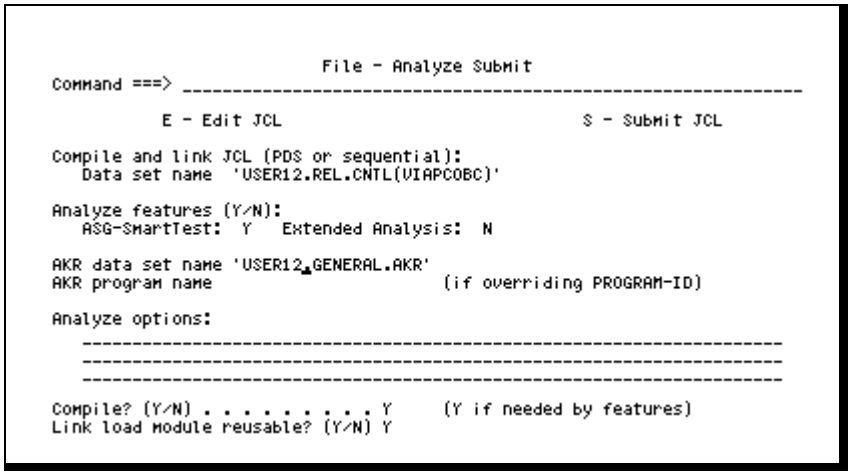
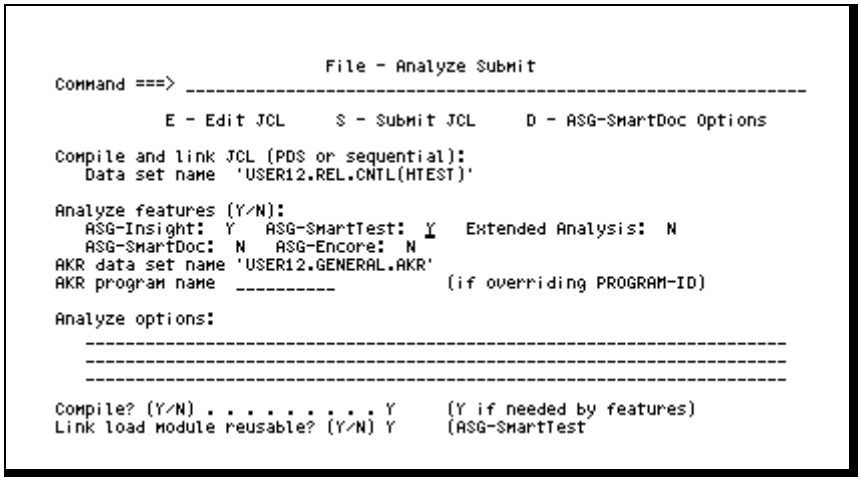


Figure 5 • File - Analyze Submit Screen (Accessed through ESW)



Publication Conventions

ASG uses these conventions in technical publications:

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

1 Introduction

This chapter contains an overview of SmartDoc, describes the program documentation produced by SmartDoc, and contains these sections:

Overview

Programmers and analysts use program documentation to learn about the internal structure and logic of programs. Quality documentation gives these IT professionals a program road-map that helps them maintain and enhance existing systems. SmartDoc provides this information through static program analysis, comprehensive reports, and diagrams.

SmartDoc gives you comprehensive COBOL program knowledge through the information it produces from static analysis and generated documentation. All documentation SmartDoc generates presents information in an easy-to-read format. Additionally, program structure charts, software metrics, and a program summary report provide you with high level program information.

SmartDoc is a part of Center, an integrated product family supporting software maintenance life cycle automation.

SmartDoc is a batch-oriented product. However, you can use an interactive online component to generate requested program documentation, set up and maintain the AKR, and display program metrics.

Program Preparation

SmartDoc prepares a program report by using one of the analyze methods described in Chapter 11, "Analyze," on page 157. The AKR stores the analysis output. Then, this information becomes an integrated component of the program documentation.

SmartDoc Reports

SmartDoc has an interactive user interface used to generate documentation, reports, and diagrams. These are the SmartDoc reports:

SmartDoc Reports	
Table of Contents	Master Index
Program Summary	Metrics report Program Metrics Software Science Volume Metric Control Variable Metric Cyclomatic Complexity Metric Essential Complexity Metric GOTOFAR Metric Perform Range Metrics Software Science Volume Metric Cyclomatic Complexity Metric GOTOFAR Metric
Advanced Source Listing	Paragraph Cross-Reference

SmartDoc Reports	
CALL Statement report	Perform Range Hierarchy Chart
Compiler/Optimizer Output report	Perform Range Usage and Interface report
Condensed Source Listing	Program Exception report
COPY Statement report	Structure Chart
Data Division report	Subset report
Enhanced Data Cross Reference report	Verb Summary

See Chapter 5, "Reports," on page 39 for detailed information on SmartDoc reports.

Interfaces

SmartDoc is a batch-oriented product that you can integrate with compile and link JCL, or you can run stand alone. It includes an interactive, online component you can use to generate program documentation, to set up and maintain the AKR, and to display program metrics.

Note:
These functions can also be accomplished if ISPF is not installed.

The SmartDoc online component now features Common User Access (CUA) screens, action bars, pull-downs, and pop-ups designed for easy access to all product features.

An action bar is the line of keywords displayed at the top of a screen. Each keyword represents a category of actions you can perform on that screen. Process the action bar by selecting an action. Select an action by moving the cursor to the desired keyword and pressing Enter.

A pull-down displays when you select an action on the action bar. On a pull-down, actions followed by three dots (...) display a pop-up when selected. Actions not followed by three dots (...) immediately activate internal commands. There are two ways to select an item on a pull-down:

- Move the cursor to the desired keyword and press Enter.
- Enter the number of the desired action in the input field and press Enter.

A pop-up is a window that displays when you either type a command or select an item on a pull-down or a pop-up. You can enter information for the requested action on a pop-up. Enter the desired data and/or option, then follow the instructions on the pop-up to process the information.

Note:

Press PF3/15 to exit either a pull-down or a pop-up without processing any actions.

Operating Systems

SmartDoc runs under MVS and uses ISPF (Version 3.1 or later) as its standard online user interface. However, you can manually set up and maintain JCL to produce reports if ISPF is unavailable. Use the procedures (PROCs) included with SmartDoc to maintain the AKR.

Printers

SmartDoc supports most printers that use a standard character set, and provides program parameters to use alternate characters if required. See Chapter 8, "Options," on page 137 for more information.

COBOL Support

SmartDoc supports these versions of COBOL:

- COBOL/370
- OS/VS COBOL
- COBOL II (including Release 3)
- CASE generated COBOL

Preprocessor Support

SmartDoc supports these preprocessor languages directly:

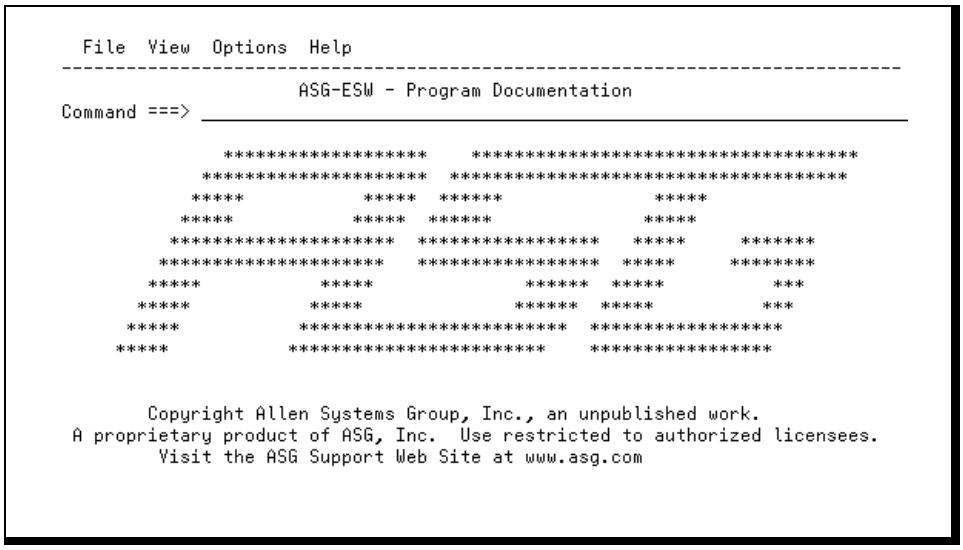
- Command Level CICS
- Command Level DL/I
- CA-IDMS
- SQL

Other preprocessor languages can be supported from the generated COBOL code.

Major SmartDoc Components

The first screen SmartDoc displays identifies major SmartDoc components (see Figure 6). The action bar is one of these, and designates the primary functional organization of the SmartDoc online component. Each action bar item displays a pull-down list of options available for that action.

Figure 6 • Primary SmartDoc Screen



Actions

These are the Primary SmartDoc screen actions:

Action	Description
File	The File action displays the File pull-down, used to analyze a program, submit SmartDoc reports, manage the AKR, and to exit SmartDoc. See Chapter 6, "File," on page 111 for additional information.
View	The View action displays the View pull-down, used to view metrics for a program in the AKR. See Chapter 7, "View," on page 131 for additional information.
Options	The Options action displays the Options pull-down used to set options such as PF key values, Log file attributes, and various options for the interactive process. See Chapter 8, "Options," on page 137 for additional information.
Help	The Help action displays the Help pull-down used to access the Online Help facility. See the online help, and Chapter 9, "Help," on page 147 and Chapter 15, "Help Facility," on page 225 for additional information.

2 Concepts

This chapter contains a detailed description of concepts and tools used by SmartDoc and contains these sections:

Terms

SmartDoc introduces new concepts and terms that are key to its operation. These concepts are described in this chapter:

- AKR
- Program metrics
- Control flow analysis
- Data flow analysis
- Subsets

- Perform ranges
- Data items
- Input and output data items
- Program structure

Application Knowledge Repository (AKR)

The AKR is either an BDAM or a VSAM file organization. The AKR has utilities you can use for allocation and maintenance. You can use the AKR utilities to allocate multiple AKRs. For more information about AKR structure and maintenance, see Chapter 13, "AKR Utilities," on page 193.

The Program Analyzer stores the program information in the AKR after it analyzes a program. This information is about program relationships, logic and data, execution paths, and COBOL intelligence. For more information about the Program Analyzer, see Chapter 11, "Analyze," on page 157.

SmartDoc generates program documentation and metrics from the information the program analysis placed in the AKR.

Program Metrics

Software metrics assess program complexity, architecture, and software quality. They allow organizations to identify programs that need enhanced, re-engineered, or additional resources.

The AKR stores the program metrics information as a separate member (i.e., it is stored separately from the actual program). If you delete the program from the AKR, the metrics information remains intact. Use the Metrics Display and Utilities screen to either rename or delete metrics.

Unlimited versions of metric data are retained for each program, providing information for complexity versus time graphs.

Software Science Volume Metric

The Software Science Volume Metric is a size (or volume) metric based on the premise that the larger the program, the more difficult it is to understand and maintain. The Software Science Volume Metric is defined by using these nine variables. For example:

$n1$= number of distinct operators in the program

$n2$ = number of distinct operands in the program

$N1$= total number of operators in the program

$N2$ = total number of operands in the program

$n = n1 + n2$ = the vocabulary of the program

$N = N1 + N2$ = the length of the program

The volume (V) of a program can then be defined as:

$V = N \log2\ n$

The effort (E) to understand a program can be defined as:

$E = V * V$

SmartDoc reports on program volume (V).

Cyclomatic Complexity Metric

The Cyclomatic Complexity Metric measures logical flow paths. This metric is defined as the number of predicates (or branch) points in the program, and operates on the premise that the number of program paths determine the complexity of the program. The number of predicates is a count of the IF, GO TO DEPENDING, etc., statements (i.e., all conditional branches in the program).

Control Variable Metric

The Control Variable Metric is similar to the Cyclomatic Complexity Metric, except the Control Variable Metric measures the number of program control variables. This metric is based on the premise that programs with equal flow paths, but more variables controlling the flow, are more difficult to understand and maintain than programs with fewer control variables. For example:

Complexity = (number of branch points)

Complexity = (number of branch points) + (number of control variables)

Essential Complexity Metric

The Essential Complexity Metric measures how much a program can be reduced, quantifying how well structured a program is. This metric is defined as the Cyclomatic Complexity Metric minus the number of reducible subgraphs in the program. Reducible subgraphs are created through unstructured constructs.

The value of the Essential Complexity metric is equivalent to the Cyclomatic Complexity Metric after removing all proper, single entry, and single exit subgraphs. For example:

Complexity = (number of branch points - reducible subgraphs)

GOTOFAR Metric

The GOTOFAR metric measures the relative frequency of long GOTOS in a program. A long GOTO is one that jumps out of the current paragraph. The GOTOFAR metric is a rational quantity expressed as a decimal fraction with a value either greater than or equal to zero and less than one. The metric is defined as this ratio, for example:

Complexity = (number of long GOTOs)/(number of PROC DIV statements)

Control Flow Analysis

The analyze job provides the control flow analysis when you perform a SmartDoc analysis. The Advanced Source Listing reports control flow analysis information when you analyze a program. The control flow analysis indicates the control path either to or from a particular point in the code. The PROCEDURE DIVISION of the Advanced Source Listing shows each source statement. Whenever SmartDoc encounters a label (PROCEDURE DIVISION, paragraph, or section label), information on how the control flows to that label is shown on the Advanced Source Listing. The Advanced Source Listing also shows where the source passed control when SmartDoc encounters a COBOL verb affecting control flow. When SmartDoc encounters the end of a paragraph or a perform range, the Advanced Source Listing shows all possible return locations.

Data Flow Analysis

The analyze job provides the data flow analysis when you perform an Extended SmartDoc analysis. Information from the SmartDoc data flow analysis is presented on the Advanced Source Listing for each data item.

When you use a data item, SmartDoc indicates all locations in the source where the current data item value may have been set. When a data item is modified, SmartDoc indicates all locations within the source where the modified value may be used next. A SmartDoc data flow analysis considers all data alias names (renames, redefines, record group items).

Subsets

SmartDoc classifies COBOL statements into subsets by grouping similar COBOL verbs together. For example, lines that contain READ, WRITE, OPEN, or CLOSE verbs can be referenced as the IO subset. The Subset report lists each of the subsets and identifies the paragraphs or the divisions containing them. The paragraph or the division name is shown with the page and line number where that subset occurs in the Advanced Source Listing.

These are the COBOL subsets and their corresponding entities:

COBOL Subset	Description
Assignment	Statements that assign a value, such as MOVE, ADD, and Compute
CALL	Statements that relate to subprogram calls, such as CALL and CANCEL
CICS	Any CICS or DL/1 Command Level statements
COBOLII	PROCEDURE DIVISION statements that are exclusively COBOL II, including CONTINUE, END, and INITIALIZE verbs
COMMENT	Statements having no run-time effect, such as all lines with an asterisk (*) in column 7, the entire IDENTIFICATION DIVISION, and NOTE statements
CONDITIONAL	Statements or the parts of statements that conditionally change the flow of control in a program such as IF, ELSE and WHEN

COBOL Subset	Description
DB2 | SQL	EXEC SQL statements
DDL	SQL Data Definition Language statements, such as CREATE, ALTER, DECLARE and DROP
DEBUG	Statements containing a DEBUG, EXHIBIT, ON, READY, or RESET verb, as well as statements containing a D in column 7
DEFINITION	Declaratives of data items including the SPECIAL-NAMES paragraph in the ENVIRONMENT DIVISION, as well as the entire DATA DIVISION
DIRECTIVES	Statements that direct the complier to take specific actions during compilation, such as BASIS, EJECT, and TITLE
DL/I | DL/1	DL/I Command Level statements
DML	SQL Data Manipulation Language statements, such as SELECT, UPATE, INSERT and COMMENT
ENTRY	The PROCEDURE DIVISION statements and all ENTRY statements
EXIT | PGMEXIT	Statements containing a STOP RUN, GOBACK, or EXIT PROGRAM verb, as well as CALL statements indicated as NORET (non-returning)
FALLTHROUGH	Statements of PERFORMed units that fall through to the next paragraph
FD	Statements containing file definitions
GOTO	Statements containing an ALTER or an GOTO verb
IDMSQ	IDMS statements
IO INPUT OUTPUT	COBOL IO statements (IO, Input, or Output respectively) including CALL statements indicated as containing IO, Input, or Output
DIVISION PARAGRAPH SECTION	Statements containing either DIVISION or SECTION headers, or PARAGRAPH labels
MATH	Statements containing ADD, SUBTRACT, MULTIPLY, DIVIDE, or COMPUTE verbs

COBOL Subset	Description
PERFORM	Statements containing the PERFORM, SORT, or MERGE verbs
SORTMERGE	Statements containing SORT, MERGE, or RELEASE verbs; a paragraph or a section name referred to in INPUT/OUTPUT PROCEDURES
01 AND 77 LEVELS	Statements containing either 01 or 77 data definition

Perform Ranges

A perform range consists of all the code that executes by following a PERFORM statement. For example:

```
PERFORM PARAGRAPH-ABC THRU PARAGRAPH-XYZ.
```

The prior statement indicates the paragraphs range begins with *PARAGRAPH-ABC* and continues through the end of *PARAGRAPH-XYZ*, including all paragraphs executed between.

Data Items

A data item can be any of these six listed types:

- Elementary dataname
- File name
- Group name
- Table name
- Table element name
- Special name

SmartDoc supports any legal COBOL reference for a data element. Also, SmartDoc identifies a redefined variable by both the specified name and the redefined name, and treats a table entry reference as a reference to the entire table. All references are reported when data items overlap and a name can refer to parts of multiple data items. For example, SmartDoc identifies references to a group item as well as the individual elements within the group. This is true of both data item modification and/or uses.

SmartDoc identifies valid data item references, as opposed to matching simple character patterns in the variable name. These valid data item references identify redefined or renamed data items, or record group items (also called aliases). SmartDoc also identifies indirect data item references. Indirect references are datanames either directly or indirectly affected by the use of a data item or a modification.

SmartDoc identifies definitions, modifications, and uses of data items. A definition is the defined data item and data item aliases specified in the DATA DIVISION. Modification is an occurrence of a data item where its value is either set or altered. Lastly, SmartDoc uses a tested or a used value to identify how a data item is being used.

Input and Output Data Items

The Perform Range Usage and Interface report identifies data items used to communicate parameters to and from a perform range. These data items are referenced as either IN or OUT. IN data items are modified outside of the perform range, then used prior to modification within the perform range. OUT data items are modified within the perform range, then used outside the perform range prior to modification. USE and MOD are data items referenced inside the perform range.

Program Structure

SmartDoc gives you the structure of a COBOL program. The Perform Range Hierarchy Chart and Structure Chart show statements comprising the general program structure. The Perform Range Hierarchy Chart shows the program structure in an area as densely as possible and indicates the relative nesting level of performs and calls. The Structure Chart graphically shows the information reported on the Perform Range Hierarchy Chart.

3 Getting Started With SmartDoc

This chapter contains the information required to run and produce SmartDoc reports and contains these sections:

Introducing SmartDoc

This chapter is for the new SmartDoc user. Review Chapter 2, "Concepts," on page 7 to understand SmartDoc terminology before following the sample SmartDoc session presented in this chapter. Also, review Chapter 5, "Reports," on page 39 to familiarize yourself with the many SmartDoc reports.

Perform these tasks before beginning the sample SmartDoc session:

- Determine the AKR used.

  If you do not know the name of the AKR to use either contact your systems programmer or allocate an AKR as directed in Chapter 13, "AKR Utilities," on page 193.

- Select a program to document.

See these chapters for more information:

- See Chapter 11, "Analyze," on page 157 for information on analyze options, program analysis and generating SmartDoc reports.
- See Chapter 12, "SmartDoc Options," on page 189 for information on SmartDoc options.

Starting SmartDoc

The method you use to invoke SmartDoc depends on your system setup. See your system administrator if you need assistance activating SmartDoc.

To use either ISPF or CLIST to start SmartDoc, follow this step:

- Use either the ISPF selection or CLIST (as indicated by your system administrator) to start SmartDoc.

The SmartDoc Primary Screen displays after you activate the session (see Figure 8 on page 17).

To use the ESW Screen to start SmartDoc

1. Initiate the ESW screen by using either the ISPF selection or CLIST as indicated by your System Administrator (see Figure 7).

2. Select Document ▸ Program and press Enter.

   The product name displays on the screen as ASG-ESW - Program Documentation.

Figure 7 • ESW Primary Screen

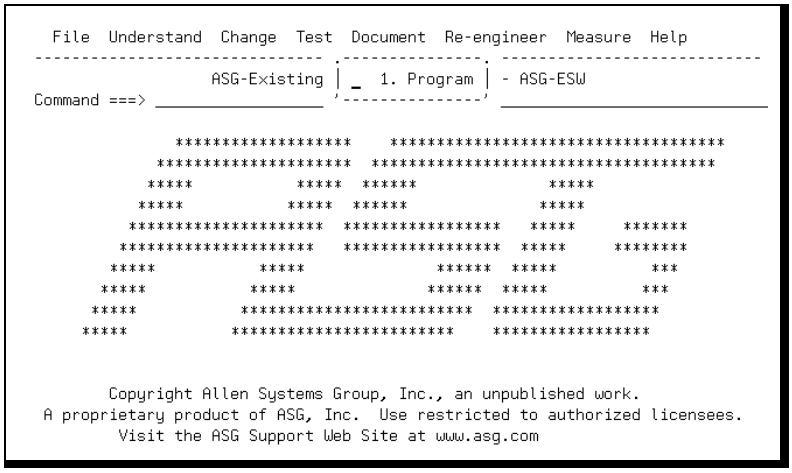
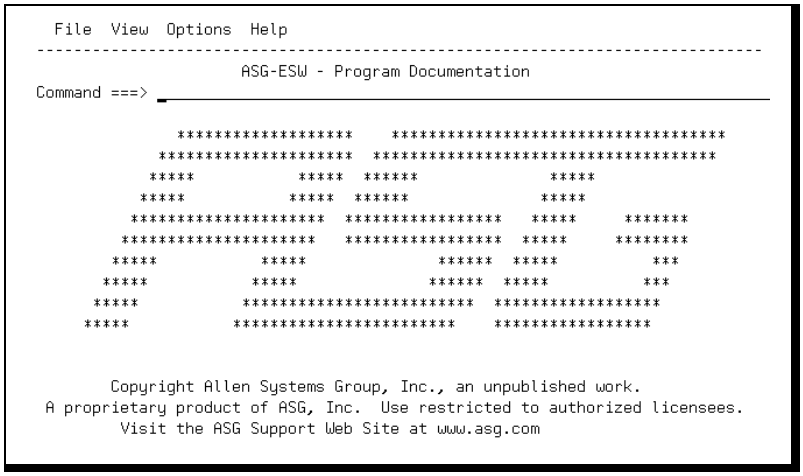


Figure 8 • Primary SmartDoc Screen



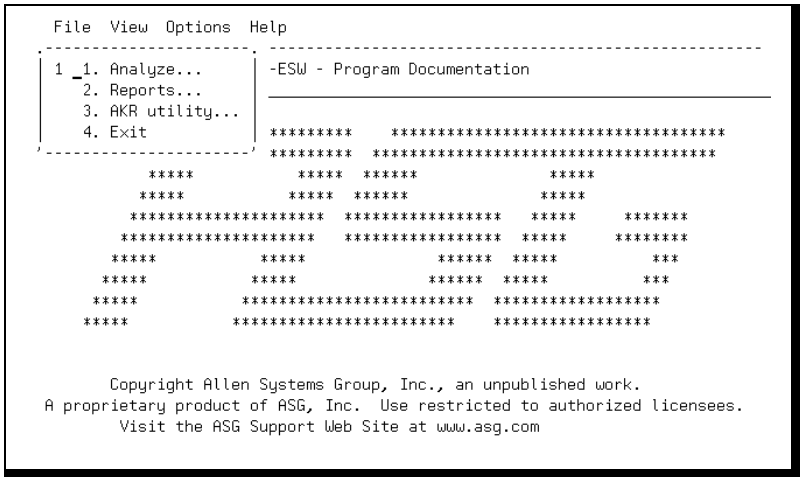
Analyzing the Program

The selected program must be analyzed by the Program Analyzer before you can produce SmartDoc reports.

To analyze a program

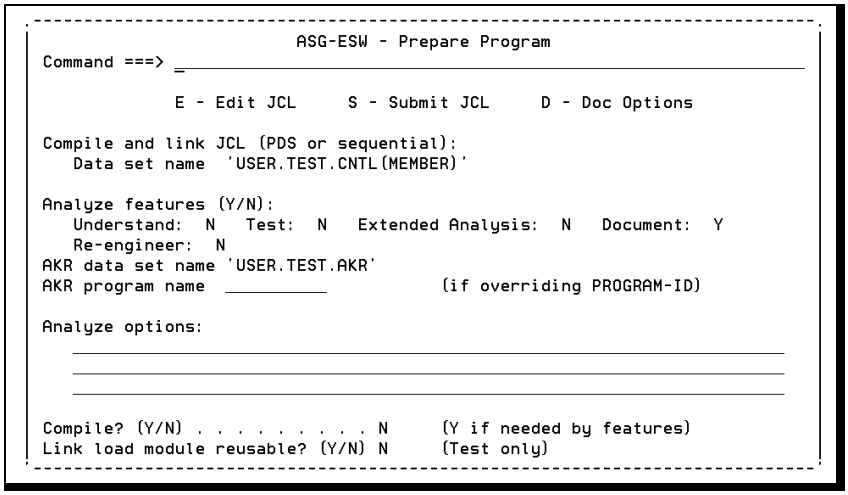
1 Select File ▸ Analyze from the SmartDoc Primary Screen and press Enter (see Figure 9). The Prepare Program pop-up displays (see Figure 10 on page 18).

Figure 9 • File Pull-down



2 Use the Prepare Program pop-up in Figure 10 to analyze the program in preparation for SmartDoc.

Figure 10 • Prepare Program Pop-up



Note:

The product names listed under the Analyze features (`Y/N`) field may differ from the names shown in the example in Figure 10 if you started SmartDoc from the ESW Primary Screen.

3 Type the dataset of the JCL used to compile and link the program in the Compile and link JCL field.

Our sample session uses ASG.VIACEN*xx*.CNTL(VIADDCVS).

4 If other ESW are products installed:

Select the desired analyze job(s) type(s) in the Analyze features field.

Or

If SmartDoc is the only product installed:

The default for SmartDoc is YES.

5 Type the AKR name in the AKR dataset name field and press Enter.

Our sample session uses USER.TEST.AKR.

6 Type `Y` in the Compile? field and press Enter.

7 Type `D` in the command area to select the File - SmartDoc Options pop-up to submit the reports simultaneously.

Generating SmartDoc Reports

Use the File - SmartDoc report pop-up to produce SmartDoc reports, to either submit or edit the JCL for the analyze job, and to specify the SmartDoc analysis.

To display the File - SmartDoc Options pop-up, complete this step:

- Select Analyze Submit ▸ option D and press Enter.

Perform an Extended SmartDoc analysis with a compile and reports from this pop-up with these tasks:

To submit an analyze job

1 Type a back slash `(/)` in the Analyze field and press Enter to submit an Analyze job.

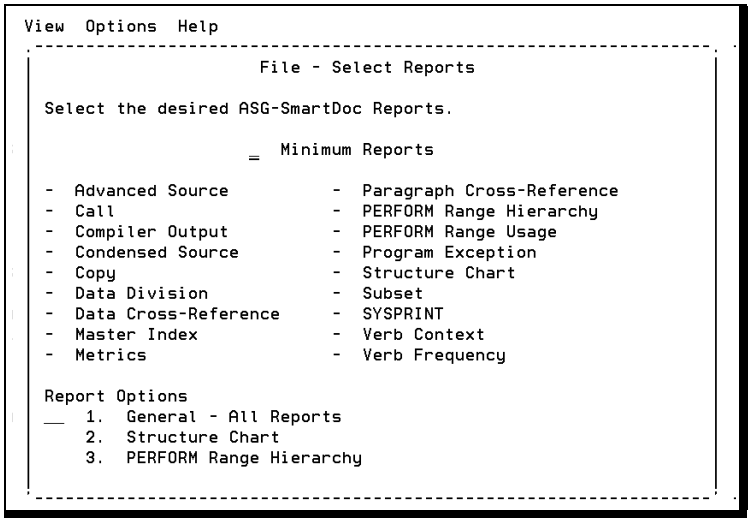
2 Type a back slash `(/)` in the Extended SmartDoc analysis field and press Enter to obtain extended analysis and reporting features.

3 Type `Y` in the Compile field and press Enter to compile the program at the time of analysis.

4 Type `1` in the Actions field and press Enter to display the File - Select reports pop-up illustrated in Figure 11 on page 20.

Use this pop-up to specify report options, and to select the reports generated following program analysis.

Figure 11 • File - Select Reports Pop-up



5 Type a non-blank character in the report field to select a report and press Enter.

To only produce the Advance Source Listing and the Enhanced Paragraph Cross-Reference reports, select the Minimum reports field. For this exercise, make sure that the Minimum reports field is not selected.

Note:

To specify format and content options for reports, select an action in the Report Options field. See "File - Select Reports Pop-up" on page 117 for additional information on these options.

You can submit other reports from this screen without performing another analyze after a program is analyzed and it resides in the AKR.

6 Press F3 to return to the File - SmartDoc Reports pop-up after selecting the reports.

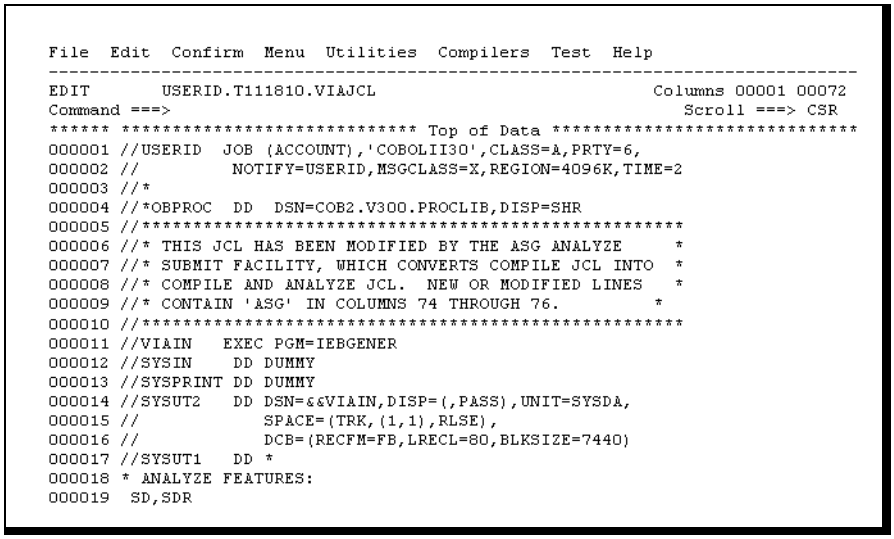
7 Type `3` in the Actions field to edit the JCL for the analyze job.

Editing the JCL

To initiate an editor session with the compile/analyze JCL for the displayed analyze job, follow this step:

- Select File - SmartDoc Report ▸ Edit JCL action and press Enter. The Editor screen (see Figure 12) displays.

Figure 12 • Editor Screen Showing Compile/Analyze JCL



To edit the JCL

1. Review the generated JCL to make sure it is correct for your environment.

2. Use the ISPF SUBMIT command to submit the job when you are satisfied with the JCL.

3. Press PF3/15 to return to SmartDoc.

   You have successfully submitted a SmartDoc analysis and report job.

4. Retrieve the output from your assigned printer when the job is complete. This is the same printer that SYSOUT output is normally printed on, unless you specified a different printer.

4 Techniques

This chapter discusses the purpose and usage techniques of SmartDoc reports and contains these sections:

Introduction

This chapter describes each report function and the way it is used in a typical programming task and then provides two solutions to the programming task. The first solution is the cumbersome programming method required without SmartDoc. The second solution is the easy SmartDoc technique.

See Chapter 5, "Reports," on page 39 for samples of all reports, and additional information about them.

Advanced Source Listing

Purpose of Report

The Advanced Source Listing augments the source with control and data flow information and gives you extended understanding of the program. This information improves your ability to navigate through the source, and reveals hidden detail. For example:

Action	Description
Programming Task	You want to add code to support a new employee type in the company's employee information system. You must understand how the program uses the EMPLOYEE-CODE. You must also ensure the existing code supports the new type and that, where needed, the new type is addressed as needed.
Without SmartDoc	Manually trace through the code to find and mark all references to EMPLOYEE-CODE. Now manually trace and mark all logic paths that may lead to where the EMPLOYEE-CODE is used. When you understand all the paths leading to usage points repeat the previous steps everywhere the EMPLOYEE-CODE value may determine the logic path used.

Action	Description
	Now you may have enough information to make changes, depending on program size. Make the changes, recompile the program, and test the changes.
The SmartDoc Technique	Use the Enhanced Data Cross-Reference report to find all references to EMPLOYEE-CODE. Use the Advanced Source Listing control and data flow information to understand how EMPLOYEE-CODE is used at each program reference location. Make any necessary code changes. Now, reanalyze, and run SmartDoc reports for the program. Review the program changes and use the Advanced Source Listing control and data flow information to ensure they work. Recompile and test the program. SmartDoc's static analysis capability enhances your ability to make changes that work correctly the first time.

Call Statement Report

Purpose of Report

The Call Statement report shows all calls to external programs, if program control returns to the current program, and where the call is located. Finally, the Call Statement report shows the parameters passed to the called program and how they are used. For example:

Action	Description
Programming Task	A program is being changed to include extra information. The revised program is going to use old data in new ways, and you need to find where the data items are being used and changed.
Without SmartDoc	Another manual search is needed. You need to find all called programs and search them, as well as searching the program being altered. You do not know whether the called program changes parameters that are later used. So, you also must search the code you are altering for every occurrence of parameters being passed to another program.
The SmartDoc Technique	Review the Call Statement report to find all calls to other programs. The Call Statement report lists both IN parameters and OUT parameters. From the information in the Call Statement report, you can turn to the Advanced Source Listing and follow the data flow information for the affected variables.

Condensed Source Listing

Purpose of Report

The Condensed Source Listing shows structurally significant code portions giving you a clear program overview. Listed divisions, sections, and paragraphs are indented. For example:

Action	Description
Programming Task	You are a new programmer assigned to maintain and enhance a large, old, and unstructured program. You need to learn the old program before making changes.
Without SmartDoc	Study and manually analyze the listings trying to gain an understanding of the program structure.
The SmartDoc Technique	Review the Condensed Source Listing showing the structurally significant program parts with noise verbs removed. The Condensed Source Listing enables you to easily understand even unstructured programs.

Copy Statement Report

Purpose of Report

The Copy Statement report shows all COBOL COPY statement occurrences, and also displays other source managers' COPY type directives (-INC, ++INCLUDE, etc.). For example:

Action	Description
Programming Task	The user is getting erroneous output. You are trying to locate its source, but cannot find the problem in the copied code.
Without SmartDoc	Trace through the code noting any COPY statements. Try to find the copied code source and examine it for any variable names duplicated in the source code.
The SmartDoc Technique	Review the Copy Statement report showing the DSNs for the actual code copied. Use that information to go directly to the code in question.

Data Division Report

Purpose of Report

The Data Division report gives you detailed information about all 01structures in a program DATA DIVISION. This is the information the Data Division report provides:

- COBOL level number
- dataname
- length
- starting and ending position
- format
- picture clause definition
- name of the COPY member where the data item is defined (if applicable)

For example:

Action	Description
Programming Task	You are assigned to export report data from the mainframe to the PC and reformat it. To reformat the data, you need to know the length, starting, and ending position of each data item appearing in the report.
Without SmartDoc	Use program and copy member listings to determine each data item length before creating the program to export and reformat the report data. Then, you must produce a report copy and manually count the columns to determine starting and ending position.
The SmartDoc Technique	Review the DATA Division report to identify the length, starting and ending position and use the information to develop a program to export and reformat the report data.

Enhanced Data Cross-Reference Report

Purpose of Report

The Enhanced Data Cross-Reference report provides a cross-reference to all program data item references (including alias definitions and references). With this information you can locate every data item reference easily, without having to manually scan the code. For example:

Action	Description
Programming Task	You wish to expand the a field length. You must understand the potential impact on other fields that use the data item before implementing this change.
Without SmartDoc	Manually review the code to find every data item reference, then list all data items that may be indirectly impacted by change. Manually review the code and build a list of each REDEFINES or RENAMES for the data item to expand. Now repeat the first step for each RENAMES or REDEFINES. Manually review the code to find the reference location for each data item, and build a indirectly affected data items list. Repeat this procedure for each listed data item until no more indirectly affected data items are found.
The SmartDoc Technique	Review the SmartDoc Enhanced Data Cross-Reference report. The report shows all data item definitions, alias definitions, usages, modifications and indirect references. Now, review the SmartDoc Advanced Source Listing and find the source lines indicated by the Enhanced Data Cross-Reference report. At each source line, determine if other data items are indirectly affected by the first data item size change, and build a list of indirectly affected data items. Repeat the previous process with the second list by using the Enhanced Data Cross-Reference report, until no more indirectly affected data items are found. The final list should contain the data item in question and any indirectly affected ones. This list contains the data items that require expansion to expand the first data item.

Program Metrics

Purpose of Report

Program Metrics show the program's complexity, program architecture, and software quality mathematically. Metrics for multiple program versions allow you to examine how complexity and quality changes over time. For example:

Action	Description
Programming Task	You have to improve the existing software and are not sure where to start.
Without SmartDoc	Edit the library and select a program at random, or perhaps, select the software with the most records.
The SmartDoc Technique	Run a SmartDoc analysis and start working with the program with the worst metrics. Program metrics tell you what software is most likely to fail over time.

Extended Example

This is an extended example of the use of Metrics for multiple program versions:

Action	Description
Programming Task	The company has twenty programs and is deciding which programs to rework.
Without SmartDoc	Start your search with the program that is causing the most problems or with the largest program.
The SmartDoc Technique	Use SmartDoc to get the Software Science Volume, Cyclomatic Complexity, and the Control Variable metrics value for all of the programs being considered.

Follow the procedure below to use the SmartDoc Technique to calculate the mean and standard deviation values. You can use these values to determine what programs would benefit most from restructuring.

To use SmartDoc to calculate metric mean and standard deviation values

1. Calculate the mean for each of the three metrics. The mean is the total metric values added together for all the programs divided by the number of programs.

2. Use the mean calculated in step 1 to calculate the standard deviation for each of the three metrics.

3 Calculate the difference in the number of standard deviations between the Cyclomatic Complexity and the Software Science Volume metric for each program.

4 Calculate the difference in the number of standard deviations between the Control Variable metric and the Software Science Volume metric for each program.

Use the calculations from step 3 and step 4 to identify programs whose number of standard deviations from the mean for the Cyclomatic Complexity metric and the Control Variable metric are more than one standard deviation greater than the number of standard deviations for the Software Science Volume metric. These programs are probably good candidates for restructuring.

Consider the programs listed in this table, for example:

	Software Science Volume		Cyclomatic Complexity			Control Variable		
Program	Metric Value	# Standard Deviations	Metric Value	# Standard Deviations	Diff.	Metric Value	Standard Deviations	Diff.
1	3000	0.03	16	1.12	1.09	25	1.14	1.11
2	2500	0.78	17	0.82	0.04	25	1.14	0.36
3	3500	0.84	21	0.39	-0.45	32	0.76	-0.08
4	2400	0.94	18	0.52	-0.42	26	0.86	-0.08
5	2800	0.29	15	1.42	1.13	24	1.41	1.12
6	2900	0.13	19	0.21	0.08	28	0.32	0.19
7	3300	0.52	20	0.09	-0.43	29	0.05	-0.47
8	3200	0.36	18	0.52	0.16	26	0.86	0.5
9	1900	1.75	27	2.21	0.46	24	1.41	-0.34
10	2600	0.62	19	0.21	-0.41	24	1.41	0.79
11	3700	1.17	19	0.21	-0.96	30	0.22	-0.95
12	2000	1.59	24	1.3	-0.29	30	0.22	-1.37
13	2200	1.26	18	0.52	-0.74	33	1.03	-0.23
14	3700	1.17	17	0.82	-0.35	34	1.3	0.13
15	4100	1.81	23	1.00	-0.81	32	0.76	-1.05
16	3900	1.49	23	1.00	-0.49	32	0.76	-0.73

	Software Science Volume		Cyclomatic Complexity			Control Variable		
Program	Metric Value	# Standard Deviations	Metric Value	# Standard Deviations	Diff.	Metric Value	Standard Deviations	Diff.
17	3300	0.52	15	1.42	0.9	32	0.76	0.24
18	3400	0.68	26	1.91	1.23	37	2.11	1.43
19	2700	0.45	20	0.09	-0.36	30	0.22	-0.23
20	2500	0.78	19	0.21	-0.57	31	0.49	-2.09
	Metric Mean	Standard Deviation for Metric Value	Metric Mean	Standard Deviation for Metric Value		Metric Mean	Standard Deviation for Metric Value	
	2980	617.74	19.7	03.3		29.2	3.7	

These programs should be reworked, for example:

- Program 18 would benefit most, and needs restructuring to improve ease of maintenance. In the program metrics table the differences between the Software Science Volume metric and the other two metrics for Program 18 are about 1.23 and 1.43.
- Program 5 might be examined next. In the program metrics table the differences between the Software Science Volume metric and the other two metrics are about 1.13 and 1.12.
- Finally, examine program 1. In the program metrics table, program 1 has a little more than one standard deviation greater Cyclomatic Complexity and Control Variable metrics than Software Science Volume metric. The rest seem proportionally complex for their size.

Paragraph Cross-Reference

Purpose of Report

The Paragraph Cross-Reference report lists every entry and exit for all paragraphs and sections contained within the program. The listing also displays paragraph entries and exits by line number. Use the report information to discern all accesses to and from a paragraph or section. For example:

Action	Description
Programming Task	You altered the calculating method for 401K contributions and a paragraph of original code is no longer needed. But, you must insure the paragraph is not used by other code before deleting it from the source.
Without SmartDoc	Search the source code for paragraph references. The paragraph can be removed after the entire source is checked.
The SmartDoc Technique	Review the paragraph in the Paragraph Cross-Reference report to see all locations where it is invoked. This eliminates costly mistakes caused by a perform or a GO TO that was overlooked when the program was changed.

Perform Range Hierarchy Chart

Purpose of Report

The Perform Range Hierarchy Chart shows the structural interdependencies of the program. This diagram shows you the program perform range execution relationships as an indented list. For example:

Action	Description
Programming Task	You are assigned to maintain and enhance a large program. You need to learn the program before making changes.
Without SmartDoc	Study and manually analyze the listings to understand the program structure.
The SmartDoc Technique	Review the Perform Range Hierarchy Chart to see the program structure displayed in a dense, semi-graphical representation. The Perform Range Hierarchy Chart lists the control flow transfers from one perform range to another, enabling you to easily understand the program structure.

Perform Range Usage and Interface Report

Purpose of Report

The Perform Range Usage and Interface report lists the program perform ranges along with the paragraphs and sections that perform them. The data items involved with the perform range are also given, and are listed as INPUTs, OUTPUTs, USES or MODS.

This report displays all data item usage within a perform range. Use this information to make separately callable modules when re-engineering code. The Perform Range Usage and Interface report also tells you where perform ranges are invoked, and how perform range data changes can impact other program areas. For example:

Action	Description
Programming Task	You are adding a new feature to existing code, and are told to remove the file-reading portion and make a separate module called from both its original location as well as from new code. As situations change, logic must change to match it. A conditional statement needs changed, and the impact on later uses of the affected variable needs examining.
Without SmartDoc	Evaluate the source code paragraph that does the reading to locate all of the perform range input or the output data items so that they can be passed to and from the new module. Also, all data items that are either used or modified need to become part of the new module's Data Division. All of the old paragraph's invocation locations need altered to the appropriate call. Find all outgoing variables affected by the conditional. Walk through the code and try to determine what perform range uses the variable next and the affect of the alterations. Trace out from each of those perform ranges and see what affect there is.
The SmartDoc Technique	A reference to the reading perform range, in this case a paragraph only, in the Perform Range Usage and Interface report immediately shows all the data items involved, eliminating the laborious task of finding them yourself, and removing the potential for errors. Also, all the invocation locations are displayed in the PERFORMED BY portion of the report. Use the Perform Range Usage and Interface report to immediately see what variables enter and exit the perform range. See "Advanced Source Listing" on page 24 for an example of how the variables are used. Then follow the data flow information for the affected variables to see what change is needed.

Program Exception Report

Purpose of Report

The Program Exception report provides program processing flow information, including:

- Unstructured exits from a perform range
- Live exits
- Recursion
- Uninitialized uses
- Modifications without uses
- Dead code
- Dead data

Information from the Program Exception report reveals important code features that may cause undesired results and/or abends. Having these possible errors centrally located shortens the debugging process and identifies trouble spots for further investigation. For example:

Action	Description
Programming Task	You made changes to an existing program and the program is constantly timing out, indicating an infinite loop. Many variables were added or had their uses altered during code changes. Now test the changes.
Without SmartDoc	Finding the recursive cycle involves decrypting the dump from the abend, then examining the source until the loop is found. Execute the program in as many ways as possible, trying to test for every possible case.
The SmartDoc Technique	Run the Program Exception report before executing the program. Note that a recursive cycle is indicated, including the actual statements along with the line numbers of each. Fix the infinite loop before it becomes a problem. Review the Program Exception report to see where likely errors occur before attempting program execution. The Program Exception report lists potential problems such as uninitialized data and live exits. These can then be fixed to assure greater program reliability.

Structure Chart

Purpose of Report

The Structure Chart displays a program structure graphically. This diagram gives a clear, concise program overview. For example:

Action	Description
Programming Task	You want to add a new feature to an old unstructured program and you need to understand the existing structure.
Without SmartDoc	Study and manually analyze the listings, and try to understand the program structure.
The SmartDoc Technique	Review the Structure Chart that displays the program structure in an easy-to-understand format. The control flow becomes immediately and readily apparent. Make a Tile Mode version for future reference.

Subset Report

Purpose of Report

The Subset report identifies logical groupings of similar COBOL verbs. The subsets help you to identify similar statement groups within the program. For example:

Action	Description
Programming Task	You must check the number and relative positioning of program comments to ensure code quality assurance standards. You decide to consolidate all the I/O into one area while updating a program.
Without SmartDoc	You have to search the source code line-by-line to locate and verify all comments, a time consuming and unproductive task. Use FIND and RFIND commands on all possible I/O verbs. Note where they are and what it takes to move them.
The SmartDoc Technique	Review the Subset report comments section to immediately identify all code comments (and their line numbers). Review the Subset report where the I/O statements are listed as a logical grouping by line number. These lines on the Advanced Source Listing show what logic leads up to them and what you need to do to overhaul them.

Verb Summary Report

Purpose of Report

The Verb Summary report cross-references COBOL verbs to where they are used within the program. This report includes the Verb Frequency Table, showing the percentage of use in proportion to all other verbs. This information provides a complete verb usage listing. For example:

Action	Description
Programming Task	A company has verb usage standard that requires a single paragraph be used to do all reads from a file, and you must ensure that a new program follows the standard. Company standards require that no GO TOs be used, and you are quickly reviewing the code.
Without SmartDoc	You search the source program line-by-line for file read references Search the code for all GO TOs, then write down where they are and repair them.
The SmartDoc Technique	Review the Verb Summary report revealing all program READ statements. The Advanced Source Listing is included so you can check file references. Review the Verb Summary report for any GO TOs.

5 Reports

This chapter describes and illustrates each SmartDoc report and contains these sections:

Introduction

SmartDoc program documentation is used to understand internal program structures and includes:

- Source code
- Source code commentary
- External program specifications
- Design documents
- Program Structure Charts
- Cross-reference maps

SmartDoc provides you with accurate, timely, and automatic program documentation. Comprehensive information about a COBOL program is presented in reports and listings you can use to effectively meet the challenge of maintaining and enhancing existing systems.

SmartDoc presents program information in a report collection that documents each program. For example, the program Structure Chart displays program logic graphically, and the software metrics enable you to evaluate and rank each program within a system.

This chapter describes and illustrates each SmartDoc report. Some reports are shown in the multiple formats produced by the different types of analysis jobs. Each analysis type provides SmartDoc with a slightly different feature set.

These are the identifiers used to indicate the type of analysis performed:

Identifier	Description
DA	An Extended SmartDoc analysis and compile were performed. This analysis provides data flow information. All features are available.
DX	An Extended SmartDoc analysis was performed (no compile). This analysis provides data flow information. DMAP information is not merged into the Data Division of the Advanced Source Listing, and statement offsets are not available in the Procedure Division of the Advanced Source Listing.
DC	A short SmartDoc analysis and compile were performed. This analysis provides all reporting capabilities excluding data flow information. Data flow information in the Advanced Source Listing is replaced with data cross-reference information. Uninitialized uses and modifications that have not been used are not noted in the Advanced Source Listing, the Enhanced Data Cross-Reference report, or the Program Exception report.
DS	A short SmartDoc analysis was performed (no compile). This analysis provides all reporting capabilities excluding data flow information and compiler information. The DS analysis provides the same information as the DC analysis, but does not provide compiler information. Specifically, DMAP information is not merged into the Data Division of the Advanced Source Listing, and statement offsets are not available in the Procedure Division of the Advanced Source Listing.

Report Headings

All SmartDoc reports contain a standard heading, as shown in Figure 13.

Figure 13 • Report Heading Example

```
ASG-SMARTDOC-OS Rx.x LVL000 (A)          ADVANCED SOURCE LISTING (B)          DDMMMYYYY HH:MM:SS PAGE 9
                                           PROGRAM: VIADDDM0 (C)                    (D)          (E)
```

Report Field Descriptions

These are the report fields and their descriptions:

Field	Description
(A)	SmartDoc release number and level
(B)	Report title
(C)	Program name the report was generated for
(D)	Date and time the report was generated
(E)	Report page number

Note:

The Program Summary report does not include the program name in the report heading. The Table of Contents and Program Summary reports do not contain page numbering.

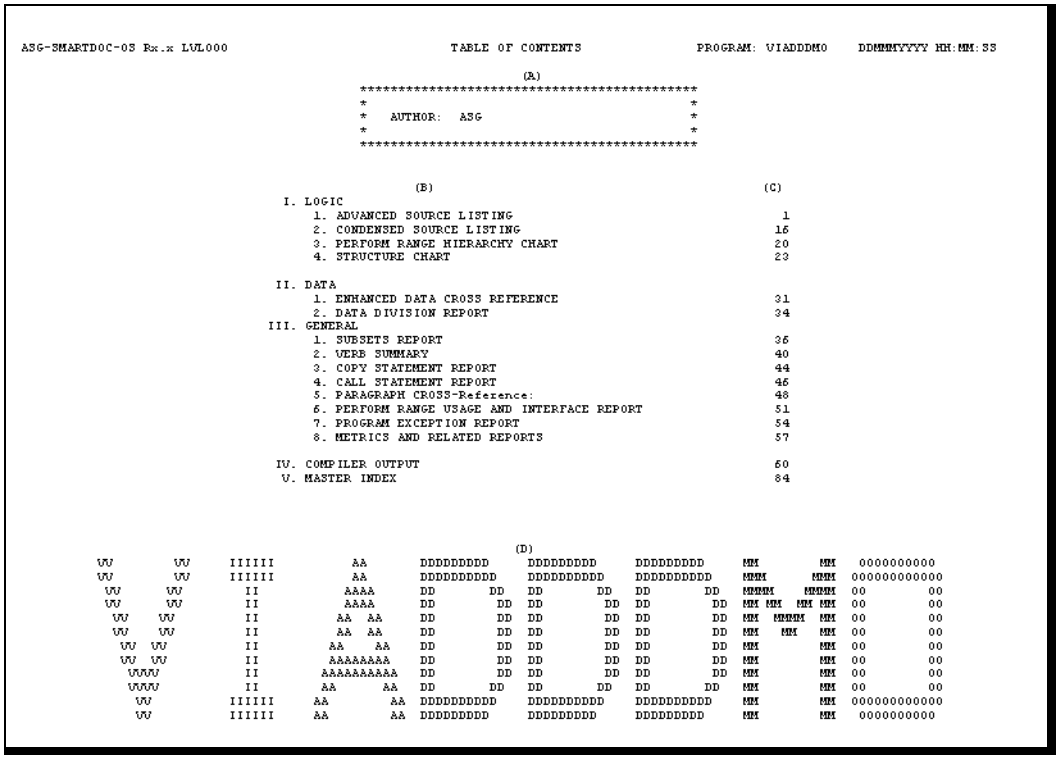
Table of Contents Report

Think of SmartDoc reports as chapters in a book describing a program. SmartDoc generates a Table of Contents (see Figure 14 on page 43) for this book and divides it into these categories:

Category		Contents
I	Logic	1. Advanced Source Listing 2. Condensed Source Listing 3. Perform Range Hierarchy Chart 4. Structure Chart
II	Data	1. Enhanced Data Cross-Reference report 2. Data Division report

Category		Contents
III	General	1. Subsets report 2. Verb Summary report 3. COPY Statement report 4. CALL Statement report 5. Paragraph Cross-Reference 6. Perform Range Usage and Interface report 7. Program Exception report 8. Metrics and related reports
IV	Compiler/Optimizer Output	
V	Master Index	

Figure 14 • Table of Contents Report



```
ASG-SMARTDOC-03 Rx.x LVL000                    TABLE OF CONTENTS              PROGRAM: VIADDDM0    DDMMMYYYY HH:MM:SS

                                                    (A)
                                  *********************************************
                                  *                                           *
                                  *   AUTHOR:  ASG                            *
                                  *                                           *
                                  *********************************************


                                                (B)                                                (C)
                                 I. LOGIC
                                      1. ADVANCED SOURCE LISTING                                    1
                                      2. CONDENSED SOURCE LISTING                                  16
                                      3. PERFORM RANGE HIERARCHY CHART                             20
                                      4. STRUCTURE CHART                                           23

                                 II. DATA
                                      1. ENHANCED DATA CROSS REFERENCE                             31
                                      2. DATA DIVISION REPORT                                      34
                                III. GENERAL
                                      1. SUBSETS REPORT                                            36
                                      2. VERB SUMMARY                                              40
                                      3. COPY STATEMENT REPORT                                     44
                                      4. CALL STATEMENT REPORT                                     46
                                      5. PARAGRAPH CROSS-Reference:                                48
                                      6. PERFORM RANGE USAGE AND INTERFACE REPORT                  51
                                      7. PROGRAM EXCEPTION REPORT                                  54
                                      8. METRICS AND RELATED REPORTS                               57

                                 IV. COMPILER OUTPUT                                               60
                                  V. MASTER INDEX                                                  84


                                                    (D)
          VV        VV     IIIIIII          AA       DDDDDDDDD     DDDDDDDDD     DDDDDDDDD     MM         MM   0000000000
          VV        VV     IIIIIII         AA       DDDDDDDDDD    DDDDDDDDDD    DDDDDDDDDD    MMM       MMM  0000000000000
           VV      VV        II           AAAA      DD        DD  DD        DD  DD        DD  MMMM     MMMM  00          00
           VV      VV        II           AAAA      DD         DD DD         DD DD         DD MM MM   MM MM  00          00
            VV    VV         II          AA  AA     DD         DD DD         DD DD         DD MM  MMMMM  MM  00          00
            VV    VV         II          AA  AA     DD         DD DD         DD DD         DD MM   MMM   MM  00          00
             VV  VV          II         AA    AA    DD         DD DD         DD DD         DD MM         MM  00          00
             VV  VV          II         AAAAAAAA    DD         DD DD         DD DD         DD MM         MM  00          00
              VVVV           II        AAAAAAAAAA   DD         DD DD         DD DD         DD MM         MM  00          00
              VVVV           II       AA      AA    DD        DD  DD        DD  DD        DD  MM         MM  00          00
               VV          IIIIIII   AA        AA  DDDDDDDDDD    DDDDDDDDDD    DDDDDDDDDD    MM         MM  0000000000000
               VV          IIIIIII   AA        AA  DDDDDDDDD     DDDDDDDDD     DDDDDDDDD     MM         MM   0000000000
```

Report Field Descriptions

These are the Table of Contents report field descriptions:

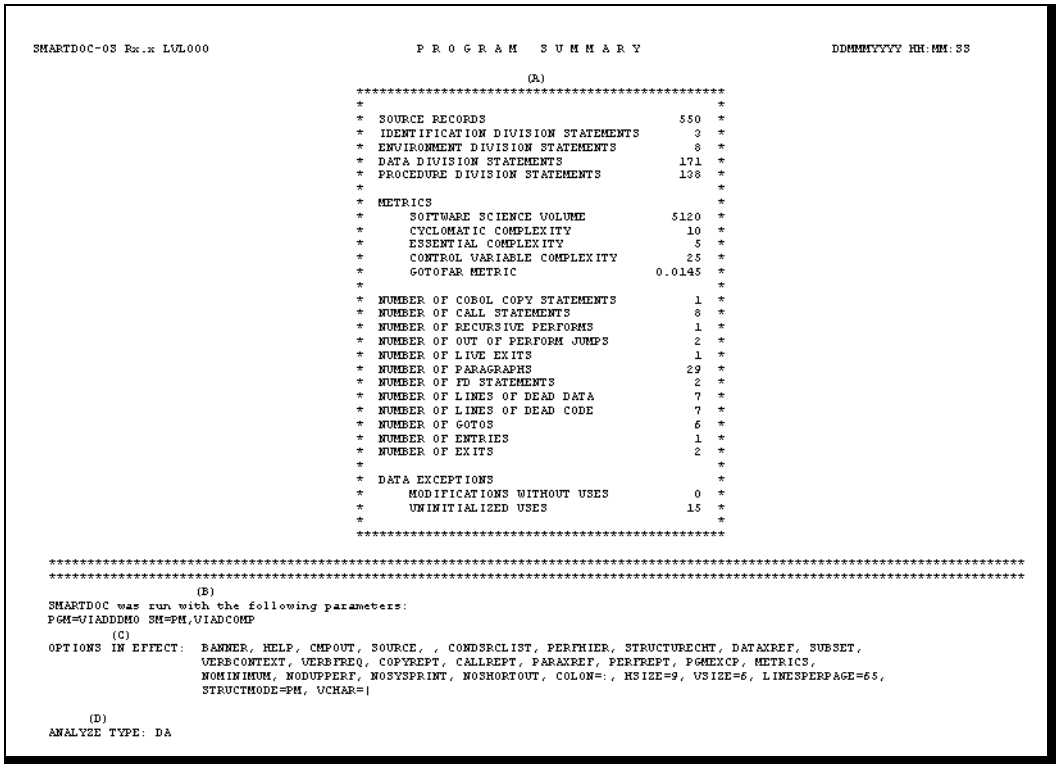
Field	Description
(A)	Program author from the Identification Division
(B)	List of reports generated for this execution of SmartDoc
(C)	Page number where the particular report begins
(D)	Program name

Program Summary

The page following the Table of Contents shows program summary information (see Figure 15 on page 45). This is the information in the Program Summary includes:

- Number of lines in the source program
- Number of statements in the IDENTIFICATION DIVISION
- Number of statements in the ENVIRONMENT DIVISION
- Number of statements in the DATA DIVISION
- Number of statements in the PROCEDURE DIVISION
- Metrics values for the program
- Numbers of various types of COBOL statements
- Number of data item modifications without subsequent uses
- Number of data item uses without initialization
- Parameters specified for this execution of SmartDoc
- Options in effect for this execution of SmartDoc
- Type of analysis performed for this program

Figure 15 • Program Summary



```
SMARTDOC-OS Rx.x LVL000                    P R O G R A M   S U M M A R Y                    DDMMMYYYY HH:MM:SS

                                                (A)
                                  ************************************************
                                  *                                              *
                                  *  SOURCE RECORDS                      550     *
                                  *  IDENTIFICATION DIVISION STATEMENTS    3     *
                                  *  ENVIRONMENT DIVISION STATEMENTS       8     *
                                  *  DATA DIVISION STATEMENTS            171     *
                                  *  PROCEDURE DIVISION STATEMENTS       138     *
                                  *                                              *
                                  *  METRICS                                     *
                                  *       SOFTWARE SCIENCE VOLUME       5120     *
                                  *       CYCLOMATIC COMPLEXITY           10     *
                                  *       ESSENTIAL COMPLEXITY             5     *
                                  *       CONTROL VARIABLE COMPLEXITY     25     *
                                  *       GOTOFAR METRIC                0.0145   *
                                  *                                              *
                                  *  NUMBER OF COBOL COPY STATEMENTS      1     *
                                  *  NUMBER OF CALL STATEMENTS            8     *
                                  *  NUMBER OF RECURSIVE PERFORMS         1     *
                                  *  NUMBER OF OUT OF PERFORM JUMPS       2     *
                                  *  NUMBER OF LIVE EXITS                 1     *
                                  *  NUMBER OF PARAGRAPHS                29     *
                                  *  NUMBER OF FD STATEMENTS              2     *
                                  *  NUMBER OF LINES OF DEAD DATA         7     *
                                  *  NUMBER OF LINES OF DEAD CODE         7     *
                                  *  NUMBER OF GOTOS                      6     *
                                  *  NUMBER OF ENTRIES                    1     *
                                  *  NUMBER OF EXITS                      2     *
                                  *                                              *
                                  *  DATA EXCEPTIONS                             *
                                  *       MODIFICATIONS WITHOUT USES      0     *
                                  *       UNINITIALIZED USES             15     *
                                  *                                              *
                                  ************************************************

*****************************************************************************************************************************************************
*****************************************************************************************************************************************************
                        (B)
SMARTDOC was run with the following parameters:
PGM=VIADDDM0 SM=PM,VIADCOMP
          (C)
OPTIONS IN EFFECT:  BANNER, HELP, CMPOUT, SOURCE, , CONDSRCLIST, PERFHIER, STRUCTURECHT, DATAXREF, SUBSET,
                    VERBCONTEXT, VERBFREQ, COPYREPT, CALLREPT, PARAXREF, PERFREPT, PGMEXCP, METRICS,
                    NOMINIMUM, NODUPPERF, NOSYSPRINT, NOSHORTOUT, COLON=:, HSIZE=9, VSIZE=6, LINESPERPAGE=65,
                    STRUCTMODE=PM, VCHAR=|

       (D)
ANALYZE TYPE: DA
```

Report Field Descriptions

These are the Program Summary report field descriptions:

Field	Description
(A)	Summary information, including the number of statements in each division, metrics values for the program, numbers of various COBOL statements in the program, and the number of data exceptions **Note:** Number of COBOL copy statements is the number of copy statements resolved by the COBOL compiler only.
(B)	List of the VIAIN DD statement parameters in effect for this execution of SmartDoc
(C)	Options in effect for this execution of SmartDoc
(D)	Type of analysis performed for this program

Advanced Source Listing

The Advanced Source Listing provides you with information for tracing program logic, and for understanding the program data flow that enhances and replaces the COBOL compiler listing. The source code prints in the same format as the original source file with the exception of the SKIP and EJECT compiler directives (these function in the same manner as in the compiler listing). Two formats are produced: one for the DATA DIVISION and a second for the PROCEDURE DIVISION.

For COBOL II Release 3 and later programs containing internal subprograms, SmartDoc produces a separate Advanced Source Listing for each subprogram.

Advanced Source Listing for the DATA DIVISION

SmartDoc produces two types of DATA DIVISION reports depending on the type of analysis performed.

Analysis with Compile

The source code prints when the Analysis job runs with a compile, and displays the extracted compiler DMAP information for each data item merged into the listing on the right side of the report (either a DC or a DA analysis). This is the DMAP information:

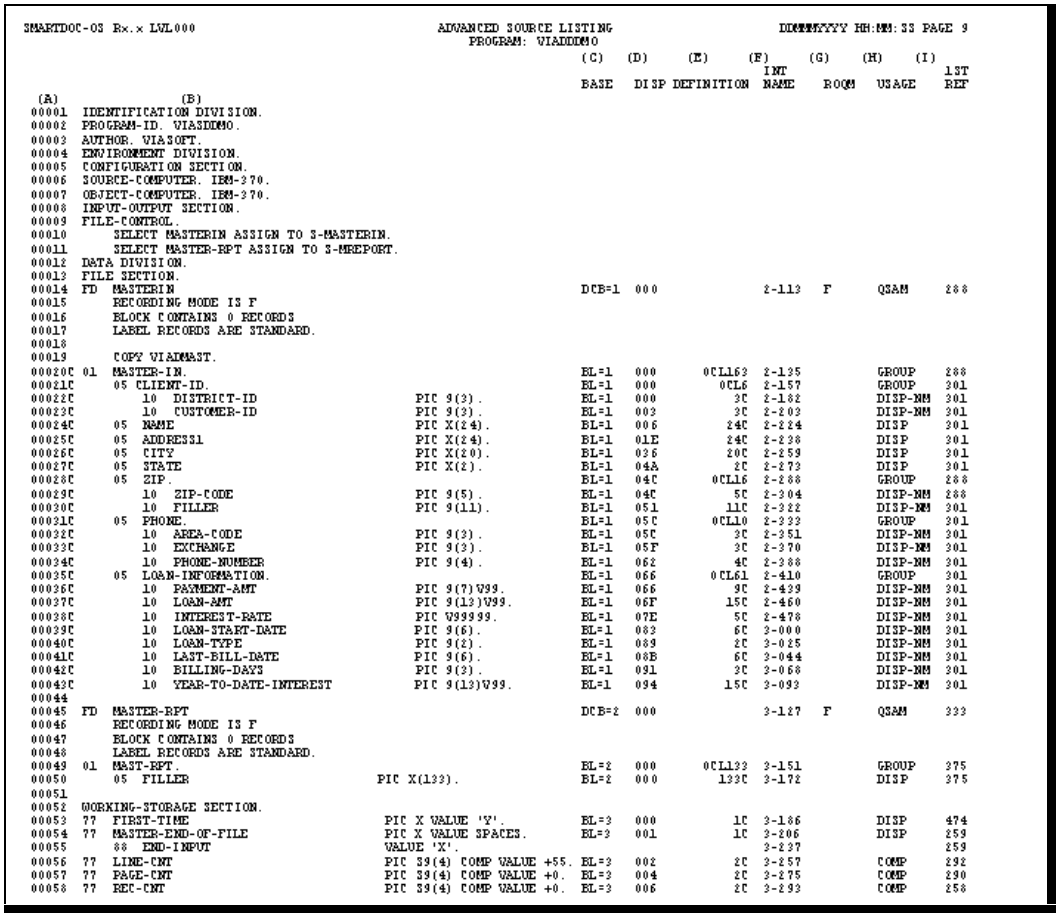
Field	Description
BASE	Base locator cell for a COBOL data item used internally to reference the data item
DISPLACEMENT	Relative offset of the data item within the base locator cell
DEFINITION	Assembly language internal definition of a data item
INTERNAL NAME (applies only to VS-COBOL)	Name assigned to a data item by the COBOL compiler
COMPILER FLAGS	Compiler flags, generated by the COBOL compiler, identify types of data items. Listed below are the flags for OS/VS COBOL and for COBOL II. OS/VS COBOL: R = dataname redefines another dataname. O = an OCCURS clause was specified for the dataname. Q = dataname is either the object or contains the object of the DEPENDING ON option in the OCCURS clause.

Field	Description
	M = the format of the records in the file is: F Fixed V Variable U Undefined S Spanned I = an input CD in a teleprocessing application. O = an output CD in a teleprocessing application. COBOL II: D = dataname is the object of an OCCURS DEPENDING clause. E = dataname is EXTERNAL. F = file is fixed length. FB = file is fixed length and blocked. G = a GLOBAL. O = an OCCURS clause. OG = the group has its own length definition. R = dataname redefines another dataname. S = a spanned file. U = an undefined format file. V = a variable length file. VB = a variable length blocked file.
USAGE	Indicates the access method for an FD, or a description of how a data item is used

The first reference line number in the PROCEDURE DIVISION is indicated, unless the data item is not used. If the data item is not used, the first reference line number is indicated as DEADDATA. The first reference refers to the data item in source sequence.

Figure 16 on page 48 shows an Advanced Source Listing for the DATA DIVISION, produced by a DC or a DA type of analysis.

Figure 16 • Advanced Source Listing (DC or DA Analysis) DATA DIVISON



```
SMARTDOC-OS Rx.x LVL000                    ADVANCED SOURCE LISTING                    DDMMMYYYY HH:MM:SS PAGE 9
                                              PROGRAM: VIADDM90
                                                                  (C)   (D)     (E)     (F)     (G)    (H)    (I)
                                                                                          INT                    1ST
                                                                  BASE  DISP DEFINITION NAME    ROOM USAGE    REF
 (A)              (B)
00001  IDENTIFICATION DIVISION.
00002  PROGRAM-ID. VIASDM90.
00003  AUTHOR. VIASOFT.
00004  ENVIRONMENT DIVISION.
00005  CONFIGURATION SECTION.
00006  SOURCE-COMPUTER. IBM-370.
00007  OBJECT-COMPUTER. IBM-370.
00008  INPUT-OUTPUT SECTION.
00009  FILE-CONTROL.
00010      SELECT MASTERIN ASSIGN TO S-MASTERIN.
00011      SELECT MASTER-RPT ASSIGN TO S-MREPORT.
00012  DATA DIVISION.
00013  FILE SECTION.
00014  FD  MASTERIN                                             DCB=1 000            2-113  F      QSAM     288
00015      RECORDING MODE IS F
00016      BLOCK CONTAINS 0 RECORDS
00017      LABEL RECORDS ARE STANDARD.
00018
00019      COPY VIADMAST.
00020C 01  MASTER-IN.                                          BL=1  000     0CL169 2-125          GROUP    288
00021C     05 CLIENT-ID.                                      BL=1  000      0CL6  2-157          GROUP    301
00022C         10  DISTRICT-ID                 PIC 9(3).      BL=1  000       3C   2-182          DISP-NM  301
00023C         10  CUSTOMER-ID                 PIC 9(3).      BL=1  003       3C   2-203          DISP-NM  301
00024C     05  NAME                            PIC X(24).     BL=1  006      24C   2-224          DISP     301
00025C     05  ADDRESS1                        PIC X(24).     BL=1  01E      24C   2-238          DISP     301
00026C     05  CITY                            PIC X(20).     BL=1  036      20C   2-259          DISP     301
00027C     05  STATE                           PIC X(2).      BL=1  04A       2C   2-273          DISP     301
00028C     05  ZIP.                                          BL=1  04C     0CL16  2-288          GROUP    288
00029C         10  ZIP-CODE                    PIC 9(5).      BL=1  04C       5C   2-304          DISP-NM  288
00030C         10  FILLER                      PIC 9(11).     BL=1  051      11C   2-322          DISP-NM  301
00031C     05  PHONE.                                        BL=1  05C     0CL10  2-333          GROUP    301
00032C         10  AREA-CODE                   PIC 9(3).      BL=1  05C       3C   2-351          DISP-NM  301
00033C         10  EXCHANGE                    PIC 9(3).      BL=1  05F       3C   2-370          DISP-NM  301
00034C         10  PHONE-NUMBER                PIC 9(4).      BL=1  062       4C   2-388          DISP-NM  301
00035C     05  LOAN-INFORMATION.                              BL=1  066     0CL61  2-410          GROUP    301
00036C         10  PAYMENT-AMT                 PIC 9(7)V99.   BL=1  066       9C   2-429          DISP-NM  301
00037C         10  LOAN-AMT                    PIC 9(13)V99.  BL=1  06F      15C   2-460          DISP-NM  301
00038C         10  INTEREST-RATE               PIC V99999.    BL=1  07E       5C   2-478          DISP-NM  301
00039C         10  LOAN-START-DATE             PIC 9(6).      BL=1  083       6C   3-000          DISP-NM  301
00040C         10  LOAN-TYPE                   PIC 9(2).      BL=1  089       2C   3-025          DISP-NM  301
00041C         10  LAST-BILL-DATE              PIC 9(6).      BL=1  08B       6C   3-044          DISP-NM  301
00042C         10  BILLING-DAYS                PIC 9(3).      BL=1  091       3C   3-068          DISP-NM  301
00043C         10  YEAR-TO-DATE-INTEREST       PIC 9(13)V99.  BL=1  094      15C   3-093          DISP-NM  301
00044
00045  FD  MASTER-RPT                                           DCB=2 000            3-127  F      QSAM     333
00046      RECORDING MODE IS F
00047      BLOCK CONTAINS 0 RECORDS
00048      LABEL RECORDS ARE STANDARD.
00049  01  MAST-RPT.                                          BL=2  000     0CL133 3-151          GROUP    375
00050      05  FILLER                      PIC X(133).        BL=2  000      133C  3-172          DISP     375
00051
00052  WORKING-STORAGE SECTION.
00053  77  FIRST-TIME                       PIC X VALUE 'Y'.  BL=3  000       1C   3-186          DISP     474
00054  77  MASTER-END-OF-FILE               PIC X VALUE SPACES. BL=3 001      1C   3-206          DISP     259
00055      88  END-INPUT                    VALUE 'X'.                             3-237                   259
00056  77  LINE-CNT                         PIC 39(4) COMP VALUE +55. BL=3 002  2C   3-257          COMP     292
00057  77  PAGE-CNT                         PIC 39(4) COMP VALUE +0.  BL=3 004  2C   3-275          COMP     290
00058  77  REC-CNT                          PIC 39(4) COMP VALUE +0.  BL=3 006  2C   3-293          COMP     258
```

Report Field Descriptions Analysis with Compile

These are the Analysis with Compile report field descriptions:

Field	Description
(A)	Line numbers generated by the analyze job. All line number references throughout the SmartDoc reports are based on these numbers.
(B)	Source lines
(C)	Base locator cell used internally to reference the data item
(D)	The displacement or the relative offset of the data item within the base locator cell
(E)	Assembly language internal definition of the data item
(F)	Internal name assigned to the data item by the COBOL compiler (applies only to VS-COBOL)
(G)	Compiler generated flags that identify the type of data item listed
(H)	Indicates the usage of the data item or of the file as defined in the COBOL program
(I)	The first reference of the data item in source sequence

Analysis without Compile

In this analysis, the first reference line number in the PROCEDURE DIVISION is indicated when the Analysis job runs without a compile (a DS or a DX analysis), unless the data item is not used. If the data item is not used the first reference line number is indicated as DEADDATA. The first reference refers to the source sequence data item. Figure 17 shows the Advanced Source Listing produced by a DS or a DX type of analysis.

Figure 17 • Advanced Source Listing (DS or DX Analysis) - DATA DIVISION

```
ASG-SMARTDOC-OS Rx.x LVL000                    ADVANCED SOURCE LISTING                    DDMMMYYYY HH:MM:SS PAGE 9
                                                  PROGRAM: VIADDDM0
                                                                          (C)
                                                                          1ST
                                                                          REF
 (A)              (B)
00001  IDENTIFICATION DIVISION.
00002  PROGRAM-ID. VIADDDM0.
00003  AUTHOR. ASG.
00004  ENVIRONMENT DIVISION.
00005  CONFIGURATION SECTION.
00006  SOURCE-COMPUTER. IBM-370.
00007  OBJECT-COMPUTER. IBM-370.
00008  INPUT-OUTPUT SECTION.
00009  FILE-CONTROL.
00010      SELECT MASTERIN ASSIGN TO S-MASTERIN.
00011      SELECT MASTER-RPT ASSIGN TO S-MREPORT.
00012  DATA DIVISION.
00013  FILE SECTION.
00014  FD  MASTERIN                                                          (288)
00015      RECORDING MODE IS F
00016      BLOCK CONTAINS 0 RECORDS
00017      LABEL RECORDS ARE STANDARD.
00018
00019      COPY VIADMAST.
00020  01  MASTER-IN.                                                       (288)
00021      05 CLIENT-ID.                                                   (301)
00022          10  DISTRICT-ID                  PIC 9(3).                 (301)
00023          10  CUSTOMER-ID                  PIC 9(3).                 (301)
00024      05  NAME                             PIC X(24).                (301)
00025      05  ADDRESS1                         PIC X(24).                (301)
00026      05  CITY                             PIC X(20).                (301)
00027      05  STATE                            PIC X(2).                 (301)
00028      05  ZIP.                                                       (288)
00029          10  ZIP-CODE                     PIC 9(5).                 (288)
00030          10  FILLER                       PIC 9(11).                (301)
00031      05  PHONE.                                                     (301)
00032          10  AREA-CODE                    PIC 9(3).                 (301)
00033          10  EXCHANGE                     PIC 9(3).                 (301)
00034          10  PHONE-NUMBER                 PIC 9(4).                 (301)
00035      05  LOAN-INFORMATION.                                          (301)
00036          10  PAYMENT-AMT                  PIC 9(7)V99.              (301)
00037          10  LOAN-AMT                     PIC 9(13)V99.             (301)
00038          10  INTEREST-RATE                PIC V99999.               (301)
00039          10  LOAN-START-DATE              PIC 9(6).                 (301)
00040          10  LOAN-TYPE                    PIC 9(2).                 (301)
00041          10  LAST-BILL-DATE               PIC 9(6).                 (301)
00042          10  BILLING-DAYS                 PIC 9(3).                 (301)
00043          10  YEAR-TO-DATE-INTEREST        PIC 9(13)V99.             (301)
00044
00045  FD  MASTER-RPT                                                        (333)
00046      RECORDING MODE IS F
00047      BLOCK CONTAINS 0 RECORDS
00048      LABEL RECORDS ARE STANDARD.
00049  01  MAST-RPT.                                                        (375)
00050      05  FILLER                       PIC X(133).                   (375)
00051
00052  WORKING-STORAGE SECTION.
00053  77  FIRST-TIME                        PIC X VALUE 'Y'.             (474)
00054  77  MASTER-END-OF-FILE                PIC X VALUE SPACES.          (259)
00055      88  END-INPUT                     VALUE 'X'.                   (259)
00056  77  LINE-CNT                          PIC 39(4) COMP VALUE +55.    (292)
00057  77  PAGE-CNT                          PIC 39(4) COMP VALUE +0.     (290)
00058  77  REC-CNT                           PIC 39(4) COMP VALUE +0.     (258)
```

Report Field Descriptions Analysis without Compile

These are the Analysis without Compile report field descriptions:

Field	Description
(A)	Line numbers generated by the analyze job. All line number references throughout the SmartDoc reports are based on these numbers.
(B)	The source lines.
(C)	The first reference of the data item in source sequence.

Advanced Source Listing for the PROCEDURE DIVISION

The source code PROCEDURE DIVISION prints information about each line on the right side of the report. This information includes these items:

Control flow information. Control flow information shows how a statement is executed (PERFORM, GO TO, FALLTHRU, ENTRY, etc.), and how the flow continues to the next logical statement (PERFORM, GO TO, FALLTHRU, EXIT, RETURN, etc.). Use this information when tracing the logical flow of the program. Unexecutable code is indicated as DEADCODE.

Data flow information. Data flow information displays for each data item in a source line. The location of the data item definition is shown each time the data item is referenced. The location of the previous modification that set the current value displays if the reference is a USE. If the reference is a MODIFICATION, the location of the next use of the value is shown. Use this information when tracing data item usage.

Data items. Data items used without first being initialized are referenced as NO-VALUE. Data items modified without subsequently using the assigned value are referenced as NO-USE.

Special registers. Special registers are referenced as D-COBOL. Special registers are not included in the DATA DIVISION, therefore they are outside the bounds of the program. SmartDoc is unable to determine processing outside the bounds of the program, and refers to the special registers as being defined by COBOL.

CICS defined data items. CICS defined data items are referenced as D-CICS when a program containing command level CICS is analyzed without using the CICS preprocessor.

Data cross-reference information. If you did not perform an extended program analysis, data flow information is unavailable and data cross-reference information is shown instead (See Chapter 11, "Analyze," on page 157 for information on the analyze features.) Data cross-reference information lists the previous and next reference to the address space named by a data item, in source sequence. Cross-references are based on generated line numbers in the report.

Occasionally, control and data flow information exceeds available report space causing an overflow. An overflow causes all control and data flow information for that source line to move to the overflow area at the bottom of the page where the source line appears. Insufficient space at the page bottom of the page causes all previous overflow for that page to print, then the new overflow prints on the next page.

A symbol legend prints at the bottom of each report page. These symbols correspond to the control flow and data flow information shown to the right of each PROCEDURE DIVISION source statement.

These are the control flow symbols:

Symbol	Description
<-	Indicates where control was passed from
->	Indicates where control is being passed
- G	Control is transferred by a GO TO
I	Control is transferred by an internal call, for COBOL II Release 3 and later programs
P	Control is transferred by a PERFORM
R	Control RETURNs to the specified line
C	Control is transferred by an ON CONDITION exception or by a DECLARATIVE
FALLTHRU	Indicates processing falls through to the next statement
PGM EXIT	Control EXITs the program
PGM ENTRY	Program execution begins here
DEADCODE	The source line contains unexecutable code

These are the data flow symbols:

Symbol	Description
D	The reference is to the definition of the data item
M	The reference is to a previous modification of the data item
U	The reference is to the next use of the data item
E	The reference is to a possible external modification. An external modification occurs when a data item is passed to a called program and is modified by the called program.
D-CICS	A data item defined by CICS
D-COBOL	A special register. This data item also includes implicitly used registers. For example, CALL statements are marked as D-COBOL because the return code register is affected by the CALL.
D-IDMS	A data item defined by IDMS.
D-SQL	A data item defined by SQL.

When a source line contains multiple data items, each data item reference is enclosed in parentheses and listed in the order it appears in the source.

Data flow analysis information is unavailable when a short SmartDoc analysis is performed. Cross-referenced information is shown with these symbols:

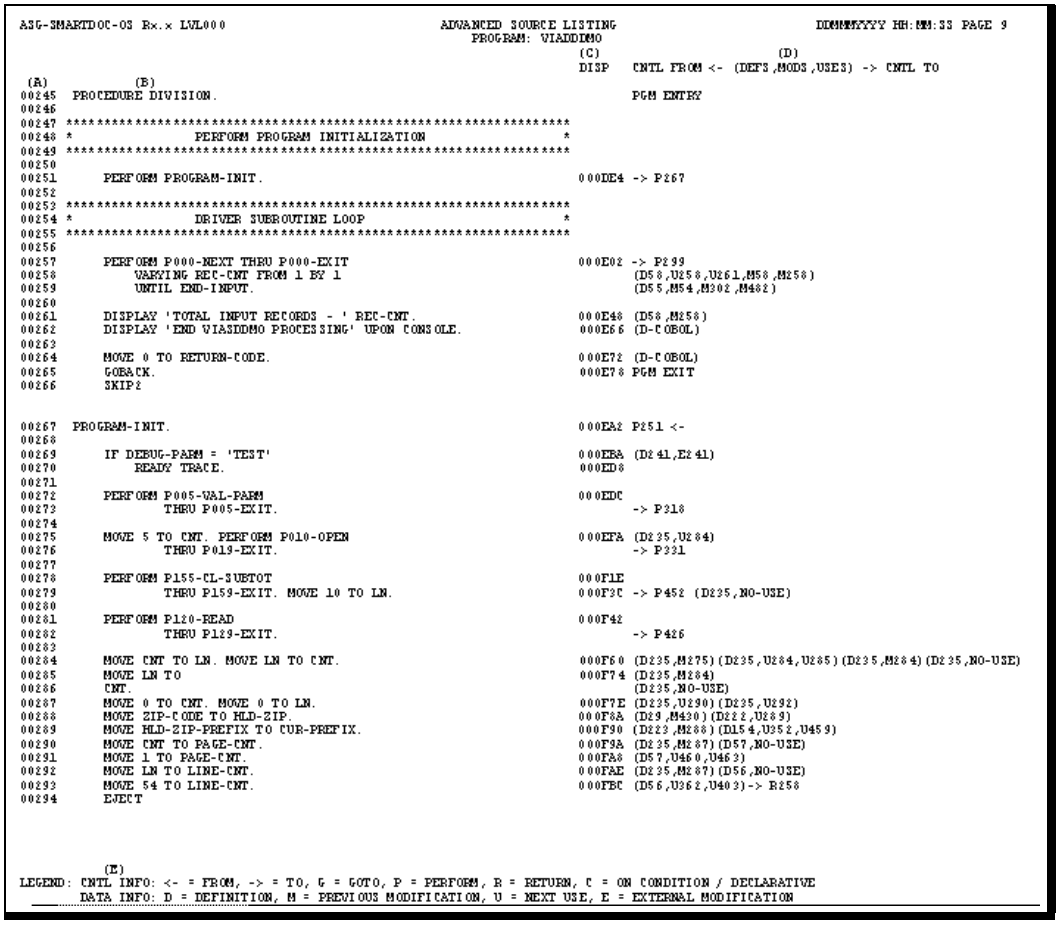
- D—the reference is to the definition of the data item
- P—the previous reference of the data item
- N—the next reference of the data item

These figures illustrate the Advanced Source Listing for the PROCEDURE DIVISION:

- The Advanced Source Listing produced by a DA analysis is shown in Figure 18 on page 54.
- The Advanced Source Listing produced by a DX analysis is shown in Figure 19 on page 56.
- The Advanced Source Listing produced by a DC analysis is shown in Figure 20 on page 58.
- The Advanced Source Listing produced by a DS analysis is shown in Figure 21 on page 60.

Extended SmartDoc Analysis with Compile (DA Analysis)

Figure 18 • Advanced Source Listing (DA Analysis) - PROCEDURE DIVISION



```
ASG-SMARTDOC-03 Rx.x LVL000                    ADVANCED SOURCE LISTING                    DDMMMYYYY HH:MM:SS PAGE 9
                                                  PROGRAM: VIADDM0
                                                                        (C)                    (D)
                                                                        DISP   CNTL FROM <- (DEFS,MODS,USES) -> CNTL TO
 (A)          (B)
00245  PROCEDURE DIVISION.                                                    PGM ENTRY
00246
00247 ******************************************************************
00248 *                PERFORM PROGRAM INITIALIZATION                  *
00249 ******************************************************************
00250
00251      PERFORM PROGRAM-INIT.                                        000DE4 -> P267
00252
00253 ******************************************************************
00254 *                DRIVER SUBROUTINE LOOP                          *
00255 ******************************************************************
00256
00257      PERFORM P000-NEXT THRU P000-EXIT                             000E02 -> P299
00258          VARYING REC-CNT FROM 1 BY 1                                     (D58,U258,U261,M58,M258)
00259          UNTIL END-INPUT.                                                (D55,M54,M302,M482)
00260
00261      DISPLAY 'TOTAL INPUT RECORDS - ' REC-CNT.                    000E48 (D58,M258)
00262      DISPLAY 'END VIASDDM0 PROCESSING' UPON CONSOLE.              000E66 (D-COBOL)
00263
00264      MOVE 0 TO RETURN-CODE.                                       000E72 (D-COBOL)
00265      GOBACK.                                                      000E78 PGM EXIT
00266      SKIP2


00267  PROGRAM-INIT.                                                    000EA2 P251 <-
00268
00269      IF DEBUG-PARM = 'TEST'                                       000EBA (D241,E241)
00270          READY TRACE.                                             000ED8
00271
00272      PERFORM P005-VAL-PARM                                        000EDC
00273              THRU P005-EXIT.                                             -> P318
00274
00275      MOVE 5 TO CNT. PERFORM P010-OPEN                             000EFA (D235,U284)
00276              THRU P019-EXIT.                                             -> P331
00277
00278      PERFORM P155-CL-SUBTOT                                       000F1E
00279              THRU P159-EXIT. MOVE 10 TO LN.                       000F3C -> P452 (D235,NO-USE)
00280
00281      PERFORM P120-READ                                            000F42
00282              THRU P129-EXIT.                                             -> P426
00283
00284      MOVE CNT TO LN. MOVE LN TO CNT.                              000F60 (D235,M275)(D235,U284,U285)(D235,M284)(D235,NO-USE)
00285      MOVE LN TO                                                   000F74 (D235,M284)
00286      CNT.                                                                (D235,NO-USE)
00287      MOVE 0 TO CNT. MOVE 0 TO LN.                                 000F7E (D235,U290)(D235,U292)
00288      MOVE ZIP-CODE TO HLD-ZIP.                                    000F8A (D29,M430)(D222,U289)
00289      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX.                           000F90 (D223,M288)(D154,U352,U459)
00290      MOVE CNT TO PAGE-CNT.                                        000F9A (D235,M287)(D57,NO-USE)
00291      MOVE 1 TO PAGE-CNT.                                          000FA8 (D57,U460,U463)
00292      MOVE LN TO LINE-CNT.                                         000FAE (D235,M287)(D56,NO-USE)
00293      MOVE 54 TO LINE-CNT.                                         000FBC (D56,U362,U403)-> R258
00294      EJECT



          (E)
LEGEND: CNTL INFO: <- = FROM, -> = TO, G = GOTO, P = PERFORM, R = RETURN, C = ON CONDITION / DECLARATIVE
        DATA INFO: D = DEFINITION, M = PREVIOUS MODIFICATION, U = NEXT USE, E = EXTERNAL MODIFICATION
```

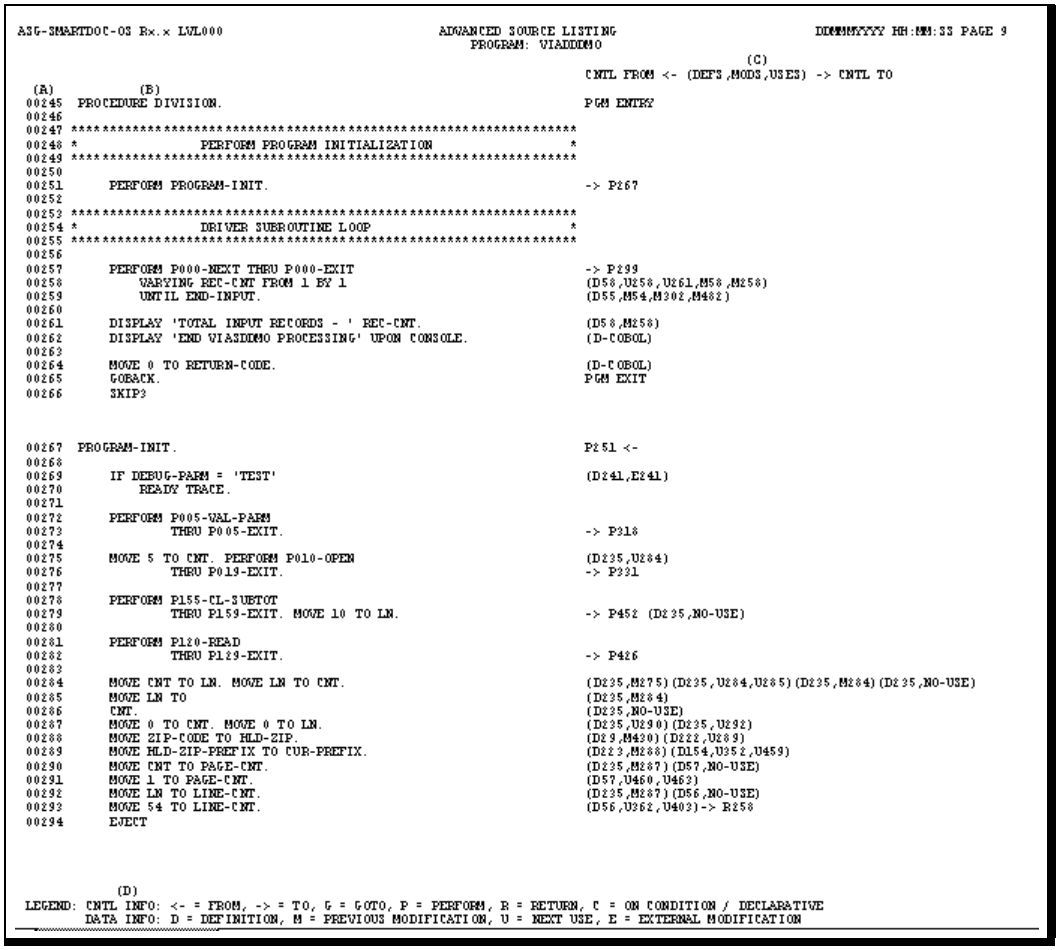
Report Field Descriptions

These are the Extended SmartDoc with Compile (DA Analysis) report field descriptions:

Field	Description
(A)	The line numbers generated by the analyze job. All line number references throughout the SmartDoc reports are based on these numbers.
(B)	The PROCEDURE DIVISION source lines.
(C)	The displacement of the statement into the CSECT. Use this information when debugging a program. For example, when a program abends, you can calculate the displacement by using the PSW address and the load point to determine the source statement where the problem actually occurred.
(D)	The control flow and data flow information. Control flow information shows how a statement executes and how the flow continues to the next logical statement. Data flow information shows the definition, previous modification of a data item, and the next data item use.
(E)	The symbol legend used to identify control flow and data flow.

Extended SmartDoc Analysis without Compile (DX Analysis)

Figure 19 • Advanced Source Listing (DX Analysis) - PROCEDURE DIVISION



```
ASG-SMARTDOC-03 Rx.x LVL000                    ADVANCED SOURCE LISTING                    DDMMMYYYY HH:MM:SS PAGE 9
                                                   PROGRAM: VIADDDMO
                                                                                              (C)
                                                                                 CNTL FROM <- (DEFS,MODS,USES) -> CNTL TO
 (A)          (B)
00245  PROCEDURE DIVISION.                                                      PGM ENTRY
00246
00247 ***************************************************************
00248 *                PERFORM PROGRAM INITIALIZATION                *
00249 ***************************************************************
00250
00251      PERFORM PROGRAM-INIT.                                                -> P267
00252
00253 ***************************************************************
00254 *                DRIVER SUBROUTINE LOOP                        *
00255 ***************************************************************
00256
00257      PERFORM P000-NEXT THRU P000-EXIT                                     -> P299
00258          VARYING REC-CNT FROM 1 BY 1                                      (D58,U258,U261,M58,M258)
00259          UNTIL END-INPUT.                                                 (D55,M54,M302,M482)
00260
00261      DISPLAY 'TOTAL INPUT RECORDS - ' REC-CNT.                            (D58,M258)
00262      DISPLAY 'END VIASDDMO PROCESSING' UPON CONSOLE.                      (D-COBOL)
00263
00264      MOVE 0 TO RETURN-CODE.                                               (D-COBOL)
00265      GOBACK.                                                              PGM EXIT
00266      SKIP3


00267  PROGRAM-INIT.                                                            P251 <-
00268
00269      IF DEBUG-PARM = 'TEST'                                               (D241,E241)
00270          READY TRACE.
00271
00272      PERFORM P005-VAL-PARM
00273              THRU P005-EXIT.                                              -> P318
00274
00275      MOVE 5 TO CNT. PERFORM P010-OPEN                                     (D235,U284)
00276              THRU P019-EXIT.                                              -> P331
00277
00278      PERFORM P155-CL-SUBTOT
00279              THRU P159-EXIT. MOVE 10 TO LN.                               -> P452 (D235,NO-USE)
00280
00281      PERFORM P120-READ
00282              THRU P129-EXIT.                                              -> P426
00283
00284      MOVE CNT TO LN. MOVE LN TO CNT.                                      (D235,M275)(D235,U284,U285)(D235,M284)(D235,NO-USE)
00285      MOVE LN TO                                                           (D235,M284)
00286      CNT.                                                                 (D235,NO-USE)
00287      MOVE 0 TO CNT. MOVE 0 TO LN.                                         (D235,U290)(D235,U292)
00288      MOVE ZIP-CODE TO HLD-ZIP.                                            (D29,M420)(D222,U289)
00289      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX.                                   (D223,M288)(D154,U352,U459)
00290      MOVE CNT TO PAGE-CNT.                                                (D235,M287)(D57,NO-USE)
00291      MOVE 1 TO PAGE-CNT.                                                  (D57,U460,U462)
00292      MOVE LN TO LINE-CNT.                                                 (D235,M287)(D56,NO-USE)
00293      MOVE 54 TO LINE-CNT.                                                 (D56,U362,U403)-> R258
00294      EJECT



           (D)
LEGEND: CNTL INFO: <- = FROM, -> = TO, G = GOTO, P = PERFORM, R = RETURN, C = ON CONDITION / DECLARATIVE
        DATA INFO: D = DEFINITION, M = PREVIOUS MODIFICATION, U = NEXT USE, E = EXTERNAL MODIFICATION
```

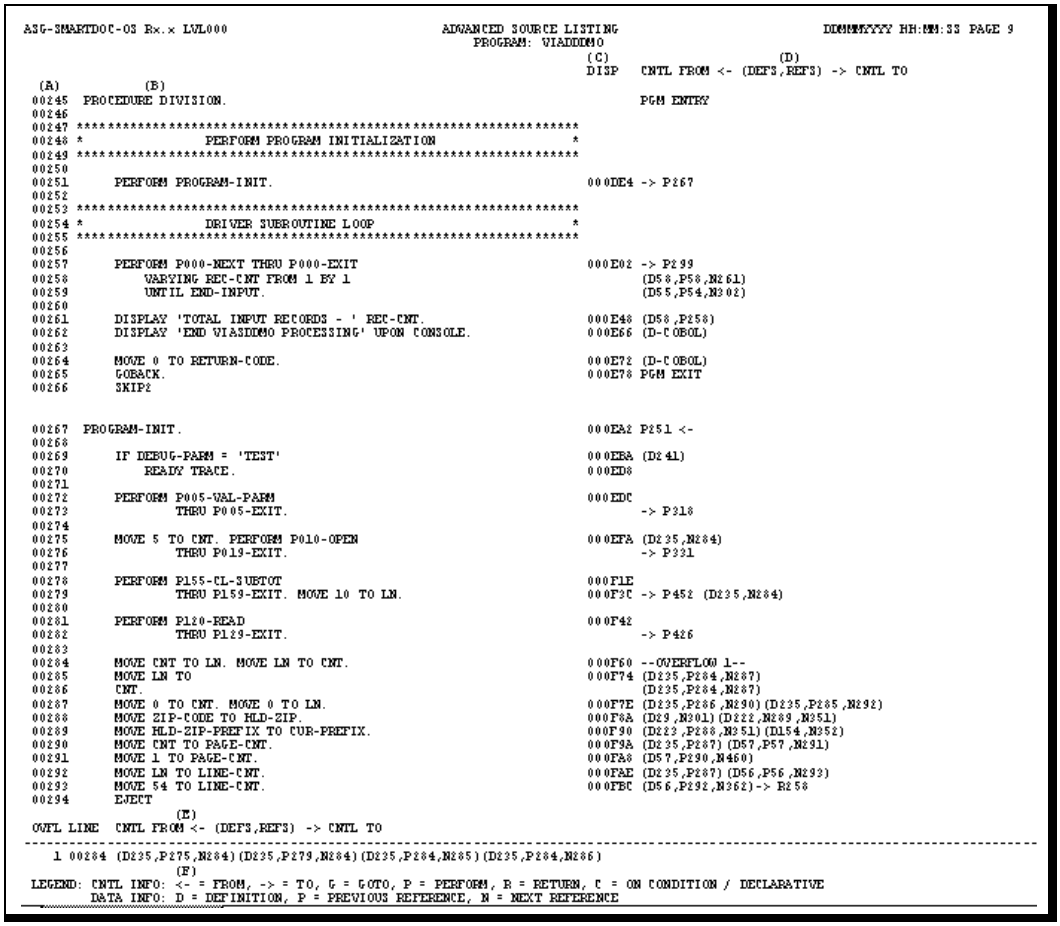
Report Field Descriptions

These are the Extended SmartDoc Analysis without Compile (DX Analysis) report fields:

Field	Description
(A)	This field shows the line numbers the analyze job generated. All line number references throughout the SmartDoc reports are based on these numbers.
(B)	This field shows the PROCEDURE DIVISION source lines.
(C)	This field shows the control flow and data flow information. Control flow information shows how a statement gets executed and how the flow continues to the next logical statement. Data flow information shows the definition, previous modification of a data item, and the next data item use.
(D)	This field shows the legend of symbols used to identify control flow and data flow.

Short SmartDoc Analysis with Compile (DC Analysis)

Figure 20 • Advanced Source Listing (DC Analysis) - PROCEDURE DIVISION



```
ASG-SMARTDOC-OS Rx.x LVL000                    ADVANCED SOURCE LISTING                          DDMMMYYYY HH:MM:SS PAGE 9
                                                   PROGRAM: VIADDM30
                                                                          (C)                        (D)
                                                                          DISP   CNTL FROM <- (DEFS,REFS) -> CNTL TO
 (A)          (B)
00245  PROCEDURE DIVISION.                                                      PGM ENTRY
00246
00247 ***************************************************************
00248 *                PERFORM PROGRAM INITIALIZATION                *
00249 ***************************************************************
00250
00251      PERFORM PROGRAM-INIT.                                       000DE4 -> P267
00252
00253 ***************************************************************
00254 *               DRIVER SUBROUTINE LOOP                         *
00255 ***************************************************************
00256
00257      PERFORM P000-NEXT THRU P000-EXIT                            000E02 -> P299
00258          VARYING REC-CNT FROM 1 BY 1                                    (D58,P58,N261)
00259          UNTIL END-INPUT.                                               (D55,P54,N302)
00260
00261      DISPLAY 'TOTAL INPUT RECORDS - ' REC-CNT.                   000E48 (D58,P258)
00262      DISPLAY 'END VIASDDM30 PROCESSING' UPON CONSOLE.            000E66 (D-COBOL)
00263
00264      MOVE 0 TO RETURN-CODE.                                      000E72 (D-COBOL)
00265      GOBACK.                                                     000E78 PGM EXIT
00266      SKIP2


00267  PROGRAM-INIT.                                                   000EA2 P251 <-
00268
00269      IF DEBUG-PARM = 'TEST'                                      000EBA (D241)
00270          READY TRACE.                                            000ED8
00271
00272      PERFORM P005-VAL-PARM                                       000EDC
00273              THRU P005-EXIT.                                            -> P318
00274
00275      MOVE 5 TO CNT. PERFORM P010-OPEN                            000EFA (D235,N284)
00276              THRU P019-EXIT.                                            -> P331
00277
00278      PERFORM P155-CL-SUBTOT                                      000F1E
00279              THRU P159-EXIT. MOVE 10 TO LN.                      000F3C -> P452 (D235,N284)
00280
00281      PERFORM P120-READ                                           000F42
00282              THRU P129-EXIT.                                            -> P426
00283
00284      MOVE CNT TO LN. MOVE LN TO CNT.                             000F60 --OVERFLOW 1--
00285      MOVE LN TO                                                  000F74 (D235,P284,N287)
00286      CNT.                                                               (D235,P284,N287)
00287      MOVE 0 TO CNT. MOVE 0 TO LN.                                000F7E (D235,P286,N290)(D235,P285,N292)
00288      MOVE ZIP-CODE TO HLD-ZIP.                                   000F8A (D29,N301)(D222,N289,N351)
00289      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX.                          000F90 (D222,P288,N351)(D154,N352)
00290      MOVE CNT TO PAGE-CNT.                                       000F9A (D235,P287)(D57,P57,N291)
00291      MOVE 1 TO PAGE-CNT.                                         000FA8 (D57,P290,N460)
00292      MOVE LN TO LINE-CNT.                                        000FAE (D235,P287)(D56,P56,N293)
00293      MOVE 54 TO LINE-CNT.                                        000FBC (D56,P292,N362)-> R258
00294      EJECT
                (E)
OVFL LINE  CNTL FROM <- (DEFS,REFS) -> CNTL TO
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
   1 00284 (D235,P275,N284)(D235,P279,N284)(D235,P284,N285)(D235,P284,N286)
           (F)
LEGEND: CNTL INFO: <- = FROM, -> = TO, G = GOTO, P = PERFORM, R = RETURN, C = ON CONDITION / DECLARATIVE
        DATA INFO: D = DEFINITION, P = PREVIOUS REFERENCE, N = NEXT REFERENCE
```

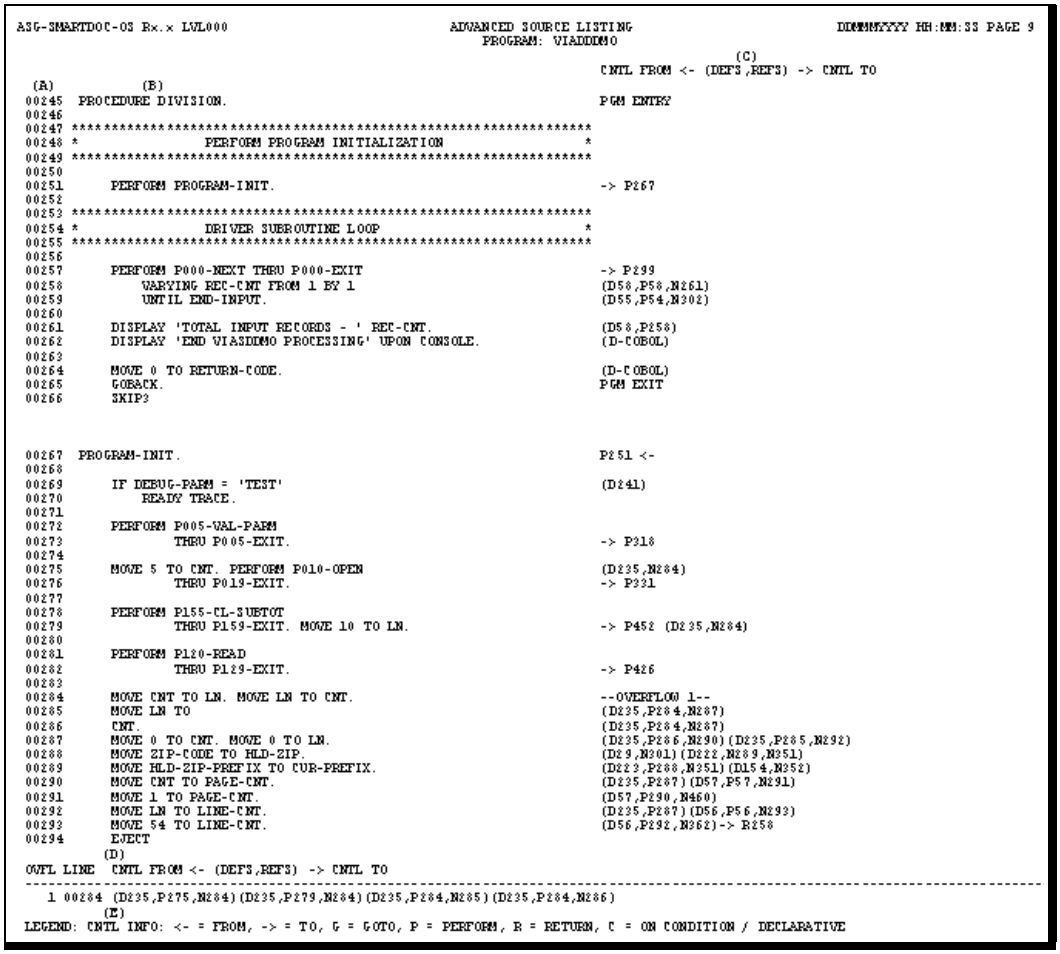
Report Field Descriptions

These are the Short SmartDoc Analysis with Compile (DC Analysis) report fields:

Field	Description
(A)	This field shows the line numbers the analyze job generated. All line number references throughout the SmartDoc reports are based on these numbers.
(B)	This field shows the PROCEDURE DIVISION source lines.
(C)	This field shows the displacement of the statement into the CSECT. Use this information when debugging a program. For example, when a program abends, you can calculate the displacement by using the PSW address and the load point to determine the source statement where the problem actually occurred.
(D)	This field shows the control flow and data cross-reference information. Control flow information shows how a statement executes and how the flow continues to the next logical statement. Data cross-reference information shows the definition, previous reference, and the next data item reference.
(E)	This field shows the overflow information from the statement containing more information than fits on the line with the PROCEDURE DIVISION source statement.
(F)	This field shows the legend of symbols used to identify control flow and data flow.

Short SmartDoc Analysis without Compile (DS Analysis)

Figure 21 • Advanced Source Listing (DS Analysis) - PROCEDURE DIVISION



```
ASG-SMARTDOC-03 Rx.x LVL000                    ADVANCED SOURCE LISTING                    DDMMMYYYY HH:MM:SS PAGE 9
                                                  PROGRAM: VIADDM90
                                                                                  (C)
                                                                                  CNTL FROM <- (DEFS,REFS) -> CNTL TO
 (A)          (B)
00245  PROCEDURE DIVISION.                                                       PGM ENTRY
00246
00247 ************************************************************************
00248 *                  PERFORM PROGRAM INITIALIZATION                      *
00249 ************************************************************************
00250
00251      PERFORM PROGRAM-INIT.                                                -> P267
00252
00253 ************************************************************************
00254 *                  DRIVER SUBROUTINE LOOP                              *
00255 ************************************************************************
00256
00257      PERFORM P000-NEXT THRU P000-EXIT                                     -> P299
00258          VARYING REC-CNT FROM 1 BY 1                                      (D58,P58,N261)
00259          UNTIL END-INPUT.                                                 (D55,P54,N302)
00260
00261      DISPLAY 'TOTAL INPUT RECORDS - ' REC-CNT.                            (D58,P258)
00262      DISPLAY 'END VIASDDM90 PROCESSING' UPON CONSOLE.                     (D-COBOL)
00263
00264      MOVE 0 TO RETURN-CODE.                                               (D-COBOL)
00265      GOBACK.                                                              PGM EXIT
00266      SKIP3


00267  PROGRAM-INIT.                                                            P251 <-
00268
00269      IF DEBUG-PARM = 'TEST'                                               (D241)
00270          READY TRACE.
00271
00272      PERFORM P005-VAL-PARM
00273              THRU P005-EXIT.                                              -> P318
00274
00275      MOVE 5 TO CNT. PERFORM P010-OPEN                                     (D235,N284)
00276              THRU P019-EXIT.                                              -> P331
00277
00278      PERFORM P155-CL-SUBTOT
00279              THRU P159-EXIT. MOVE 10 TO LN.                               -> P452 (D235,N284)
00280
00281      PERFORM P120-READ
00282              THRU P129-EXIT.                                              -> P426
00283
00284      MOVE CNT TO LN. MOVE LN TO CNT.                                      --OVERFLOW 1--
00285      MOVE LN TO                                                           (D235,P284,N287)
00286      CNT.                                                                 (D235,P284,N287)
00287      MOVE 0 TO CNT. MOVE 0 TO LN.                                         (D235,P286,N290)(D235,P285,N292)
00288      MOVE ZIP-CODE TO HLD-ZIP.                                            (D29,N301)(D222,N289,N351)
00289      MOVE HLD-ZIP-PREFIX TO CUR-PREFIX.                                   (D222,P288,N351)(D154,N352)
00290      MOVE CNT TO PAGE-CNT.                                                (D235,P287)(D57,P57,N291)
00291      MOVE 1 TO PAGE-CNT.                                                  (D57,P290,N460)
00292      MOVE LN TO LINE-CNT.                                                 (D235,P287)(D56,P56,N293)
00293      MOVE 54 TO LINE-CNT.                                                 (D56,P292,N362)-> R258
00294      EJECT
          (D)
OVFL LINE  CNTL FROM <- (DEFS,REFS) -> CNTL TO
-------------------------------------------------------------------------------------------------------------------------------
   1 00284  (D235,P275,N284)(D235,P279,N284)(D235,P284,N285)(D235,P284,N286)
          (E)
LEGEND: CNTL INFO: <- = FROM, -> = TO, G = GOTO, P = PERFORM, R = RETURN, C = ON CONDITION / DECLARATIVE
```

Report Field Descriptions

These are the Short SmartDoc Analysis without Compile (DS Analysis) report fields:

Field	Description
(A)	This field shows the line numbers the analyze job generated. All line number references throughout the SmartDoc reports are based on these numbers.
(B)	This field shows the PROCEDURE DIVISION source lines.
(C)	This field shows the control flow and data cross-reference information. Control flow information shows how a statement executes and how the flow continues to the next logical statement. Data cross-reference information shows the definition, previous reference, and the next data item reference.
(D)	This field shows the overflow information from the statement containing information than fits on the line with the PROCEDURE DIVISION source statement.
(E)	This field shows the legend of symbols used to identify control flow and data flow.

Advanced Source Listing for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, the Advanced Source Listing uses the symbol I to identify calls to internal subprograms. The symbol is followed by the line number where the control is passed. Figure 22 and Figure 23 illustrate the Advanced Source Listing for the PROCEDURE DIVISION of a COBOL II Release 3 program, showing an internally-called subprogram. An Extended SmartDoc analysis with a compile has been performed (DA analysis).

Figure 22 shows the call statement, showing that control is transferred by an internal call to line 606.

Figure 22 • Advanced Source Listing for COBOL II Release 3 - Example 1

```
ASG-SMARTDOC-OS Rx.x LVL000                          ADVANCED SOURCE LISTING                          DDMMMYYYY HH:MM:SS PAGE  999
                                                        PROGRAM: VIADDEM3

                                                                                  DISP   CNTL FROM <- (DEFS,MODS,USES) -> CNTL TO

00231  procedure division.                                                        0008D0 PGM ENTRY
00232
00272 ************************************************************************
00273 *               process the input file                                   *
00274 ************************************************************************
00275
00276  p000-next.                                                                       P243 <-
00277
00278      call 'viaddeml' using master-in                                       000A42 --OVERFLOW 1--
00279                            master-end-of-file                                     (D54,U245,U279,U396,U402,M54,M279,M452)
00280                            master-report-date.                                    (D157,U280,M280,M427)-> I606
00281
00282      perform p100-print                                                    000A84

OVFL LINE  CNTL FROM <- (DEFS,MODS,USES) -> CNTL TO
---- ----- ------------------------------------------------------------------------------------------------------------------------------
   1 00278 (D20,U278,U327,U345,U346,U347,U350,U351,U352,U355,U356,U357,U358,U359,U360,U363,U364,U365,U366,U367,U383,U385,U405,M278,
           M399)

LEGEND: CNTL INFO: <- = FROM, -> = TO, G = GOTO, I = INT. CALL, P = PERFORM, R = RETURN, C = ON CONDITION / DECLARATIVE
        DATA INFO: D = DEFINITION, M = PREVIOUS MODIFICATION, U = NEXT USE, E = EXTERNAL MODIFICATION
```

Figure 23 shows the code, beginning on line 606, control was transferred to in the previous figure.

Figure 23 • Advanced Source Listing for COBOL II Release 3 - Example 2

```
ASG-SMARTDOC-OS Rx.x LVL000                          ADVANCED SOURCE LISTING                          DDMMMYYYY HH:MM:SS PAGE  999
                                                  PROGRAM: VIADDEM3     SUBPROGRAM: VIADDEM1

                                                                                  DISP   CNTL FROM <- (DEFS,MODS,USES) -> CNTL TO

00606  procedure division using master-in,                                         I278 <-
00607                           master-end-of-file,
00608                           master-report-date.
00609
00610      if first-time = 'Y'                                                    0014E6 (D526,M526)
00611          open output daily-totals                                           0014F2 (D511,U611,M611,M617)
00612          move init-amts to loan-amt                                         00150A (D527,M527)(D592,U633,U637)
00613          move ' ' to first-time      else                                   00151E (D526,NO-USE)-> FALLTHRU
00614          if end-input                                                       001526 (D601,NO-VALUE)
00615          perform p200-print-totals                                          001532
00616                  thru p299-exit                                                   -> P648
00617          close daily-totals                                                 00154A (D511,U611,M669)
00618          go to p009-end-program                                             001552 -> G623
00619      else
00620         perform p100-compute-totals                                         00156E
00621              thru p199-exit.                                                     -> P631 -> FALLTHRU
00622
00623  p009-end-program.                                                               FALLTHRU,G618 <-
00624      goback.                                                                -> R282


LEGEND: CNTL INFO: <- = FROM, -> = TO, G = GOTO, I = INT. CALL, P = PERFORM, R = RETURN, C = ON CONDITION / DECLARATIVE
        DATA INFO: D = DEFINITION, M = PREVIOUS MODIFICATION, U = NEXT USE, E = EXTERNAL MODIFICATION
```

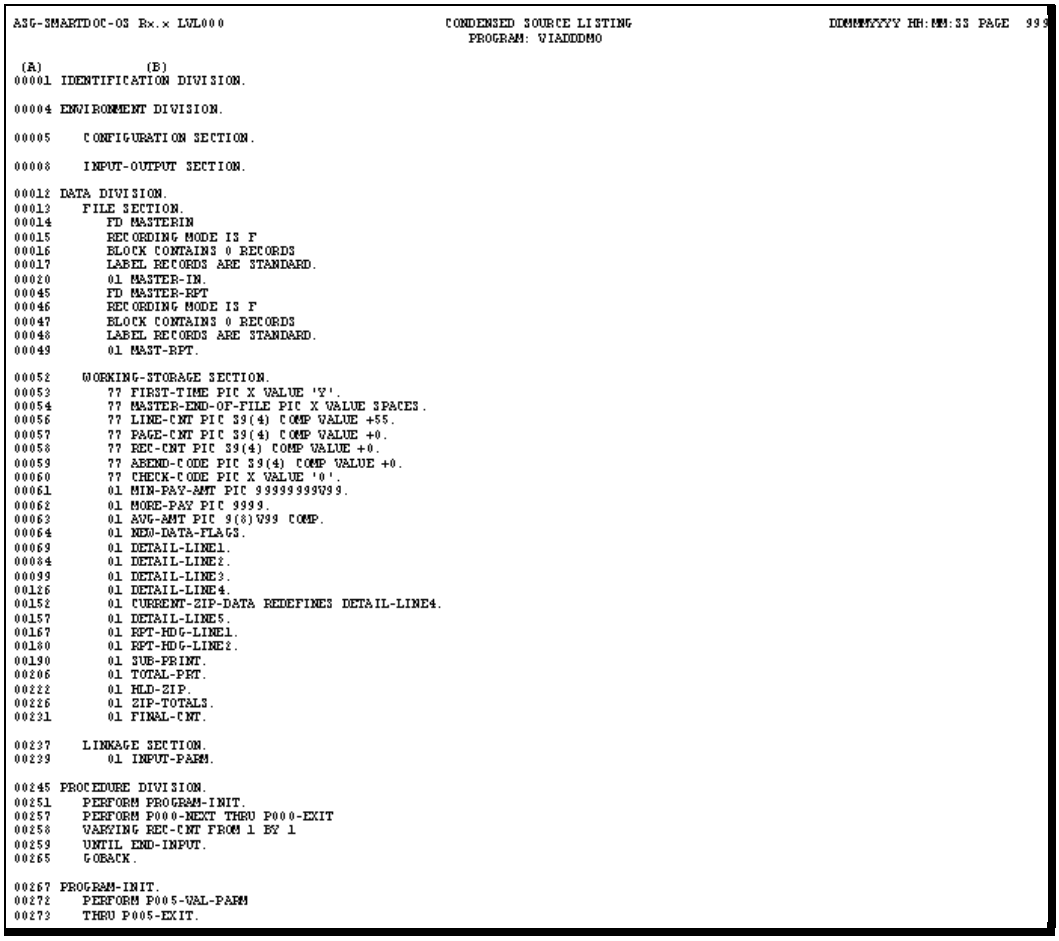
Condensed Source Listing

The Condensed Source Listing (see Figure 24 on page 64) shows only the structurally significant portions of the program. It eliminates structurally insignificant source code statements and shows the remaining code indented hierarchically according to divisions, sections, and paragraphs. The IDENTIFICATION, ENVIRONMENT, DATA, and PROCEDURE divisions are shown on the Condensed Source Listing. For COBOL II Release 3 and later programs, each internally-called subprogram is started on a new page.

This is the list of items included on the Condensed Source Listing:

Item	Significant Program Portion
IDENTIFICATION DIVISION	Division header
ENVIRONMENT DIVISION	Division header Section headers
DATA DIVISION	Division header Section headers FD, SD, RD, CD, 01 and 77 level data items
PROCEDURE DIVISION	Division header Section and paragraph labels GO TO statements PERFORM statements CALL statements ENTRY statements CICS statements that are structurally significant (e.g., LINK) STOP RUN statements GOBACK statements PGM EXIT statements

Figure 24 • Condensed Source Listing



```
ASG-SMARTDOC-OS Rx.x LVL000                    CONDENSED SOURCE LISTING                    DDMMMYYYY HH:MM:SS PAGE  999
                                                  PROGRAM: VIADDDM0

 (A)            (B)
00001 IDENTIFICATION DIVISION.

00004 ENVIRONMENT DIVISION.

00005     CONFIGURATION SECTION.

00008     INPUT-OUTPUT SECTION.

00012 DATA DIVISION.
00013     FILE SECTION.
00014         FD MASTERIN
00015         RECORDING MODE IS F
00016         BLOCK CONTAINS 0 RECORDS
00017         LABEL RECORDS ARE STANDARD.
00020         01 MASTER-IN.
00045         FD MASTER-RPT
00046         RECORDING MODE IS F
00047         BLOCK CONTAINS 0 RECORDS
00048         LABEL RECORDS ARE STANDARD.
00049         01 MAST-RPT.

00052     WORKING-STORAGE SECTION.
00053         77 FIRST-TIME PIC X VALUE 'Y'.
00054         77 MASTER-END-OF-FILE PIC X VALUE SPACES.
00056         77 LINE-CNT PIC S9(4) COMP VALUE +55.
00057         77 PAGE-CNT PIC S9(4) COMP VALUE +0.
00058         77 REC-CNT PIC S9(4) COMP VALUE +0.
00059         77 ABEND-CODE PIC S9(4) COMP VALUE +0.
00060         77 CHECK-CODE PIC X VALUE '0'.
00061         01 MIN-PAY-AMT PIC 99999999V99.
00062         01 MORE-PAY PIC 9999.
00063         01 AVG-AMT PIC 9(8)V99 COMP.
00064         01 NEW-DATA-FLAGS.
00069         01 DETAIL-LINE1.
00084         01 DETAIL-LINE2.
00099         01 DETAIL-LINE3.
00126         01 DETAIL-LINE4.
00152         01 CURRENT-ZIP-DATA REDEFINES DETAIL-LINE4.
00157         01 DETAIL-LINE5.
00167         01 RPT-HDG-LINE1.
00180         01 RPT-HDG-LINE2.
00190         01 SUB-PRINT.
00206         01 TOTAL-PRT.
00222         01 HLD-ZIP.
00226         01 ZIP-TOTALS.
00231         01 FINAL-CNT.

00237     LINKAGE SECTION.
00239         01 INPUT-PARM.

00245 PROCEDURE DIVISION.
00251     PERFORM PROGRAM-INIT.
00257     PERFORM P000-NEXT THRU P000-EXIT
00258     VARYING REC-CNT FROM 1 BY 1
00259     UNTIL END-INPUT.
00265     GOBACK.

00267 PROGRAM-INIT.
00272     PERFORM P005-VAL-PARM
00273     THRU P005-EXIT.
```

Report Field Descriptions

These are the Condensed Source Listing report fields:

Field	Description
(A)	This field shows the source code line numbers.
(B)	This field shows the significant source code statements that show the structure of the program.

Perform Range Hierarchy Chart

The Perform Range Hierarchy Chart (see Figure 25 on page 66) shows the perform range execution relationships (not the execution sequence of the program) in an indented list. Each relative PERFORM nesting level is also shown. When SmartDoc produces the diagram, these various conditions can occur:

- If the list is too long (vertically) to fit on a page, it continues on the next page.
- If the list is too wide (horizontally) to fit on a page, the item that does not fit and its subordinates appear on the next page with no other information. The location where the item would have appeared contains a reference to the page where it does appear (by name and page number).
- If a perform range is called from multiple places, and it is not a terminal PERFORM (a PERFORM that does not perform anything else) the perform range is either shown or is given a cross-reference (based on the DUPPERF option). The cross-reference is by name and page number to the first occurrence where the full structure is defined. The DUPPERF option determines if a perform range is repeated on the Perform Range Hierarchy Chart each time it is called.

Note:

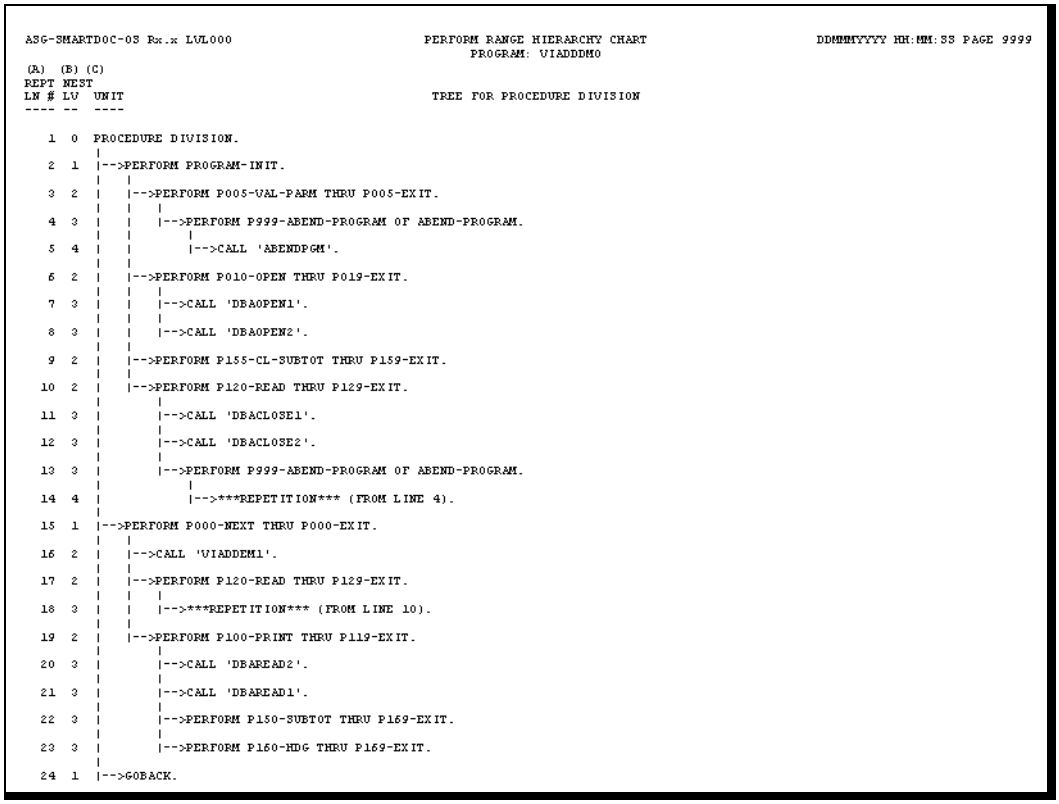
The DUPPERF option can produce a lengthy Perform Range Hierarchy Chart if the program contains many perform ranges called from multiple places.

For COBOL II Release 3 and later programs containing internal subprograms, a separate Perform Range Hierarchy Chart is produced for each subprogram.

Use these options to produce the Perform Range Hierarchy Chart:

Option	Description
Gotos option	GO TO and ALTER statements are also included in the report.
Conditionals option	Conditionals that affect the PERFORM, CALL, GO TO and ALTER statements are also included. When you select the Conditionals option, the Gotos option is automatically selected.

Figure 25 • Perform Range Hierarchy Chart



```
ASG-SMARTDOC-03 Rx.x LVL000                    PERFORM RANGE HIERARCHY CHART                    DDMMMYYYY HH:MM:SS PAGE 9999
                                                     PROGRAM: VIADDDM0

(A)  (B) (C)
REPT NEST
LN # LV  UNIT                                   TREE FOR PROCEDURE DIVISION
---- --  ----

   1  0  PROCEDURE DIVISION.
         |
   2  1  |-->PERFORM PROGRAM-INIT.
         |   |
   3  2  |   |-->PERFORM P005-VAL-PARM THRU P005-EXIT.
         |   |   |
   4  3  |   |   |-->PERFORM P999-ABEND-PROGRAM OF ABEND-PROGRAM.
         |   |   |       |
   5  4  |   |   |       |-->CALL 'ABENDPGM'.
         |   |
   6  2  |   |-->PERFORM P010-OPEN THRU P019-EXIT.
         |   |   |
   7  3  |   |   |-->CALL 'DBAOPEN1'.
         |   |   |
   8  3  |   |   |-->CALL 'DBAOPEN2'.
         |   |
   9  2  |   |-->PERFORM P155-CL-SUBTOT THRU P159-EXIT.
         |   |
  10  2  |   |-->PERFORM P120-READ THRU P129-EXIT.
         |       |
  11  3  |       |-->CALL 'DBACLOSE1'.
         |       |
  12  3  |       |-->CALL 'DBACLOSE2'.
         |       |
  13  3  |       |-->PERFORM P999-ABEND-PROGRAM OF ABEND-PROGRAM.
         |       |       |
  14  4  |       |       |-->***REPETITION*** (FROM LINE 4).
         |
  15  1  |-->PERFORM P000-NEXT THRU P000-EXIT.
         |   |
  16  2  |   |-->CALL 'VIADDEM1'.
         |   |
  17  2  |   |-->PERFORM P120-READ THRU P129-EXIT.
         |   |   |
  18  3  |   |   |-->***REPETITION*** (FROM LINE 10).
         |   |
  19  2  |   |-->PERFORM P100-PRINT THRU P119-EXIT.
         |       |
  20  3  |       |-->CALL 'DBAREAD2'.
         |       |
  21  3  |       |-->CALL 'DBAREAD1'.
         |       |
  22  3  |       |-->PERFORM P150-SUBTOT THRU P169-EXIT.
         |       |
  23  3  |       |-->PERFORM P160-HDG THRU P169-EXIT.
         |
  24  1  |-->GOBACK.
```

Report Field Descriptions

These are the Perform Range Hierarchy Chart fields:

Field	Description
(A)	This field shows the line number as generated for the Perform Range Hierarchy Chart.
(B)	This field shows the perform range level including all nested perform ranges.
(C)	This field refers to each perform range and the code included in that range as a unit.

Perform Range Hierarchy Chart for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, the Perform Range Hierarchy Chart (see Figure 26) places the word INTERNAL after the call statement to identify internally-called subprograms.

Figure 26 • Perform Range Hierarchy Chart for COBOL II Release 3

```
ASG-SMARTDOC-03 Rx.x LVL000                    PERFORM RANGE HIERARCHY CHART                    DDMMMYYYY HH:MM:SS PAGE  999
                                                        PROGRAM: VIADDEM3

REPT NEST
LN # LV  UNIT                                   TREE FOR PROCEDURE DIVISION
---- --  ----

   1  0  PROCEDURE DIVISION.
         |
   2  1  |-->PERFORM PROGRAM-INIT.
         |   |
   3  2  |   |-->PERFORM P005-VAL-PARM THRU P005-EXIT.
         |   |   |
   4  3  |   |   |-->CALL P999-ABEND-PROGRAM - INTERNAL.
         |   |
   5  2  |   |-->PERFORM P010-OPEN THRU P019-EXIT.
         |   |   |
   6  3  |   |   |-->CALL 'DBAOPEN1'.
         |   |   |
   7  3  |   |   |-->CALL 'DBAOPEN2'.
         |   |
   8  2  |   |-->PERFORM P155-CL-SUBTOT THRU P159-EXIT.
         |   |
   9  2  |   |-->PERFORM P120-READ THRU P129-EXIT.
         |       |
  10  3  |       |-->CALL P999-ABEND-PROGRAM - INTERNAL.
         |       |
  11  3  |       |-->CALL 'DBACLOSE1'.
         |       |
  12  3  |       |-->CALL 'DBACLOSE2'.
         |
  13  1  |-->PERFORM P000-NEXT THRU P000-EXIT.
         |   |
  14  2  |   |-->CALL VIADDEM1 - INTERNAL.
         |   |
  15  2  |   |-->PERFORM P120-READ THRU P129-EXIT.
         |   |   |
  16  3  |   |   |-->***REPETITION*** (FROM LINE 10).
         |   |
  17  2  |   |-->PERFORM P100-PRINT THRU P119-EXIT.
         |       |
  18  3  |       |-->PERFORM P150-SUBTOT THRU P169-EXIT.
         |       |
  19  3  |       |-->PERFORM P160-HDG THRU P169-EXIT.
         |       |
  20  3  |       |-->CALL 'DBAREAD2'.
         |       |
  21  3  |       |-->CALL 'DBAREAD1'.
         |
  22  1  |-->GOBACK.
```

Perform Range Hierarchy Chart - Gotos and Conditionals Options

To include GO TO and ALTER statements in the report, follow this step:

- Select the Perform Range Hierarchy Chart Gotos option.

To include the structurally relevant conditional statements associated with the PERFORM, CALL, GO TO, and ALTER statements in the report, follow this step:

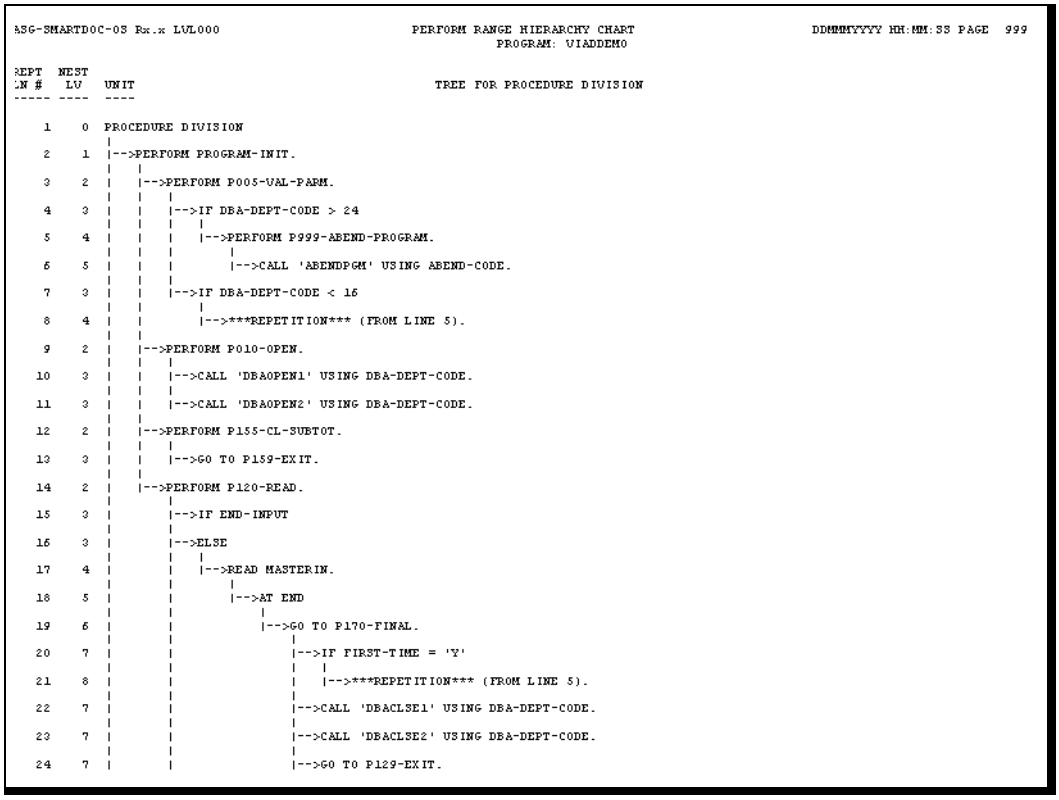
- Select the Perform Range Hierarchy Chart Conditionals option.

To automatically select the Gotos option, follow this step:

- Select the Conditionals option.

Figure 27 shows the Perform Range Hierarchy Chart, including the Gotos and Conditionals options.

Figure 27 • Perform Range Hierarchy Chart - Gotos & Conditionals Options



```
ASG-SMARTDOC-03 Rx.x LVL000                    PERFORM RANGE HIERARCHY CHART                    DDMMMYYYY HH:MM:SS PAGE  999
                                                        PROGRAM: VIADDEM0

REPT  NEST
LN #   LV   UNIT                                    TREE FOR PROCEDURE DIVISION
----- ----  ----

    1    0  PROCEDURE DIVISION
    2    1  |-->PERFORM PROGRAM-INIT.
    3    2  |   |-->PERFORM P005-VAL-PARM.
    4    3  |   |   |-->IF DBA-DEPT-CODE > 24
    5    4  |   |   |   |-->PERFORM P999-ABEND-PROGRAM.
    6    5  |   |   |       |-->CALL 'ABENDPGM' USING ABEND-CODE.
    7    3  |   |   |-->IF DBA-DEPT-CODE < 16
    8    4  |   |       |-->***REPETITION*** (FROM LINE 5).
    9    2  |   |-->PERFORM P010-OPEN.
   10    3  |   |   |-->CALL 'DBAOPEN1' USING DBA-DEPT-CODE.
   11    3  |   |   |-->CALL 'DBAOPEN2' USING DBA-DEPT-CODE.
   12    2  |   |-->PERFORM P155-CL-SUBTOT.
   13    3  |   |   |-->GO TO P159-EXIT.
   14    2  |   |-->PERFORM P120-READ.
   15    3  |       |-->IF END-INPUT
   16    3  |       |-->ELSE
   17    4  |       |   |-->READ MASTERIN.
   18    5  |       |       |-->AT END
   19    6  |       |           |-->GO TO P170-FINAL.
   20    7  |       |               |-->IF FIRST-TIME = 'Y'
   21    8  |       |               |   |-->***REPETITION*** (FROM LINE 5).
   22    7  |       |               |-->CALL 'DBACLSE1' USING DBA-DEPT-CODE.
   23    7  |       |               |-->CALL 'DBACLSE2' USING DBA-DEPT-CODE.
   24    7  |       |               |-->GO TO P129-EXIT.
```

Structure Chart

The Structure Chart (see Figure 28 on page 69) presents the program execution order graphically, and shows the relationships between the logical units of the program. Logical units are either called programs and performed paragraphs or sections.

Note:

The Structure Chart does not show the information in execution sequence.

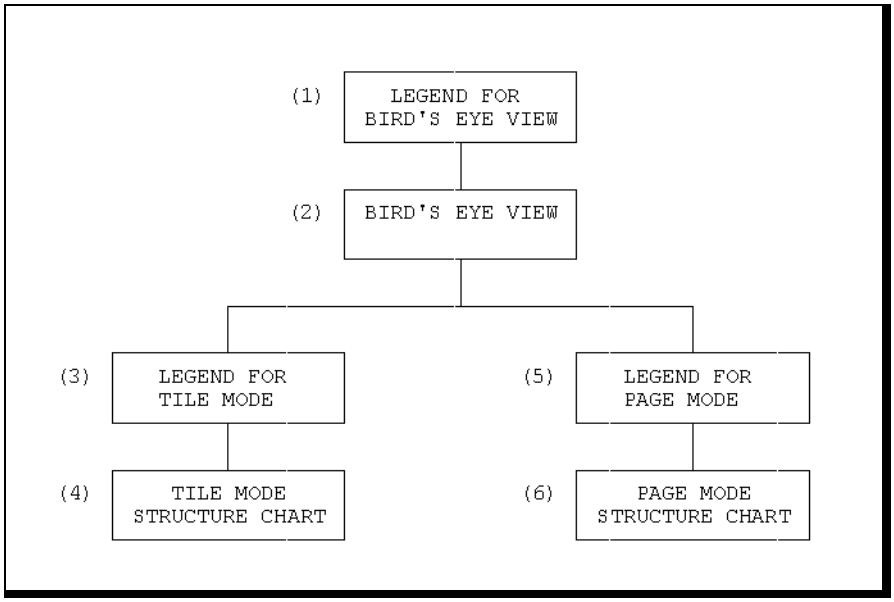
Structure Charts are produced in either the Page or the Tile Mode. Page Mode produces a Structure Chart that fits in a notebook. Tile Mode produces a large Structure Chart in pieces that form a single chart when taped together. Additionally, the Structure Chart can be produced in Bird's Eye View, a highly condensed format that shows the entire program in a minimum number of pages.

Use these options to produce the Structure Chart:

Option	Description
Gotos option	GO TO and ALTER statements are included in the report.
Conditionals option	Conditionals that affect the PERFORM, CALL, GO TO and ALTER statements are also included. Select the Conditionals option to automatically select the Gotos option.

Figure 28 shows the different Structure Chart output types.

Figure 28 • Structure Chart Components



These notes correspond to the numbers in Figure 28.

Field	Name	Description
1, 3, 5	Legend Pages	These pages give you instructions on how to use the chart that immediately follows.
2	Bird's Eye View	A Structure Chart produced in Tile Mode format with a box size of 1 x 1 (in characters).

Field	Name	Description
4	Tile Mode Structure Chart	A Structure Chart consisting of pages that can be taped together.
6	Page Mode Structure Chart	A Structure Chart that fits in a notebook. Wherever possible, this type of Structure Chart shows a parent and all of its subordinates (called either paragraphs or sections) on the same page. If the parent and its subordinates do not fit on a page, they are shown on a separate page as a substructure.

The Structure Chart presents each logical unit as a box with lines either going to or coming from it. The lines indicate the execution order. When you specify Page Mode TO PAGE, the page numbers are shown instead of the box, and a connected box is continued on the next page. A diagram continued from another page is shown as FROM PAGE(S) followed by the page number(s).

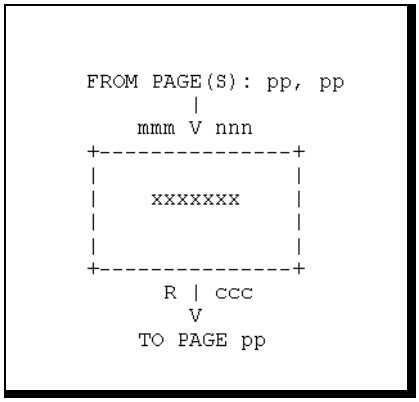
A perform range called from multiple places that is not a terminal PERFORM (a PERFORM that does not perform anything else) is either shown or given a cross-reference (based on the DUPPERF option). This cross-reference is by name and page number to the first occurrence where the full structure is defined. The DUPPERF option determines if a perform range is repeated on the Structure Chart each time it is called. Using the DUPPERF option can produce a lengthy Structure Chart if the program contains many perform ranges called from multiple places.

Note:

The DUPPERF option only applies to Structure Charts produced in Tile Mode. NODUPPERF is implicit for Structure Charts produced in Page Mode.

Figure 29 identifies the format of the logical units of the Structure Chart.

Figure 29 • Structure Chart Format



Report Field Description

These are the Structure Chart Format fields:

Field	Description
FROM PAGE(S)	If this unit is called from other pages, this text lists the calling pages.
pp	This field shows the page number(s).
mmm	This field shows the number of times the unit is called from this parent.
nnn	This field shows the total number of calling units.
xxxxxxx	This field shows the name of the unit.
R	This field designates a recursive perform.
ccc	This field shows the total number of called units appearing on other pages.
TO PAGE	If the called units do not appear on this page, this text lists the page of the called unit.

For COBOL II Release 3 and later programs containing internal subprograms, SmartDoc produces a separate Structure Chart for each subprogram.

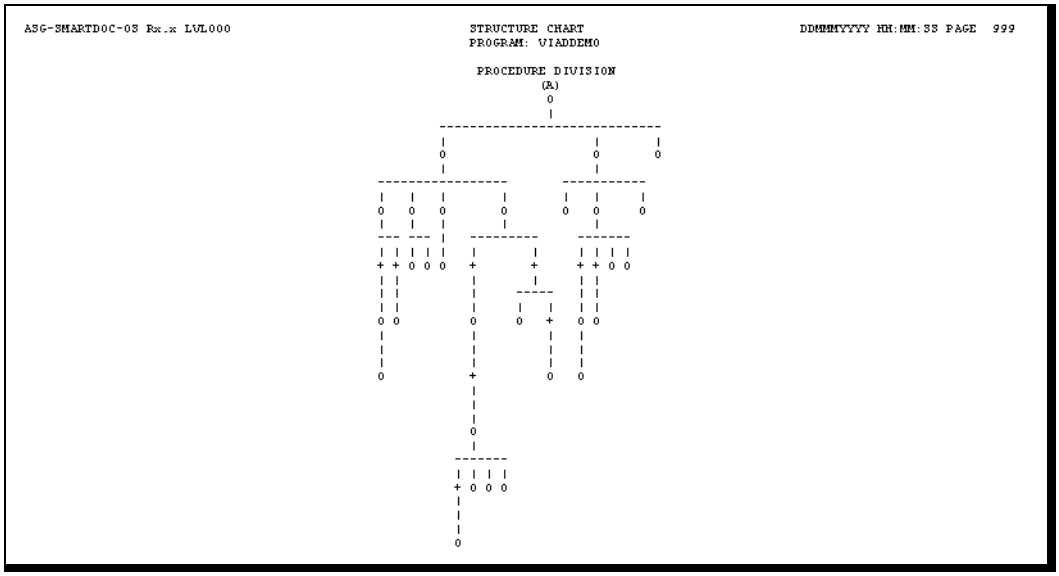
Figure 30 on page 72, Figure 31 on page 73, and Figure 32 on page 74 illustrate Bird's Eye View, Page Mode, Page Mode (Overflow Substructure) and Tile Mode Structure Charts, respectively.

Bird's Eye View

The Bird's Eye View is a Tile Mode Structure Chart that condenses each box to one character. This table lists the characters the Bird's Eye View displays, and their meanings:

Field	Description
O	This field shows a perform unit on the Structure Chart. This can be a perform range, GO TO, CALL, or a conditional statement.
+	This field shows a conditional control transfer by a GO TO, CALL, or a perform range.
V	The unit is a reuse (or recursive) use of a unit previously defined on the diagram.

Figure 30 • Structure Chart - Bird's Eye View



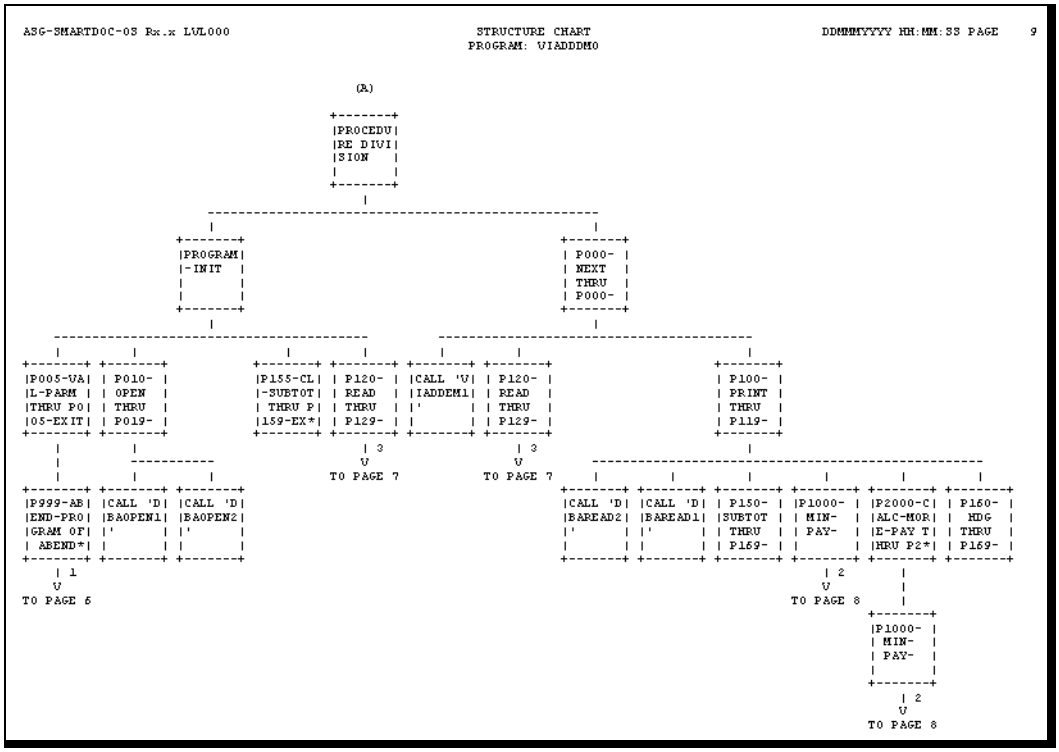
Report Field Description

A. This field shows the Bird's Eye View program structure.

Page Mode

Page Mode produces the Structure Chart in a format designed to be placed in a notebook, as shown in Figure 31.

Figure 31 • Structure Chart - Page Mode



Report Field Description

A. This field show the graphical program structure.

Page Mode with Gotos and Conditionals Options

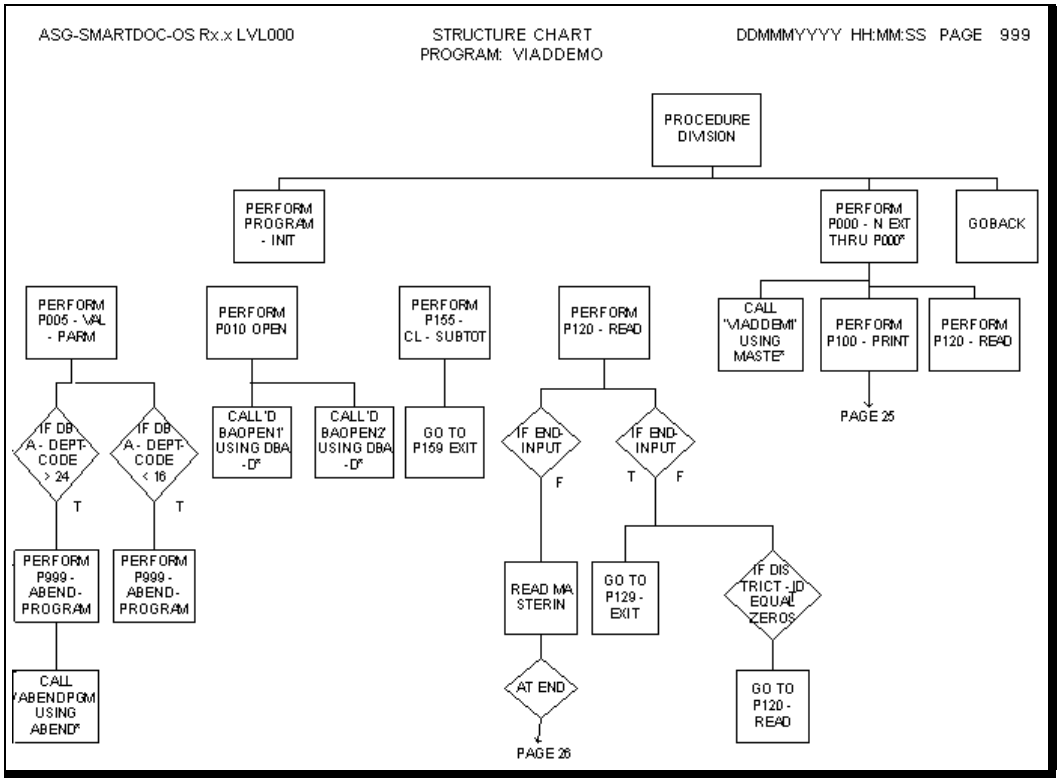
To include the GO TO and ALTER statements in the report, select the Structure Chart Gotos option.

To include the structurally relevant conditional statements associated with the PERFORM, CALL, GO TO, and ALTER statements in the report, select the Structure Chart Conditionals option.

To automatically select the Gotos option, select the Conditionals option.

Figure 32 shows the Page Mode Structure Chart, including the Gotos and Conditionals options.

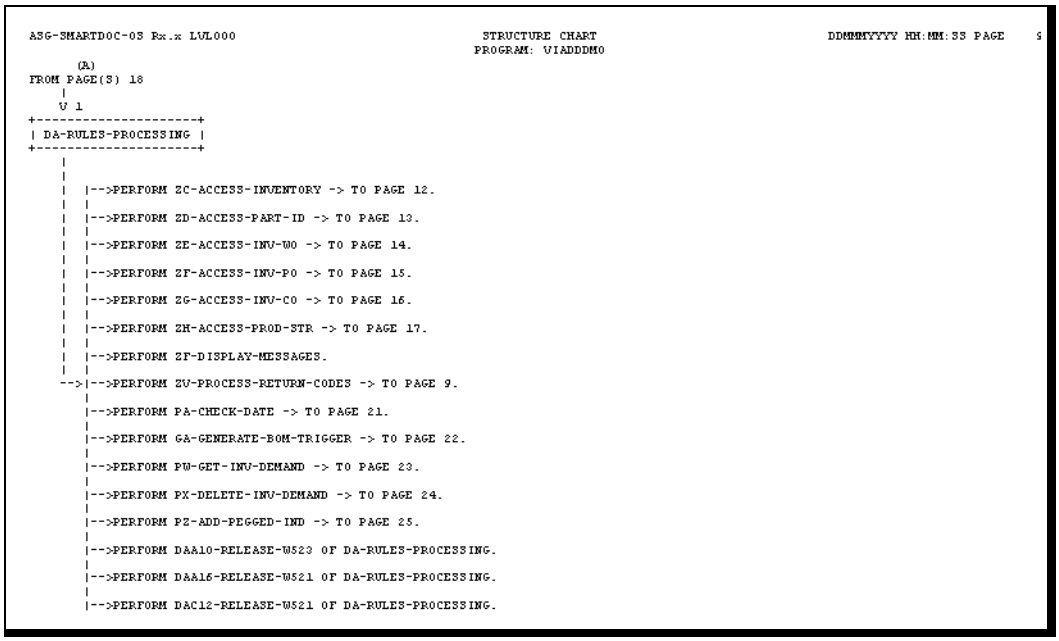
Figure 32 • Structure Chart - Page Mode with GOTOs & Conditionals



Page Mode Overflow Substructure

The Overflow Substructure shown in Figure 33 lists the called paragraphs or sections of a logical unit that do not fit on the appropriate page of the Structure Chart.

Figure 33 • Structure Chart - Page Mode (Overflow Substructure)



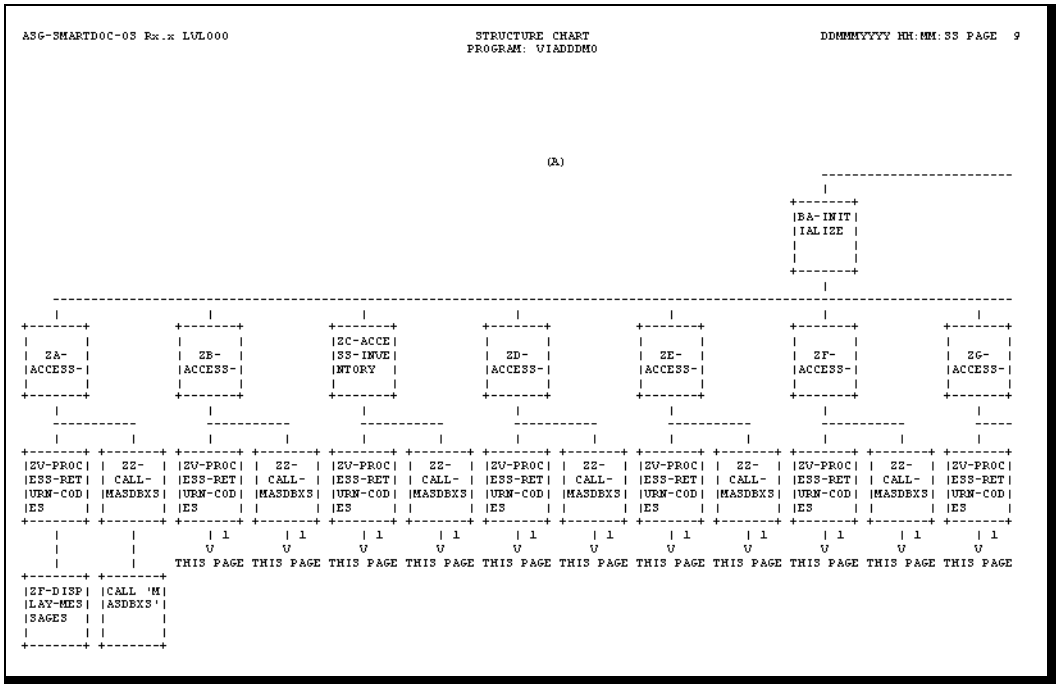
Report Field Description

A. This field show the parent and its subordinates as an overflow substructure, with connecting FROM and TO pages indicated.

Tile Mode

Tile Mode produces the Structure Chart shown in Figure 34 in a format designed to be taped together into one large diagram of the entire program.

Figure 34 • Structure Chart - Tile Mode



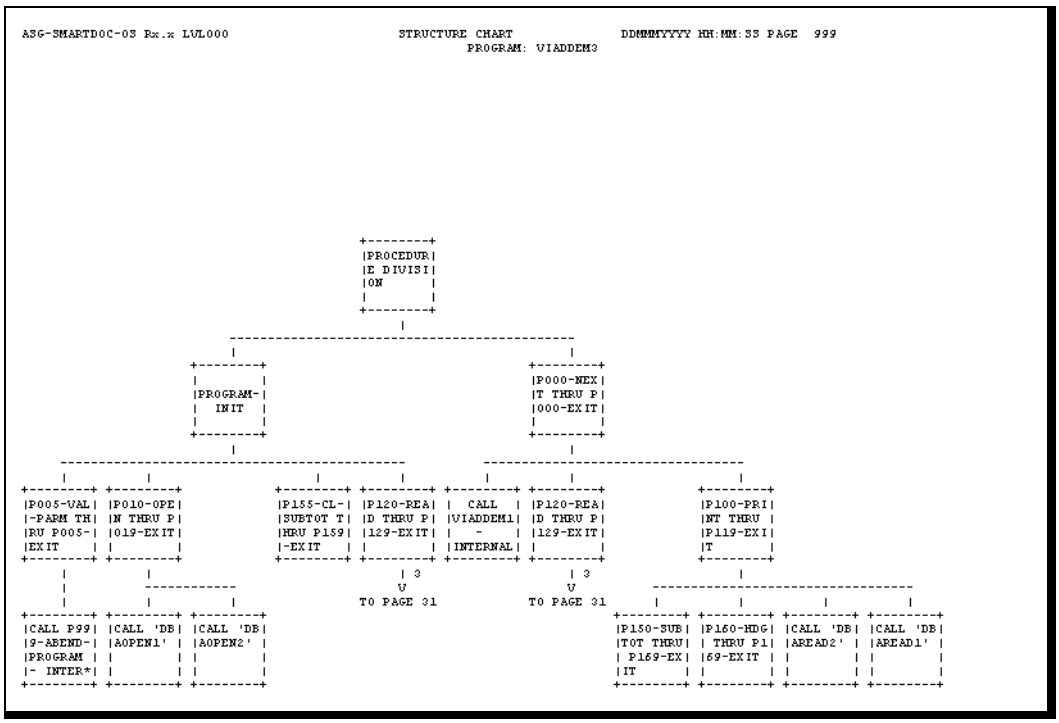
Report Field Description

A. Graphical source code hierarchy

Structure Chart for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, the Structure Chart places the word INTERNAL after the call statement in any appropriate box to identify internally-called subprograms (see Figure 35).

Figure 35 • Structure Chart for COBOL II Release 3



Enhanced Data Cross-Reference Report

The Enhanced Data Cross-Reference report (see Figure 36 on page 79) cross-references all program data entities, and lists all data items and related definitions alphabetically, including special registers, figurative constants, and literals. This report references each data item to where it is either directly or indirectly used or modified. The locations where aliases are defined, modified, or used are also indicated.

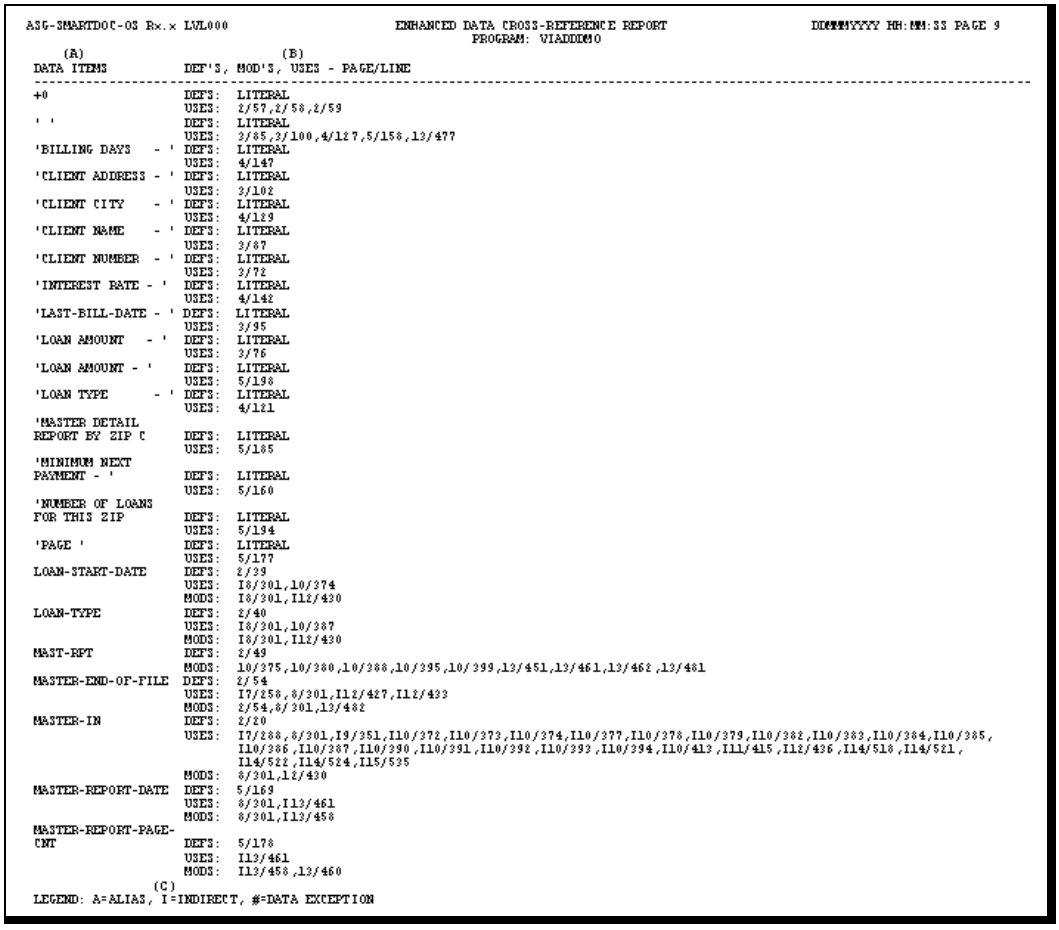
The Enhanced Data Cross-Reference report helps you to answer your data questions. Each data item is referenced to locations where it is either used or modified. The report indicates data that was neither used or modified as DEADDATA. Additionally, each data item is referenced to aliases (i.e., renames or redefines) eliminating the manual searches through the listing to locate where a data field is either used or modified.

A symbol legend prints at the bottom of each report page. These symbols correspond to the reference information shown to the right of each data item. These are the descriptions for each of these symbols:

Field	Description
A	This field references a data item alias definition (either renames or redefines).
I	This field references an indirect modification of the data item value if it is on the line with MODS. This occurs due to a rename, or to a secondary name modification such as a renamed or a redefined data item. The reference is to an indirect use of the data item value if it is on the line with USE. This occurs due to the value being used by a secondary name such as a renamed or a redefined data item.
#	This field refers to a data item that has not been initialized, or to a data item that was modified without either a subsequent reference or use of that value.

For COBOL II Release 3 and later programs containing internal subprograms, SmartDoc produces a separate Enhanced Data Cross-Reference report for each subprogram. The reports identify Global data items as GLOBAL.

Figure 36 • Enhanced Data Cross-Reference Report



```
ASG-SMARTDOC-OS Rx.x LVL000                ENHANCED DATA CROSS-REFERENCE REPORT                DDMMMYYYY HH:MM:SS PAGE 9
                                                PROGRAM: VIADDM40
    (A)                        (B)
DATA ITEMS           DEF'S, MOD'S, USES - PAGE/LINE
-------------------------------------------------------------------------------------------------------------------------------
+0                   DEFS:  LITERAL
                     USES:  2/57,2/58,2/59
' '                  DEFS:  LITERAL
                     USES:  3/85,3/100,4/127,5/158,13/477
'BILLING DAYS   - '  DEFS:  LITERAL
                     USES:  4/147
'CLIENT ADDRESS - '  DEFS:  LITERAL
                     USES:  3/102
'CLIENT CITY    - '  DEFS:  LITERAL
                     USES:  4/129
'CLIENT NAME    - '  DEFS:  LITERAL
                     USES:  3/87
'CLIENT NUMBER  - '  DEFS:  LITERAL
                     USES:  3/72
'INTEREST RATE - '   DEFS:  LITERAL
                     USES:  4/142
'LAST-BILL-DATE - '  DEFS:  LITERAL
                     USES:  3/95
'LOAN AMOUNT   - '   DEFS:  LITERAL
                     USES:  3/76
'LOAN AMOUNT - '     DEFS:  LITERAL
                     USES:  5/138
'LOAN TYPE      - '  DEFS:  LITERAL
                     USES:  4/121
'MASTER DETAIL
REPORT BY ZIP C      DEFS:  LITERAL
                     USES:  5/185
'MINIMUM NEXT
PAYMENT - '          DEFS:  LITERAL
                     USES:  5/160
'NUMBER OF LOANS
FOR THIS ZIP         DEFS:  LITERAL
                     USES:  5/194
'PAGE '              DEFS:  LITERAL
                     USES:  5/177
LOAN-START-DATE      DEFS:  2/29
                     USES:  I8/301,10/374
                     MODS:  I8/301,I12/430
LOAN-TYPE            DEFS:  2/40
                     USES:  I8/301,10/387
                     MODS:  I8/301,I12/430
MAST-RPT             DEFS:  2/49
                     MODS:  10/375,10/380,10/388,10/395,10/399,13/451,13/461,13/462,13/481
MASTER-END-OF-FILE   DEFS:  2/54
                     USES:  I7/258,8/301,I12/427,I12/433
                     MODS:  2/54,8/301,13/482
MASTER-IN            DEFS:  2/20
                     USES:  I7/288,8/301,I9/351,I10/372,I10/373,I10/374,I10/377,I10/378,I10/379,I10/382,I10/383,I10/384,I10/385,
                            I10/386,I10/387,I10/390,I10/391,I10/392,I10/393,I10/394,I10/413,I11/415,I12/426,I14/518,I14/521,
                            I14/522,I14/524,I15/535
                     MODS:  8/301,12/430
MASTER-REPORT-DATE   DEFS:  5/169
                     USES:  8/301,I13/461
                     MODS:  8/301,I13/458
MASTER-REPORT-PAGE-
CNT                  DEFS:  5/178
                     USES:  I13/461
                     MODS:  I13/458,13/460
              (C)
LEGEND: A=ALIAS, I=INDIRECT, #=DATA EXCEPTION
```

Report Field Descriptions

These are the Enhanced Data Cross-Reference report field descriptions:

Field	Description
(A)	This field shows an alphabetical program data item listing, including figurative constants, string constants, and COBOL special registers.
(B)	Each data item is cross-referenced based on whether it is defined, used, or modified. If the data item is a literal, the reference shows `LITERAL` followed by where that literal is used. References are shown as p/n where p is the page number of the Advanced Source Listing and n is the line number on that page where the data item occurs.
(C)	This field shows the legend that indicates the symbol value corresponding to the cross-reference information.

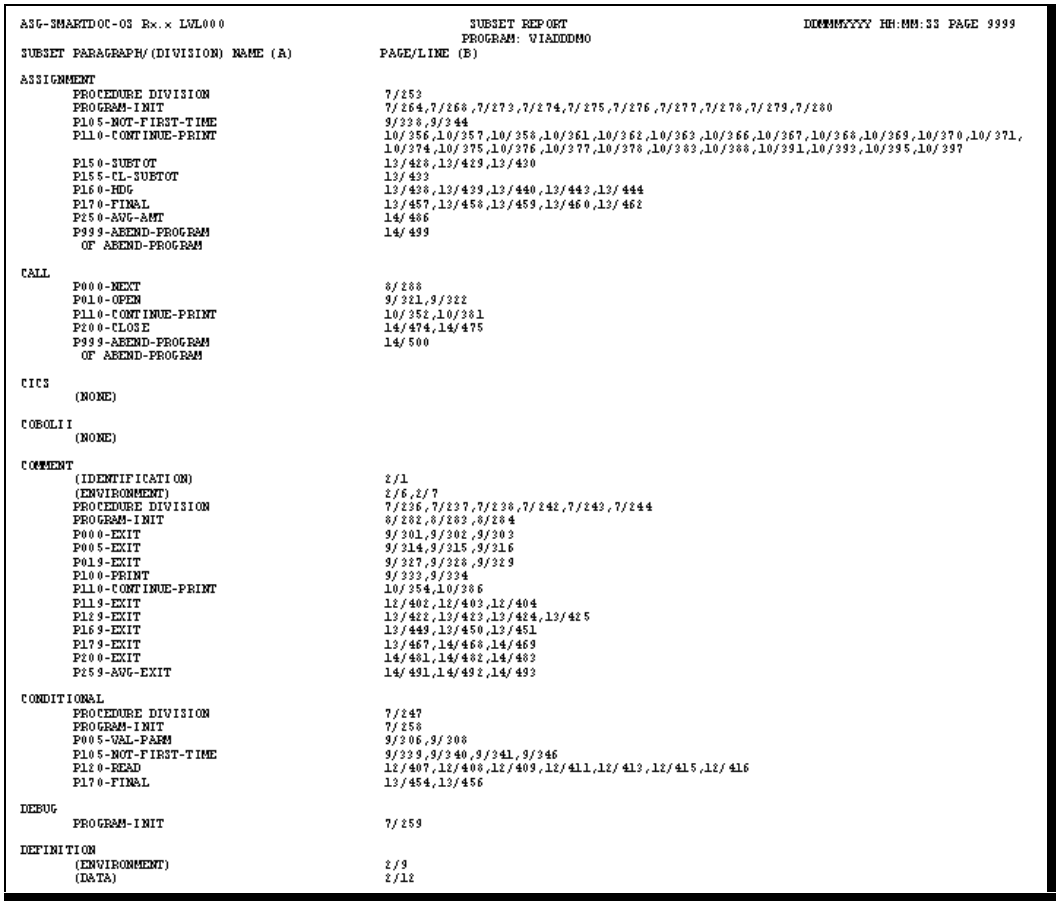
Subset Report

The Subset report (see Figure 37 on page 81) identifies logical groups of subsets. A subset is a group of statements containing similar COBOL verbs. For example, lines that contain READ, WRITE, OPEN, or CLOSE verbs can be referenced as an IO subset. The Subset report shows the paragraph page and line number where a particular subset is located. If there are no source statements for a subset, (NONE) appears for that subset.

For COBOL II Release 3 and later programs, the containing Paragraph/Division is qualified by the containing program.

See the table contained in "Subsets" on page 11 for a list and description of subsets supported by SmartDoc.

Figure 37 • Subset Report



```
ASG-SMARTDOC-03 Rx.x LVL000                    SUBSET REPORT                    DDMMMYYYY HH:MM:SS PAGE 9999
                                               PROGRAM: VIADDDM0
SUBSET PARAGRAPH/(DIVISION) NAME (A)           PAGE/LINE (B)

ASSIGNMENT
       PROCEDURE DIVISION                      7/253
       PROGRAM-INIT                            7/264,7/268,7/273,7/274,7/275,7/276,7/277,7/278,7/279,7/280
       P105-NOT-FIRST-TIME                     9/338,9/344
       P110-CONTINUE-PRINT                     10/356,10/357,10/358,10/361,10/362,10/363,10/366,10/367,10/368,10/369,10/370,10/371,
                                               10/374,10/375,10/376,10/377,10/378,10/383,10/388,10/391,10/393,10/395,10/397
       P150-SUBTOT                             13/428,13/429,13/430
       P155-CL-SUBTOT                          13/433
       P160-HDG                                13/438,13/439,13/440,13/443,13/444
       P170-FINAL                              13/457,13/458,13/459,13/460,13/462
       P250-AVG-AMT                            14/486
       P999-ABEND-PROGRAM                      14/499
        OF ABEND-PROGRAM

CALL
       P000-NEXT                               8/288
       P010-OPEN                               9/321,9/322
       P110-CONTINUE-PRINT                     10/352,10/381
       P200-CLOSE                              14/474,14/475
       P999-ABEND-PROGRAM                      14/500
        OF ABEND-PROGRAM

CICS
       (NONE)

COBOLII
       (NONE)

COMMENT
       (IDENTIFICATION)                        2/1
       (ENVIRONMENT)                           2/6,2/7
       PROCEDURE DIVISION                      7/236,7/237,7/238,7/242,7/243,7/244
       PROGRAM-INIT                            8/282,8/283,8/284
       P000-EXIT                               9/301,9/302,9/303
       P005-EXIT                               9/314,9/315,9/316
       P019-EXIT                               9/327,9/328,9/329
       P100-PRINT                              9/333,9/334
       P110-CONTINUE-PRINT                     10/354,10/386
       P119-EXIT                               12/402,12/403,12/404
       P129-EXIT                               13/422,13/423,13/424,13/425
       P169-EXIT                               13/449,13/450,13/451
       P179-EXIT                               13/467,14/468,14/469
       P200-EXIT                               14/481,14/482,14/483
       P259-AVG-EXIT                           14/491,14/492,14/493

CONDITIONAL
       PROCEDURE DIVISION                      7/247
       PROGRAM-INIT                            7/258
       P005-VAL-PARM                           9/306,9/308
       P105-NOT-FIRST-TIME                     9/339,9/340,9/341,9/346
       P120-READ                               12/407,12/408,12/409,12/411,12/413,12/415,12/416
       P170-FINAL                              13/454,13/456

DEBUG
       PROGRAM-INIT                            7/259

DEFINITION
       (ENVIRONMENT)                           2/9
       (DATA)                                  2/12
```

Report Field Descriptions

These are the Subset report field descriptions:

Field	Description
(A)	This field shows the subset group followed by the paragraphs and/or division names containing statements with that particular subset type. When there are no statements for the indicated subset, (NONE) is shown.
(B)	This field show the reference to the page and line number on the Advanced Source Listing where the statement containing that subset type is located.

Subset Report for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, the Subset report (see Figure 38) lists the program name before each Paragraph/Division name.

Figure 38 • Subset Report for COBOL II Release 3

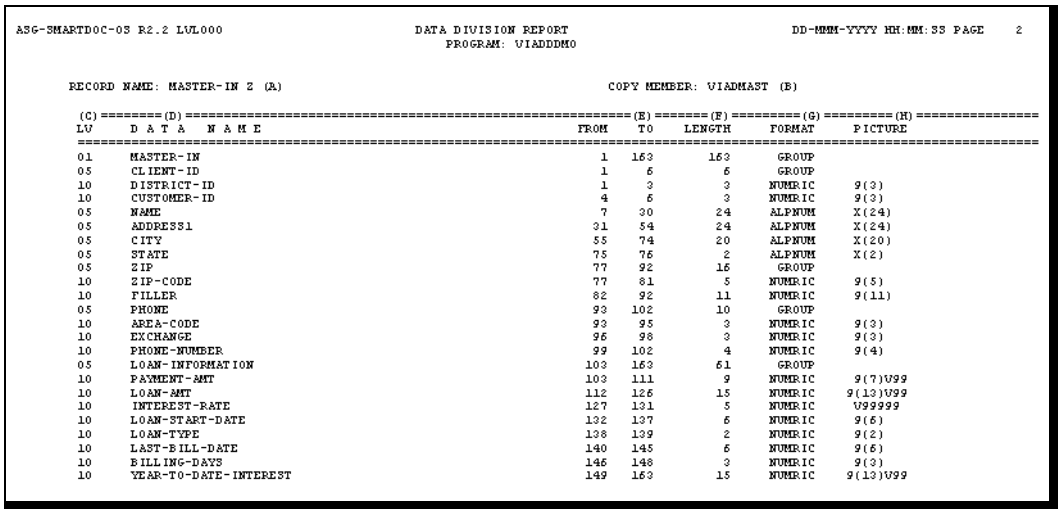
```
ASG-SMARTDOC-OS Rx.x LVL000                          SUBSET REPORT                          DDMMMYYYY
                                                     PROGRAM: VIADDEM3
SUBSET PARAGRAPH/(DIVISION) NAME                PAGE/LINE

ASSIGNMENT
       VIADDEM3/PROCEDURE DIVISION              6/250
       VIADDEM3/PROGRAM-INIT                    6/267,6/268,6/269,6/270
       VIADDEM3/P005-VAL-PARM                   7/297
       VIADDEM3/P105-NOT-FIRST-TIME             7/327,7/333
       VIADDEM3/P110-CONTINUE-PRINT             8/345,8/346,8/347,8/350,8/351,8/352,8/355,8/356,8/357,8/358,8/359,
                                                8/365,8/366,8/367,8/372,8/377,8/380,8/382,8/384,8/386
       VIADDEM3/P150-SUBTOT                     9/417,9/418,9/419
       VIADDEM3/P155-CL-SUBTOT                  9/422
       VIADDEM3/P160-HDG                        9/427,9/428,9/429,9/432,9/433
       VIADDEM3/P170-FINAL                      10/444,10/447,10/448,10/449,10/450,10/452
       VIADDEM3/P250-AVG-AMT                    10/476
       P999-ABEND-PROGRAM/PROCEDURE             12/495
        DIVISION
       VIADDEM1/PROCEDURE DIVISION              15/612,15/613
       VIADDEM1/P100-COMPUTE-TOTALS             15/636,15/637,15/638
       VIADDEM1/P200-PRINT-TOTALS               15/650,15/657,15/658,15/662,16/663,16/667,16/668

CALL
       VIADDEM3/P000-NEXT                       6/278
       VIADDEM3/P005-VAL-PARM                   7/298
       VIADDEM3/P010-OPEN                       7/310,7/311
       VIADDEM3/P110-CONTINUE-PRINT             8/341,8/370
       VIADDEM3/P170-FINAL                      10/445
       VIADDEM3/P200-CLOSE                      10/464,10/465
       P999-ABEND-PROGRAM/PROCEDURE             12/496
        DIVISION

CICS
       (NONE)

COBOLII
       (NONE)

COMMENT
       VIADDEM3/(IDENTIFICATION)                2/1
       VIADDEM3/(ENVIRONMENT)                   2/6,2/7
       VIADDEM3/PROCEDURE DIVISION              6/233,6/234,6/235,6/239,6/240,6/241
       VIADDEM3/PROGRAM-INIT                    6/272,6/273,6/274
       VIADDEM3/P000-EXIT                       7/291,7/292,7/293
       VIADDEM3/P005-EXIT                       7/303,7/304,7/305
       VIADDEM3/P019-EXIT                       7/316,7/317,7/318
       VIADDEM3/P100-PRINT                      7/322,7/323
       VIADDEM3/P110-CONTINUE-PRINT             8/343,8/375
       VIADDEM3/P119-EXIT                       9/391,9/392,9/393
       VIADDEM3/P129-EXIT                       9/411,9/412,9/413,9/414
       VIADDEM3/P169-EXIT                       9/438,9/439,9/440
       VIADDEM3/P179-EXIT                       10/457,10/458,10/459
       VIADDEM3/P200-EXIT                       10/471,10/472,10/473
       VIADDEM3/P259-AVG-EXIT                   11/481
       P999-ABEND-PROGRAM/PROCEDURE             12/490,12/491,12/492,13/502
        DIVISION
       VIADDEM1/P009-END-PROGRAM                15/627,15/628,15/629
       VIADDEM1/P199-EXIT                       15/644,15/645,15/646

CONDITIONAL
       VIADDEM3/PROCEDURE DIVISION              6/244
       VIADDEM3/P005-VAL-PARM                   7/296
       VIADDEM3/P105-NOT-FIRST-TIME             7/328,7/329,7/330,7/335
```

Data Division Report

The Data Division report (see Figure 39) provides detailed information about all 01 structures in the DATA DIVISION of a program. This is the information on the Data Division report:

- COBOL level number
- Dataname
- Length
- Starting and ending position
- Format
- Picture clause definition
- COPY member name where the data item is defined (if applicable)

Figure 39 • Data Division Report



ASG-SMARTDOC-03 R2.2 LVL000 DATA DIVISION REPORT DD-MMM-YYYY HH:MM:SS PAGE 2
PROGRAM: VIADDDM0

RECORD NAME: MASTER-IN 2 (A) COPY MEMBER: VIADMAST (B)

(C) LV	(D) D A T A N A M E	(E) FROM	TO	(F) LENGTH	(G) FORMAT	(H) PICTURE
01	MASTER-IN	1	163	163	GROUP	
05	CLIENT-ID	1	6	6	GROUP	
10	DISTRICT-ID	1	3	3	NUMERIC	9(3)
10	CUSTOMER-ID	4	6	3	NUMERIC	9(3)
05	NAME	7	30	24	ALPNUM	X(24)
05	ADDRESS1	31	54	24	ALPNUM	X(24)
05	CITY	55	74	20	ALPNUM	X(20)
05	STATE	75	76	2	ALPNUM	X(2)
05	ZIP	77	92	16	GROUP	
10	ZIP-CODE	77	81	5	NUMERIC	9(5)
10	FILLER	82	92	11	NUMERIC	9(11)
05	PHONE	93	102	10	GROUP	
10	AREA-CODE	93	95	3	NUMERIC	9(3)
10	EXCHANGE	96	98	3	NUMERIC	9(3)
10	PHONE-NUMBER	99	102	4	NUMERIC	9(4)
05	LOAN-INFORMATION	103	163	61	GROUP	
10	PAYMENT-AMT	103	111	9	NUMERIC	9(7)V99
10	LOAN-AMT	112	126	15	NUMERIC	9(13)V99
10	INTEREST-RATE	127	131	5	NUMERIC	V99999
10	LOAN-START-DATE	132	137	6	NUMERIC	9(6)
10	LOAN-TYPE	138	139	2	NUMERIC	9(2)
10	LAST-BILL-DATE	140	145	6	NUMERIC	9(6)
10	BILLING-DAYS	146	148	3	NUMERIC	9(3)
10	YEAR-TO-DATE-INTEREST	149	163	15	NUMERIC	9(13)V99

Report Field Descriptions

These are the Data Division report field descriptions:

Field	Description
(A)	This field shows the name of the record containing the specified data items.
(B)	This field shows the name of the copy member where the record is defined, if applicable.

Field	Description
(C)	This field shows the COBOL level number for the data item. These three levels are reported: All 01-49 levels except 01s with no subitems All 88 levels All 66 levels
(D)	This field shows the name of the data item reported (displays up to 32 characters).
(E)	This field shows the location of the data item. The starting position is listed in the FROM column; the ending position is listed in the TO column.
(F)	This field shows the length of the specified data item.
(G)	This field shows the format specified for the data item. These formats may appear: GROUP, ALPHA (alphabetic), ALPNUM (alphanumeric), NUMRIC (numeric).
(H)	This field shows the PICTURE definition for the data item, if applicable.

Verb Summary Report

The Verb Summary report (see Figure 40 on page 85) cross-references COBOL verbs to the lines where they are used. You can include a Verb Frequency Table at the end of the Verb Summary report. This report presents how many times each verb is used, and the percentage of its use in relation to the total verb count.

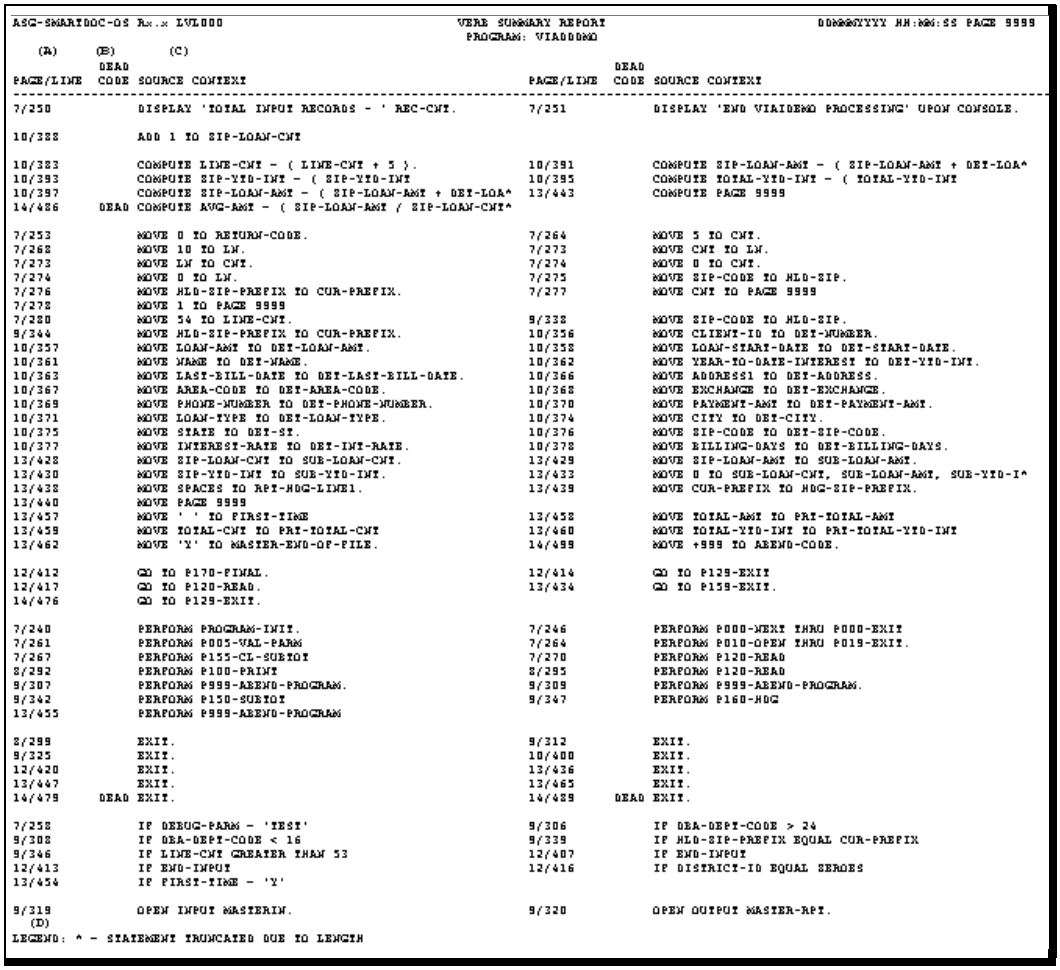
Use this report to determine how much a particular verb, or a set of verbs, is used. The Verb Frequency Table helps you determine if the program is using verbs according to standards. For example, many standards require that you use a single paragraph to read a file, and that the paragraph perform each time the file is read. If the report lists multiple READ verbs, the program might not adhere to the established standards.

The Verb Summary report displays verb usage with context information. Often, a report containing this information is half the size of the source listing. Regardless, a large program can still produce a lengthy Verb Summary report. To limit report size, set the parameters to suppress either the Verb Summary report portion or the Verb Frequency Table portion.

For COBOL II Release 3 and later programs containing internal subprograms, SmartDoc produces a separate Verb Summary report for each subprogram.

Figure 40 on page 85 and Figure 41 on page 86 illustrate the Verb Summary report and the Verb Frequency Table.

Figure 40 • Verb Summary Report



```
ASG-SMARTDOC-OS Rx.x LVL000                    VERB SUMMARY REPORT                    DDmmmYYYY HH:MM:SS PAGE 9999
                                                PROGRAM: VIA000M0

   (A)     (B)       (C)
           DEAD                                                    DEAD
PAGE/LINE  CODE SOURCE CONTEXT                          PAGE/LINE  CODE SOURCE CONTEXT
-------------------------------------------------------------------------------------------------------------------
7/250            DISPLAY 'TOTAL INPUT RECORDS - ' REC-CNT.        7/251            DISPLAY 'END VIAIDEM0 PROCESSING' UPON CONSOLE.

10/382           ADD 1 TO SIP-LOAN-CNT

10/383           COMPUTE LINE-CNT = ( LINE-CNT + 5 ).             10/391           COMPUTE SIP-LOAN-AMT = ( SIP-LOAN-AMT + DET-LOA^
10/393           COMPUTE SIP-YTD-INT = ( SIP-YTD-INT              10/395           COMPUTE TOTAL-YTD-INT = ( TOTAL-YTD-INT
10/397           COMPUTE SIP-LOAN-AMT = ( SIP-LOAN-AMT + DET-LOA^ 13/443           COMPUTE PAGE 9999
14/486     DEAD  COMPUTE AVG-AMT = ( SIP-LOAN-AMT / SIP-LOAN-CNT^

7/253            MOVE 0 TO RETURN-CODE.                           7/264            MOVE 5 TO CNT.
7/262            MOVE 10 TO LN.                                   7/273            MOVE CNT TO LN.
7/273            MOVE LN TO CNT.                                  7/274            MOVE 0 TO CNT.
7/274            MOVE 0 TO LN.                                    7/275            MOVE SIP-CODE TO HLD-SIP.
7/276            MOVE HLD-SIP-PREFIX TO CUR-PREFIX.               7/277            MOVE CNT TO PAGE 9999
7/278            MOVE 1 TO PAGE 9999
7/280            MOVE 54 TO LINE-CNT.                             9/332            MOVE SIP-CODE TO HLD-SIP.
9/344            MOVE HLD-SIP-PREFIX TO CUR-PREFIX.               10/356           MOVE CLIENT-ID TO DET-NUMBER.
10/357           MOVE LOAN-AMT TO DET-LOAN-AMT.                   10/358           MOVE LOAN-START-DATE TO DET-START-DATE.
10/361           MOVE NAME TO DET-NAME.                           10/362           MOVE YEAR-TO-DATE-INTEREST TO DET-YTD-INT.
10/363           MOVE LAST-BILL-DATE TO DET-LAST-BILL-DATE.       10/366           MOVE ADDRESS1 TO DET-ADDRESS.
10/367           MOVE AREA-CODE TO DET-AREA-CODE.                 10/368           MOVE EXCHANGE TO DET-EXCHANGE.
10/369           MOVE PHONE-NUMBER TO DET-PHONE-NUMBER.           10/370           MOVE PAYMENT-AMT TO DET-PAYMENT-AMT.
10/371           MOVE LOAN-TYPE TO DET-LOAN-TYPE.                 10/374           MOVE CITY TO DET-CITY.
10/375           MOVE STATE TO DET-ST.                            10/376           MOVE ZIP-CODE TO DET-ZIP-CODE.
10/377           MOVE INTEREST-RATE TO DET-INT-RATE.              10/378           MOVE BILLING-DAYS TO DET-BILLING-DAYS.
13/428           MOVE ZIP-LOAN-CNT TO SUB-LOAN-CNT.               13/429           MOVE ZIP-LOAN-AMT TO SUB-LOAN-AMT.
13/430           MOVE ZIP-YTD-INT TO SUB-YTD-INT.                 13/433           MOVE 0 TO SUB-LOAN-CNT, SUB-LOAN-AMT, SUB-YTD-I^
13/432           MOVE SPACES TO RPT-HDG-LINE1.                    13/439           MOVE CUR-PREFIX TO HDG-ZIP-PREFIX.
13/440           MOVE PAGE 9999
13/457           MOVE ' ' TO FIRST-TIME                           13/458           MOVE TOTAL-AMT TO PRT-TOTAL-AMT
13/459           MOVE TOTAL-CNT TO PRT-TOTAL-CNT                  13/460           MOVE TOTAL-YTD-INT TO PRT-TOTAL-YTD-INT
13/462           MOVE 'Y' TO MASTER-END-OF-FILE.                  14/488           MOVE +999 TO ABEND-CODE.

12/412           GO TO P170-FINAL.                                12/414           GO TO P125-EXIT
12/417           GO TO P120-READ.                                 13/434           GO TO P155-EXIT.
14/476           GO TO P125-EXIT.

7/240            PERFORM PROGRAM-INIT.                            7/246            PERFORM P000-NEXT THRU P000-EXIT
7/261            PERFORM P005-VAL-PARM                            7/264            PERFORM P010-OPEN THRU P019-EXIT.
7/267            PERFORM P155-CL-SUBTOT                           7/270            PERFORM P120-READ
8/292            PERFORM P100-PRINT                               8/295            PERFORM P120-READ
9/307            PERFORM P999-ABEND-PROGRAM.                      9/309            PERFORM P999-ABEND-PROGRAM.
9/342            PERFORM P150-SUBTOT                              9/347            PERFORM P160-HDG
13/455           PERFORM P999-ABEND-PROGRAM

8/299            EXIT.                                            9/312            EXIT.
9/325            EXIT.                                            10/400           EXIT.
12/420           EXIT.                                            13/436           EXIT.
13/447           EXIT.                                            13/465           EXIT.
14/479     DEAD  EXIT.                                      DEAD  14/489           EXIT.

7/252            IF DEBUG-PARM = 'TEST'                           9/306            IF DBA-DEPT-CODE > 24
9/302            IF DBA-DEPT-CODE < 16                            9/339            IF HLD-SIP-PREFIX EQUAL CUR-PREFIX
9/346            IF LINE-CNT GREATER THAN 53                      12/407           IF END-INPUT
12/413           IF END-INPUT                                     12/416           IF DISTRICT-ID EQUAL ZEROES
13/454           IF FIRST-TIME = 'Y'

9/319            OPEN INPUT MASTERIN.                             9/320            OPEN OUTPUT MASTER-RPT.
   (D)
LEGEND: ^ - STATEMENT TRUNCATED DUE TO LENGTH
```

Report Field Descriptions

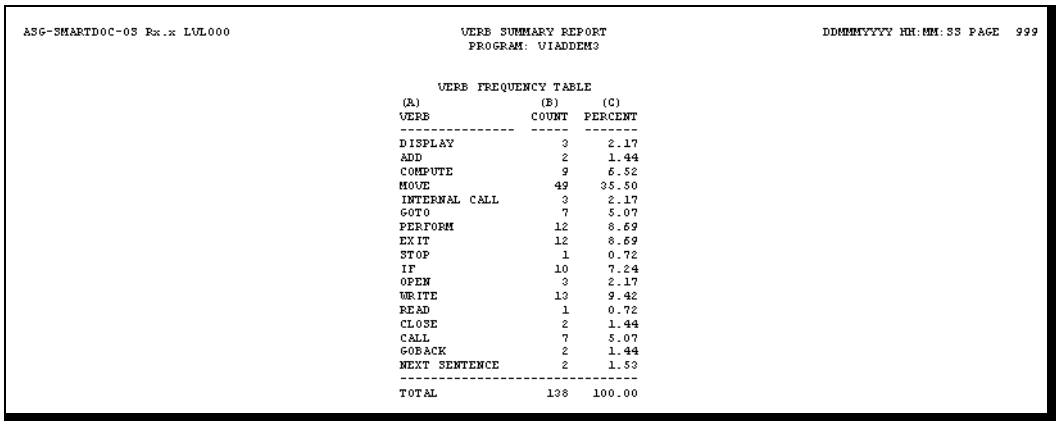
These are the Verb Summary report field descriptions:

Field	Description
(A)	This field references the Advanced Source Listing page and line number where that verb is used.
(B)	This field shows a statement containing dead code, or data contains DEAD in this field.
(C)	This field shows the verb along the source statement context that fits on the report line.
(D)	This field shows the legend that indicates an asterisk (*) is placed at the end of the source line in the context area when the entire statement cannot be shown.

Verb Frequency Table

The Verb Frequency Table (see Figure 41) presents a count of the times each verb is used, and the percentage of its use in relation to the total verb count.

Figure 41 • Verb Summary Report - Verb Frequency Table



```
ASG-SMARTDOC-OS Rx.x LVL000          VERB SUMMARY REPORT                DDMMMYYYY HH:MM:SS PAGE  999
                                      PROGRAM: VIADDEM3

                    VERB FREQUENCY TABLE
(A)                          (B)      (C)
VERB                       COUNT  PERCENT
-------------------------  -----  -------
DISPLAY                        3     2.17
ADD                            2     1.44
COMPUTE                        9     6.52
MOVE                          49    35.50
INTERNAL CALL                  3     2.17
GOTO                           7     5.07
PERFORM                       12     8.69
EXIT                          12     8.69
STOP                           1     0.72
IF                            10     7.24
OPEN                           3     2.17
WRITE                         13     9.42
READ                           1     0.72
CLOSE                          2     1.44
CALL                           7     5.07
GOBACK                         2     1.44
NEXT SENTENCE                  2     1.53
-------------------------------------------
TOTAL                        138   100.00
```

Report Field Descriptions

These are the Verb Frequency table field descriptions:

Field	Description
(A)	Each verb in the program is listed in this field. INTERNAL CALL appears only for COBOL II Release 3 and later programs.
(B)	This field indicates the number of times each verb was used in the program.
(C)	This field shows a verb usage percentage in relation to the total verb count.

Copy Statement Report

The Copy Statement report shown in Figure 42 on page 87 lists COPY directives used within the program, including these:

- COBOL COPY statement
- Librarian -INC statement
- Panvalet ++INCLUDE statement
- COPY IDMS statement

- COPYDD statement
- EXEC SQL INCLUDE statement
- User-specified source managers

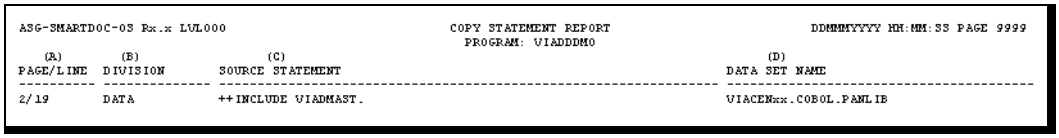
Use this report to recognize program portions that may be contained in external datasets. This report presents information in these four ways:

- The actual statement
- Page and line number containing the statement
- Division where the statement occurs
- Name of the dataset containing the copied or the included member

For COBOL II Release 3 and later programs, the division containing the statement is qualified by the containing program.

The copy or the include member expands in the Advanced Source Listing.

Figure 42 • Copy Statement Report



```
ASG-SMARTDOC-03 Rx.x LVL000                    COPY STATEMENT REPORT                    DDMMMYYYYY HH:MM:SS PAGE 9999
                                                 PROGRAM: VIADDDM0
    (A)        (B)              (C)                                                          (D)
PAGE/LINE  DIVISION       SOURCE STATEMENT                                          DATA SET NAME
---------- -------------- ------------------------------------------------------- ----------------------------------------
2/19       DATA           ++INCLUDE VIADMAST.                                     VIACENxx.COBOL.PANLIB
```

Report Field Descriptions

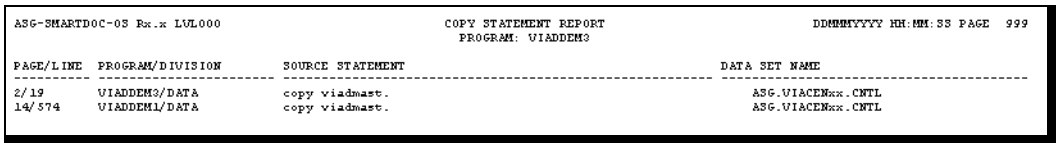
These are the Copy Statement report field descriptions:

Field	Description
(A)	This field shows the Advanced Source Listing page and line number where the statement occurs.
(B)	This field shows the DIVISION where the statement occurs.
(C)	This field shows the source statement.
(D)	This field shows the dataset name from where the member is copied.

Copy Statement Report for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, the Copy Statement report lists (see Figure 43) the program name before each Division name.

Figure 43 • Copy Statement Report for COBOL II Release 3



```
ASG-SMARTDOC-OS Rx.x LVL000                    COPY STATEMENT REPORT                    DDMMMYYYY HH:MM:SS PAGE  999
                                                 PROGRAM: VIADDEM3

PAGE/LINE  PROGRAM/DIVISION        SOURCE STATEMENT                                    DATA SET NAME
---------- ----------------------- --------------------------------------------------- ------------------------------------
2/19       VIADDEM3/DATA           copy viadmast.                                          ASG.VIACENxx.CNTL
14/574     VIADDEM1/DATA           copy viadmast.                                          ASG.VIACENxx.CNTL
```

Call Statement Report

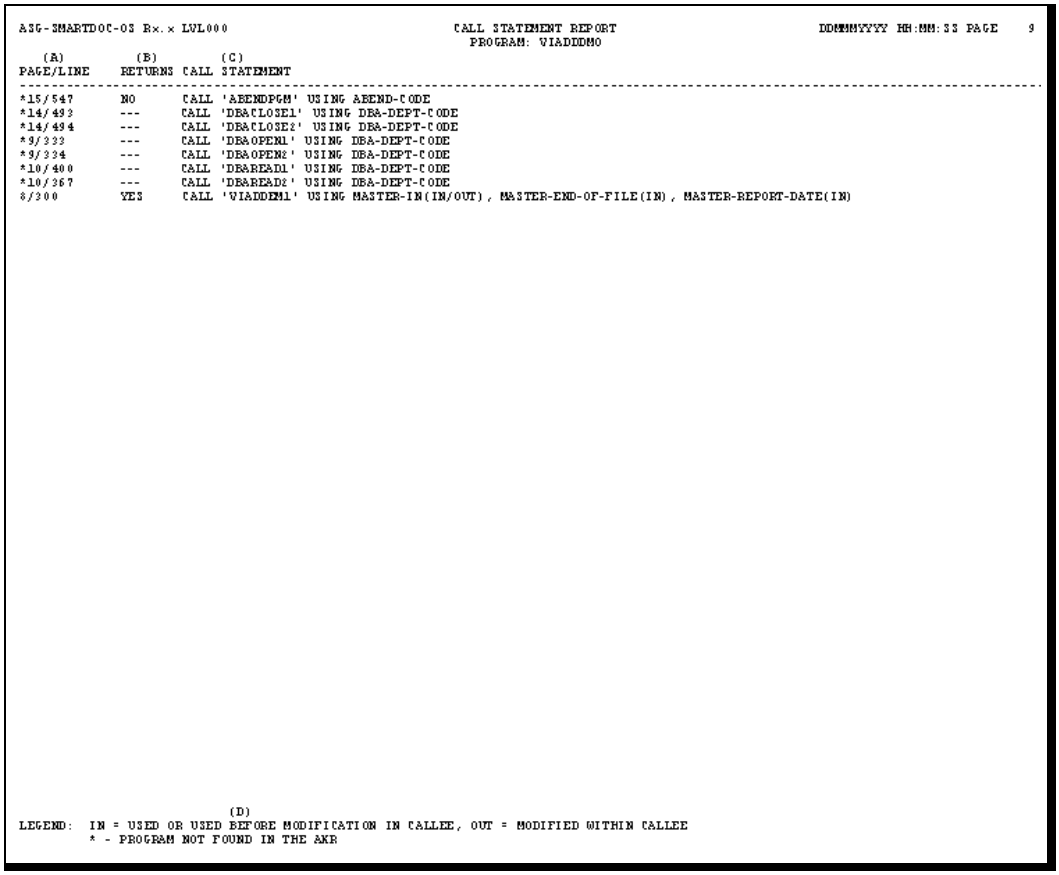
The Call Statement report (see Figure 44 on page 89) lists the source code lines that use the COBOL CALL statement. Use this report to determine how control passes to other programs. Information on this report is presented in these three ways:

- The actual CALL statement
- Page and line number containing the statement
- Arguments in the USING clause

If the called program exists in the AKR, the Call Statement report shows if it is a returning or a non-returning (RET or NORET). If you mark a program or a group of programs as either returning or non-returning during installation, the report marks them as either RET or NORET. Called programs that do not exist in the AKR are shown with an asterisk (*) preceding the page and line number. This report also indicates whether each item in the USING clause is an IN or an OUT parameter. IN parameters are used before any modifications occur to the passed value of the parameter. OUT parameters are modified by the called program. CALL statements that cannot be executed display as DEADCODE.

For COBOL II Release 3 and later programs, the Call Statement marks CALLs by the type of call (either internal or external).

Figure 44 • Call Statement Report



```
ASG-SMARTDOC-03 Rx.x LVL000                          CALL STATEMENT REPORT                          DDMMMYYYY HH:MM:SS PAGE    9
                                                       PROGRAM: VIADDDM0
    (A)          (B)        (C)
PAGE/LINE     RETURNS CALL STATEMENT
-------------------------------------------------------------------------------------------------------------------------------
*15/547       NO       CALL 'ABENDPGM' USING ABEND-CODE
*14/493       ---      CALL 'DBACLOSE1' USING DBA-DEPT-CODE
*14/494       ---      CALL 'DBACLOSE2' USING DBA-DEPT-CODE
*9/333        ---      CALL 'DBAOPEN1' USING DBA-DEPT-CODE
*9/334        ---      CALL 'DBAOPEN2' USING DBA-DEPT-CODE
*10/400       ---      CALL 'DBAREAD1' USING DBA-DEPT-CODE
*10/367       ---      CALL 'DBAREAD2' USING DBA-DEPT-CODE
8/300         YES      CALL 'VIADDEM1' USING MASTER-IN(IN/OUT), MASTER-END-OF-FILE(IN), MASTER-REPORT-DATE(IN)









                                  (D)
LEGEND:  IN = USED OR USED BEFORE MODIFICATION IN CALLEE, OUT = MODIFIED WITHIN CALLEE
         * - PROGRAM NOT FOUND IN THE AKR
```

Report Field Descriptions

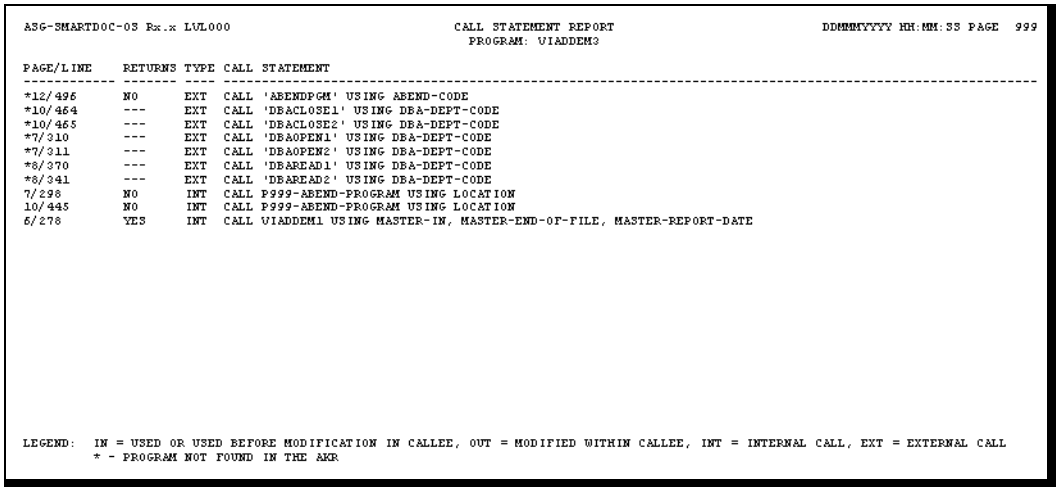
These are the Call Statement report field descriptions:

Field	Description
(A)	This field shows the page and line number where the CALL statement occurs in the Advanced Source Listing. An asterisk (*) preceding a page and line number indicates the CALLed program does not exist in the AKR.
(B)	A `Yes` displayed in this field indicates the CALLed program returns. SmartDoc determines if a program is either returning or non-returning based on the RET and NORET parameters.
(C)	This field shows the source statement containing the CALL verb.
(D)	This field shows the legend of symbols used to indicate whether an item in the USING clause is an IN or an OUT parameter.

Call Statement Report for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, SmartDoc adds the TYPE field to the Call Statement report (see Figure 45). The TYPE field identifies each call as either an INT (internal call) or an EXT (external call).

Figure 45 • Call Statement Report for COBOL II Release 3



```
ASG-SMARTDOC-OS Rx.x LVL000                    CALL STATEMENT REPORT                    DDMMMYYYY HH:MM:SS PAGE  999
                                                 PROGRAM: VIADDEM3

PAGE/LINE    RETURNS TYPE CALL STATEMENT
------------ ------- ---- ------------------------------------------------------------------------------------------------
*12/496      NO      EXT  CALL 'ABENDPGM' USING ABEND-CODE
*10/464      ---     EXT  CALL 'DBACL03E1' USING DBA-DEPT-CODE
*10/465      ---     EXT  CALL 'DBACL03E2' USING DBA-DEPT-CODE
*7/310       ---     EXT  CALL 'DBAOPEN1' USING DBA-DEPT-CODE
*7/311       ---     EXT  CALL 'DBAOPEN2' USING DBA-DEPT-CODE
*8/370       ---     EXT  CALL 'DBAREAD1' USING DBA-DEPT-CODE
*8/341       ---     EXT  CALL 'DBAREAD2' USING DBA-DEPT-CODE
7/298        NO      INT  CALL P999-ABEND-PROGRAM USING LOCATION
10/445       NO      INT  CALL P999-ABEND-PROGRAM USING LOCATION
6/278        YES     INT  CALL VIADDEM1 USING MASTER-IN, MASTER-END-OF-FILE, MASTER-REPORT-DATE




LEGEND:   IN = USED OR USED BEFORE MODIFICATION IN CALLEE, OUT = MODIFIED WITHIN CALLEE, INT = INTERNAL CALL, EXT = EXTERNAL CALL
          * - PROGRAM NOT FOUND IN THE AKR
```

Paragraph Cross-Reference Report

The Paragraph Cross-Reference report (see Figure 46 on page 91) lists the paragraphs and the sections in the program and shows how they execute. Use this report to determine how a paragraph gets executed, and the paragraphs it executes. Control is passed to another paragraph based on these seven factors:

- GO TO statements
- PERFORM statements
- ALTER statements
- Fall through logic
- PERFORM and internal CALL return processing
- ON CONDITION statements
- Internal CALL information (COBOL II Release 3 only)

Paragraphs that either transfer control or reference another paragraph are listed by page and line number. Unreferenced paragraphs are shown as DEADCODE.

For COBOL II Release 3 and later programs containing subprograms, SmartDoc produces a separate Paragraph Cross-Reference report for each subprogram. The Paragraph Cross-Reference report also includes Internal CALL information.

Figure 46 • Paragraph Cross-Reference Report

```
ASG-SMARTDOC-OS Rx.x LVL000                    PARAGRAPH CROSS-REFERENCE                         DDMMMYYYY HH:MM:SS PAGE 9
                                                    PROGRAM: VIADDMO
              (A)                        (B)           (C)                        (D)            (E)
 TARGET PARAGRAPH/SECT. NAME (PAGE/LINE) HOW      COMES FROM (PAGE/LINE)          HOW      GOES TO (PAGE/LINE)
----------------------------------------------------------------------------------------------------------------------
 ABEND-PROGRAM (15/544)                  FALL P2000-EXIT (15/538)                 FALL P999-ABEND-PROGRAM OF
                                                                                       ABEND-PROGRAM (15/546)

 PROCEDURE DIVISION (7/245)              RETN P000-EXIT (8/312)                   PERF P000-NEXT (8/299)
                                         RETN PROGRAM-INIT (7/293)                PERF PROGRAM-INIT (7/267)

 PROGRAM-INIT (7/267)                    RETN P129-EXIT (12/440)                  PERF P155-CL-SUBTOT (13/452)
                                         RETN P159-EXIT (13/456)                  PERF P120-READ (12/426)
                                         RETN P019-EXIT (9/338)                   PERF P010-OPEN (9/331)
                                         RETN P005-EXIT (9/325)                   PERF P005-VAL-PARM (9/318)
                                         PERF PROCEDURE DIVISION (7/251)          RETN PROCEDURE DIVISION (7/257)

 P000-EXIT (8/311)                       FALL P000-NEXT (8/308)                   RETN PROCEDURE DIVISION (7/258)
                                         RETN P129-EXIT (12/440)

 P000-NEXT (8/299)                       RETN P119-EXIT (11/420)                  PERF P120-READ (12/426)
                                         PERF PROCEDURE DIVISION (7/257)          PERF P100-PRINT (9/344)
                                                                                  FALL P000-EXIT (8/311)

 P005-EXIT (9/324)                       FALL P005-VAL-PARM (9/322)               RETN PROGRAM-INIT (7/275)

 P005-VAL-PARM (9/318)                   PERF PROGRAM-INIT (7/272)                PERF P999-ABEND-PROGRAM OF
                                                                                       ABEND-PROGRAM (15/546)
                                                                                  PERF P999-ABEND-PROGRAM OF
                                                                                       ABEND-PROGRAM (15/546)
                                                                                  FALL P005-EXIT (9/324)

 P010-OPEN (9/331)                       PERF PROGRAM-INIT (7/275)                FALL P019-EXIT (9/337)

 P019-EXIT (9/337)                       FALL P010-OPEN (9/335)                   RETN PROGRAM-INIT (7/278)

 P100-PRINT (9/344)                      PERF P000-NEXT (8/305)                   FALL P105-NOT-FIRST-TIME (9/349)

 P1000-EXIT (14/526)                     GOTO P1000-MIN-PAY-DUE (14/519)          FALL P2000-CALC-MORE-PAY (14/532)

 P1000-MIN-PAY-DUE (14/515)              PERF P2000-CALC-MORE-PAY (15/534)        RETN P2000-CALC-MORE-PAY (15/535)
                                         PERF P105-NOT-FIRST-TIME (9/357)         GOTO P1000-EXIT (14/526)
                                                                                  RETN P105-NOT-FIRST-TIME (9/358)

 P105-NOT-FIRST-TIME (9/349)             RETN P2000-EXIT (15/538)                 PERF P2000-CALC-MORE-PAY (14/532)
                                         RETN P1000-MIN-PAY-DUE (14/524)          PERF P1000-MIN-PAY-DUE (14/515)
                                         RETN P169-EXIT (13/467)                  PERF P160-HDG (13/457)
                                         FALL P100-PRINT (9/344)                  PERF P150-SUBTOT (13/447)
                                                                                  FALL P110-CONTINUE-PRINT (10/366)

 P110-CONTINUE-PRINT (10/366)            FALL P105-NOT-FIRST-TIME (9/363)         FALL P119-EXIT (11/419)
                                         RETN P169-EXIT (13/467)

 P119-EXIT (11/419)                      FALL P110-CONTINUE-PRINT (11/417)        RETN P000-NEXT (8/308)

 P120-READ (12/426)                      GOTO P120-READ (12/437)                  GOTO P170-FINAL (13/473)
                                         PERF P000-NEXT (8/308)                   GOTO P129-EXIT (12/439)
                                         PERF PROGRAM-INIT (7/281)                FALL P129-EXIT (12/439)
                                                                                  GOTO P120-READ (12/426)

 P129-EXIT (12/439)                      GOTO P200-CLOSE (14/496)                 RETN P000-EXIT (8/311)

   (F)
 LEGEND:  FALL = FALLTHRU, PERF = PERFORM, COND = ON CONDITION, RETN = RETURN, GOTO = GO TO
```

Report Field Descriptions

These are the Paragraph Cross-Reference report fields:

Field	Description
(A)	This field shows each paragraph or section name in the program is listed with the page and line number where it occurs in the Advanced Source Listing.
(B)	This field indicates how control was passed to the paragraph (PERFORM, FALLTHRU, GOTO, etc.).
(C)	This field shows the paragraph or section where control passed, along with the page and line number where it occurs in the Advanced Source Listing.
(D)	This field indicates how control is being passed to another paragraph (PERFORM, FALLTHRU, GOTO, etc.).

Field	Description
(E)	This field shows the paragraph or section where control is being passed.
(F)	This field shows the legend that describes how control is either passed to or from the paragraph.

Paragraph Cross-Reference for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, the Paragraph Cross-Reference report (see Figure 47) qualifies the target paragraph/section name with the program name.

Figure 47 • Paragraph Cross-Reference for COBOL II Release 3

```
ASG-SMARTDOC-OS Rx.x LVL000                    PARAGRAPH CROSS-REFERENCE                        DDMMMYYYY HH:MM:SS PAGE  999
                                                    PROGRAM: VIA0DEM2

TARGET PROG/PARA/SECT. NAME (PAGE/LINE)  HOW  COMES FROM (PAGE/LINE)                  HOW  GOES TO (PAGE/LINE)
-------------------------------------------------------------------------------------------------------------------------------
PROCEDURE DIVISION (6/231)               RETN P000-EXIT (7/289)                      PERF P000-NEXT (6/276)
                                         RETN PROGRAM-INIT (6/270)                   PERF PROGRAM-INIT (6/253)
                                              PROGRAM ENTRY                               PROGRAM EXIT

PROCEDURE DIVISION OF                    CALL P170-FINAL (10/445)
9999-ABEND-PROGRAM (12/489)              CALL P005-VAL-PARM (7/295)

PROCEDURE DIVISION OF VIA0DEM1 (15/606)  CALL P000-NEXT (6/278)

PROGRAM-INIT (6/253)                     RETN P129-EXIT (9/409)                      PERF P155-CL-SUBTOT (9/421)
                                         RETN P159-EXIT (9/425)                      PERF P120-READ (9/395)
                                         RETN P019-EXIT (7/314)                      PERF P010-OPEN (7/307)
                                         RETN P005-EXIT (7/301)                      PERF P005-VAL-PARM (7/295)
                                         PERF PROCEDURE DIVISION (6/237)             RETN PROCEDURE DIVISION (6/243)

P000-EXIT (7/288)                        FALL P000-NEXT (7/285)                      RETN PROCEDURE DIVISION (6/244)
                                         RETN P129-EXIT (9/409)

P000-NEXT (6/276)                        RETN P119-EXIT (8/388)                      PERF P120-READ (9/395)
                                         RETN P009-END-PROGRAM OF VIA0DEM1 (15/624)  PERF P100-PRINT (7/320)
                                         PERF PROCEDURE DIVISION (6/243)             FALL P000-EXIT (7/288)
                                                                                     CALL PROCEDURE DIVISION OF VIA0DEM1 (15/606)

P005-EXIT (7/300)                        FALL P005-VAL-PARM (7/295)                  RETN PROGRAM-INIT (6/258)

P005-VAL-PARM (7/295)                    PERF PROGRAM-INIT (6/255)                   FALL P005-EXIT (7/300)
                                                                                     CALL PROCEDURE DIVISION
                                                                                          OF 9999-ABEND-PROGRAM (12/489)

P009-END-PROGRAM OF VIA0DEM1 (0/-1)                                                  RETN P000-NEXT (6/282)

P010-OPEN (7/307)                        PERF PROGRAM-INIT (6/258)                   FALL P019-EXIT (7/313)

P019-EXIT (7/313)                        FALL P010-OPEN (7/311)                      RETN PROGRAM-INIT (6/261)

P100-PRINT (7/320)                       PERF P000-NEXT (6/282)                      FALL P105-NOT-FIRST-TIME (7/325)

P105-NOT-FIRST-TIME (7/325)              RETN P169-EXIT (9/436)                      PERF P160-HDG (9/426)
                                         FALL P100-PRINT (7/320)                     PERF P150-SUBTOT (9/416)
                                                                                     FALL P110-CONTINUE-PRINT (7/339)

P110-CONTINUE-PRINT (7/339)              FALL P105-NOT-FIRST-TIME (7/336)            FALL P119-EXIT (8/388)
                                         RETN P169-EXIT (9/436)

P119-EXIT (8/388)                        FALL P110-CONTINUE-PRINT (8/386)            RETN P000-NEXT (7/285)

P120-READ (9/395)                        GOTO P120-READ (9/406)                      GOTO P170-FINAL (10/442)
                                         PERF P000-NEXT (7/285)                      GOTO P129-EXIT (9/408)
                                         PERF PROGRAM-INIT (6/264)                   FALL P129-EXIT (9/408)
                                                                                     GOTO P120-READ (9/395)

P129-EXIT (9/408)                        GOTO P200-CLOSE (10/466)                    RETN P000-EXIT (7/288)
                                         GOTO P120-READ (9/403)                      RETN PROGRAM-INIT (6/267)
                                         FALL P120-READ (9/406)

P150-SUBTOT (9/416)                      PERF P105-NOT-FIRST-TIME (7/331)            FALL P155-CL-SUBTOT (9/421)


LEGEND:  FALL - FALLTHRU, PERF - PERFORM, COND - ON CONDITION, RETN - RETURN, GOTO - GO TO, CALL - INTERNAL CALL
```

Perform Range Usage and Interface Report

The Perform Range Usage and Interface report lists the paragraphs and sections that invoke a perform range. A perform range consists of all the code executes by following a PERFORM statement. The report shows each perform range along with a cross-reference to the paragraph(s) that invokes it.

This report also shows the INPUTS and OUTPUTS of each perform range. INPUTS (shown on report as `IN`) refers to the data items modified before entry into the performed paragraph and used within the perform range. OUTPUTS (shown on the report as `OUT`) refers to the data items modified within the perform range, then subsequently used elsewhere in the program. USE and MOD are data items referenced inside the perform range.

Paragraphs that call for perform range execution are identified as well. This report provides understanding of how perform ranges are used, and how data in perform ranges affects other program areas.

Use the Perform Range Usage and Interface report as a re-engineering tool to determine how to turn a perform range into a separately callable program.

If an Extended SmartDoc analysis is not performed, INPUTS and OUTPUTS are unavailable and only USE and MOD information is shown.

Perform Range Usage and Interface for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs containing subprograms, SmartDoc produces a separate Perform Range Usage and Interface report for each subprogram.

These figures illustrate the Perform Range Usage and Interface report:

- Figure 48 on page 94 shows the Perform Range Usage and Interface report produced by a DX or a DA type of analysis.
- Figure 49 on page 95 shows the Perform Range Usage and Interface report produced by a DS or a DC type of analysis.

Extended Analysis (DX or DA)

Figure 48 shows the Perform Range Usage and Interface report produced by an extended analysis, either with a compile (DA) or without a compile (DX).

Figure 48 • Perform Range Usage & Interface Report (DX or DA Analysis)

```
ASG-SMARTDOC-OS Rx.x LVL000                     PERFORM RANGE USAGE AND INTERFACE REPORT                 DDMMMYYYY HH:MM:SS PAGE 9
                                                             PROGRAM: VIA000MO
             (A)                        (B)                                            (C)
 PERFORM RANGE NAME (PAGE/LINE)     PERFORMED BY                                   INPUTS, OUTPUTS, USES AND MODS (PAGE/LINE)
-------------------------------------------------------------------------------------------------------------------------------------
 PROGRAM-INIT (7/267)               PROCEDURE DIVISION (7/251)                     IN: ZIP-CODE (2/29)
                                                                                   OUT: LINE-CNT (2/56), PAGE-CNT (2/57),
                                                                                   OUT: CUR-PREFIX (5/154)
                                                                                   USE: ZIP-CODE (2/29), HLD-ZIP-PREFIX (6/223),
                                                                                   USE: LN (6/235), CNT (6/235),
                                                                                   USE: DEBUG-PARM (6/241)
                                                                                   MOD: LINE-CNT (2/56), PAGE-CNT (2/57),
                                                                                   MOD: CUR-PREFIX (5/154), HLD-ZIP (6/222),
                                                                                   MOD: LN (6/235), CNT (6/235)

 P000-NEXT THRU P000-EXIT (8/299)   PROCEDURE DIVISION (7/257)                     IN: MASTER-IN (2/20),
                                                                                   IN: MASTER-END-OF-FILE (2/54),
                                                                                   IN: MASTER-REPORT-DATE (5/169)
                                                                                   OUT: MASTER-IN (2/20),
                                                                                   OUT: MASTER-END-OF-FILE (2/54)

 P005-VAL-PARM THRU P005-EXIT       PROGRAM-INIT (7/272)                           USE: DBA-DEPT-CODE (6/242)
 (9/318)

 P010-OPEN THRU P019-EXIT (9/331)   PROGRAM-INIT (7/275)                           OUT: MASTERIN (2/14), DBA-DEPT-CODE (6/242)
                                                                                   USE: MASTERIN (2/14), MASTER-RPT (2/45)
                                                                                   MOD: MASTERIN (2/14), MASTER-RPT (2/45)

 P100-PRINT THRU P119-EXIT (9/344)  P000-NEXT (8/305)                              IN: CLIENT-ID (2/21), NAME (2/24),
                                                                                   IN: ADDRESS1 (2/25), CITY (2/26), STATE (2/27)
                                                                                   IN: ZIP-CODE (2/29), AREA-CODE (2/32),
                                                                                   IN: EXCHANGE (2/33), PHONE-NUMBER (2/34),
                                                                                   IN: PAYMENT-AMT (2/36), LOAN-AMT (2/37),
                                                                                   IN: INTEREST-RATE (2/38),
                                                                                   IN: LOAN-START-DATE (2/39), LOAN-TYPE (2/40),
                                                                                   IN: LAST-BILL-DATE (2/41), BILLING-DAYS (2/42)
                                                                                   IN: YEAR-TO-DATE-INTEREST (2/43),
                                                                                   IN: LINE-CNT (2/56), MIN-PAY-AMT (3/61),
                                                                                   IN: MORE-PAY (3/62), DETAIL-LINE1 (3/69),
                                                                                   IN: DETAIL-LINE2 (3/84), DETAIL-LINE3 (3/99),
                                                                                   IN: DETAIL-LINE4 (4/126), CUR-PREFIX (5/154),
                                                                                   IN: DETAIL-LINE5 (5/157),
                                                                                   IN: DBA-DEPT-CODE (6/242)
                                                                                   OUT: DET-ZIP-CODE (4/138), CUR-PREFIX (5/154),
                                                                                   OUT: ZIP-LOAN-CNT (6/227), ZIP-LOAN-AMT (6/228
                                                                                   OUT: ZIP-YTD-INT (6/229), TOTAL-CNT (6/232),
                                                                                   OUT: TOTAL-YTD-INT (6/234),
                                                                                   OUT: DBA-DEPT-CODE (6/242)
                                                                                   USE: CLIENT-ID (2/21), NAME (2/24),
                                                                                   USE: ADDRESS1 (2/25), CITY (2/26), STATE (2/27
                                                                                   USE: ZIP-CODE (2/29), AREA-CODE (2/32),
                                                                                   USE: EXCHANGE (2/33), PHONE-NUMBER (2/34),
                                                                                   USE: PAYMENT-AMT (2/36), LOAN-AMT (2/37),
                                                                                   USE: INTEREST-RATE (2/38),
                                                                                   USE: LOAN-START-DATE (2/39), LOAN-TYPE (2/40),
                                                                                   USE: LAST-BILL-DATE (2/41), BILLING-DAYS (2/42
                                                                                   USE: YEAR-TO-DATE-INTEREST (2/43),
                                                                                   USE: LINE-CNT (2/56), MIN-PAY-AMT (3/61),
                                                                                   USE: MORE-PAY (3/62), DETAIL-LINE1 (3/69),
                                                                                   USE: DET-LOAN-AMT (3/77), DETAIL-LINE2 (3/84),
                                                                                   USE: DETAIL-LINE3 (3/99), DETAIL-LINE4 (4/126)
                                                                                   USE: CUR-PREFIX (5/154), DETAIL-LINE5 (5/157),
                                     (D)
 LEGEND: IN - USED, OR USED PRIOR TO ANY MODIFICATION IN THE PERFORM RANGE.
        OUT - MODIFIED WITHIN THE PERFORM RANGE AND SUBSEQUENTLY USED OUTSIDE THE PERFORM RANGE.
        USE - DATA ITEMS THAT ARE USED WITHIN THE PERFORM RANGE.
        MOD - DATA ITEMS THAT ARE MODIFIED WITHIN THE PERFORM RANGE.
```

Report Field Descriptions

These are the Perform Range Usage & Interface report (a DX or a DA analysis) report fields:

Field	Description
(A)	This field shows each perform range in the program is listed.
(B)	This field shows the paragraph or section that invokes the PERFORM. The page and line number where the PERFORM is invoked is also shown.

Field	Description
(C)	This field shows each INPUT, OUTPUT, USE and MOD of the perform range is shown with the page and line number where it occurs.
(D)	This field shows the legend of symbols used to indicate a data item as an input or an output.

Short Analysis (DS or DC)

Figure 49 shows the Perform Range Usage and Interface report produced by an Extended SmartDoc analysis, with a compile (DC) or without a compile (DS).

Figure 49 • Perform Range Usage & Interface Report (DS or DC Analysis)

```
ASG-SMARTDOC-QS Rx.x LVL000                PERFORM RANGE USAGE AND INTERFACE REPORT                DDMMMYYYY HH:MM:SS PAGE 9
                                                PROGRAM: VIA000M0
            (A)                          (B)                                        (C)
PERFORM RANGE NAME (PAGE/LINE)      PERFORMED BY                        USES AND MODIFICATIONS (PAGE/LINE)
---------------------------------- ------------------------------------ ------------------------------------------
PROGRAM-INIT (7/267)               PROCEDURE DIVISION (7/251)           USE: SIP-CODE (2/29), HLD-SIP-PREFIX (6/223),
                                                                        USE: LN (6/235), CNT (6/235),
                                                                        USE: DEBUG-PARM (6/241)
                                                                        MOD: LINE-CNT (2/56), PAGE-CNT (2/57),
                                                                        MOD: CUR-PREFIX (5/154), HLD-SIP (6/222),
                                                                        MOD: LN (6/235), CNT (6/235)

P000-NEXT THRU P000-EXIT (8/299)   PROCEDURE DIVISION (7/257)           NONE

P005-VAL-PARM THRU P005-EXIT       PROGRAM-INIT (7/272)                 USE: DEA-DEPT-CODE (6/242)
(9/318)

P010-OPEN THRU P019-EXIT (9/331)   PROGRAM-INIT (7/275)                 USE: MASTERIN (2/14), MASTER-RPT (2/45)
                                                                        MOD: MASTERIN (2/14), MASTER-RPT (2/45)

P100-PRINT THRU P119-EXIT (9/344)  P000-NEXT (8/305)                   USE: CLIENT-ID (2/21), NAME (2/24),
                                                                        USE: ADDRESS1 (2/25), CITY (2/26), STATE (2/27
                                                                        USE: SIP-CODE (2/29), AREA-CODE (2/32),
                                                                        USE: EXCHANGE (2/33), PHONE-NUMBER (2/34),
                                                                        USE: PAYMENT-AMT (2/36), LOAN-AMT (2/37),
                                                                        USE: INTEREST-RATE (2/38),
                                                                        USE: LOAN-START-DATE (2/39), LOAN-TYPE (2/40),
                                                                        USE: LAST-BILL-DATE (2/41), BILLING-DAYS (2/42
                                                                        USE: YEAR-TO-DATE-INTEREST (2/43),
                                                                        USE: LINE-CNT (2/56), MIN-PAY-AMT (3/61),
                                                                        USE: MORE-PAY (3/62), DETAIL-LINE1 (3/69),
                                                                        USE: DET-LOAN-AMT (3/77), DETAIL-LINE2 (3/84),
                                                                        USE: DETAIL-LINE3 (3/99), DETAIL-LINE4 (4/126)
                                                                        USE: CUR-PREFIX (5/154), DETAIL-LINE5 (5/157),
                                                                        USE: HLD-SIP-PREFIX (6/223),
                                                                        USE: SIP-LOAN-CNT (6/227), SIP-LOAN-AMT (6/228
                                                                        USE: SIP-YTD-INT (6/229), TOTAL-CNT (6/232),
                                                                        USE: TOTAL-YTD-INT (6/234)
                                                                        MOD: MAST-RPT (2/49), MASTER-RPT (2/45),
                                                                        MOD: LINE-CNT (2/56), MORE-PAY (3/62),
                                                                        MOD: DET-NUMBER (3/73), DET-LOAN-AMT (3/77),
                                                                        MOD: DET-START-DATE (3/81), DET-NAME (3/88),
                                                                        MOD: DET-YTD-INT (3/92),
                                                                        MOD: DET-LAST-BILL-DATE (3/96),
                                                                        MOD: DET-ADDRESS (3/103), DET-AREA-CODE (3/107
                                                                        MOD: DET-EXCHANGE (3/110),
                                                                        MOD: DET-PHONE-NUMBER (3/113),
                                                                        MOD: DET-PAYMENT-AMT (3/118),
                                                                        MOD: DET-LOAN-TYPE (4/123), DET-CITY (4/130),
                                                                        MOD: DET-ST (4/134), DET-SIP-CODE (4/138),
                                                                        MOD: DET-INT-RATE (4/144),
                                                                        MOD: DET-BILLING-DAYS (4/149),
                                                                        MOD: CUR-PREFIX (5/154), DET-PROJ-PAY (5/161),
                                                                        MOD: DET-PAY-MORE (5/165), HLD-SIP (6/222),
                                                                        MOD: SIP-LOAN-CNT (6/227), SIP-LOAN-AMT (6/228
                                                                        MOD: SIP-YTD-INT (6/229), TOTAL-CNT (6/232),
                                                                        MOD: TOTAL-YTD-INT (6/234)

P1000-MIN-PAY-DUE (14/515)         P105-MOV-FIRST-TIME (9/357)          USE: LOAN-AMT (2/37), MIN-PAY-AMT (3/61)
                                   P2000-CALC-MORE-PAY (14/534)         MOD: AZERO-CODE (2/59), MIN-PAY-AMT (3/61),
                                                                        MOD: MORE-PAY (3/62)

                          (D)
LEGEND: USE - DATA ITEMS THAT ARE USED WITHIN THE PERFORM RANGE.
        MOD - DATA ITEMS THAT ARE MODIFIED WITHIN THE PERFORM RANGE.
```

Report Field Descriptions

These are the Perform Range Usage & Interface report (DS or DC analysis) fields:

Field	Description
(A)	This field shows each perform range in the program is listed.
(B)	This field shows the paragraph or section that invokes the PERFORM. The page and line number where the PERFORM is invoked is also shown.
(C)	This field shows each use and modification of the perform range is shown with the page and line number where it occurs.
(D)	This field shows the legend of symbols used to indicate if a data item is an input or an output.

Program Exception Report

The Program Exception report lists conditions that may cause the program to either fail or execute differently than intended. These are the items reported on the Program Exception report:

Field	Description
MODIFICATION WITHOUT USE	A value other than the assigned value modified a data item (or a subfield of it).
UNINITIALIZED USE	A data item (or a subfield of it) was used without first being initialized. **Note:** Data items initialized by external calls to non-COBOL programs may appear as uninitialized uses.
RECURSION	Recursion generally occurs when a paragraph or a perform range performs itself. This programming technique can lead to endless loops.
OUT OF PERFORM JUMP	An exit is made from a perform range, due to the use of a GO TO, bypassing the normal perform range return to the line following the perform statement.

Field	Description
LIVE EXITS	Live exits are exits from perform ranges left dangling by overlapping PERFORMs and GOTOs in the original performed paragraphs. Internally, the COBOL compiler creates a jump statement to return to the caller at the end of a perform range. When the jump is actually performed, the jump statement is cleared. If an exit is made from the perform range before the jump statement is cleared (the normal flow is changed), the internal jump statement is left live. If an overlapping perform range is subsequently executed, the jump statement set by the original perform range is still live and causes the program to jump back to the location of the first caller to that perform range. This results in the program not executing as expected.
DEADDATA	Unreferenced data items or data items are only referenced in DEADCODE. If a dead data item is included by a COPY statement, it is marked as DEAD IN COPY. Otherwise, items are marked as DEAD NOT IN COPY.
DEADCODE	PROCEDURE DIVISION statements never executed under any conditions. If a dead statement is included by a COPY statement, it is marked as DEAD IN COPY. Otherwise, it is marked as DEAD NOT IN COPY.

Program Exception Report for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, SmartDoc qualifies data items and paragraphs by the containing program within the Program Exception report.

These figures illustrate the Program Exception report:

- The Program Exception report produced by a DX or a DA type of analysis is shown in Figure 50 on page 98.
- The Program Exception report produced by a DS or a DC type of analysis is shown in Figure 51 on page 100.

Extended Analysis (DX or DA)

Figure 50 shows the Program Exception report produced by an Extended SmartDoc analysis with a compile (DA) or without a compile (DX).

Figure 50 • Program Exception Report (DX or DA Analysis)

```
ASG-SMARTDOC-OS Rx.x LVL000                        PROGRAM EXCEPTION REPORT                       DDMMMYYYY HH:MM:SS PAGE 9
                                                      PROGRAM: VIA0000D
                                                             (A)
                  MODIFICATIONS WITHOUT SUBSEQUENT USE(S) - USES WITHOUT PRIOR INITIALISATION(S) WITHIN THIS PROGRAM
        (B)                                        (C)          (D)
DATA ITEM (PAGE/LINE)                          MOD/USE    (PAGE/LINE)
---------------------------------------- ----------------------------------------------------------------------------------
ABEND-CODE (3/59)                              MOD (3/59), MOD (15/548)
CHECK-CODE (3/60)                              MOD (3/60)
CNT (6/235)                                    MOD (7/284), MOD (7/286)
DBA-DEPT-CODE (6/242)                          *LINKAGE SECTION ITEM*  USE (9/319), USE (9/321), USE (9/334)
DEBUG-PARM (6/241)                             *LINKAGE SECTION ITEM*  USE (7/269)
LINE-CNT (3/56)                                MOD (3/56), MOD (7/292)
LN (6/235)                                     MOD (7/279)
MASTER-REPORT-PAGE-CNT (5/178)                 MOD (13/458)
MORE-PAY (3/62)                                MOD (9/358)
PAGE-CNT (3/57)                                MOD (3/57), MOD (7/290)
SUB-LOAN-AMT (5/199)                           MOD (13/453)
SUB-LOAN-CNT (5/195)                           MOD (13/453)
SUB-YTD-INT (5/203)                            MOD (13/453)
TOTAL-AMT (6/233)                              USE (13/478)
                                                             (E)
                                                     LIST OF RECURSION
       (F)
RECURSIVE CYCLE 1
--------------------------------------------------------------------------------------------------------------------------
PROCEDURE DIVISION

    PERFORM P1000-MIN-PAY-DUE 9/357

        PERFORM P1000-MIN-PAY-DUE 14/534

                                                             (G)
                                                LIST OF OUT-OF-PERFORM JUMPS
       (H)                                            (I)          (J)
PERFORM RANGE NAME                                  PAGE/LINE   JUMP STATEMENT
---------------------------------------------------- ---------- ------------------------------------------------------------
P1000-MIN-PAY-DUE                                   14/519      GO TO P1000-EXIT.
P120-READ THRU P129-EXIT                            12/432      GO TO P170-FINAL.

                                                             (K)
                                                     LIST OF LIVE EXITS
          (L)                                                          (M)                                         (N)
PERFORM RANGE ENCOUNTERS                                   PERFORM RANGE'S LIVE EXIT                             PAGE/LINE
------------------------------------------------------------- ------------------------------------------------------ -----
P1000-MIN-PAY-DUE                                          P2000-CALC-MORE-PAY THRU P2000-EXIT                    14/538

                                                              (O)
                                                     LIST OF DEAD DATA (NOT IN COPY)
   (P)
PAGE/LINE
--------------------------------------------------------------------------------------------------------------------------
3/60,3/63,3/64,3/65,3/66,3/67,6/240

                                                     LIST OF DEAD DATA (IN COPY)
PAGE/LINE
--------------------------------------------------------------------------------------------------------------------------
NO DEAD DATA FOUND

                                                             (Q)
                                                    LIST OF DEAD CODE (NOT IN COPY)
      (R)                                     (S)
PARAGRAPH NAME                               PAGE/LINE
---------------------------------------   --------------------------------------------------------------------------------
P200-EXIT                                    14/498,14/499
P250-AVG-AMT                                 14/505,14/506
P259-AVG-EXIT                                14/508,14/509

                                                    LIST OF DEAD CODE (IN COPY)
PARAGRAPH NAME                               PAGE/LINE
---------------------------------------   --------------------------------------------------------------------------------
NO DEAD CODE FOUND
```

Report Field Descriptions

These are the Program Exception report (DX or DA analysis) fields:

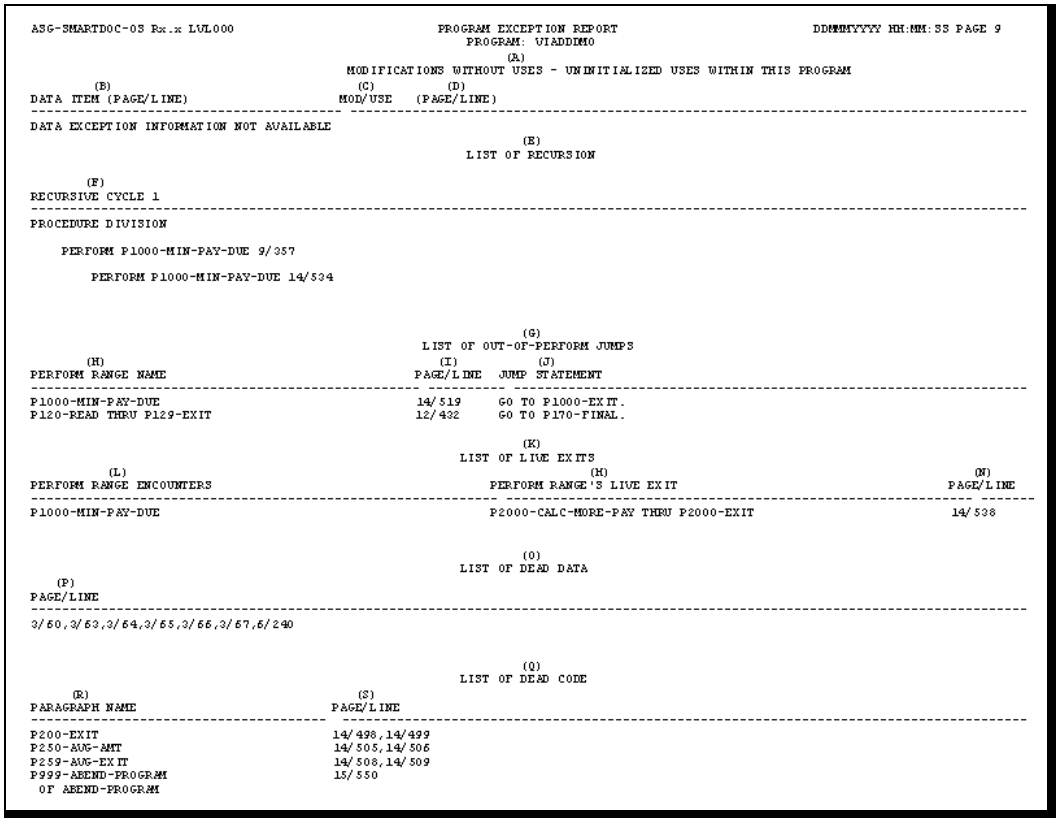
Field	Description
(A)	This field shows modifications without subsequent use(s) - uses without prior initialization(s) within this program section heading.
(B)	This field shows all data items that have been used without being initialized, and data items that have not been subsequently used after being modified.
(C)	This field shows all data items that have been used without being initialized are indicated with USE. Data items modified without subsequently using the value are indicated with MOD.
(D)	This field shows the page and line number on the Advanced Source Listing where the data item is located.
(E)	This field shows the List of Recursion section heading.
(F)	This field shows each perform range or each paragraph that performs itself is listed along with the page and line number on the Advanced Source Listing where the recursion occurs.
(G)	This field shows the List of Out-of-Perform Jumps section heading.
(H)	This field shows each perform range that contains an out of perform jump is identified.
(I)	This field shows the page and line number on the Advanced Source Listing where the PERFORM statement that caused the out of perform jump is located.
(J)	This field shows the statement that caused the out of perform jump.
(K)	This field shows the List of Live Exits section heading.
(L)	This field shows each perform range that contains a live exit is indicated.
(M)	This field shows the statement that caused the live exit.
(N)	This field shows the page and line number on the Advanced Source Listing where the perform range containing the live exit is located.
(O)	List of Dead Data section heading.
(P)	This field shows the each page and line number containing dead data is indicated.

Field	Description
(Q)	This field shows the List of Dead Code section heading.
(R)	This field shows the each paragraph that contains dead code is identified.
(S)	This field shows the page and line number on the Advanced Source Listing where the paragraph containing dead code is located.

Short Analysis (DS or DC)

Figure 51 shows the Program Exception report produced by a short SmartDoc analysis either with a compile (DC) or without a compile (DS).

Figure 51 • Program Exception Report (DS or DC Analysis)



```
ASG-SMARTDOC-03 Rx.x LVL000                    PROGRAM EXCEPTION REPORT                    DDMMMYYYY HH:MM:SS PAGE 9
                                                  PROGRAM: VIADDIM0
                                                      (A)
                                  MODIFICATIONS WITHOUT USES - UNINITIALIZED USES WITHIN THIS PROGRAM
        (B)                          (C)       (D)
DATA ITEM (PAGE/LINE)              MOD/USE   (PAGE/LINE)
-------------------------------- ---------- ----------------------------------------------------------------------------------------------------
DATA EXCEPTION INFORMATION NOT AVAILABLE

                                                      (E)
                                                LIST OF RECURSION

       (F)
RECURSIVE CYCLE 1
-------------------------------------------------------------------------------------------------------------------------------------------------
PROCEDURE DIVISION

    PERFORM P1000-MIN-PAY-DUE 9/357

        PERFORM P1000-MIN-PAY-DUE 14/534



                                                      (G)
                                           LIST OF OUT-OF-PERFORM JUMPS
       (H)                                    (I)       (J)
PERFORM RANGE NAME                         PAGE/LINE   JUMP STATEMENT
-------------------------------------------- --------- ----------------------------------------------------------------------------------------
P1000-MIN-PAY-DUE                            14/519      GO TO P1000-EXIT.
P120-READ THRU P129-EXIT                     12/432      GO TO P170-FINAL.

                                                      (K)
                                                LIST OF LIVE EXITS
         (L)                                                  (M)                                                          (N)
PERFORM RANGE ENCOUNTERS                          PERFORM RANGE'S LIVE EXIT                                          PAGE/LINE
-------------------------------------------------- ---------------------------------------------------------------------- ---------
P1000-MIN-PAY-DUE                                  P2000-CALC-MORE-PAY THRU P2000-EXIT                                  14/538

                                                      (O)
                                                LIST OF DEAD DATA
   (P)
PAGE/LINE
-------------------------------------------------------------------------------------------------------------------------------------------------
3/50,3/53,3/54,3/55,3/56,3/57,5/240

                                                      (Q)
                                                LIST OF DEAD CODE
      (R)                                (S)
PARAGRAPH NAME                         PAGE/LINE
-------------------------------------- ----------------------------------------------------------------------------------------------------
P200-EXIT                              14/498,14/499
P250-AUG-AMT                           14/505,14/506
P259-AUG-EXIT                          14/508,14/509
P999-ABEND-PROGRAM                     15/550
 OF ABEND-PROGRAM
```

Report Field Descriptions

These are the Program Exception report (a DS or a DC analysis) fields:

Field	Description
(A)	This field shows modifications without uses - uninitialized uses within this program section heading.
(B)	This field shows all data items that have been used without being initialized, and data items that have not been subsequently used after being modified.
(C)	This field shows all data items that have been used without being initialized with USE. Data items modified without subsequently using the value are indicated with MOD.
(D)	This field shows the page and line number on the Advanced Source Listing where the data item is located.
(E)	This field shows the List of Recursion section heading.
(F)	This field shows each perform range or each paragraph that performs itself along with the page and line number on the Advanced Source Listing where the recursion occurs.
(G)	This field shows the List of Out-of-Perform Jumps section heading.
(H)	This field shows each perform range that contains an out of perform jump.
(I)	This field shows the page and line number on the Advanced Source Listing where the PERFORM statement that caused the out of perform jump is located.
(J)	This field shows the statement that caused the out of perform jump.
(K)	This field shows the List of Live Exits section heading.
(L)	This field shows each perform range that contains a live exit is indicated.
(M)	This field shows the statement that caused the live exit.
(N)	This field shows the page and line number on the Advanced Source Listing where the perform range containing the live exit is located.
(O)	This field shows the List of Dead Data section heading.
(P)	This field shows each page and line number containing dead data.
(Q)	This field shows the List of Dead Code section heading.

Field	Description
(R)	This field shows each perform range or each paragraph that contains dead code is identified.
(S)	This field shows the page and line number on the Advanced Source Listing where the perform range or the paragraph containing dead code is located.

Program Exception Report for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, the Program Exception report (see Figure 52) displays the program name before each data item, perform range name, etc., where appropriate.

Figure 52 • Program Exception Report for COBOL II Release 3

```
ASG-SmartDoc-OS Rx.x LVL000                    PROGRAM EXCEPTION REPORT                    DDMMMYYYY HH:MM:SS PAGE  999
                                                  PROGRAM: VIA00E03

                                        MODIFICATIONS WITHOUT USES - UNINITIALIZED USES WITHIN THIS PROGRAM

PROGRAM/DATA ITEM (PAGE/LINE)            MOD/USE   (PAGE/LINE)
-------------------------------------------------------------------------------------------------------------------------------
VIA00E03/DBA-DEPT-CODE (5/228)           *LINKAGE SECTION ITEM*  USE (7/296), USE (7/310)
VIA00E03/DETAIL-LINE1 (3/67)             USE (8/348)
VIA00E03/DETAIL-LINE2 (3/82)             USE (8/353)
VIA00E03/DETAIL-LINE4 (4/124)            USE (8/368)
VIA00E01/END-INPUT (14/601)              USE (15/614)
P999-ABEND-PROGRAM/LOCATION (11/488)     USE (12/494)
VIA00E01/MASTER-REPORT-DATE (14/603)     USE (15/650)
VIA00E03/MASTER-RPT (2/45)               USE (7/309)
VIA00E03/MASTERIN (2/14)                 USE (7/308)
VIA00E01/RPT-HDG-LINE1 (13/530)          USE (15/651)
VIA00E03/RPT-HDG-LINE2 (4/168)           USE (9/431)
VIA00E03/SUB-PRINT (5/178)               USE (9/420)
VIA00E03/TOTAL-AMT (5/221)               USE (10/448)
VIA00E03/TOTAL-PRT (5/194)               USE (10/451)
VIA00E01/YEAR-TO-DATE-INTEREST (14/598)  USE (15/639)
VIA00E03/ZIP-LOAN-AMT (5/216)            USE (10/476)
VIA00E03/ZIP-LOAN-CNT (5/215)            USE (10/476)

                                                    LIST OF RECURSION

-------------------------------------------------------------------------------------------------------------------------------
NO RECURSIVE PERFORMS IN THIS PROGRAM


                                                 LIST OF OUT-OF-PERFORM JUMPS

PROGRAM/PERFORM RANGE NAME               PAGE/LINE   JUMP STATEMENT
-------------------------------------------------------------------------------------------------------------------------------
VIA00E03/P120-READ THRU P129-EXIT        9/401       go to p170-final.


                                                      LIST OF LIVE EXITS

PROGRAM/PERFORM RANGE ENCOUNTERS                 PERFORM RANGE'S LIVE EXIT                                        PAGE/LINE
-------------------------------------------------------------------------------------------------------------------------------
NO LIVE EXITS ENCOUNTERED IN THIS PROGRAM


                                                      LIST OF DEAD DATA

PAGE/LINE
-------------------------------------------------------------------------------------------------------------------------------
3/60,3/61,3/62,3/63,3/64,3/65,5/226

                                                      LIST OF DEAD CODE

PROGRAM/PARAGRAPH NAME                   PAGE/LINE
---------------------------------------  -------------------------------------------------------------------------------------
VIA00E03/P200-EXIT
VIA00E03/P200-EXIT                       10/468
VIA00E03/P200-EXIT                       10/469
VIA00E03/P250-AVG-AMT                    10/475
VIA00E03/P250-AVG-AMT                    10/476
VIA00E03/P259-AVG-EXIT                   10/478
VIA00E03/P259-AVG-EXIT                   10/479
P999-ABEND-PROGRAM/PROCEDURE             12/498
 DIVISION
```

Metrics Report

Software metrics assess the complexity, program architecture, and software quality that indicate the condition or state of each program. They allow you to identify programs that need either enhancement or re-engineering.

The AKR stores program metrics information as a separate member, away from the program. Also, the AKR retains unlimited versions of metric data for each program, providing the information for the complexity versus time graphs. However, due to page limitations, the metrics report lists only the 25 most recent versions.

The Metrics report provides metrics information by program and by PERFORM range. These program metrics are included:

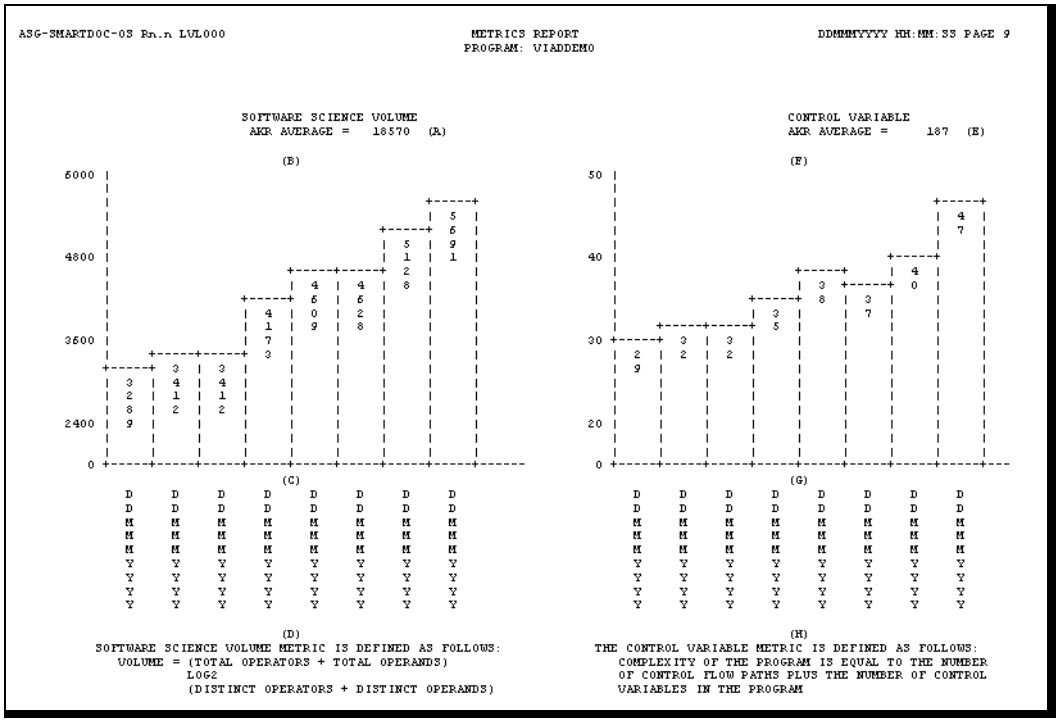
- Software Science Volume
- Cyclomatic Complexity
- Essential Complexity
- Control Variable
- GOTOFAR

The report displays the program metric information graphically. PERFORM Range metrics included in the report are, Cyclomatic, Essential, and GOTOFAR. The PERFORM range metrics information is in table format. (See Chapter 10, "Metrics," on page 153 for metric definitions.)

Software Science Volume and Control Variable Program Metrics

Figure 53 on page 104 shows the Software Science Volume and Control Variable Metric graph sections of the Metrics report. Metrics are reported by program. (See "Software Science Volume Metric" on page 9 and "Control Variable Metric" on page 9 for additional information on the metrics calculations.)

Figure 53 • Metrics Report - Software Science Volume & Control Variable Program



Report Field Descriptions

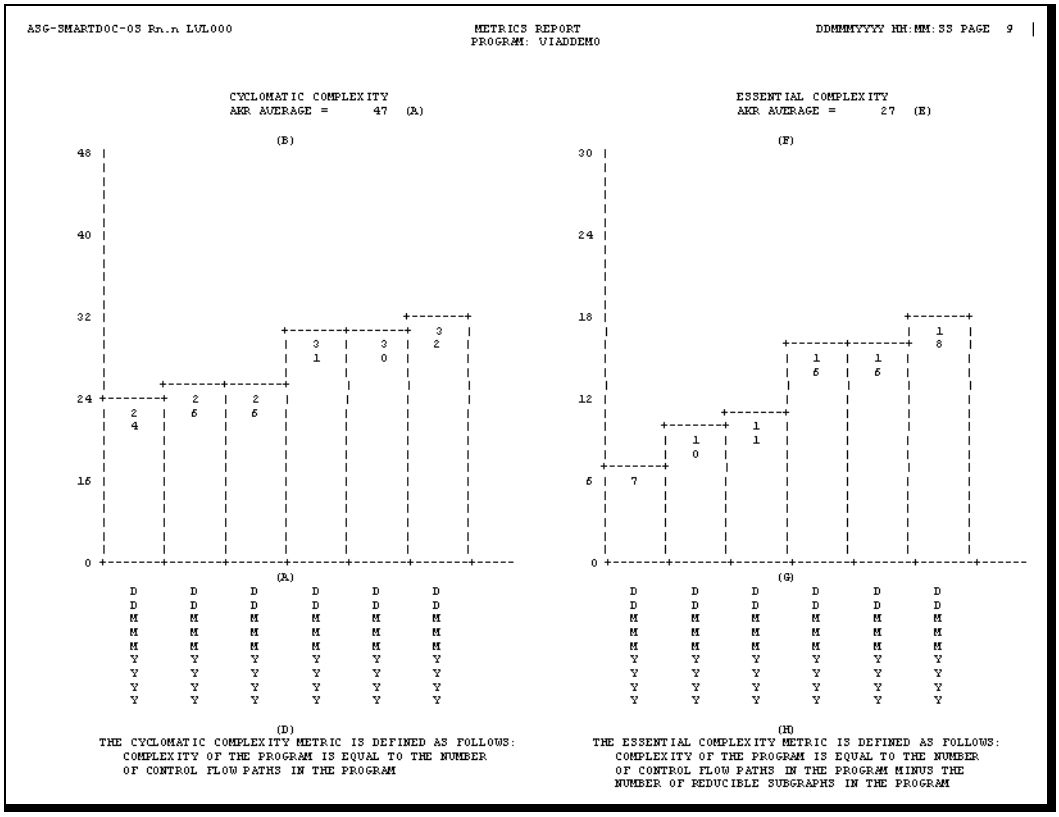
These are the Metrics report (software science volume & control variable program) fields:

Field	Description
(A)	This field shows the average Software Science Volume Metric value, based on the most recent version of program metrics in the AKR.
(B)	This field shows a graph of the Software Science Volume Metric. Each column on the graph represents one version of the program.
(C)	This field shows the date each program version was analyzed.
(D)	This field shows a brief description of the Software Science Volume Metric.
(E)	This field shows the average Control Variable Metric value, based on the most recent version of program metrics in the AKR.
(F)	This field shows a graph of the Control Variable Metric. Each column on the graph represents one version of the program.
(G)	This field shows the date each program version was analyzed.
(H)	This field shows a brief description of the Control Variable Metric.

Cyclomatic Complexity and Essential Complexity Program Metrics

Figure 54 shows the Cyclomatic Complexity Metric and the Essential Complexity Metric sections of the Metrics report. Metrics information is reported by program. (See "Cyclomatic Complexity Metric" on page 9 and "Essential Complexity Metric" on page 10 for additional information on the metrics calculations.)

Figure 54 • Metrics Report - Cyclomatic Complexity & Essential Complexity Program



Report Field Descriptions

These are the Metrics report (cyclomatic complexity & essential complexity program) fields:

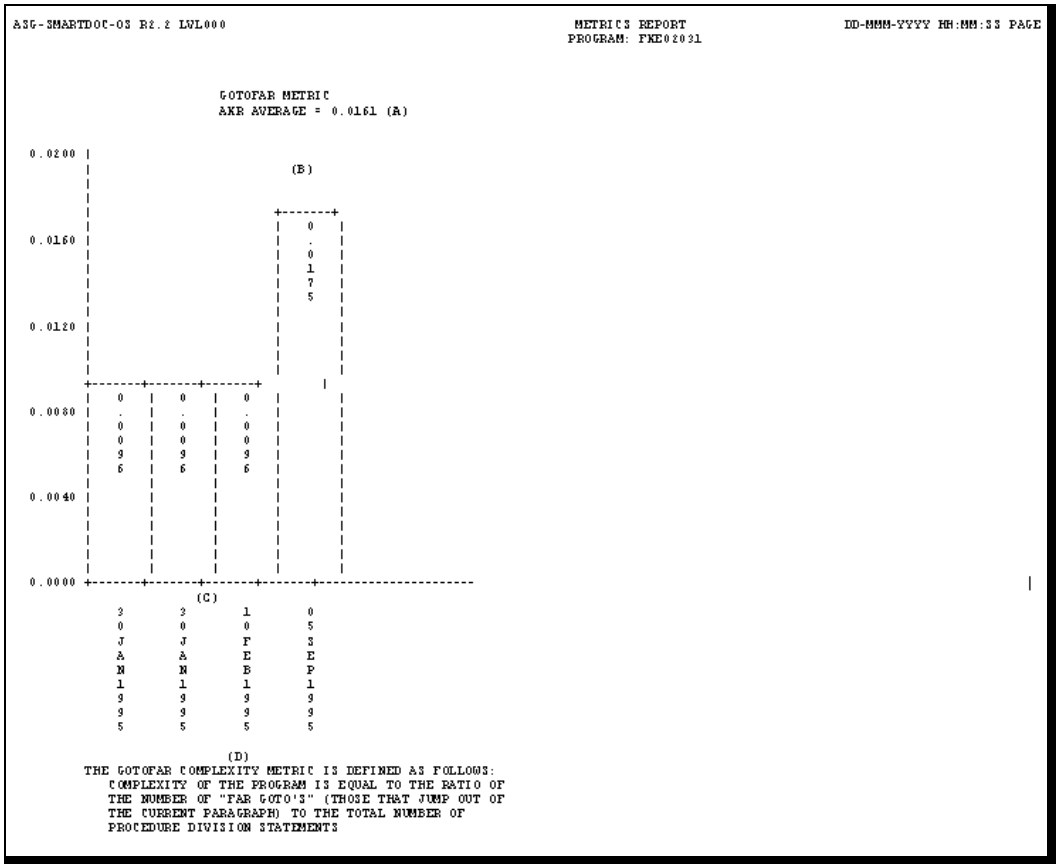
Field	Description
(A)	This field shows the average Cyclomatic Complexity value, based on the versions of program metrics in the AKR.
(B)	This field shows a graph of the Cyclomatic Complexity Metric. Each column on the graph represents one version of the program.
(C)	This field shows the date each program version was analyzed.
(D)	This field shows a brief description of the Cyclomatic Complexity Metric.

Field	Description
(E)	This field shows the average Essential Complexity value, based on the versions of program metrics in the AKR.
(F)	This field shows a graph of the Essential Complexity Metric. Each column on the graph represents one version of the program.
(G)	This field shows the date each program version was analyzed.
(H)	This field shows a brief description of the Essential Complexity Metric.

GOTOFAR Program Metric

Figure 55 shows the GOTOFAR Program Metric report. (See "GOTOFAR Metric" on page 10 for additional information on the GOTOFAR metric calculation.)

Figure 55 • Metrics Report - GOTOFAR Program Metrics



Report Field Descriptions

These are the Metrics report (GOTOFAR program metrics) fields:

Field	Description
(A)	This field shows the average GOTOFAR value, based on the versions of program metrics in the AKR.
(B)	This field shows a graph of the GOTOFAR Metric. Each column on the graph represents one version of the program.
(C)	This field shows the date each program version was analyzed.
(D)	This field shows a brief description of the GOTOFAR Metric.

PERFORM Range Metric Report

Figure 56 shows the PERFORM Range Metrics section of the Metrics report. Metrics are reported by PERFORM range.

Figure 56 • Metrics Report - PERFORM Range Metrics Report Field Descriptions

```
ASG-SMARTDOC-03 R2.2 LVL000          METRICS REPORT (BY PERFORM RANGE)          DD-MMM-YYYY HH:MM:SS PAGE
                                            PROGRAM:  FKE02031

                                                  (B)      (C)      (D)
                (A)                            CYCLO-   ESSEN-   GOTO-
PERFORM RANGE NAME (PAGE/LINE)                 MATIC    TIAL     FAR
-----------------------------------------------------   ------   ------   ------

FKE02031 (SL/1)                                     1        1   0.0000
0050-HOUSEKEEP (SL/606)                             7        1   0.0000
0088-WQ-TABLE-INIT (SL/656)                         1        1   0.0000
0100-LOAD-M-READ (SL/661)                           8        7   0.1316
0195-READ-SALARY (SL/711)                           2        1   0.0000
0200-CONCLUSION (SL/716)                            1        1   0.0000
1000-MAINLINE (SL/720)                              1        1   0.0000
1500-STRIP-MSTR (SL/741)                            2        1   0.0000
2000-FORMAT-M-READ (SL/749)                         4        1   0.0000
3300-READ-MSTR (SL/777)                             2        1   0.0000
5000-PRINT-REPORT (SL/782)                          2        1   0.0000
5100-PRINT-M-RETURN (SL/796)                       13        1   0.0000
5700-L1-BREAK (SL/911)                              3        1   0.0000
5800-L2-BREAK (SL/950)                              2        1   0.0000
5950-RETURN-REC (SL/970)                            2        1   0.0000
6400-PUT-REPORT (SL/975)                            2        1   0.1250
6405-COUNT OF 6400-PUT-REPORT THRU 640... (SL/983)  3        1   0.0000
6500-HEADERS (SL/993)                               1        1   0.0000
6600-STUFF-HEADING (SL/1008)                        3        1   0.0000
6700-READ-ORG-FILE (SL/1032)                        2        1   0.0000
6900-STUFF-TRAILER (SL/1038)                        3        1   0.0000
7100-CALC-DIFF-DATE (SL/1048)                       3        1   0.0000
7200-ADD-INCR-DATE (SL/1065)                        2        1   0.0000
8990-ABORT (SL/1081)                                1        1   0.0000
```

Report Field Descriptions

These are the Metrics report (PERFORM range metrics report field descriptions) fields:

Field	Description
(A)	This field lists the PERFORM range names for the calculated metrics.
(B)	This field lists the Cyclomatic Complexity calculated for each PERFORM range.
(C)	This field lists the Essential Complexity value calculated for each PERFORM range.
(D)	This field lists the GOTOFAR metric calculated for each PERFORM Range.

Compiler/Optimizer Output

The Compiler/Optimizer Output incorporates the compiler and CA-Optimizer output into a SmartDoc report. This output consists of the memory map, PMAP or LIST (if requested), statistics, error messages, and unmerged DMAP items.

Figure 57 • Compiler/Optimizer Output

```
ASG-SMARTDOC-03 Rx.x LVL000                          COMPILER OUTPUT REPORT                          DDMMMYYYYY HH:MM:SS PAGE     9
                                                       PROGRAM: VIADDDM0
                                (A)
                XSASW CELLS                  00AEC
                XSA CELLS                    00AEC
                PARAM CELLS                  00AEC
                RPTSAV AREA                  00B24
                CHECKPT CTR                  00B24

 LITERAL POOL (HEX)

 00D08 (LIT+0)     00010000  F0F5F1F0  F0003602  4C015CF0  F0F0F000  0035F021
 00D20 (LIT+24)    20202020  20202020  20202020  4B2020F0  21202020  2020204B
 00D38 (LIT+48)    20204BF0  21202020  20000005  1C0FF0F0  F0F0F0F0  F0F0F0F0
 00D50 (LIT+72)    F0F0F0F0  F0F00004  F0202120  20202020  20202020  20202020
 00D68 (LIT+96)    20204B20  20F02021  20202020  20202020  2020204B  2020F0F0
 00D80 (LIT+120)   F0F0F1F0  F0F0F0F0  09999999C  9999999C  025C036C  020C060C
 00D98 (LIT+144)   03E7001B  8000C5D5  C440E5C9  C1E2C4C4  D4D640D7  D9D6C3C5
 00DB0 (LIT+168)   E2E2C9D5  C7040040  20000000  00E3C5E2  E3100000  001C0000
 00DC8 (LIT+192)   00481400  0000

      DISPLAY LITERALS (BCD)

 00DCE (LTL+198)  'TOTAL INPUT RECORDS - '


                    PGT                      00B30

                OVERFLOW CELLS               00B30
                VIRTUAL CELLS                00B30
                PROCEDURE NAME CELLS         00B78
                GENERATED NAME CELLS         00BCC
                DCB ADDRESS CELLS            00CA4
                VNI CELLS                    00CAC
                LITERALS                     00D08
                DISPLAY LITERALS             00DCE


  REGISTER ASSIGNMENT

    REG 6   BL =3
    REG 7   BL =1
    REG 8   BL =2

 WORKING-STORAGE STARTS AT LOCATION 000A0 FOR A LENGTH OF 00508.
```

Report Field Description

A. This field shows the memory map and statistics for the program.

Master Index

The Master Index (see Figure 58 on page 110) is an alphabetical listing of all named entities in the SmartDoc reports. This report helps you determine where a particular entity is located in the SmartDoc reports. Use this report as the starting point for investigating a program item.

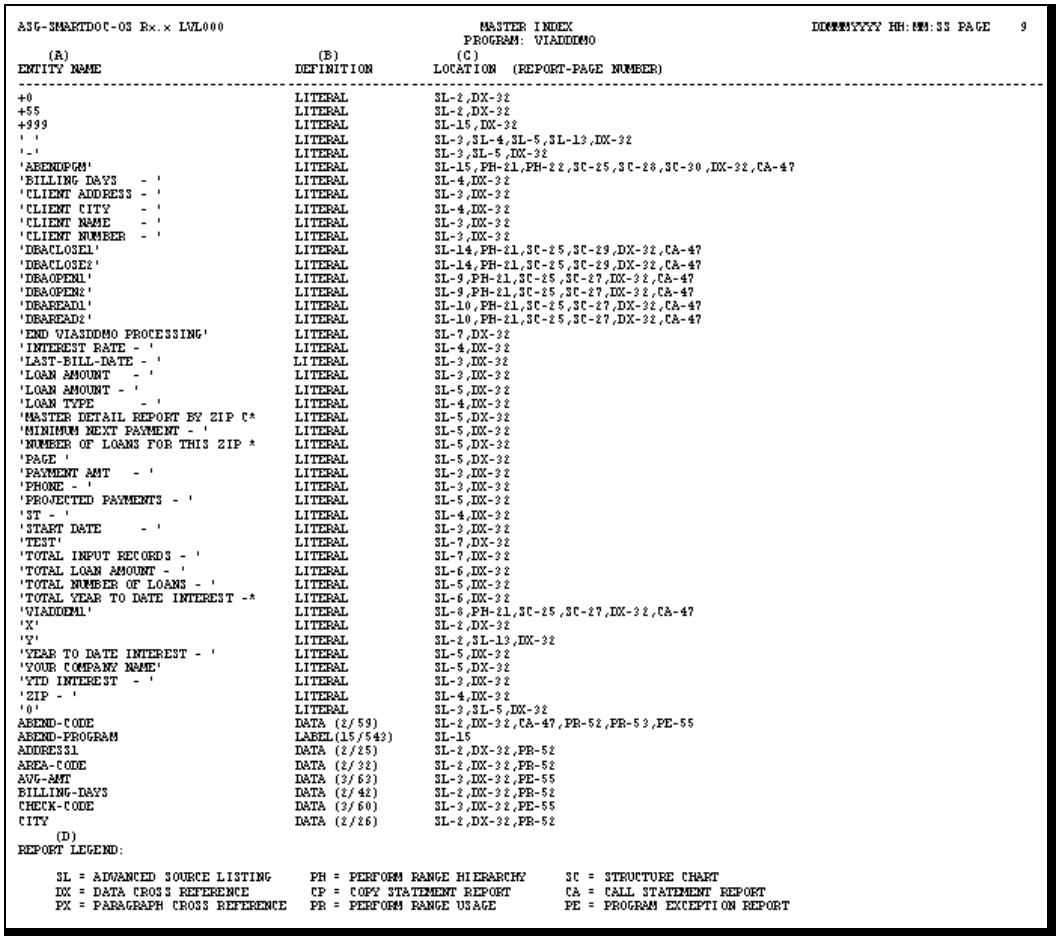
This information is included on the Master Index:

- All named entities
- Figurative constants
- Literals
- Labels
- Data items
- Items implicitly defined by the environment (compiler, preprocessors, etc.)

Master Index for COBOL II Release 3 and Later Programs

For COBOL II Release 3 and later programs, Master Index entries are qualified (where needed) by the containing program to eliminate ambiguities.

Figure 58 • Master Index



```
ASG-SMARTDOC-03 R*.* LVL000                          MASTER INDEX                                    DDMMMYYYY HH:MM:SS PAGE    9
                                                   PROGRAM: VIADDM0
    (A)                          (B)                (C)
ENTITY NAME                      DEFINITION         LOCATION  (REPORT-PAGE NUMBER)
-------------------------------------------------------------------------------------------------------------------------------------
+0                               LITERAL            SL-2,DX-32
+55                              LITERAL            SL-2,DX-32
+999                             LITERAL            SL-15,DX-32
' '                              LITERAL            SL-3,SL-4,SL-5,SL-13,DX-32
'-'                              LITERAL            SL-3,SL-5,DX-32
'ABENDPGM'                       LITERAL            SL-15,PH-21,PH-22,SC-25,SC-28,SC-30,DX-32,CA-47
'BILLING DAYS   - '              LITERAL            SL-4,DX-32
'CLIENT ADDRESS - '              LITERAL            SL-3,DX-32
'CLIENT CITY    - '              LITERAL            SL-4,DX-32
'CLIENT NAME    - '              LITERAL            SL-3,DX-32
'CLIENT NUMBER  - '              LITERAL            SL-3,DX-32
'DBACLOSE1'                      LITERAL            SL-14,PH-21,SC-25,SC-29,DX-32,CA-47
'DBACLOSE2'                      LITERAL            SL-14,PH-21,SC-25,SC-29,DX-32,CA-47
'DBAOPEN1'                       LITERAL            SL-9,PH-21,SC-25,SC-27,DX-32,CA-47
'DBAOPEN2'                       LITERAL            SL-9,PH-21,SC-25,SC-27,DX-32,CA-47
'DBAREAD1'                       LITERAL            SL-10,PH-21,SC-25,SC-27,DX-32,CA-47
'DBAREAD2'                       LITERAL            SL-10,PH-21,SC-25,SC-27,DX-32,CA-47
'END VIASDDM0 PROCESSING'        LITERAL            SL-7,DX-32
'INTEREST RATE  - '              LITERAL            SL-4,DX-32
'LAST-BILL-DATE - '              LITERAL            SL-3,DX-32
'LOAN AMOUNT    - '              LITERAL            SL-3,DX-32
'LOAN AMOUNT - '                 LITERAL            SL-5,DX-32
'LOAN TYPE      - '              LITERAL            SL-4,DX-32
'MASTER DETAIL REPORT BY ZIP C*  LITERAL            SL-5,DX-32
'MINIMUM NEXT PAYMENT - '        LITERAL            SL-5,DX-32
'NUMBER OF LOANS FOR THIS ZIP *  LITERAL            SL-5,DX-32
'PAGE '                          LITERAL            SL-5,DX-32
'PAYMENT AMT    - '              LITERAL            SL-3,DX-32
'PHONE - '                       LITERAL            SL-3,DX-32
'PROJECTED PAYMENTS - '          LITERAL            SL-5,DX-32
'ST - '                          LITERAL            SL-4,DX-32
'START DATE     - '              LITERAL            SL-3,DX-32
'TEST'                           LITERAL            SL-7,DX-32
'TOTAL INPUT RECORDS - '         LITERAL            SL-7,DX-32
'TOTAL LOAN AMOUNT - '           LITERAL            SL-6,DX-32
'TOTAL NUMBER OF LOANS - '       LITERAL            SL-5,DX-32
'TOTAL YEAR TO DATE INTEREST -*  LITERAL            SL-6,DX-32
'VIADDEM1'                       LITERAL            SL-8,PH-21,SC-25,SC-27,DX-32,CA-47
'X'                              LITERAL            SL-2,DX-32
'Y'                              LITERAL            SL-2,SL-13,DX-32
'YEAR TO DATE INTEREST - '       LITERAL            SL-5,DX-32
'YOUR COMPANY NAME'              LITERAL            SL-5,DX-32
'YTD INTEREST   - '              LITERAL            SL-3,DX-32
'ZIP - '                         LITERAL            SL-4,DX-32
'0'                              LITERAL            SL-3,SL-5,DX-32
ABEND-CODE                       DATA (2/59)        SL-2,DX-32,CA-47,PR-52,PR-53,PE-55
ABEND-PROGRAM                    LABEL(15/543)      SL-15
ADDRESS1                         DATA (2/25)        SL-2,DX-32,PR-52
AREA-CODE                        DATA (2/32)        SL-2,DX-32,PR-52
AVG-AMT                          DATA (3/63)        SL-3,DX-32,PE-55
BILLING-DAYS                     DATA (2/42)        SL-2,DX-32,PR-52
CHECK-CODE                       DATA (3/60)        SL-3,DX-32,PE-55
CITY                             DATA (2/26)        SL-2,DX-32,PR-52
    (D)
REPORT LEGEND:

     SL = ADVANCED SOURCE LISTING     PH = PERFORM RANGE HIERARCHY     SC = STRUCTURE CHART
     DX = DATA CROSS REFERENCE        CP = COPY STATEMENT REPORT       CA = CALL STATEMENT REPORT
     PX = PARAGRAPH CROSS REFERENCE   PR = PERFORM RANGE USAGE         PE = PROGRAM EXCEPTION REPORT
```

Report Field Descriptions

These are the Master Index fields:

Field	Description
(A)	This field lists each entity in the program.
(B)	This field indicates the type of entity such as a literal, figurative constant, label, data item, etc. The page and line number where the definition of that entity is located is also shown.
(C)	This field shows a symbol for the report and the page number where the entity is referenced.
(D)	This field shows the legend of symbols used to identify the report on the referenced entity.

6 File

This chapter describes the File pull-down and contains these sections:

File Pull-down

Use the File pull-down (see Figure 59 on page 112) to perform these actions:

- Analyze a program
- Generate SmartDoc reports
- Analyze programs
- Exit SmartDoc

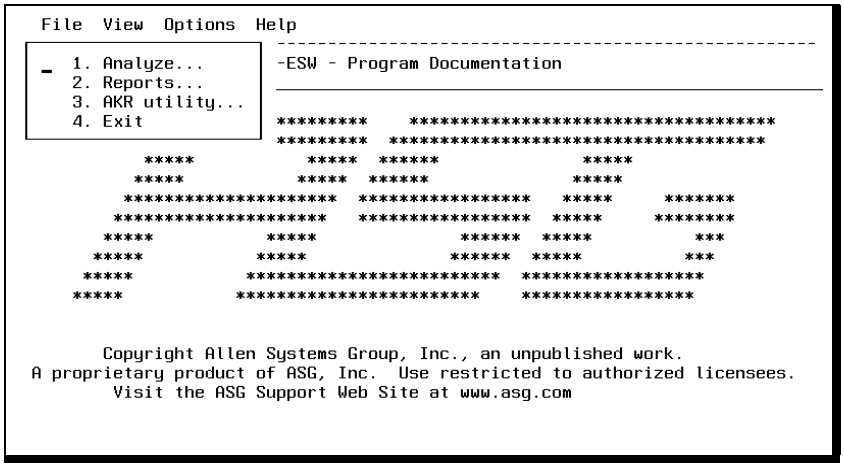
To display the File pull-down, follow this step:

- Select File on the action bar and press Enter (see Figure 59 on page 112).

Note:

The File pull-down contains different actions on the Program Metrics View screen. See Figure 71 on page 133 for more information.

Figure 59 • File Pull-down



Actions

These are the actions available on the File Pull-down:

Field	Description
1. Analyze	Displays the Analyze Submit pop-up (see Figure 60 on page 113) used to submit a compile/analyze job
2. Reports	Displays the File - SmartDoc Options pop-up (see Figure 61 on page 116) used to specify what reports are generated, and to customize the generated reports
3. AKR utility	Displays the File - AKR Utility pop-up (see Figure 66 on page 123) used to display the program directory, to allocate or to expand an AKR, to rename a program, and to delete a program
4. Exit	Exits SmartDoc

Analyze Submit Pop-up

Use the Analyze Submit pop-up (see Figure 60 on page 113) to analyze and place a program in the AKR.

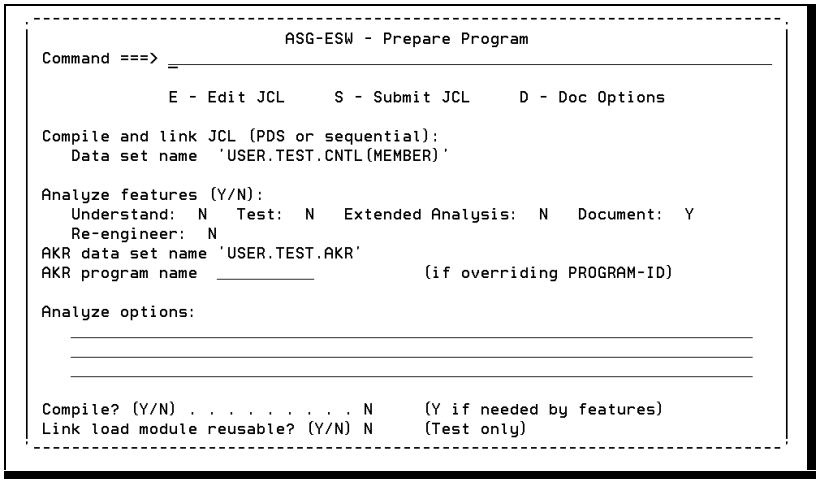
To display the Analyze Submit pop-up, follow this step:

- Select File ▸ Analyze and press Enter (see Figure 59).

  Or

  Type the ANALYZE command on any screen and press Enter.

Figure 60 • Analyze Submit Pop-up



Options

These are the Analyze Submit pop-up options:

Field	Description
E - Edit JCL	Enter `E` to review or to change the compile/analyze JCL, if necessary. Select the E option to generate the JCL to edit from the JCL member you specified in the dataset name field (using the rules outlined in the Automatic JCL Modifications section). The generated JCL displays on the Edit screen. When editing is complete, enter the ISPF SUBMIT command to execute the edited JCL. Optionally, you can use the CREATE command to saved the edited JCL in a partitioned dataset. Otherwise, changes made at this time are not saved.
S - Submit JCL	Enter `S` to submit the JCL to compile/analyze the specified program. The JCL submitted is generated from the JCL member specified in the dataset name field, applying the rules outlined in the Automatic JCL Modifications section.
D - SmartDoc Options	This field displays only if SmartDoc is installed. Enter `D` to display the File -SmartDoc Options pop-up (see Figure 61 on page 116) used to request an Extended SmartDoc analysis and to specify what reports (if any) to generate.

Fields

These are the Analyze Submit pop-up fields:

Field	Description	
Compile and link JCL (PDS or sequential)	Dataset name	Required. The PDS member or the sequential dataset containing the JCL to compile and link the program. If the JCL resides in a source manager such as Librarian or Panvalet, use the VIASUB edit macro to submit the compile/analyze job.
Analyze Features		
	Understand	Optional. This field displays only if Insight is installed. This analysis provides the logic and program execution flow capabilities of Insight. If Insight is the only product installed, this field contains `YES` and cannot be changed. The default is `Y`.
	Test	Optional. This field displays only if SmartTest is installed. Enter `Y` to perform a SmartTest compile/analysis. This analysis provides the testing and debugging information required by SmartTest. If SmartTest is the only product installed, this field contains `YES` and cannot be changed. The default is `Y`.
	Extended Analysis	Optional. This field displays only if SmartTest is installed. This type of analysis provides comprehensive program analyzing capabilities in addition to the testing and debugging capabilities of SmartTest. The default is `Y`. An Extended SmartDoc analysis is specified on the File - SmartDoc Options pop-up (see Figure 61 on page 116).
	Document	Optional. This field 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 `YES` and cannot be changed. The default is `Y`.
	Re-engineer	Optional. `YES` specifies that a Encore compile/analysis is performed. This type of analysis provides the logic and program execution flow capabilities of Encore. The default is `N`.
AKR dataset name	Optional. The AKR that contains the information for the analyzed program.	

Field	Description
AKR program name	Optional. Enter an alias name for the analyze process to use when saving its results in the AKR. If you do not enter a value in this field, the results of the analyze job are saved in the AKR with the same name as the PROGRAM-ID statement name in the COBOL source. If an AKR program name is entered, the analyzed program is saved in the AKR with that name, and as an alias of the PROGRAM-ID. If the program contains ENTRY points, the analyze job saves a member for each ENTRY point in the AKR with an alias of the PROGRAM-ID. This field is only used for the AKR program name and does not change the COBOL program name in the source.
Analyze options	Optional. This field is used to override analyze options. Default options for the analyze job are established at installation time. Analyze options that can be entered in this field are described in Chapter 11, "Analyze," on page 157.
Compile?	Optional. A program does not need to be compiled if Insight, Encore, or SmartDoc are the only features specified. To suppress the compile step, type `N` in this field. This field is forced to a value of `Y` if SmartTest and/or Extended analysis is selected.
Link load module reusable?	Optional. This field is used to test a program dynamically loaded under SmartTest and is tested with RUN MONITOR. You need to mark the load module as reusable so that the Breakpoints are retained across calls. The default is `Y`.

File - SmartDoc Report Pop-up

Use the File - SmartDoc Options pop-up (see Figure 61 on page 116) to specify analyze options and display the File - Select Reports pop-up.

To display the File - SmartDoc Options pop-up, follow this step:

- Select Analyze Submit pop-up ▸ D and press Enter (see Figure 60 on page 113).

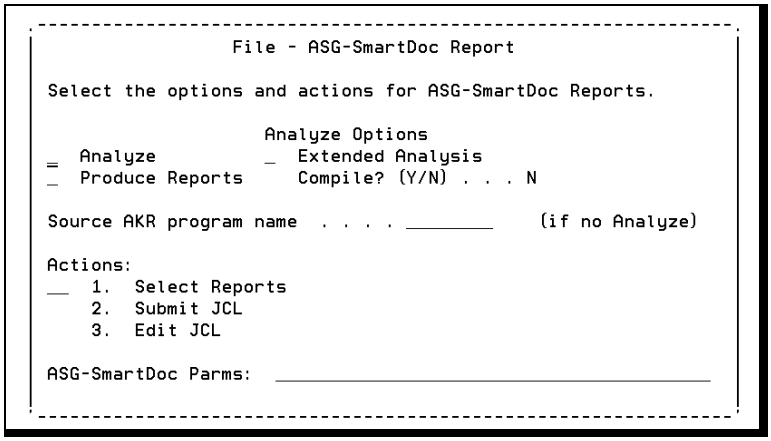
  Or

  Select File ▸ Reports and press Enter (see Figure 59 on page 112).

Note:

You can perform a SmartDoc analysis with another product analysis. Also, you can perform a SmartDoc analysis either with or without generating the SmartDoc reports. Generate the reports or the additional reports later, without analyzing the program again.

Figure 61 • File - SmartDoc Report Pop-up



Fields

These are the File - SmartDoc Report pop-up fields:

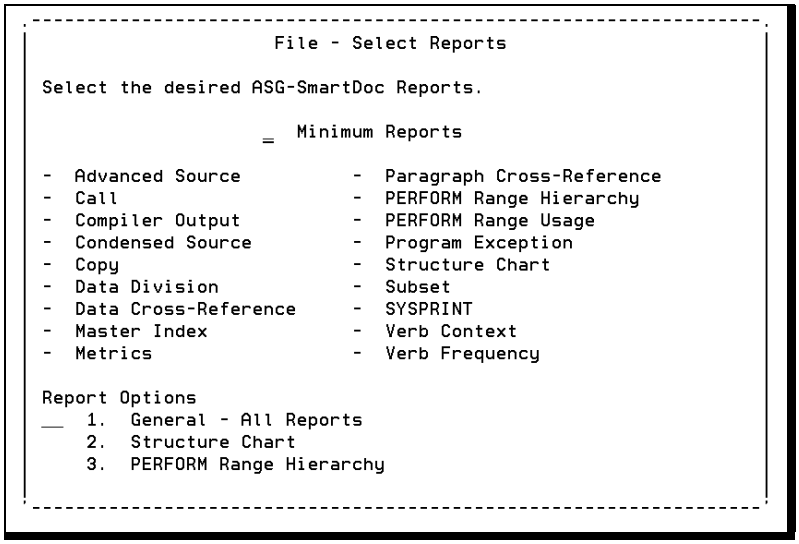
Field	Description	
Analyze Options	Enter a non-blank character to select an analyze option:	
	Analyze	Specifies that SmartDoc analyzes the current program. If a program in the current AKR was already analyzed by SmartDoc, you can generate reports without running another analysis job. In this case, do not select Analyze.
	Extended Analysis	
	Product Reports	Specifies data flow analysis be performed for SmartDoc reports. Select this option to generate the reports selected on the File - Select Reports pop-up when the program is analyzed. If you only want to run a compile/analyze, leave this option blank.
Compile	Enter either `Y` or `N` to execute a compile before the current program is analyzed. Enter `N` if you want to analyze a program without generating reports.	
Source AKR program name	Enter the program to generate SmartDoc reports. This field is only required when a SmartDoc analysis was done and the program resides in the AKR.	

Field	Description
Actions	Enter the desired action number: Select Reports — Displays the File - Select Reports pop-up to specify the reports generated. S - Submit JCL — Submit the JCL to analyze and/or generate reports for the specified program. E - Edit JCL — Displays the JCL for the compile/analyze job in the ISPF editor for review. When editing is complete, enter the ISPF SUBMIT to execute the JCL. Optionally, you can use the ISPF CREATE command to save the edited JCL in a partitioned dataset. Otherwise, the changes you make now are not saved. **Note:** The VIASUB edit macro cannot be used now since it has been updated.
SmartDoc Parms	Enter any desired SmartDoc execution options. To concatenate options, separate them by a comma or a space.

File - Select Reports Pop-up

Use the File - Select Reports pop-up (see Figure 62) to select and customize the generated reports.

Figure 62 • File - Select Reports Pop-up



Fields

Type a non-blank character in one of these fields to generate a SmartDoc report:

Field	Description
Minimum Reports	This field generates the Advanced Source Listing and the Enhanced Data Cross-Reference report for the specified program. The Compiler/Optimizer Output is also generated if a compile was performed. All other reports are suppressed.
Advanced Source Call	This field generates the Advanced Source Listing for the specified program.
Call	This field generates the Call Statement report for the specified program. `N` suppresses the report.
Compiler Output	This field generates the Compiler/Optimizer Output for the specified program.
Condensed source	This field generates the Condensed Source Listing for the specified program.
Copy	This field generates the Copy Statement report for the specified program.
Data Division	This field generates the Data Division report for the specified program.
Data cross-reference	This field generates the Enhanced Data Cross-Reference report.
Master Index	This field generates the Master Index for the specified program.
Metrics	This field generates the Metrics report for the specified program.
Paragraph cross-reference	This field generates the Paragraph Cross-Reference report.
PERFORM Range Hierarchy	This field generates the Perform Range Hierarchy Chart for the specified program.
PERFORM Range Usage	This field generates the PERFORM Range Usage report for the specified program.
Program Exception report	This field generates the Program Exception report for specified program.
Structure Chart	This field generates the Structure Chart for the specified program.

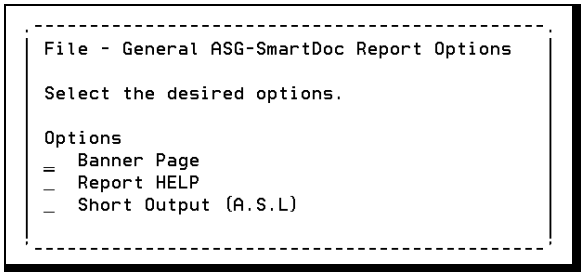
Field	Description	
Subset	This field generates the Subset report for the specified program.	
SYSPRINT	Selecting this field causes the ESW monitor to create a separate compiler output file. Input this file to a post-processor or use it for other user-specified processing.	
Verb Context	Selecting this field generates context information and includes it on the Verb Summary report.	
Verb Frequency	Selecting this field generates the Verb Frequency Table and includes it in the Verb Summary report.	
Report Options	Enter the number of one of these options:	
	1. General - All Reports	Displays the File - General SmartDoc Options pop-up to specify options for report output.
	2. Structure Chart	Displays the File - Structure Chart Options pop-up to specify the content and format of the Structure Chart.
	3. PERFORM Range Hierarchy	Displays the File - PERFORM Range Hierarchy pop-up to specify the content of the PERFORM Range Hierarchy Chart.

See Chapter 12, "SmartDoc Options," on page 189 for more option information.

File - General SmartDoc Report Options Pop-up

Use the File - General SmartDoc Report Options pop-up (see Figure 63) to specify if reports are to contain a banner page and report help. You can also condense the Advanced Source Listing to only include SmartDoc cross-reference information.

Figure 63 • File - General SmartDoc Report Options Pop-up Options



Options

To select the desired options, type a non-blank character one of these fields preceding the option:

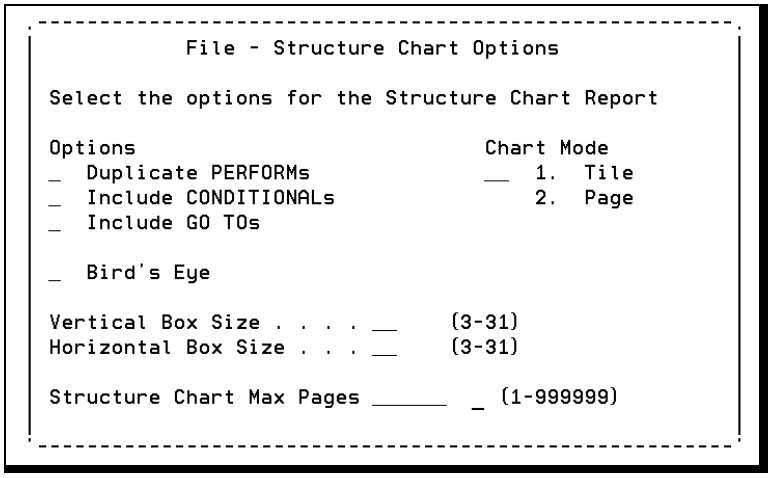
Field	Description
Banner Page	This field includes a banner page that precedes the Table of Contents for the generated reports.
Report Help	This field includes descriptive information about the report and its contents on the first page of the report.
Short Output	This field specifies that only cross-reference information appear on the Advanced Source Listing when an extended SmartDoc analysis is performed. Select this option to improve the Advance Source Listing readability when analyzing large programs that might produce many overflow lines.

See Chapter 12, "SmartDoc Options," on page 189 for more option information.

File - Structure Chart Options

Use the File - Structure Chart Options pop-up (see Figure 64) to specify the content and format of the Structure Chart.

Figure 64 • File - Structure Chart Options Pop-up



Options

To select an option, type a non-blank character one of these option fields:

Field	Description
Duplicate PERFORMs	This field specifies that PERFORM ranges be duplicated on the PERFORM Range Hierarchy report and the Structure Chart for each use. If Duplicate PERFORMs is not selected, the PERFORM range is shown where it is first used. Subsequent uses refer to the first use.
Include CONDITIONALs	This field includes the structurally relevant conditional statements that affect the PERFORM, CALL, GO TO, and ALTER statements on the Structure Chart. If the Include CONDITIONALs option is selected, Include GO TOs must also be selected.
Include GO TOs	This field includes GO TO and ALTER statements on the Structure Chart, in addition to PERFORM and CALL statements.
Bird's Eye	This field generates the Bird's Eye View representation of the Structure Chart. This report is shown in Tile Mode with each box condensed to one character.
Chart Mode	Enter the desired mode number. Select Tile Mode to generate a chart designed to be taped together into one diagram. Select Page Mode to generate a chart designed to be placed in a notebook.

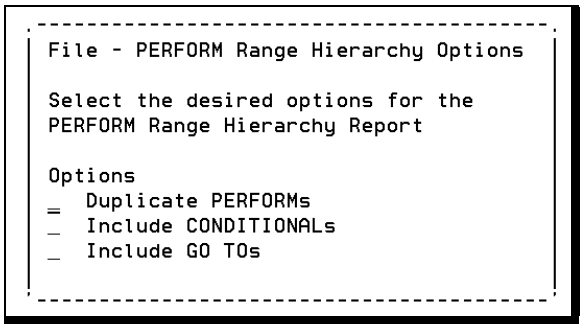
Field	Description
Vert. box size (3-31)	This field specifies the height (in lines) of each box on the Structure Chart. The minimum value that can be entered is 3; 31 is the maximum. The maximum value can be constrained by physical limitations, such as the number of lines per page. The default value for this field is 6.
Horiz. box size (3-31)	This field specifies the width (in characters) of each box on the Structure Chart. The minimum value that can be entered is 3 (6 for DBCS); the maximum value is 31. The maximum value can be constrained by physical limitations, such as the number of lines per page. The default value for this field is 9.
Structure Chart Max Pages	This field specifies the maximum number of pages generated for a Structure Chart. The value may be in the range 1 to 999999, inclusive. The default is 999999.

See Chapter 12, "SmartDoc Options," on page 189 for more option information.

File - PERFORM Range Hierarchy Options Pop-up

Use the File - PERFORM Range Hierarchy Options Pop-up (see Figure 65) to duplicate PERFORMs, include CONDITIONALs, and include GO TOs.

Figure 65 • File - PERFORM Range Hierarchy Options Pop-up



Options

To select an option, type a non-blank character in one of these option fields:

Field	Description
Duplicate PERFORMS	This field specifies that PERFORM ranges be duplicated on the PERFORM Range Hierarchy report for each use. If Duplicate PERFORMS is not selected, the PERFORM range is shown where it is first used. Subsequent uses refer to the first use.
Include CONDITIONALs	This field includes the structurally relevant conditional statements that affect the PERFORM, CALL, GO TO, and ALTER statements on the PERFORM Range Hierarchy report. If the Include CONDITIONALs option is selected, Include GO TOs must also be selected.
Include GO TOs	This field includes GO TO and ALTER statements on the PERFORM Range Hierarchy Chart, in addition to PERFORM and CALL statements.

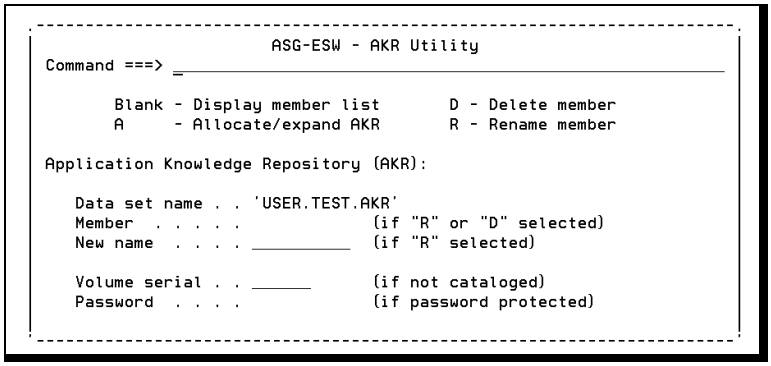
File - AKR Utility Pop-up

Use the File - AKR Utility pop-up (see Figure 66) to display the program directory, to allocate or expand an AKR, to rename a program, and to delete a program.

To display the File - AKR Utility pop-up:

- Select File ▸ AKR utility from the SmartDoc Primary Screen and press Enter.

Figure 66 • File - AKR Utility Pop-up



Options

These are the File - AKR Utility pop-up options:

Field	Description
Blank - Display member list	This field displays the File - AKR Directory pop-up that lists the program directory. You can also use the File - AKR Directory pop-up (see Figure 67 on page 125) to delete and to rename programs.
A - Allocate/expand AKR	Select option A to allocate a new AKR or to expand an existing AKR. Enter the AKR name in the Dataset name field. The File - AKR Allocate/Expand pop-up (see Figure 68 on page 129) displays. If expanding an existing AKR, type `YES` in the Expand existing AKR field on that pop-up.
D - Delete member	Select option D to delete a program. Prior to selecting option D, type the AKR and program names in the Dataset name and Program fields.
R - Rename member	Select option R to rename a program. Prior to selecting option R, type the AKR and program names in the Dataset name and Program fields; and type the new name in the New name field.

Fields

These are the File - AKR Utility pop-up fields:

Field	Description
Dataset name	Required. Enter the name of the AKR directory. If the TSO ID qualifier is the same as the user ID, type the group, type, and program without quotes. If the TSO ID qualifier is different than the user ID, type the project, group, type, and program within quotes. Alternately, the program name may be typed in the Program field.
Member	Required when either deleting or renaming a program. Enter the name of the program in the AKR; otherwise, type the program name in the Program field or in the Dataset name field.
Newname	Required when renaming a program. Enter the new name of the program. The name must be between 1 and 10 alphanumeric characters.

Field	Description
Volume serial	Required if the dataset specified in the Dataset name field is not cataloged. Enter the volume serial number. If the dataset is cataloged, this field is optional.
Password	Required if the dataset is protected. Enter the dataset password.

File - AKR Directory Pop-up

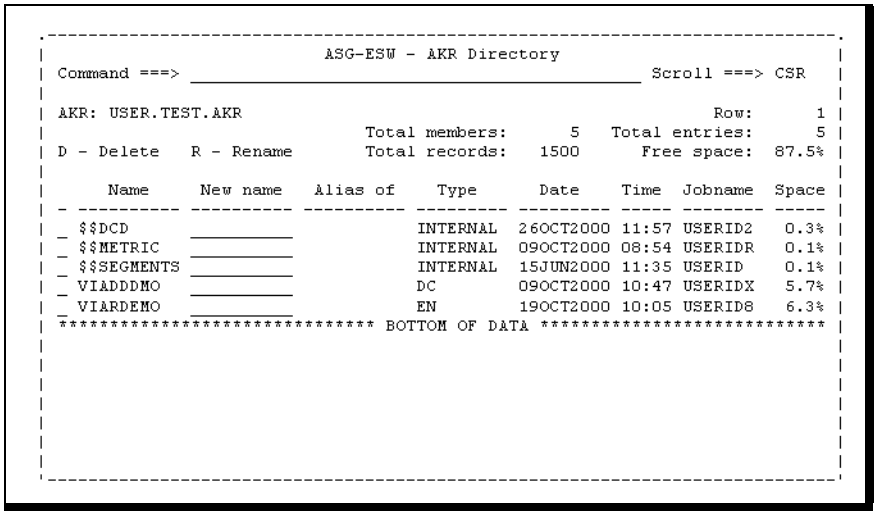
The File - AKR Directory pop-up (see Figure 67) lists all members in the specified AKR.

To display the File - AKR Directory pop-up, follow this step:

- Press Enter on the File - AKR Utility pop-up with no entry in the command input area.

Scroll this pop-up to view members not visible, or enter the L (Locate) command with a character string to view a specific member. The pop-up is scrolled to the member that most closely matches the character string.

Figure 67 • File - AKR Directory Pop-up



Fields

These are the File - AKR Directory pop-up fields:

Field	Description
AKR	This field shows the complete dataset name of the requested AKR, as entered on the File - AKR Utility pop-up (see Figure 66 on page 123).
Row	This field shows the relative number in the AKR of the first member displayed on this pop-up.
Total members	This field shows indicates the total number of members in this AKR, not including aliases.
Total records	This field shows the number of records allocated to this AKR, typed in the Space Amount field on the File - AKR Allocate/Expand pop-up (see Figure 68 on page 129) when the AKR was either allocated or expanded. If you enter the Space Units as Tracks or as Cylinders instead of Records, the amount is converted to Records and displays in the Total records field.
Total Entries	This field shows the total number of members in this AKR, including aliases.
Free space	This field shows the amount of available space in this AKR, rounded to the nearest .1 percent.
line command area	The line command area, to the left of the Name field, accepts these line commands: D - Delete Type `D` to the left of the program to delete it from the AKR. Alias programs cannot be deleted. Programs can also be deleted on the File - AKR Utility pop-up (see Figure 66 on page 123). R - Rename Type `R` to the left of the program rename it. Then type the new name in the New name field. Note that alias programs cannot be renamed. When a primary program name is changed, the alias name is automatically changed. Programs can also be renamed on the File - AKR Utility pop-up (see Figure 66 on page 123).

Field	Description
Name	This field shows the AKR member name. Program names are taken from the PROGRAM-ID statement. If the analyzed program contains an ENTRY point, Name is the ENTRY point name. If you overrode the name in the PROGRAM-ID statement when you submitted the analyze job, Name is the name typed in the AKR program name field on the Analyze Submit pop-up. See the Analyze Submit pop-up (see Figure 60 on page 113) for more information on entries in the AKR.
New name	Required when renaming a member. Enter the new member name. The member name must be 1 through 10 alphanumeric characters.
Alias Of	If an analyzed program contains an ENTRY point, Alias Of is the name of the program containing the ENTRY point. If you overrode the name in the PROGRAM-ID statement when the analyze job was submitted, Alias Of is the name typed in the AKR program name field on the Analyze Submit pop-up. An alias program cannot be deleted or renamed. The alias name is automatically changed when the primary program is either deleted or renamed. See the Analyze Submit pop-up (see Figure 60 on page 113) for more information on alias names in the AKR.
Type	This field shows the type of analysis performed on the member. The type of analysis is specified on the Analyze Submit pop-up (see Figure 60 on page 113) and can be one of these entries: AL — Indicates an Alliance analysis was performed. ST — Indicates a SmartTest analysis was performed. IN — Indicates a Insight analysis was performed. The IN value displays as a result of a Insight analyze job. This value is entered on the Analyze Submit pop-up and can only be specified if the Insight product is installed. DS — Indicates a SmartDoc analysis was performed. DC — Indicates a SmartDoc analysis with a COBOL compile was performed. DX — Indicates an Extended SmartDoc analysis was performed. DA — Indicates an Extended SmartDoc analysis with a COBOL compile was performed.

Field	Description	
	ASM	Indicates the program is an Assembler source program.
	EN	Indicates a Encore analysis was performed. The Renaissance RN analysis type is currently still a valid analysis type, but will not be supported in future releases of Encore.
	RC	Indicates a Recap analysis was performed.
	PROFILE	This is a SmartTest feature that indicates the member contains profile information. Generally, the member name is the user ID for which the profile was created. Profile information is automatically saved when the SmartTest Session Tailoring screen is used to specify the testing environment options. After program level testing options are specified on the Session Tailoring screen (and saved in the AKR profile member), the profile member is used by SmartTest when a test session is initiated. A profile is only used (and updated) by the user for whom it is created. Any modifications to the Session Tailoring screen are automatically reflected in the profile member.
	METRICS	This is a SmartDoc feature that indicates the member contains metrics information. The Name field contains $$METRIC or the name you assigned to the metrics by using the Rename function. The $$METRIC member contains metrics that have been calculated for the programs in this AKR.
	INTERNAL	This is a Encore feature that indicates the member contains logic segment information. The Name field contains $$SEGMENTS or the name you assigned to the member by using the Rename function. The $$SEGMENTS member contains the logic segment information.
Date	The date the program was analyzed.	
Time	This field shows the time the program was analyzed.	
Jobname	This field shows the JOB NAME used to analyze the program.	
Space	This field shows the percentage of space the program is using on this AKR, rounded to the nearest .1 percent.	

File - AKR Allocate/Expand Pop-up

Use the File - AKR Allocate/Expand pop-up (see Figure 68) to allocate a new AKR, or to expand an existing AKR.

To display the File - AKR Allocate/Expand pop-up

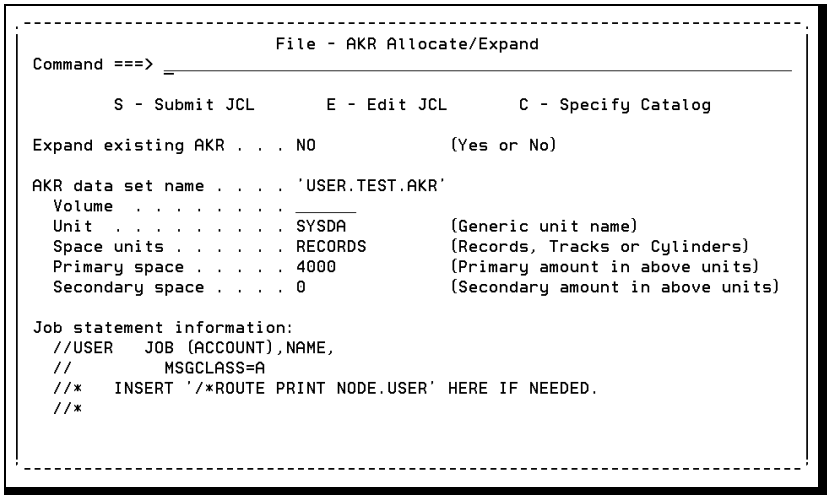
1 Select File ▸ AKR Utility and press Enter (see Figure 66 on page 123).

2 Type `A` and press Enter.

To expand an existing AKR, type `YES` in the Expand existing AKR field on this pop-up. To allocate a new AKR, type `NO` in this field.

The space needed for the AKR depends on the size and the number of COBOL programs analyzed and placed in it. See the Space amount field in the Fields section below for more information.

Figure 68 • File - AKR Allocate/Expand Pop-up with SMS and a VSAM AKR



Options

These are the File - AKR Allocate/Expand pop-up options:

Field	Description
S - Submit JCL	Type S to submit the JCL to allocate or to expand the AKR shown in the AKR dataset name field.
E - Edit JCL	Type E to edit the JCL to allocate or to expand the AKR shown in the AKR dataset name field.
C - Specify Catalog	Type C to display the AKR Catalog Information pop-up used to specify the dataset name and password of a private catalog to contain the AKR.

Fields

These are the File - AKR Allocate/Expand pop-up fields:

Field	Description
Expand existing AKR	Required. Enter YES to expand the AKR shown in the AKR dataset name field. Enter NO to allocate a new AKR. The default is NO.
AKR dataset name	The AKR name specified on the File - AKR Utility pop-up (see Figure 66 on page 123). **Note:** Management Class, Storage Class, and Data Class provide various parameters for newly allocated datasets. These parameters apply only if you have SMS active at your site. Your system administrator determines the valid entries for these parameters.
Management Class	Optional. Enter the management class for a new AKR. The default is MGMTCLAS.
Storage Class	Optional. Enter the storage class for a new AKR. The default is STORCLAS.
Volume	Required. Enter the volume serial number where the AKR is to reside.
Data Class	Optional. Enter the data class for a new AKR. Data class is valid only if SMS is active at your site. The default is DATACLAS.
Space units	Optional. Enter the type of units to allocate. The units must be records, tracks, or cylinders. The default is records.

Field	Description
Primary space/Secondary space	Required if you are not using SMS. Enter the amount of space to allocate in the type of units specified in the Primary Space Units field. Enter the amount of secondary space if required. The space needed for the AKR depends on the size and number of programs analyzed and placed in it. See the *Center Installation Guide* for space estimates.
Unique	Optional. Enter `NO` if the AKR is a sub-allocation in a VSAM data space on the volume. The default is `YES`.
Job statement information	Optional. Enter appropriate JOB statement information for your site.

7 View

This chapter discusses the View pull-down used to view metrics data for the programs that reside in the AKR and contains these sections:

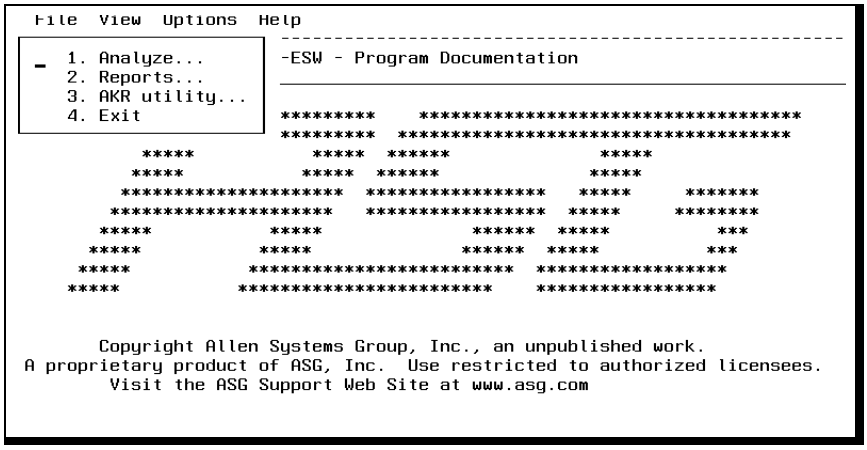
View Pull-down

Use View pull-down (see Figure 69) to view metrics data for the programs that reside in the AKR.

To display the View pull-down, follow this step:

- Select View on the action bar and press Enter.

Figure 69 • View Pull-Down



Action

Metrics View. Displays the View - Open Metrics Repository pop-up (see Figure 70) to specify the program metrics viewed.

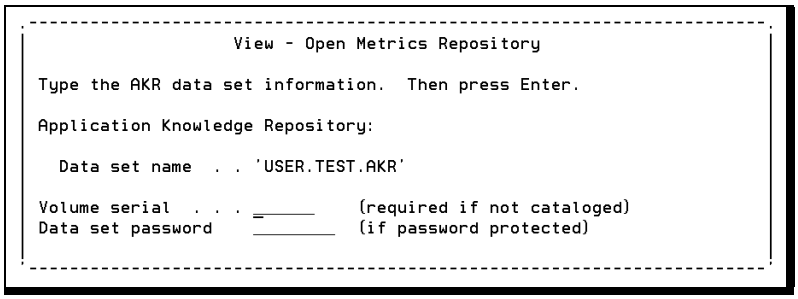
View - Open Metrics Repository Pop-up

Use the View - Open Metrics Repository pop-up (see Figure 70) to specify the AKR for the displayed metrics.

To display the View - Open Metrics Repository pop-up, follow this step:

- Select View ▸ Metrics View and press Enter (see Figure 69 on page 131).

Figure 70 • View - Open Metrics Repository Pop-up



Fields

These are the View - Open Metrics Repository pop-up fields:

Field	Description

Data set name	If the TSO ID qualifier is the same as the user ID, enter the library, type, and program without quotes.
	`TEST.AKR(pgmname)`
	If the TSO ID qualifier is different than the user ID, enter the project, library, type, and program within quotes.
	`'TSOID.TEST.AKR(`*`pgmname`*`)'`

Volume serial	Required if the dataset is not cataloged. Enter the volume serial number. If the dataset is cataloged, this field is optional.
Data set password	Required if the dataset is protected. Enter the dataset password.

Program Metrics View Screen

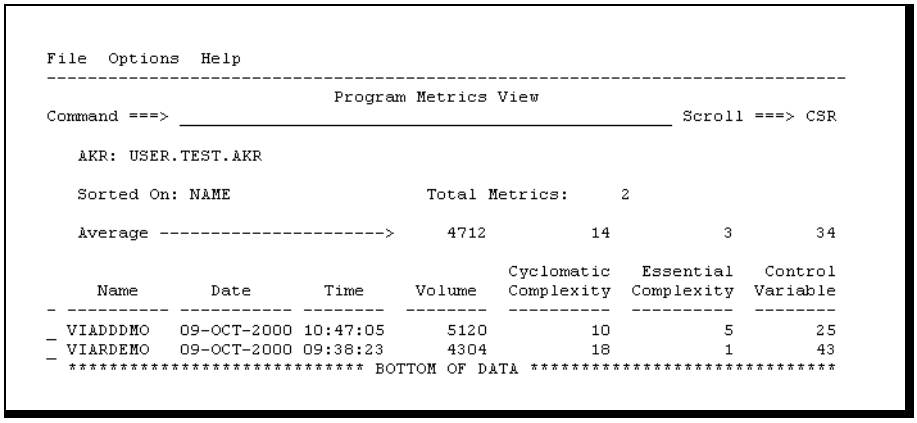
Use the Program Metrics View screen (see Figure 71) to display the latest metrics generated for a program.

To display the Program Metrics View screen, follow this step:

- Type the AKR dataset name on the View - Open Metrics Repository pop-up (see Figure 70 on page 132) and press Enter.

Use this screen to delete or to rename the metrics for a particular program.

Figure 71 • Program Metrics View Screen



Note:

The Program Metrics View screen contains a shortened action bar.

Action Bar

These are the actions available on the Program Metrics View screen action bar:

Action	Description
File	The File pull-down contains the Exit Metrics Display action. This action displays the View - Open Metrics Repository pop-up. Specify another AKR here. Press PF3/15 to exit.
Options	The Options pull-down deletes, renames, or sorts the program metrics. The Options pull-down contains these actions: Delete Metrics: After selecting the program metrics name to delete, use this action to display the Options - Delete Confirmation pop-up to delete the metrics. On the Options - Delete Confirmation pop-up, either enter `Y` to delete the metrics or `N` to cancel the deletion. Rename: After selecting the program metrics to rename, use this action to display the Options - Rename Metrics pop-up to rename the metrics. On the OptionsRename Metrics pop-up, enter the new name and select the desired option to rename the metrics only, or to rename the program and the metrics. Sort Metrics: Use this action to display the Options - Sort Metrics pop-up to change the order of the metrics listed on the Program Metrics View screen. On the Options - Sort Selection pop-up, specify the column to sort on. Sorting of metrics are in ascending order.
Help	The Help pull-down accesses the online help facility. See the online help and Chapter 9, "Help," on page 147 and Chapter 15, "Help Facility," on page 225 for more information.

Fields

These are the Program Metrics View screen fields:

Field	Description
AKR	This field shows the AKR specified on the View - Open Metrics Repository pop-up (see Figure 70 on page 132) displays in this field.
Sorted On	This field shows the column on which the display is sorted. This can either be the Name field or one of the metric types shown.
Total Metrics	This field shows indicates the total number of programs with metrics in this AKR.

Field	Description
AKR Average	This field shows the average metrics value of all programs in the AKR for each metrics type.
line command area	This field shows the line command area, to the left of the Name field, accepts this line command: S Enter an S to the left of each program to delete or to rename, then use the Options pull-down to choose the appropriate action. See the Action Bar section of Figure 70 on page 132 for more information.
Name	This field shows the programs and metrics that reside in this AKR.
Date	This field shows the date the program was analyzed.
Time	This field shows the time the program was analyzed.
Volume	This field shows the most recent Software Science Volume metric calculated for the program.
Cyclomatic Complexity	This field shows the most recent Cyclomatic Complexity metric calculated for the program.
Essential Complexity	This field shows the most recent Essential Complexity metric calculated for the program.
Control Variable	This field shows the most recent Control Variable metric calculated for the program.

8 Options

This chapter discusses the Options pull-down used to access the pop-ups that customize the SmartDoc environment and contains these sections:

Options Pull-down

Use the Options pull-down (see Figure 72 on page 138) to access the pop-ups used to customize the SmartDoc environment. Customizing the SmartDoc environment includes defining and processing the Log file, and determining the values of the PF keys.

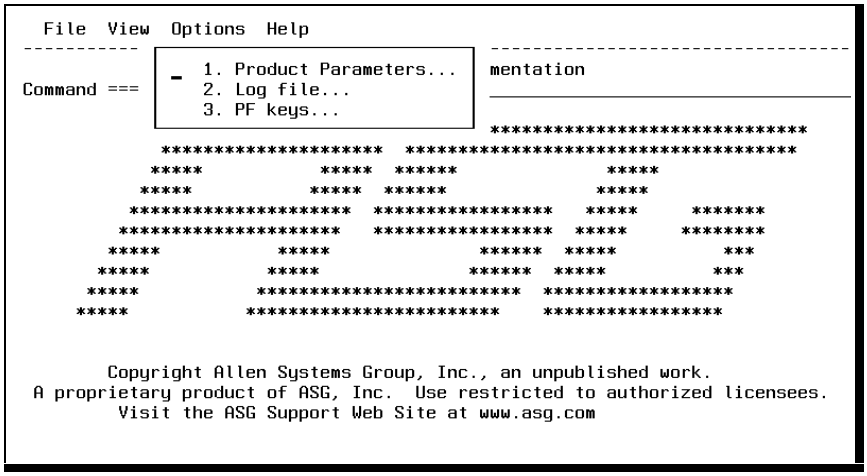
To display the Options pull down, follow this step:

- Select Options on the action bar and press Enter.

Note:

The Options pull-down contains different actions on the Program Metrics Display screen. See Figure 71 on page 133 for more information.

Figure 72 • Options Pull-down



Actions

These are the Options pull-down actions:

Action	Description
1. Product Parameters	This action displays the Options - Product Parameters pop-up (see Figure 73 on page 139) that controls the online operation of SmartDoc. This includes whether an alarm sounds when an error message displays, and defines the Log file.
2. Log file	This action displays the Options - Log File Definition pop-up (see Figure 74 on page 140) that processes the Log file. Use this file for system message logging and error handling.
3. PF keys	This action displays the Options - PF Key Definition pop-up (see Figure 75 on page 142) that displays or that changes the PF keys used with SmartDoc.

Options - Product Parameters Pop-up

Use the Options - Product Parameters pop-up (see Figure 73) to set parameters that affect the online operation of SmartDoc, and to allocate the SmartDoc Log file.

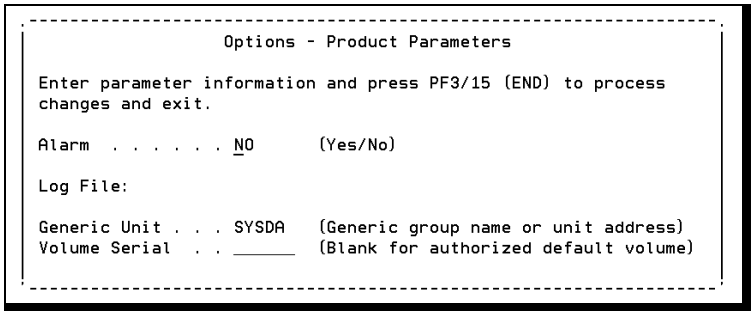
To display the Options - Product Parameters pop-up, follow this step:

- Select Options ▸ Product Parameters and press Enter.

  Or

  Enter the PARMDEF command on any screen and press Enter.

Figure 73 • Options - Product Parameters Pop-up



Fields

These are the Options - Product Parameters pop-up fields:

Field		Description
Alarm		Required. This field controls the audible alarm on the terminal. If YES, the alarm sounds when displaying an error message. If NO, the alarm does not sound. The default is YES.
LOG FILE	Generic Unit	Required. Enter the device type for the Log file allocated upon entry into SmartDoc. The Log file is used for error messages and log commands. Specify file characteristics on the Options - Log File Definition pop-up (see Figure 74 on page 140).
	Volume Serial	Optional. Enter the volume serial number for the Log file.

Options - Log File Definition Pop-up

Use the Options - Log File Definition pop-up (see Figure 74) to set values for allocating, formatting, and processing the SmartDoc Log dataset. SmartDoc uses this file for error messages and log commands.

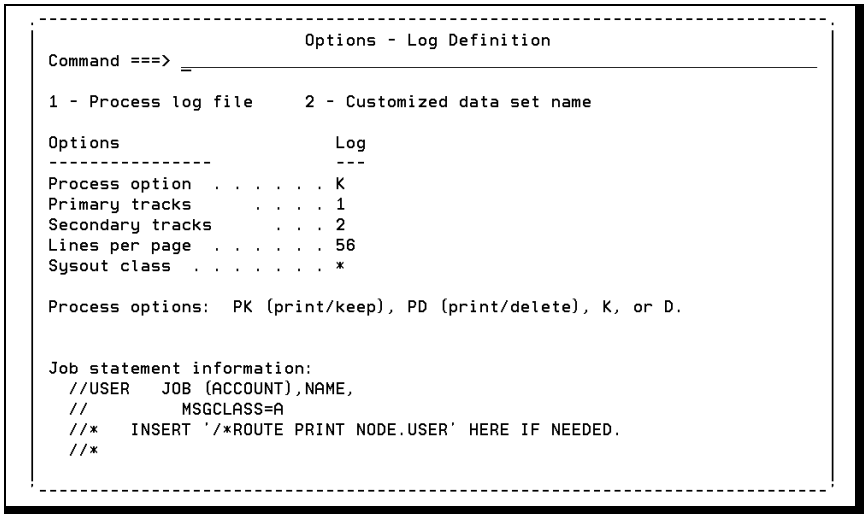
To display the Options - Log File Definition pop-up, follow this step:

- Select Options ▸ Log file and press Enter.

  Or

  Enter the PRINTLOG command on any screen and press Enter.

Figure 74 • Options - Log File Definition Pop-up



Options

These are the Options - Log File Definition pop-up options:

Option	Description
1 - Process log file	Either verify or change the Options, then select 1 to print and/or de-allocate the Log file. A new file is allocated to collect additional data, if required.

Option	Description
	If either the PK or PD Process option is specified, you must enter Job statement information prior to selecting this option. It is not necessary to exit SmartDoc to print the Log file.
2 - Customized data set name	Displays the Options - Log Customization pop-up which enables you to specify a dataset name where the Log file will be saved. By default, these files are allocated as: USERID.*yyyxxxxx*.VIALOG where: *yyy* is the product ID. *xxxxx* is a sequential number from 000001 to 99999.

Fields

These are the Options - Log File Definition pop-up fields:

Field	Description
Process option	Required. Enter one of the listed processing options. The default is PD.
Primary tracks	Required. Enter the number of primary tracks to allocate. A size change does not take effect until the next allocation occurs. The default is 1.
Secondary tracks	Required. Enter the number of secondary tracks to allocate. A size change does not take effect until the next allocation occurs. The default is 1.
Lines per page	Required. Enter the number of print lines per page. Typical maximum values are 60 for six lines per inch and 80 for eight lines per inch. The default is 56.
Sysout class	Required. Enter the SYSOUT class value. The default is asterisk (*) which sends the SYSOUT to the destination specified in the MSGCLASS parameter on the JOB statement.
Process options	This field lists the available options for the Log file: PK Print and keep PD Print and delete K Keep without printing D Delete without printing

Field	Description
LOG FILE IS ALLOCATED	This message displays when the Log file has been properly allocated. If the message does not appear, check the assignments on the Options - Product Parameters pop-up (see Figure 73 on page 139).
Job statement information	Enter the appropriate JOB statement information for your installation. These JCL statements are required if either the PK or PD Process option is specified.

Options - PF Key Definition Pop-up

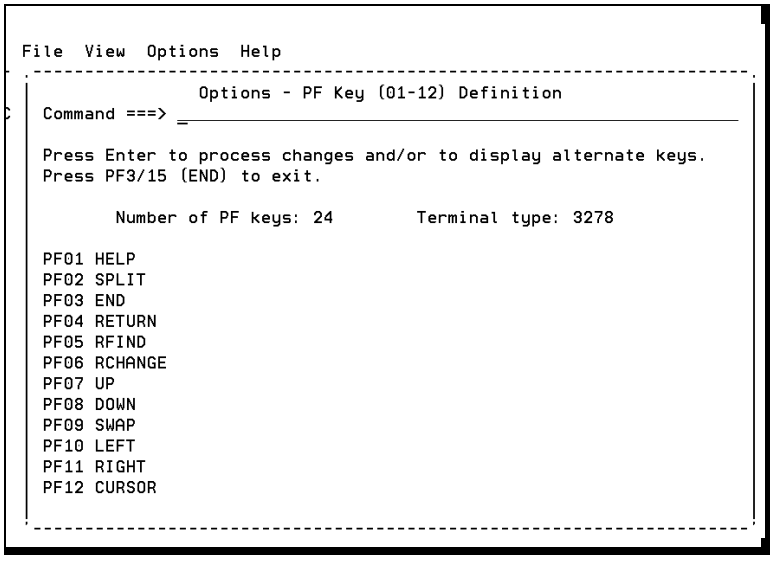
Use the Options - PF Key Definition pop-up (see Figure 75) to display and/or redefine PF key values.

To display the Options - PF Key Definition pop-up

1 Select Options ▸ PF keys and press Enter.

2 Enter the KEYS command on any screen and press Enter. PF keys 1 through 12 display initially. A similar screen exists for PF keys 13 through 24. Press Enter to display.

Figure 75 • Options - PF Key Definition Pop-up



Fields

These are the Options - PF Key Definition pop-up fields:

Field	Description
Number of PF keys	This field shows the number of supported PF keys.
Terminal type	The number in this field indicates the type of terminal being used. SmartDoc supports these 3270 type terminals: Model 2 (24 lines x 80 columns) Model 3 (27 lines x 80 columns) Model 4 (43 lines x 80 columns) Model 5 (27 lines x 133 columns)
PF1 - PF12	The value assigned to PF keys 1 through 12. You can assign any command or any data value to a PF key.
PF13 - PF24	The value assigned to PF keys 13 through 24.

9 Help

This chapter discusses the Help pull-down used to access the Online Help facility and contains these sections:

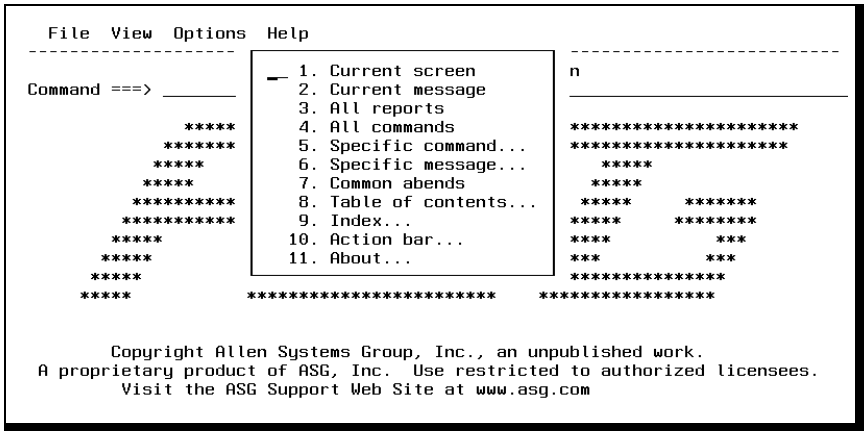
Help Pull-down

Use the Help pull-down (see Figure 76) to access the Online Help facility.

To display the Help pull-down, follow this step:

- Select Help on the action bar and press Enter.

Figure 76 • Help Pull-down



Actions

These are the Help pull-down actions:

Action	Description
1. Current screen	This action displays help for the screen or the pop-up currently displayed.
2. Current message	This action displays help for the message currently displayed.
3. All reports	This action displays a complete list of all SmartDoc reports, where information about a specific report is displayed by selecting the appropriate number.
4. All commands	This action displays a complete list of all SmartDoc primary commands, where information about a specific command is displayed by selecting the appropriate number.
5. Specific command	This action displays the Help - Specific ASG-ESW Command pop-up (see Figure 77 on page 149) used to obtain help about a specific SmartDoc primary command.
6. Specific message	This action displays the Help - Specific ASG-ESW Message Number pop-up (see Figure 78 on page 150) used to obtain help about a specific message number.
7. Common abends	This action displays the Abends screen, where information about a specific abend is displayed by selecting the appropriate number. Select topic 2 on this screen to display the ASG-ESW Abend Codes screen, which lists all the ESW user abends and their explanations.
8. Table of contents	This action displays the online help table of contents, listing general information about SmartDoc. See Chapter 15, "Help Facility," on page 225 for more information and an example of the online help table of contents.
9. Index	This action displays the first online help index screen. See Chapter 15, "Help Facility," on page 225 for more information and an example of the online help index.
10. Action bar	This action displays the Help Tutorial for the SmartDoc action bar.
11. About	This action displays the Help - About pop-up (see Figure 79 on page 151) that lists information about the currently installed levels of SmartDoc and Center.

Usage Notes

See "Screen Help" on page 227 for more information about the Online Help Facility.

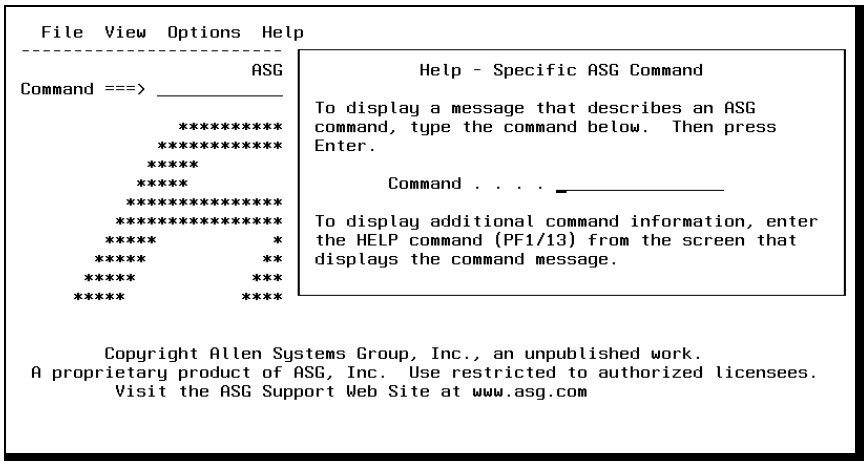
Help - Specific ASG-ESW Command Pop-up

Use the Help - Specific ASG-ESW Command pop-up (see Figure 77) to obtain help about a specific SmartDoc primary command.

To display the Help - Specific ASG-ESW Command pop-up, follow this step:

- Select Help ▸ Specific command and press Enter.

Figure 77 • Help - Specific ASG-ESW Command Pop-up



Fields

Command. Required. Enter a SmartDoc primary command. The Help Tutorial for that command displays. See Chapter 15, "Help Facility," on page 225 for more information.

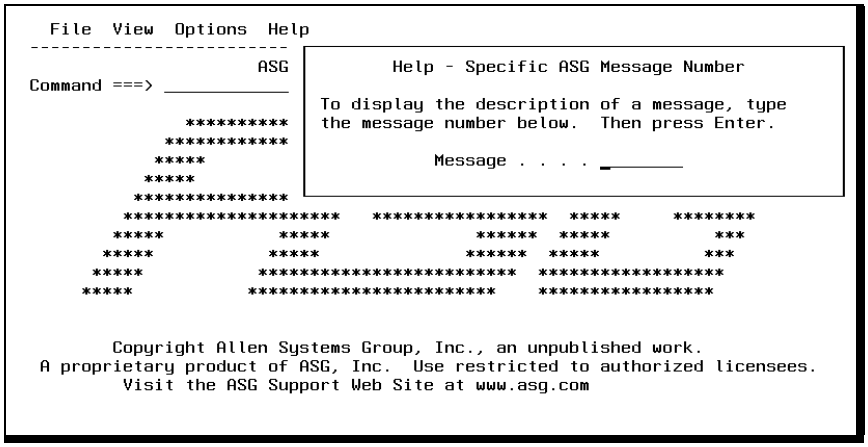
Help - Specific ASG-ESW Message Number Pop-up

Use the Help - Specific ASG-ESW Message Number pop-up (see Figure 78) to display help for a specific ESW message.

To display the Help - Specific ASG-ESW Message Number pop-up, follow this step:

- Select Help ▸ Specific message and press Enter.

Figure 78 • Help - Specific ASG-ESW Message Number Pop-up



Fields

Message. Required. Enter a ESW message number. The Help Explanation and Action screen for that message displays. See Chapter 15, "Help Facility," on page 225 for more information on the Help Explanation and Action screen.

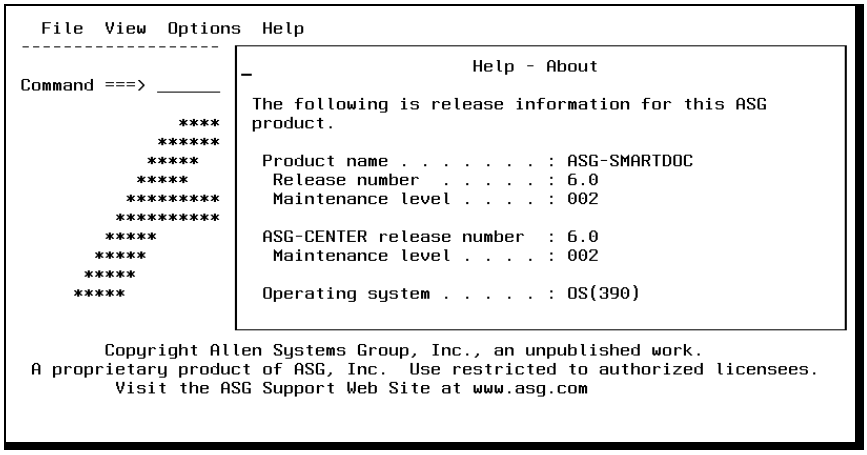
Help - About Pop-up

Use the Help - About Pop-up (see Figure 79) to display information about the currently-installed releases of SmartDoc and Center.

To display the Help - About Pop-up, follow this step:

- Select Help ▸ About and press Enter.

Figure 79 • Help - About Pop-up



Fields

These are the Help - About pop-up fields:

Field	Description
Product name	This field shows the name SmartDoc or the name of the ESW product you are currently running.
Release number	This field shows the currently installed release number of SmartDoc.
Maintenance level	This field shows the currently installed SmartDoc level number.
ASG-Center release number	This field shows the currently installed release number of Center.
Center maintenance level	This field shows the currently installed Center level number.
Operating system	This field shows the operating system you are running.

Note:

The ASG Service Desk requests this information when you contact them for assistance. This information can also be displayed by entering the PRODLVL command on any screen.

10 Metrics

This chapter discusses the SmartDoc generated metrics used in the management of the program maintenance life cycle and contains these sections:

Introduction to Metrics

Metrics gives you information about the complexity and quality of a program that helps you manage the program maintenance life cycle. Metrics are best used when you are comparing programs of a similar nature. For example, you should not compare a batch program that writes reports with a CICS accounts payable program.

Listed below is a sample procedure for using SmartDoc metrics to manage the maintenance life cycle of a program.

To use SmartDoc metrics to manage the maintenance life cycle of a program

1. Use SmartDoc to analyze each program.
2. From the Program Metrics View pop-up (see Figure 71 on page 133), sort the display on each of the metric types, and print the results.
3. Group the programs by purpose and calculate an average for each metrics group, or use the average shown on the Program Metrics View pop-up.
4. Calculate a standard deviation for each metric in the group (percent of average). For additional information, see Chapter 4, "Techniques," on page 23.

5 Calculate the number of standard deviations from the average for each metric on each program. For additional information, see Chapter 4, "Techniques," on page 23.

6 Review programs with more than one standard deviation above average for the Software Science Volume metric for re-engineering into smaller, independently called programs.

7 Programs that have more above average Cyclomatic Complexity and Control Variable standard deviations than their Software Science Volume values are particularly complex for their size. Consider re-engineering or restructuring these programs, as they can be extremely difficult to maintain and enhance.

When you analyze a new program (metrics calculated), compare its metric values against existing programs. Use this information to assess the long term maintenance resources required to maintain the new program.

Storing Program Metrics

The AKR stores software metrics information as a separate member (i.e., stores it separately from the actual program). The metrics information remains intact if the program is deleted from the AKR. When you use the AKR Utilities to rename a program the corresponding metrics member is not renamed. To rename or to delete either a program or metrics, use the Program Metrics View pop-up.

Unlimited versions of metric data are retained for each program. However, due to page size limitations, only the twenty most recent versions are shown on the metrics reports.

Renaming or Deleting Metrics When ISPF is Not Installed

SmartDoc provides you with options for either renaming or deleting metrics when ISPF is not installed. Enter these options in the DPARM parameter of the VIASANJC JCL. Use the DPARM parameter to specify all SmartDoc reporting options when ISPF is unavailable.

To rename metrics for a program, use the RENAME(*pgm*) option along with the PGM=*pgmnm* option, where pgm is the new name for the metrics and pgmnm is the name of the metrics in the AKR to be renamed. Only the metrics are renamed when you use the RENAME(*pgm*) option. To rename the program, use the VIASAKRU JCL.

To delete metrics for a program, use the DELMET option along with the PGM=*pgmnm* option, where *pgmnm* is the name of the metrics in the AKR being deleted. The program itself is not deleted when you use the DELMET option.

Note:

When you use the RENAME or the DELMET options, use only the SmartDoc report generation (SDR) feature to execute SmartDoc. RENAME and DELMET prevent SmartDoc from generating reports and cannot be used if reports are to be generated.

11 Analyze

This chapter discusses the analyze process used by SmartDoc and contains these sections:

Overview

Note:
A program must be analyzed before ESW products can provide information about it.

The analyze process gathers information about the program, such as program relationships, logic, data, and execution paths. The AKR stores this information and makes it available to ESW products in online and batch environments.

Analyzing a COBOL Program

The analyze process is similar to a COBOL compile. The process has these three primary inputs:

- Source COBOL program (including copy books)
- JCL used to compile and link the COBOL program
- Options and features for tailoring the analyze steps

Each input item is described in the next sections.

Analyze Input Descriptions

COBOL Source Program

Like the compiler, the analyze process requires basic program standards. Basic program standards include:

- The COBOL language as specified in the IBM OS/VS COBOL, COBOL II, and COBOL/370 Language Reference guides is accepted by the analyze job. The analyze job correctly processes any program compiled without either warnings or errors by the IBM OS/VS COBOL or the COBOL II compilers.
- OS/VS COBOL programs that receive conditional (C), error (E), or disaster (D) messages from the IBM compiler cannot be successfully analyzed.
- COBOL II and COBOL/370 programs that receive error (E), severe (S), or unrecoverable (U) messages from the IBM compiler cannot be successfully analyzed.

The program analyzer resource estimates for processing various sizes of COBOL programs are shown below.

The information in these tables resulted from the running of Analyze under this criteria:

- Version - Center R5.0
- CPU Type - 3090-600 running MVS/ESA
- Disk Type - 3390
- Analyze Parms - BUFMAXKB=4096 KB
- Compiler Parms - BUF=256 KB, SIZ=1024

SmartDoc Short Analysis Resource Estimates

Source Lines	Virtual Memory Size	XA Memory Size	CPU Time MM:SS	AKR Blocks	VIAUT2 Cyls
1000	1060 KB	12100 KB	0:04	150	2
2000	1060 KB	12200 KB	0:10	300	3
5000	1060 KB	12500 KB	0:30	750	6
10000	1060 KB	12800 KB	1:00	1500	12
20000	1060 KB	140000 KB	2:00	3000	24
50000	1060 KB	16000KB	5:00	6000	50

SmartDoc Extended Analysis Resource Estimates

Source Lines	Virtual Memory Size	XA Memory Size	CPU Time MM:SS	AKR Blocks	VIAUT2 Cyls
1000	256 KB	12100 KB	0:03	250	2
2000	256 KB	12800 KB	0:15	600	5
5000	256 KB	16000 KB	1:00	2000	15
10000	256 KB	20000 KB	5:00	5000	40
20000	256 KB	28000 KB	15:00	10000	80
50000	4096 KB	48000 KB	60:00	30000	240

Compile/Link JCL

The compile/link JCL is the complete JCL used to compile the program. Specifically, the JCL should contain steps to accomplish these actions:

- Fetch the source from the source manager (such as Librarian or Panvalet)
- Execute the preprocessors
- Invoke the compiler with the appropriate options and COPY libraries
- Invoke the linkage editor if desired

Analyze Options and Features

Use these analyze features to indicate the type of analysis performed:

Analysis	Features
Encore analysis	This analysis provides the information required for code extraction and execution flow capabilities
SmartTest analysis	This analysis provides the testing and debugging information required by SmartTest
Extended SmartTest analysis	This analysis provides comprehensive program analyzing capabilities in addition to the testing and debugging capabilities of a SmartTest analysis
Insight analysis	This analysis provides logic and execution flow capabilities
SmartDoc analysis	This analysis provides the information required for SmartDoc reports
Extended SmartDoc analysis	This analysis provides data flow analysis.

Default options for the analyze process are established at installation time. When you submit the analyze job, specify the options to be overridden.

The Analyze Process

The analyze process consists of setting up and executing a batch job. The method you use depends on the environment where the analyze process is invoked, but also may depend on the access method containing the compile/link JCL. These are the four methods used to invoke the analyze process:

Method	Description
Analyze Option or Command	To display the Analyze Submit pop-up, Select the File ▸ Analyze. **Or** Enter the ANALYZE command from any screen and press Enter. Enter the required input and output information, then submit the job.
ISPF	From any ISPF screen, execute the VIASUBDS CLIST. This CLIST is executed by typing: `TSO VIASUBDS dsn parms` where: *dsn* is a PDS member or a sequential dataset containing the compile JCL *parms* represents any of the available execution parameters described in the VIASUBDS and VIASUB Parameters table contained in "Analyze Using ISPF/PDF Edit" on page 163.
ISPF/PDF Edit	Execute the VIASUB PDS edit macro. This edit macro is executed by typing this syntax: `VIASUB parms` where *parms* represents any of the available execution parameters described in the VIASUBDS and VIASUB Parameters table contained in "Analyze Using ISPF/PDF Edit" on page 163.
VIASANJC JCL	Execute the VIASANJC JCL to convert the compile and link JCL, then execute the converted JCL to analyze the program. This method is used when ISPF is not installed.

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

Compile JCL is From	Method for Executing Analyze Job
PDS or Sequential dataset	Analyze Submit Screen, VIASUBDS CLIST, VIASUB edit macro, or VIASANJC JCL
Librarian, 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

Analyze Using ISPF

Use the VIASUBDS CLIST to submit the analyze job from any ISPF screen. 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, the PANEL parameter is entered 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. A list of these parameters, with the default parameters underlined, is found in the VIASUBDS and VIASUB Parameters table contained in "Analyze Using ISPF/PDF Edit" on page 163.

Note:

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

Analyze Using ISPF/PDF Edit

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

```
VIASUB parms
```

Where *parms* is one or more parameters that control the operation of VIASUB. Typically, you use the PANEL parameter to display the Analyze Submit Parameters screen in order to enter necessary parameters. The ISPF profile saves the parameters and uses them 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 VIASUB parameters:

Parameter	Description
AKR (*xxxxx*)	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, use triple quotes. For example: `AKR ('''ASG, VIACEN`*xx*`.AKR''')`
AOPT(*xxxxx*)	Specifies options supplied to the analyze job. The COBOL II option is automatically added if the compiler specified in the input JCL is COBOL II. If you specify more than one analyze option, they should be separated by commas and enclosed in single quotes. For example: `AOPT ('XMEM, RECUR,SUBSYS=D239').` See "Analyze Options" on page 175 for information on each analyze option.
CMPL NOCMPL	CMPL indicates a COBOL compile and an analysis is 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 bypasses subsequent job steps (e.g., a link edit) based on a successful compilation. NOCMPL cannot be specified if a SmartTest analysis is being executed.

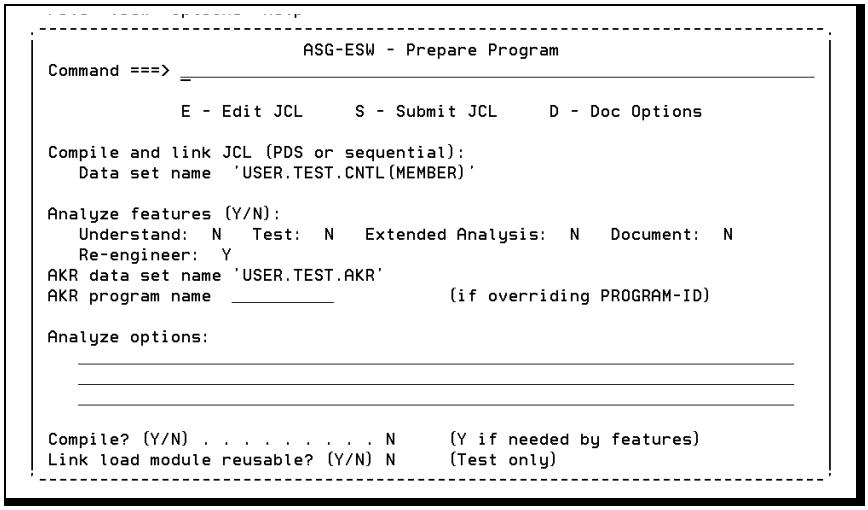
Parameter	Description
DSCHK NODSCHK	DSCHK indicates datasets needed by the resulting JCL are verified. Specifically, the AKR and the load library containing VIASMNTR are checked. When NODSCHK is specified, the AKR and the load library need not exist at the time the VIASUB or the VIASUBDS is executed. 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 to VIASUB.
EDIT	EDIT specifies that the resulting JCL is not submitted for batch processing. The PDF editor is invoked for the resulting JCL. Make the desired changes and then enter the SUBMIT command to submit the JCL. EDIT must be entered each time it is needed. Note that the edits made to the JCL are not saved. The CREATE command must be used to save the modified JCL elsewhere. The EDIT option is ignored if the Analyze Submit Parameters screen displays. In this case, the E command must be entered to edit the JCL.
INS NOINS	INS specifies that an Insight analysis is performed.
OUTPUT (*xxxxx*)	Specifies the resulting JCL is not 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. OUTPUT must be entered each time it is needed.
PANEL NOPANEL	PANEL indicates the Analyze Submit Parameters screen is displayed for analyze job parameter entry. The Analyze Submit Parameters screen displays even if a valid AKR name is specified as a parameter, or is obtained from the ISPF profile when PANEL is specified.
PGM (xxxxx)	Specifies a name when storing the program in the AKR. This name overrides the program name in the PROGRAM-ID paragraph.
PROCONLY	Indicates the JCL contains only a cataloged procedure rather than a complete job. PROCONLY suppresses the generation of the VAIAN DD statement. PROCONLY must be entered each time it is needed.
REUS NOREUS	REUS specifies that when you use SmartTest to test a program, the program is dynamically loaded and tested with RUN NOMONITOR.
ENS NOENS	ENS specifies that a Encore analysis is done.

Parameter	Description
SD NOSD	SD specifies that a SmartDoc analysis is done.
SDR NOSDR	SDR specifies that SmartDoc reports are done.
SDX NOSDX	SDX specifies that a SmartDoc Extended analysis is done.
ST NOST	ST specifies that a SmartTest analysis is done.
STX NOSTX	STX specifies that an Extended SmartTest analysis is done. When the INS and ST parameters are specified, an Extended SmartTest analysis is automatically done.

Analyze Submit Parameters Screen

Figure 80 shows the Analyze Submit Parameters screen. This screen displays when you specify the PANEL parameter while executing either VIASUBDS or VIASUB, or when you use the NOPANEL option and SmartDoc detects an error condition.

Figure 80 • Analyze Submit Parameters Screen



Options

These are the Analyze Submit Parameters screen options:

Options	Description
E - Edit JCL	Enter E to review or to change the compile/analyze JCL, if necessary. Select this option to use the rules outlined in "Automatic JCL Modifications" on page 169 to generate the JCL to be edited. The generated JCL displays on the Edit screen. When editing is complete, the ISPF SUBMIT command must be entered to submit the edited JCL for execution. Optionally, use the ISPF CREATE command to save the edited JCL in a partitioned dataset. Otherwise, any changes made at this time are not saved.
S - Submit JCL	Enter S to submit the JCL to compile/analyze the specified program. The JCL submitted is generated from the JCL specified when the VIASUBDS CLIST or the VIASUB edit macro was invoked, applying the rules outlined in "Automatic JCL Modifications" on page 169 ..
D - SmartDoc Options	This field displays only if SmartDoc is installed. Enter D to display the File - SmartDoc Report Options pop-up (see Figure 61 on page 116) used to request an Extended SmartDoc analysis and to specify what reports (if any) are generated.

Fields

These are the Analyze Submit Parameters screen fields:

Field	Description	
Analyze features	Understand	This field 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 YES and cannot be changed. The default is Y.
	Test	This field displays only if SmartTest is installed. Y indicates that a SmartTest compile/analysis is performed. This type of analysis provides the testing and debugging information required by SmartTest. If SmartTest is the only product installed, this field contains YES and cannot be changed. The default is Y.

Field	Description	
	Extended Analysis	This field displays only if SmartTest is installed. This type of analysis provides comprehensive program analyzing capabilities in addition to the testing and debugging capabilities of SmartTest. The default is `Y`. An Extended SmartDoc analysis is specified on the SmartDoc Options screen.
	Document	This field 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 `YES` and cannot be changed. The default is `Y`.
	Re-engineer	`YES` specifies that an Encore compile/analysis is performed. This type of analysis provides the logic and program execution flow capabilities of Encore. If Encore is the only product installed, this field contains `YES` and cannot be changed. The default is `N`.
AKR dataset name	The AKR that will contain the information for the analyzed program.	
AKR program name	This field specifies an alias name used by the analyze job to save its results in the AKR. If you don't enter a value in this field, the results of the analyze job save in the AKR with the same name as the PROGRAM-ID statement name in the COBOL source. This field is only used for the AKR program name and does not change the COBOL program name in the source.	
Analyze Options	Use this field to specify the analyze options you want to override. Default options for the analyze job are established at installation time. Analyze options that can be entered in this field are described in "Analyze Options" on page 175.	
Compile?	A program does not need to be compiled if Insight, Encore, or SmartDoc are the only features specified. To suppress the compile step, enter `N` in this field. This field is forced to a value of `Y` if SmartTest and/or Extended analysis is selected.	
Link load module reusable?	Use this field to test a program under SmartTest that is dynamically loaded if it is tested. It is necessary to mark the load module as reusable so that the Breakpoints are retained across calls. The default is `Y`.	

Submitting an Analyze Job When ISPF is not Installed

When ISPF is not installed, you cannot use the SmartDoc online component (Analyze Submit screen), the VIASUB edit macro, or the VIASUBDS CLIST to submit analyze jobs. In this case, either manually create JCL to analyze a program, or use the VIASANJC JCL provided with SmartDoc to convert existing compile and link JCL.

VIASANJC accepts a compile and link JCL dataset (PDS or sequential) as input. With the exception of OUTPUT, all allowed parameters are listed in the VIASUBDS and VIASUB Parameters table contained in "Analyze Using ISPF/PDF Edit" on page 163.

VIASANJC creates JCL in either a separate PDS or sequential dataset to execute these actions:

- Compile
- Link edit
- Analyze
- Symbol (processing performed by the ESW monitor)
- SmartDoc

Specify these additional parameters in the VIAIN DD statement in the converted JCL, if required:

Parameter	To Execute or to Generate
INS	Insight analysis
ST	SmartTest analysis
STX	SmartTest Extended analysis
SD	SmartDoc analysis
SDX	SmartDoc Extended analysis
SDR	SmartDoc reports
ENS	Encore

Parameter	Description	
DPARM	SmartDoc report options. To generate reports without an analysis, enter the PGM=name option in the DPARM parameter to indicate the program use.	
	SYSPRINT	Create a separate compiler output file for use by a post-processor.
	VIADCOMP	Create the SmartDoc intermediate compiler output file if the CMPOUT SmartDoc option is used.
CMPL	COBOL Compile	The automatic JCL modifications described in "Automatic JCL Modifications" on page 169 are made when you use VIASANJC to convert existing compile and link JCL.

Automatic JCL Modifications

The analysis process automatically modifies the JCL based on the specified parameters and analyze options. If problems arise, use this procedure as a checklist to manually perform the analyze process until the problem is resolved. Make changes to either the JCL or the compile procedure, or copy the compile procedure.

To manually perform the analyze process

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

```
PGM= parameter      Replace with:

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

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

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

3 Increase the SPACE= specification for the SYSUT1 DD to a minimum of `SPACE=(CYL,(5,5))`.

4 If the SYSIN DD statement contains `FREE=CLOSE`, change it to `FREE=END`.

5 To ensure the ESW load libraries are available, either add a //STEPLIB DD statement to specify the ESW load libraries, or concatenate the libraries to an existing STEPLIB DD.

6 Ensure that the JOB and the modified STEP EXEC statements have a minimum of `REGION=4096 KB`.

7 Add a VIAIN DD statement to designate the features and options used during analysis. This is the format:

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

To do this step manually, modify the COBOL parameter string to include the appropriate ESW parameter. For example:

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

Where *vopt* can be any of these parameters:

Parameter	Description
INS	Insight only analysis (no COBOL compile)
ST	SmartTest only analysis (no Extended analysis)
STX	SmartTest Extended analysis
SD	SmartDoc analysis
SDX	Extended SmartDoc analysis
SDR	SmartDoc report generation
ENS	Encore analysis (no COBOL compile)
[analyze parms]	Valid analyze parameters (using the standard analyze options)
CMPL	COBOL compile (forces the JCL to execute a COBOL compile and an analysis)
NOCMPL	Suppress the COBOL compile (JCL bypasses the compile and executes only an analyze job)
(NO)SYSPRINT	Create separate compiler output file

Parameter	Description
(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 If no ESW feature is specified (i.e., INS, ST, STX, SD, SDX, SDR, or RNS), all ESW processing is suppressed. This means the procedure executes a compile as it did before

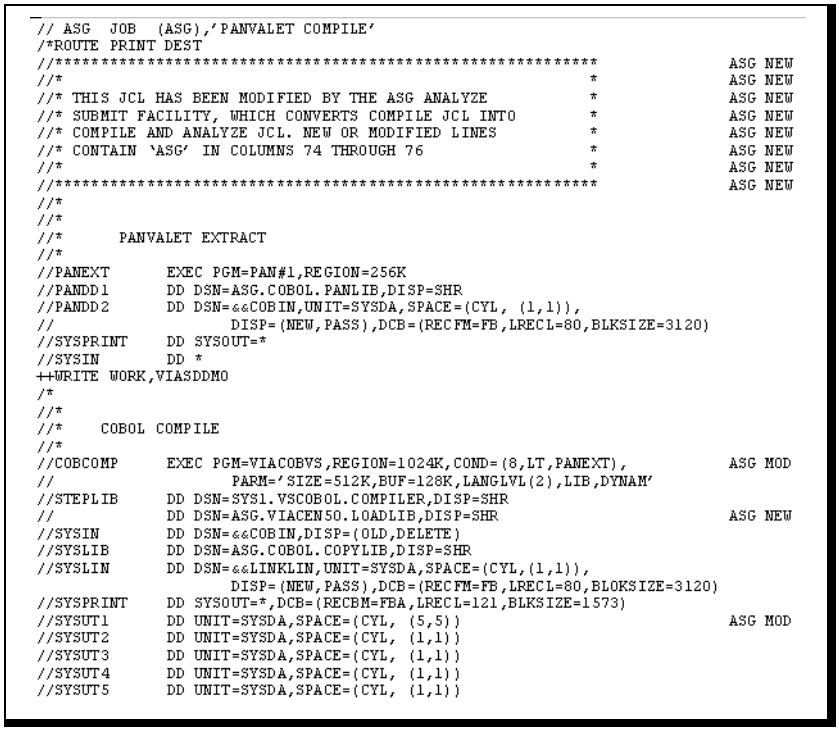
Figure 81 is an example of compile JCL as it might appear in a dataset at your site.

Figure 81 • Compile/Analyze JCL Before Modifications

```
// ASG  JOB  (ASG),'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,VIASDDMO
/*
//*
//*    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.VSCOBOL.COMPILER,DISP=SHR
//SYSIN      DD DSN=&&COBIN,DISP=(OLD,DELETE)
//SYSLIB     DD DSN=ASG.COBOL.COPYLIB,DISP=SHR
//SYSLIN     DD DSN=&&LINKLIN,UNIT=SYSDA,SPACE=(CYL, (1,1)),

DISP=(NEW,PASS),DCB=(RECFM=FB,LRECL=80,BLOKSIZE=3120)
//SYSPRINT   DD SYSOUT=*,DCB=(RECBM=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.VSCOBOL.COBLIN,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 VIASDDMO (R )
/*
```

Figure 82 shows how the compile JCL illustrated in Figure 81 on page 171 appears after you generate the compile/analyze JCL following the rules in this section. Statements you either added or modified are tagged to the right with ASG NEW and ASG MOD.

Figure 82 • Compile/Analyze JCL After Modifications



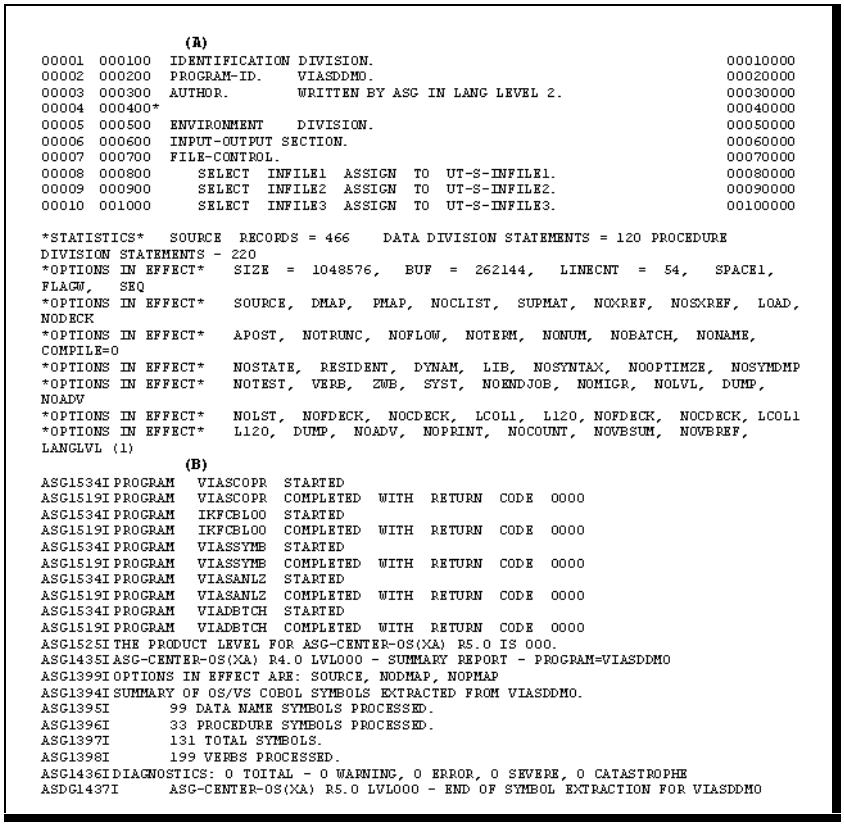
```
// ASG  JOB  (ASG),'PANVALET COMPILE'
/*ROUTE PRINT DEST
//************************************************************          ASG NEW
//*                                                          *          ASG NEW
//* THIS JCL HAS BEEN MODIFIED BY THE ASG ANALYZE            *          ASG NEW
//* 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
//************************************************************          ASG NEW
//*
//*
//*      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,VIASDDM0
/*
//*
//*    COBOL COMPILE
//*
//COBCOMP     EXEC PGM=VIACOBVS,REGION=1024K,COND=(8,LT,PANEXT),          ASG MOD
//                PARM='SIZE=512K,BUF=128K,LANGLVL(2),LIB,DYNAM'
//STEPLIB     DD DSN=SYS1.VSCOBOL.COMPILER,DISP=SHR
//            DD DSN=ASG.VIACEN50.LOADLIB,DISP=SHR                        ASG NEW
//SYSIN       DD DSN=&&COBIN,DISP=(OLD,DELETE)
//SYSLIB      DD DSN=ASG.COBOL.COPYLIB,DISP=SHR
//SYSLIN      DD DSN=&&LINKLIN,UNIT=SYSDA,SPACE=(CYL,(1,1)),
                DISP=(NEW,PASS),DCB=(RECFM=FB,LRECL=80,BLOKSIZE=3120)
//SYSPRINT    DD SYSOUT=*,DCB=(RECBM=FBA,LRECL=121,BLKSIZE=1573)
//SYSUT1      DD UNIT=SYSDA,SPACE=(CYL, (5,5))                            ASG MOD
//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))
```

Analyze Summary Report

The AKR stores the program information from the completed analyze job. The analyze job also produces a summary report of the run-time statistics and diagnostic messages. This report varies, depending on whether you specified the SOURCE or the NOSOURCE option when you submitted the analyze job.

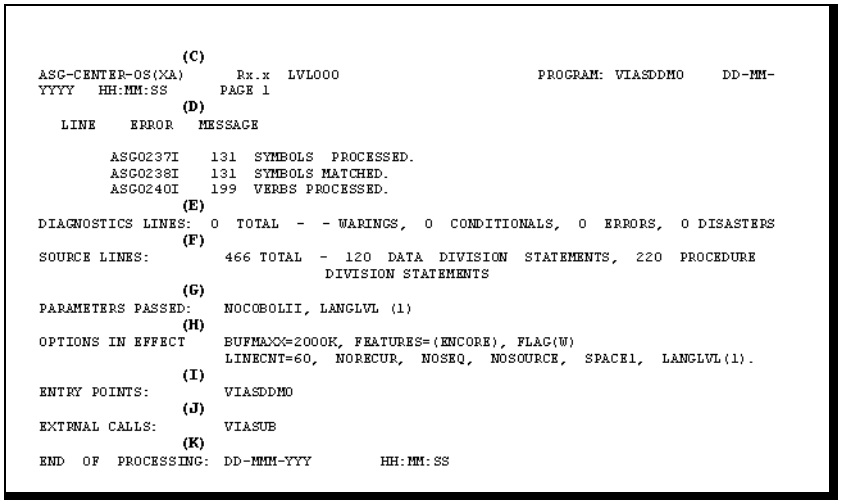
Figure 83 on page 173 and Figure 84 on page 173 show the Analyze Summary report with the SOURCE option used. The information shown on this summary is described below.

Figure 83 • Analyze Summary Report



```
                (A)
00001  000100  IDENTIFICATION DIVISION.                                    00010000
00002  000200  PROGRAM-ID.    VIASDDMO.                                   00020000
00003  000300  AUTHOR.        WRITTEN BY ASG IN LANG LEVEL 2.             00030000
00004  000400*                                                            00040000
00005  000500  ENVIRONMENT    DIVISION.                                   00050000
00006  000600  INPUT-OUTPUT SECTION.                                      00060000
00007  000700  FILE-CONTROL.                                              00070000
00008  000800      SELECT  INFILE1  ASSIGN  TO  UT-S-INFILE1.             00080000
00009  000900      SELECT  INFILE2  ASSIGN  TO  UT-S-INFILE2.             00090000
00010  001000      SELECT  INFILE3  ASSIGN  TO  UT-S-INFILE3.             00100000

*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,  SUPMAT,  NOXREF,  NOSXREF,  LOAD,
NODECK
*OPTIONS IN EFFECT*   APOST,  NOTRUNC,  NOFLOW,  NOTERM,  NONUM,  NOBATCH,  NONAME,
COMPILE=0
*OPTIONS IN EFFECT*   NOSTATE,  RESIDENT,  DYNAM,  LIB,  NOSYNTAX,  NOOPTIMZE,  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
ASG1525I THE PRODUCT LEVEL FOR ASG-CENTER-OS(XA) R5.0 IS 000.
ASG1435I ASG-CENTER-OS(XA) R4.0 LVL000 - SUMMARY REPORT - PROGRAM=VIASDDMO
ASG1399I OPTIONS IN EFFECT ARE: SOURCE, NODMAP, NOPMAP
ASG1394I SUMMARY OF OS/VS COBOL SYMBOLS EXTRACTED FROM VIASDDMO.
ASG1395I        99 DATA NAME SYMBOLS PROCESSED.
ASG1396I        33 PROCEDURE SYMBOLS PROCESSED.
ASG1397I        131 TOTAL SYMBOLS.
ASG1398I        199 VERBS PROCESSED.
ASG1436I DIAGNOSTICS: 0 TOITAL - 0 WARNING, 0 ERROR, 0 SEVERE, 0 CATASTROPHE
ASDG1437I       ASG-CENTER-OS(XA) R5.0 LVL000 - END OF SYMBOL EXTRACTION FOR VIASDDMO
```

Figure 84 • Analyze Summary Report (continued)



```
                (C)
ASG-CENTER-OS(XA)      Rx.x  LVL000                        PROGRAM: VIASDDMO     DD-MM-
YYYY   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  -  - WARINGS,  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)
                (H)
OPTIONS IN EFFECT     BUFMAXX=2000K, FEATURES=(ENCORE), FLAG(W)
                      LINECNT=60,  NORECUR,  NOSEQ,  NOSOURCE,  SPACE1,  LANGLVL(1).
                (I)
ENTRY POINTS:         VIASDDMO
                (J)
EXTRNAL CALLS:        VIASUB
                (K)
END  OF  PROCESSING:  DD-MMM-YYY        HH:MM:SS
```

Fields

These are the Analyze Summary report fields:

Field	Description
(A)	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: C Statement was inserted with a COPY statement. ** Original source statement number is out of sequence. I Statement was inserted with an INSERT statement.
(B)	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.
(C)	The Center (Analyze) release and product level is shown along with the date and time the analysis was performed.
(D)	LINE and ERROR MESSAGE - This information displays only if error conditions were encountered. If so, this area lists the line number where the error occurred and the error message.
(E)	DIAGNOSTICS LINES - This field 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 - This field lists all of the analyze options specified for this analyze job.
(H)	OPTIONS IN EFFECT - This field lists all options in effect, including default and user-specified options.
(I)	ENTRY POINTS - This field lists the entry points in this program.
(J)	EXTERNAL CALLS - This field lists the programs that this program CALLs.
(K)	END OF PROCESSING - This field lists the day, month, year, and time the analyze job completed. This date and time is also reflected in the online AKR statistics.

Analyze Options

Use Analyze options to override the default options for the Analyze job that were established when the ESW products were installed.

The Analyze job uses many of the same options as the IBM OS/VS COBOL and the COBOL II compilers. These compile-time options are available for you to use to control the output format and to describe COBOL options.

To override the installation options, enter the desired options on the File - Analyze Submit pop-up. Separate the options with a comma (,).

If you enter an invalid option, it is ignored by the analyze job. If you enter valid option more than once, the last one is processed.

Options that accept program names as parameters, with the exception of PROGRAM, accept wildcard characters. The asterisk (*) represents zero or more characters. The question mark (?) represents a single character. For example:

Parameter	Description
DBA*	All programs that begin with DBA and end with any number of other characters
D?A*	All programs that begin with D followed by any one character, followed by A, then followed by any number of other characters
DBA???	All programs that begin with DBA and end with any three characters

The analyze options are summarized in the next sections. Abbreviations are shown in uppercase (abbreviations comply with compiler standards). The Analyze Summary report printed at the end of each analyze job lists the actual options in effect, and the options that were passed to it (the override options).

Buffers

This is the buffers parameter:

Parameter	Description
BUF	Use this parameter only as an override. The amount of main storage allocated to buffers and internal tables is dynamically allocated. Use BUF if an override is necessary. The minimum BUF value is 20 KB; the maximum is 20000 KB.

For example:

```
BUF(nnnnnK)
BUF=nnnnnK
```

Where *nnnnn* is a number from 20 to 20000.

COBOL Level

These are the COBOL level parameters:

Parameter	Description
COBOL370	This parameter overrides the LANGLVL option and processes the input program as COBOL/370
COBOLII	This parameter overrides the LANGLVL option and processes the input program as COBOL II
COB2R3	This parameter overrides the LANGLVL option and processes the input program as COBOL II Release 3.0 or Release 3.1.
NOCOBOLII	This parameter overrides the LANGLVL option and processes the input program as VS COBOL

For example:

```
COBOL370
COBOLII
COB2R3
NOCOBOLII
```

The default is based on the compiler being used and is determined automatically by the submit process.

DB2 Load Library

This is the DB2 load library parameter:

Parameter	Description
DB2LIB	This parameter specifies the load library used to invoke the DB2 pre-processor at your site.

For example:

```
DB2LIB=xxxxxx.xxxxx.xxxxxx
```

Where *xxxxxx.xxxxx.xxxxxx* is the dataset name of the DB2 load library.

DB2 Application Plan

This is the DB2 application plan parameter:

Parameter	Description
DB2PLAN	This parameter specifies the ESW application plan the VIASBIND job created at installation time. You can use DB2PLAN to override the default plan name.

For example:

```
DB2PLAN=xxxxxxxx
```

Where *xxxxxxxx* is the name of the ESW application plan.

Dynamic CALLs

These are the dynamic CALLs parameter:

Parameter	Description
DYNCALL	Default. Use this parameter to specify whether SmartDoc functions use the variable name in dynamic calls as the name of the called program.
NODYNCALL	This parameter specifies that dynamic calls are not processed by the analyze, and information for them is not available to ESW product functions.

For example:

```
DYNcall
NODYNcall
```

The minimum abbreviation is either DYN or NODYN. For example, the analyze process for this code proceeds differently, depending on whether DYNCALL is specified:

```
77  MYPROG   PIC X(8).
CALL MYPROG USING PARM1, PARM2.
```

In this example, if DYNCALL is in effect, the analyze process assumes that the program being called is MYPROG, regardless of the data value that MYPROG contains at run-time. The analyze process looks up the analysis results of MYPROG in the AKR to determine whether PARM1 and PARM2 are either used or modified.

If NODYNCALL is in effect for this example, the analyze process assumes that the program being called could be anything, and treats both PARM1 and PARM2 as used and modified on the call statement.

Note:
The DYNCALL option is unrelated to the COBOL compiler option DYNAM.

Flag Messages

These are the message levels:

Parameter	Description
I	Informational
W	Warning
E	Error
S	Severe
U	Unrecoverable

These are the flag parameters:

Parameter	Description
FLAG	This parameter specifies the types of messages listed for the Analyze job
FLAGW	This parameter indicates all warning and diagnostic messages are listed. This is the default
FLAGE	This parameter indicates diagnostic messages are listed; all other messages are suppressed
FLAG(*x*)	This parameter indicates all messages of the specified level or above are listed

For example:

```
fLAGW
fLAGE
fLAG(x)
```

Where *x* is a specified message level. The minimum abbreviations are LAGW, LAGE, and LAG(*x*).

Note:

Some informational messages are produced regardless of the flag setting.

Input

These are the input parameters:

Parameter	Description
INPUT	This parameter lists the CALLed programs that contain INPUT statements. When commands that search for INPUT are issued, statements that CALL these programs are shown in the command results. The specified programs are in addition to those specified at installation time.
NOINPUT	This parameter overrides the installation default list of CALLed programs that contain INPUT statements. The specified programs are deleted from the default list.

For example:

```
Input(x,x,...x)
Input=x
NOInput(x,x,...x)
NOInput=x
```

Where *x* is a program name. The minimum abbreviation is either I or NOI, with at least one program name. Wildcard characters are allowed in the program name.

IO

These are the IO parameters:

Parameter	Description
IO	This parameter lists the CALLed programs that contain INPUT and OUTPUT statements. When commands that search for INPUT and OUTPUT are issued, statements that CALL these programs are shown in the command results. The specified programs are in addition to those specified at installation time.
NOIO	This parameter overrides the installation default list of CALLed programs that contain INPUTand OUTPUT statements.The specified programs are deleted from the default list.

For example:

```
IO(x,x,...x)
IO=x
NOIO
```

Where *x* is a program name. Wildcard characters are allowed in the program name.

Language Level

These are the language level parameters:

Parameter	Description
LANGLVL	This parameter specifies that the 1968 or 1974 American National Standard COBOL definitions are used when analyzing source elements with meanings that have changed.
LANGLVL(1)	This parameter indicates the 1968 standard is used.
LANGLVL(2)	This parameter indicates the 1974 standard (X3.23-1974) is used.

For example:

```
LANGLVL(1)
LANGLVL(2)
```

The default is based on the compiler used, and is determined automatically by the submit process.

Line Count

This is the line count parameter:

Parameter	Description
LINECNT	This parameter indicates the number of lines printed on each page of the source listing. The default is 60.

For example:

```
lineCNT=nn
```

Where *nn* is a number from 01 to 99. The minimum abbreviation is CNT, with a line count number.

Main

This is the main parameter:

Parameter	Description
MAIN	This parameter is used only as an override. If the program is the main program, use the MAIN option to treat the EXIT program as a fallthrough.

For example:

```
lineCNT=nn
MAIN
```

The EXIT PROGRAM statements in COBOL programs are treated as GOBACKs by the Analyze job, because the program is treated as a CALLed subprogram.

Maximum Number of Errors

This is the maximum number of errors parameter:

Parameter	Description
MBRERCNT	This parameter specifies the maximum number of analysis errors allowed for a member during an analyze job. If this number of errors is exceeded, the analyze job terminates processing for that member. The number specified must be between 1 and 4000. The default is set at installation.

For example:

```
MBRERCNT=nnnn
```

Where *nnnn* is a number from 1 to 4000.

Output

These are the output parameters:

Parameter	Description
OUTPUT	This parameter lists the CALLed programs that contain OUTPUT statements. When commands that search for OUTPUT are issued, statements that CALL these programs are shown in the command results. The specified programs are in addition to those specified at installation time.
NOOutput	This parameter overrides the installation default list of CALLed programs that contain OUTPUT statements. The specified programs are deleted from the default list.

For example:

```
Output(x,x,...x)
Output=x
NOOutput(x,x,...x)
NOOutput=x
```

Where *x* is a program name. The minimum abbreviation is either O or NOO, with at least one program name. Wildcard characters are allowed in the program name.

Program

This is the program parameter:

Parameter	Description
PROGRAM	This parameter overrides the name coded in the PROGRAM-ID statement.

For example:

```
PROgram(xxxxxxxxxx)
PGM=xxxxxxxxxx
```

Where *xxxxxxxxxx* is a program name up to 10 characters. The minimum abbreviation is PRO, with a program name.

Analyzed programs are stored in the AKR and identified by the program name coded in the PROGRAM-ID statement.

Recursion

These are the recursion parameters:

Parameter	Description
RECUR	This parameter specifies whether the recursion report should be included in the Analyze Summary report. If RECUR is specified and recursion is not found, a message displays indicating no recursion was detected. If RECUR is specified and recursion is found, a message is issued and the recursive code is printed on the report.
NORECUR	Default.

For example:

```
RECur
NORECur
```

The minimum abbreviation is either REC or NOREC.

Return

These are the return parameters:

Parameter	Description
RETURN	This parameter overrides the installation list of the programs or the entry points that do not return when CALLed. Override the system defaults by listing the programs or the entry points that are to return when CALLed.
NORETURN	This parameter lists the additional programs or the entry points that are not to return when CALLed. When any of these programs are CALLed by the program being analyzed, they are treated as non-returning CALLs. The specified programs are in addition to the system defaults for programs that do not return when CALLed.

For example:

```
RETurn(x,x,...x)
RETurn=x
NORETurn(x,x,...x)
NORETurn=x
```

Where *x* is a program name. The minimum abbreviation is either RET or NORET, with at least one program name. Wildcard characters are allowed in the program name.

Sequence

This is the sequence parameter:

Parameter	Description
SEQ	Default. This parameter specifies whether the analyze job checks the source module statement number sequence. A warning message prints if the statements are not in sequence. If the Source option is also specified, a flag (**) is placed between the Analyze job sequence numbers and the source sequence numbers.

For example:

```
SEQ
NOSEQ
```

Source

These are the source parameters:

Parameter	Description
SOURCE	This parameter specifies the source program is listed. Specify the SOURCE option if a full program listing is desired at Analyze time.
NOSOURCE	Default.

For example:

```
SOUrce
NOSOUrce
```

The minimum abbreviation is either SOU or NOSOU.

Spacing

These are the spacing parameters:

Parameter	Description
SPACE	This parameter specifies the spacing for the source listing generated when the SOURCE option is used.
SPACE1	Default. Use to specify single spacing.
SPACE2	This parameter specifies double spacing; that is, one blank line appears between every source line.
SPACE3	This parameter specifies triple spacing; that is, two blank lines appear between every source line.

For example:

```
spACE1
spACE2
spACE3
```

The minimum abbreviations are ACE1, ACE2, and ACE3.

SQL Authorization ID

This is the SQL authorization ID parameter:

Parameter	Description
SQLID	This parameter specifies the authorization ID or the owner name used by the analyze process to qualify unqualified table and view references in your program.

For example:

```
SQLID=nnnnnnnn
```

Where *nnnnnnnn* is an 8-character name.

```
SQLID(nnnnnnnn,nnnnnnnn,nnnnnnnn)
```

DB2 Subsystem

This is the DB2 subsystem parameter:

Parameter	Description
SUBSYS	This parameter specifies the subsystem or the location that designates the DBMS where the tables accessed by a specified program are stored. SUBSYS overrides the name provided at installation time.

For example:

```
SUBSYS=xxxx
```

Where *xxxx* is the name of the subsystem or the location of the DBMS.

Live Exit

This is the live exit parameter:

Parameter	Description
XLIVE	Use this parameter as an override and only use it with programs that contain live exits. Live exits are exits from perform ranges left dangling by either imbedded PERFORMs or GO TOs in the original performed paragraph. If XLIVE is not used, code unprocessed because of the live exit is ignored. If XLIVE is used, unprocessed code is saved.

For example:

```
XLIVE
```

Note:

Using XLIVE can significantly increase resource usage.

Memory

This is the memory parameter:

Parameter	Description
XMEM	Use only as an override. If a program is extremely large (for example, 30,000 source lines) and there is insufficient memory, increase the region space. If there is still insufficient memory, enter the XMEM option. This results in more disk I/O and additional CPU usage, but less memory consumption.

For example:

```
XMEM
```

12 SmartDoc Options

This chapter discusses SmartDoc options used to control report generation and specify various report formats and contains these sections:

Introduction

SmartDoc options control report generation and specify report formats. Use the SmartDoc Options screen to specify most options (when the online component is available). When ISPF is not installed, use the DPARM parameter to specify these options in the VIAIN DD statement of the analyze job. See Chapter 11, "Analyze," on page 157 for information on the SmartDoc Options screen and the VIAIN DD statement.

SmartDoc Options

This chapter summarizes each SmartDoc option. Each default is underlined, and abbreviations are shown in uppercase. These are the defaults that reflect the information on the installation tape:

Option	Description
BaNner NOBaNner	BaNner produces a banner page that precedes the Table of Contents for the generated reports. The default is BANNER.
BIRDSeye NOBIRDSeye	BIRDSeye generates a Bird's Eye View representation of the Structure Chart. This report is shown in Tile Mode, with each box condensed to one character. The default is BIRDSEYE.

Option	Description
CAllrept NOCAllrept	CALLrept generates the Call Statement report. The default is CALLrept.
CMpout NOCMpout	CMpout generates the Compiler/Optimizer Output if SmartDoc is executed with a compile. The default is CMpout.
CoNdsrclist NOCoNdsrclist	CoNdsrclist generates the Condensed Source Listing. The default is CoNdsrclist.
CoLon=:	CoLon defines a substitute for the colon character used on all generated reports. The default is CoLon.
CoPyrept NOCoPyrept	CoPyrept generates the Copy Statement report. The default is CoPyrept.
DataDIV NODataDIV	DataDIV generates the DATA Division report. The default is DataDIV.
DataXref NODataXref	DataXref generates the Enhanced Data Cross-Reference report. The default is DataXref.
DELMET	DELMET deletes metrics for a program when ISPF is not installed. When this option is used, SmartDoc reports cannot be generated. See Chapter 10, "Metrics," on page 153 for detailed information on this option.
DuPperf NODuPperf	DuPperf shows perform ranges multiple times on the Perform Range Hierarchy Chart or the Structure Chart (Tile Mode only) when they are used more than once in a program. **Note:** If the program contains many perform ranges called from multiple places, using the DuPperf option can produce lengthy reports. The default is NODuPperf.
HCond NOHCond	HCond generates the Perform Range Hierarchy Chart with the Conditionals option. The default is NOHCond.
HelP NOHelP	HelP is used to include descriptive help information about a generated report on the first page of that report. The default is HelP.

Option	Description
HGoto NOHGoto	HGoto generates the Perform Range Hierarchy Chart with the Gotos option. The default is HGoto.
HsiZe=9	HsiZe specifies the horizontal size (in characters) of the boxes on the Structure Chart. The minimum value is 3 (6 for DBCS); 31 is the maximum. The maximum box size can be constrained by physical limitations, such as the page size (LiNesperpag). The default is 9.
LiNesperpage=60	LiNesperpag specifies the number of lines printed on each report page of the reports. The HsiZe and VsiZe values combine with the LiNesperpag value to determine the maximum box size. The default is 60.
MasterINDEX NOMasterINDEX	MasterINDEX generates the Master Index for the SmartDoc reports. The default is NOMasterINDEX.
MeTrics NOMeTrics	MeTrics generates the Metrics report. The default is NOMeTrics.
MiNimum NOMiNimum	MiNimum specifies that only the Advanced Source Listing and Enhanced Data Cross Reference report are generated. The Compiler/Optimizer Output is also generated if a compile is performed. The default is NOMiNimum.
ParaXref NOParaXref	ParaXref generates the Paragraph Cross-Reference report. The default is ParaXref.
PerfHier NOPerfHier	PerfHier generates the Perform Range Hierarchy Chart. The default is PerfHier.
PeRfrept NOPeRfrept	PeRfrept generates the Perform Range Usage and Interface report. The default is PeRfrept.
PgmExcp NOPgmExcp	PgmExcp generates the Program Exception report. The default is PgmExcp.
ReName	ReName is used to rename metrics for a program when ISPF is not installed. When this option is used, SmartDoc reports cannot be generated. See Chapter 10, "Metrics," on page 153 for detailed information on this option.
SCond NOSCond	SCond generates the Structure Chart with the Conditionals option. The default is NOSCond.

Option	Description
SGoto NOSGoto	SGoto generates the Structure Chart with the Gotos option. The default is SGoto.
ShOrtout NOShOrtout	ShOrtout is used to show only the cross reference information on the Advanced Source Listing when an Extended SmartDoc analysis is performed. Very large programs can produce many overflow lines in the Advanced Source Listing. Using this option can reduce the number of overflows, thus making the Advanced Source Listing more readable. The default is NOShOrtout.
SouRcelist NOSouRcelist	SouRcelist generates the Advanced Source Listing. The default is SouRcelist.
StructMode=TM | PM	StructMode specifies the type of Structure Chart produced. TM produces a Structure Chart in Tile Mode. PM produces a Structure Chart in Page Mode. The default is TM.
STructurecht NOSTructurecht	STructurecht generates the Structure Chart. The default is STructurecht.
SubSet NOSubSet	SubSet generates the Subset report. The default is SubSet.
SysPrint NOSysPrint	SysPrint causes the ESW monitor to create a separate compiler output file. Either input this file to a post processor or use it for other user-specified processing. The default is NOSysPrint.
VCHAR=¦ VE=¦	VCHAR specifies the substitution character to replace vertical bars on the Structure Chart and Perform Range Hierarchy Chart. The default is ¾ (vertical bar).
VerbContext NOVerbContext	VerbContext generates the Verb Summary report with the context portion of the report included. The default is VerbContext.
VerbFreq NOVerbFreq	VerbFreq generates the Verb Summary report with the Verb Frequency Table portion of the report included. The default is VerbFreq.
VsiZe=6	VsiZe specifies the vertical size of the boxes (in lines) on the Structure Chart. The minimum value is 3; 31 is the maximum, inclusive. The actual maximum box size can be constrained by physical limitations, such as the page size (LiNesperpage). The default is 6.

13 AKR Utilities

This chapter describes the online and batch AKR utilities used by SmartDoc and contains these sections:

Introduction to AKR Management

The AKR is the repository for all of the information used by the ESW family. The AKR stores analyzed programs for use by ESW. You can either define a single AKR for use by all ESW users, or define separate AKRs for use by departments, groups, or individual users.

ESW gives you both online and batch utilities for managing the AKR. This section explains the online AKR utilities (see "Online AKR Utilities" on page 194), then the batch AKR utilities (see "Batch AKR Utilities" on page 194).

AKR Structure

The AKR is a BDAM or a VSAM file organization.

See the *ASG-Center Installation Guide* for additional information about the AKR.

Online AKR Utilities

The online AKR utilities include these elements:

Element	Description
File - AKR Utility pop-up	Use to rename or to delete a program, or to display the AKR Directory.
File - AKR Directory pop-up	Use to view all programs in an AKR. Use this pop-up to rename or to delete a group of programs. Statistics about the AKR are also shown on this pop-up.
File - AKR Allocate/Expand pop-up	Use to allocate a new AKR or to expand an existing AKR. **Note:** When you use the Allocate/Expand utility, the default AKR organization type is applied. For example, if your site's default AKR type is sequential, any new AKR allocated is created as a sequential file, and any non-sequential AKR expanded is reorganized as a sequential file.

See the online help for examples of each of these pop-ups.

Batch AKR Utilities

Use the Batch AKR Utility to maintain the AKR without using ISPF. These are the commands available in the batch utility:

Command	Description
CONVERT	Use to convert the members you selected from a previous ESW product release level to the current release level.
COPY	Use to copy the members you selected from one AKR to another.

Command	Description
DELETE	This command deletes the members you selected from the AKR.
EXPORT	Use to create metrics and function-point CDF files. **Note:** EXPORT is available only to Recap users.
HELP	Use to print the AKR Utility Help report.
INIT	Use to format a previously-defined dataset into an AKR format.
MOVE	Use to copy the members you selected from one AKR to another and deletes them from the original AKR.
PRINT	Use to print either AKR directory information or COBOL source listings for the selected AKR members.
PUNCH	Use to produce either a file that contains the AKR directory information, or the COBOL source code for the AKR members you selected.

Job Control Statements

The Batch AKR Utility uses the JCL statements shown in Figure 85. The VIAAKRIN and VIAAKROT DD statements describe AKRs used for AKR Utility processing. The VIASYSIN DD control cards consist of the necessary Batch AKR commands (see "Batch AKR Utilities" on page 194). Review the description for each command to determine what DD statements are affected.

Figure 85 • Batch AKR JCL Statements

```
//UTILITY  EXEC  PGM=VIASAKRU,REGION=3000K,PARM=''
//STEPLIB  DD    DISP=SHR,DSN=(ASG load library)
//VIAAKRIN DD    DISP=SHR,DSN=(Input AKR)
//VIAAKROT DD    DISP=SHR,DSN=(Output AKR)
//VIAPRINT DD    SYSOUT=A (Print file description)
//VIAPUNCH DD    SYSOUT=B (Punch file description)
//VIALOG   DD    SYSOUT=A (Log file)
//VIASYSIN DD    *
<control cards>
//
```

Control Cards

The control cards following the VIASYSIN DD statement pass commands to the Batch AKR Utility. Control cards must conform to these standards:

- Command information must be contained in columns 1 through 72 of the control card.
- Only one command can be entered on each control card.
- Only one control card may be used per command.

All control cards with command disposition and command summaries are printed to the VIALOG AKR Utility Log file. Blank control cards are ignored.

AKR Commands

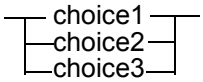
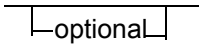
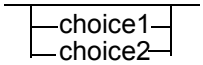
Command Format

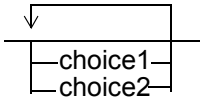
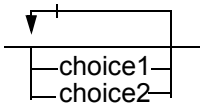
Commands that use member names accept special characters to signify generic names. An asterisk (*) represents zero or more characters. A question mark (?) represents one character. For example:

Parameter	Description
DBA*	Use to specify all members that begin with DBA and end with any other characters.
D?A*	Use to specify all members that begin with D followed by one character, followed by an A, then followed by any other characters.
DBA???	Use to specify all members that begin with DBA and end with any three characters.
LASTUSED	Use 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.
REPLACE	Use to specify that members are to be replaced on the output AKR.
NOREPLACE	Default. Use to prevent members from being replaced on the output AKR.

Command Syntax

These descriptions include the format and briefly explain the command parameters:

Item	Description
ABBREViations	Command abbreviations are shown in uppercase letters; lowercase letters in the command are optional.
lowercase	Lowercase values indicate user-supplied variable information.
UPPERCASE	Uppercase words indicate either commands or keywords.
Bold	Bold operands are available only if Insight is installed and a Insight analysis has been run on the COBOL program being tested.
Underline	The default value of an operand is underlined.
|	A vertical bar separates synonymous commands or operands.
	A right ending arrow indicates that the command syntax is continued on the next line.
	A right beginning arrow indicates the command syntax is continued from the previous line.
	Right and left ending arrows indicate the end of the command syntax.
	An operand or a keyword appearing on the main command line is required.
	Stacked operands on the main line indicate a choice of one required item.
	An operand or a keyword appearing below the main command line is optional.
	Stacked operands below the main line show a choice of one optional item.

Item	Description
	A returning arrow indicates that more than one operand can be chosen.
	A returning arrow with a plus sign (+) indicates that operands can be concatenated by placing a + between them.

Batch AKR Comments

The Batch AKR Comments command includes a comment with the commands. This is the Batch AKR Comments command syntax:

Operands

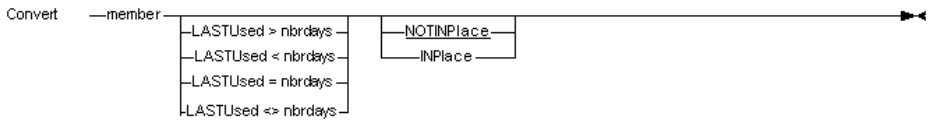
Comment. User-supplied text.

Usage Notes

Blank control cards are ignored.

CONVERT Batch AKR Command

The CONVERT Batch AKR command converts selected members that were analyzed by a prior release of ESW products to the current release level. This is the CONVERT Batch AKR command syntax:



Operands

These are the CONVERT Batch AKR command operands:

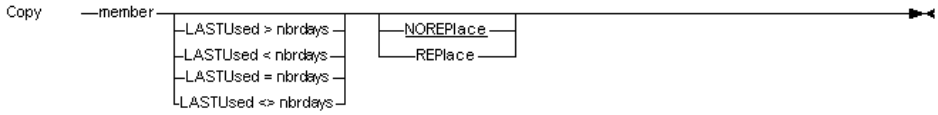
Operand	Description
member	This operand can be a specific or a generic name as described in "Command Format" on page 196.
LASTUsed	Use to select members based on the number of days since they were last used, as described in "Command Format" on page 196.
NOTINPlace	Use to specify that a member with the same name on the receiving AKR should be replaced with the member from the sending AKR.
INPlace	Use to specify that a member is converted and kept within the AKR named in the VIAAKRIN DD statement. This option should be used with caution. Either consult your systems programmer or the ASG Service Desk.

Usage Notes

Use this command to copy members 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 195.

COPY Batch AKR Command

The COPY Batch AKR command copies selected members from one AKR to another. This is the COPY Batch AKR command syntax:



Operands

These are the COPY Batch AKR command operands:

Operand	Description
member	This operand can be a specific or a generic name as described in "Command Format" on page 196.
LASTUsed	Use to select members based on the number of days since they were last used, as described in "Command Format" on page 196.

Operand	Description
NOREPlace	Use to prevent existing members from being replaced by members with the same name. This is the default.
REPlace	Use to replace members that have the same name on the receiving AKR.

Usage Notes

Use this command to copy members from the AKR specified in the VIAAKRIN DD statement to the AKR specified in the VIAAKROT DD statement.

Note:

When you use the Allocate/Expand utility, the default AKR organization type is applied. For example, let's say your site's default AKR type is sequential. Therefore, any new AKR you allocate is created as a sequential file, and any non-sequential AKR that you expand is reorganized as a sequential file.

DELETE Batch AKR Command

The DELETE Batch AKR command erases selected members from the AKR. This is the DELETE Batch AKR command syntax:

Operands

These are the DELETE Batch AKR command operands:

Operand	Description
member	This operand can be a specific or a generic name as described in "Command Format" on page 196.
LASTUsed	Use to select members based on the number of days since they were last used, as described in "Command Format" on page 196.

Usage Notes

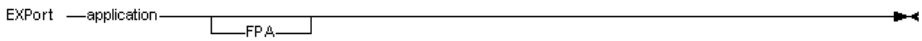
Note:

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

You cannot use this command to delete members that begin with VIA (all ESW test members begin with VIA). To delete these members, use the online AKR Utility function described in "Online AKR Utilities" on page 194.

EXPORT Batch AKR Command

The EXPORT Batch AKR command creates metrics and function point CDF files. This is the EXPORT Batch AKR command syntax:



Operands

These are the EXPORT Batch AKR command operands:

Operand	Description
application	This operand can be a specific or a generic name as described in "Command Format" on page 196.
FPA	Use to generate only function-point information. If FPA is not specified, both metrics and function point information are generated.

Usage Notes

Note:

EXPORT is available only for Recap users.

HELP Batch AKR Command

The HELP Batch AKR Command prints a description of the Batch AKR Utility and allowable commands. This is the HELP Batch AKR command syntax:

Operands

None.

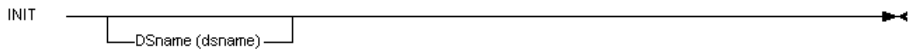
Usage Notes

- Use a question mark (?) as an alternate command.
- The HELP command has no operands.

The Help report is printed to the SYSOUT specified in the VIAPRINT DD statement.

INIT Batch AKR Command

The INIT Batch AKR command initializes a new AKR. This is an internal command that is used by the online AKR Utility Allocation function. See "Online AKR Utilities" on page 194 for additional information. This is the INIT Batch AKR command syntax:



Operand

This is the INIT Batch AKR command operand:

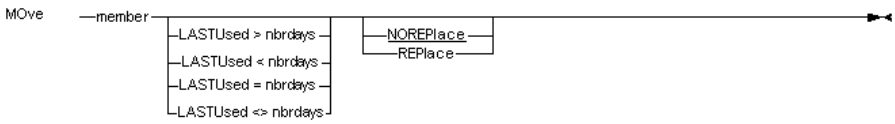
Operand	Description
DSname(dsname)	Use to specify the dataset name for the new AKR.

Usage Notes

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

MOVE Batch AKR Command

The MOVE Batch AKR command moves selected members from one AKR to another. This command copies specified members to the receiving AKR and erases them from the sending AKR. This is the MOVE Batch AKR command syntax:



Operands

These are the MOVE Batch AKR command operands:

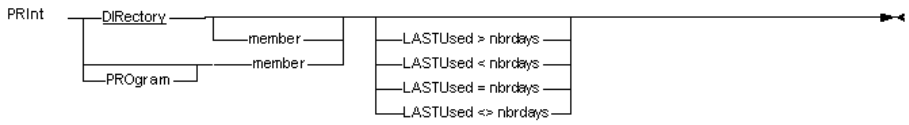
Operand	Description
member	This operand can be a specific or a generic name as described in "Command Format" on page 196.
LASTUsed	Use to select members based on the number of days since they were last used, as described in "Command Format" on page 196.
NOREPlace	Use to prevent existing members from being replaced by members with the same name. This is the default.
REPlace	Use to replace members that have the same name on the receiving AKR.

Usage Notes

This command moves members from the AKR you specified in the VIAAKRIN DD statement to the AKR you specified in the VIAAKROT DD statement.

PRINT Batch AKR Command

The PRINT Batch AKR command prints the AKR directory information for the entire AKR, a specified member, or the source code for a specified member. This is the PRINT Batch AKR command syntax:



Operands

These are the PRINT Batch AKR command operands:

Operand	Description
blank	If the PRINT batch AKR command is entered with no operand, the AKR directory information is printed.
DIRectory	Use to print AKR directory information. This is the default. If a member is specified, only the AKR directory information for that member is printed.

Operand	Description
PROgram	Use to print the COBOL source for the specified AKR member. The generated COBOL source listing contains expansions of all COPYBOOKs and/or INCLUDEs, as well as the results of any source preprocessors such as CICS macro expansion.
member	This operand can be a specific or a generic name as described in "Command Format" on page 196.
LASTUsed	Use to select a member based on the number of days since it was last used, as described in "Command Format" on page 196.

Usage Notes

This command extracts either the directory information or the COBOL source from the AKR you specified in the VIAAKRIN DD statement and prints it to the SYSOUT you specified in the VIAPRINT DD statement. See "AKR Utility Directory Report" on page 207 for additional information.

PUNCH Batch AKR Command

The PUNCH Batch AKR command produces a file that contains AKR directory information for the entire AKR, a specified member, or the source code for a specified member. This is the PUNCH Batch AKR command syntax:

Operands

These are the PUNCH Batch AKR command operands:

Operand	Description
blank	If the PUNCH batch AKR command is entered with no operand, a file is produced containing the AKR directory information.
DIRectory	Use to produce a file that contains directory information. This is the default. If a member is specified, the AKR directory information for that member only is printed.
PROgram	Use to produce a file that contains COBOL source code for the specified AKR member. The generated COBOL source contains expansions of all COPYBOOKs and/or INCLUDEs, as well as the results of any source preprocessors such as CICS macro expansion.

Operand	Description
member	This operand can be a specific or a generic name as described in "Command Format" on page 196.
LASTUsed	Use to select a member based on the number of days since it was last used, as described in "Command Format" on page 196.

Usage Notes

This command extracts either the directory information or the COBOL source from the AKR you specified in the VIAAKRIN DD statement and writes it to the file you specified in the VIAPUNCH DD statement. The file that is produced file 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. The NAME parameter contains AKRDIR`nn` for directory information, where `nn` is a consecutively assigned number.

See "Punch Directory File" on page 207 for more information.

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	DDMMMYYYY
Analyze job name	8	Character
Analyze CPU	4	Character
Analyze product level	8	Character
Last reference date	9	DDMMMYYYY
Last reference user ID	8	Character
Last reference CPU	4	Character

Batch AKR Reports

Examples of these reports are provided in this section:

- AKR Utility Log (see Figure 86)
- AKR Utility Directory report (see Figure 87 on page 207)
- File produced by the PUNCH DIRECTORY command (see Figure 88 on page 207)

AKR Utility Log

The VIALOG AKR Utility Log provides a summary of the commands issued to the Batch AKR Utility (see Figure 86). This log contains this information:

- 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 86 • AKR Utility Log

```
ASG-CENTER-OS  RX.X LVL000       AKR UTILITY LOG              DDMMMYYYY  HH:MM:SS    Page 1


*************************************************************************000140000
*  PRODUCE A REPORT CONTAINING DIRECTORY INFORMATION FOR ALL           *000150000
*       MEMBERS OF ASG-ENCORE.AKR (VIAAKRIN) THAT HAVE                 *000160000
*       NOT BEEN REFERENCED IN THE LAST 7 DAYS.                        *000170000
*************************************************************************000180000
*                                                                       000190000
PRINT DIRECTORY * LASTUSE > 7                                           000200000

   ASG1289I  8  DIRECTORY ENTRIES SUCCESSFULLY PRINTED.


                                                                        000210000
*************************************************************************000220000
*  PRODUCE A REPORT CONTAINING DIRECTORY INFORMATION FOR ALL           *000230000
*       MEMBERS OF.ASG-ENCORE.AKR (VIAAKRIN) THAT HAVE                 *000240000
*       NOT BEEN REFERENCED IN THE LAST 7 DAYS.                        *000250000
*************************************************************************000260000
*                                                                       000270000
PUNCH DIRECTORY * LASTUSE > 7                                           000280000

   ASG1290I  8  DIRECTORY ENTRIES SUCCESSFULLY PUNCHED.


   ASG1314I  *** END OF VIASYSIN ***

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


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

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

AKR Utility Directory Report

The AKR Utility Directory report (see Figure 87) lists the PRINT DIRECTORY command results and writes to the VIAPRINT DD 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 87 • AKR Utility Directory Report

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


   AKR:  ASG.VIACENnn.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 CPUA IN030000  DDMMMYYYY     HH:MM:SS     ASG      CPUA
 PYRL0085  17  16 DDMMYYYY    HH:MM:SS ASGA CPUA IN030000  DDMMMYYYY     HH:MM:SS     ASG      CPUA
 PYRL0105  17   8 DDMMYYYY    HH:MM:SS ASGA CPUA IN030000
 SR0005A   493  8 DDMMYYYY    HH:MM:SS ASGA CPUC IN030000
 W550044   66  12 DDMMYYYY    HH:MM:SS ASGA CPUC IN030000  DDMMMYYYY     HH:MM:SS     ASG      CPUA
 XRSCL070  1714 12 DDMMYYYY    HH:MM:SS ASGA CPUA IN030000  DDMMMYYYY     HH:MM:SS     ASG      CPUA
 XRSCL100  15   9 DDMMYYYY    HH:MM:SS ASGZ CPUC IN030000
 XRSCL200  41  10 DDMMYYYY    HH:MM:SS ASGZ CPUC IN030000  DDMMMYYYY     HH:MM:SS     ASG      CPUA

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

Punch Directory File

The Punch Directory (see Figure 88) writes to the VIAPUNCH DD file when the PUNCH DIRECTORY command processes. The file is formatted in standard IBM IEBUPDTE Utility format. The first card,./ADD..., is an IEBUPDTE control card that indicates these cards are 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. See the PUNCH Batch AKR command for the AKR Punch Directory File format. (See "Punch Directory File" on page 207.)

Figure 88 • AKR Punch Directory File

```
./  ADD NAME=AKRDIR1,LIST=ALL

ACTG0018  40  16DDMMYYYYASGACPUAIN030000DDMMMYYYYVIASOFT   CPUA
ACTG0018  17  16DDMMYYYYASGACPUAIN030000DDMMMYYYYVIASOFT   CPUA
ACTG0018  17   8DDMMYYYYASGACPUAIN030000
ACTG0018 493   8DDMMYYYYASGACPUCIN030000
ACTG0018  66  12DDMMYYYYASGACPUCIN030000DDMMMYYYYVIASOFT   CPUA
ACTG0018 1714 12DDMMYYYYASGACPUAIN030000DDMMMYYYYVIASOFT   CPUA
ACTG0018  15   9DDMMYYYYASGZCPUCIN030000
ACTG0018  41  10DDMMYYYYASGZCPUCIN030000DDMMMYYYYVIASOFT   CPUA
./  ENDUP
```

Allocating and Expanding AKRs without ISPF

The Batch AKR Utility can allocate and expand VSAM AKRs without using ISPF. ESW provides the VIASAKRA JCL to allocate an AKR, and the VIASAKRX JCL to expand an AKR. Figure 89 and Figure 90 on page 209 illustrate the VIASAKRA JCL. These figures illustrate the VIASAKRX JCL:

- Figure 91 on page 210
- Figure 92 on page 211
- Figure 93 on page 211

Figure 89 • VIASAKRA JCL for a VSAM AKR (1 of 2)

```
// ASG  JOB  ( ),   'ALLOC / INIT AKR'
//*ROUTE  PRINT  XXXXXX.XXXXX
//*
//*
*********************************************************************
//*  ASG,  INC.              ASG-CENTER  RX.X          MMM, YYYY     *
//*   *                                                              *
//*   *                                                              *
//*   *  JCL  PROCEDURE TO ALLOCATE AND INITIALIZE A ASG             *
//*   *  APPLICATION  KNOWLEDGE  REPOSITORY  (AKR)                   *
//* *******************************************************************
//*
//VIASIDAP  PROC  SYSOUT='*'       PRINTED OUTPUT MESSAGE CLASS
//           ASG='ASG'             HIGH LEVEL NODE FOR ASG DATA SETS
//           CENTER='VIACEN50'     MIDDLE NODE FOR ASG DATA SETS
//*
//*   ************************************************
//*   *  DEFINE  A  NEW  ASG  AKR  FILE                *
//*   ************************************************
//*
//DEFAKR   EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT  DD SYSOUT=&SYSOUT
//SYSOUT    DD SYSOUT=&SYSOUT
//SYSIN     DD DDNAME=SYSIN
//*
//*   ************************************************
//*   *  INITIALIZE THE NEWLY ALLOCATED AKR FILE       *
//*   ************************************************
//*
//INITAKR  EXEC PGM=VIASAKRU,REGION=2048K,COND=(0,LT,DEFAKR)
//STEPLIB   DD DSN=&VASG..&CENTER..LOADLIB,DISP=SHR
//VIASYSIN  DD DDNAME=SYSIN
//VIALOG    DD SYSOUT=&SYSOUT
//SYSUDUMP  DD SYSOUT=&SYSOUT
//*
//*   ************************************************
//*   *  DELETE NEW AKR (ONLY IF INITAKR FAILS)        *
//*   ************************************************
//*
//DELETE   EXEC PGM=IDCAMS,REGION=512K,
//            COND=(EVEN, (0,LT,DEFAKR), (0, EQ,INITAKR))
//SYSPRINT  DD SYSOUT=&SYSOUT
//SYSOUT    DD SYSOUT=&SYSOUT
//SYSIN     DD DDNAME=SYSIN
//*
//          PEND
//*
//VIASAKRA  EXEC  VIASIDAP
//*
```

Figure 90 • VIASAKRA JCL for a VSAM AKR (2 of 2)

```
//DEFAKR.SYSIN  DD  *
 DEFINE  CLUSTER
          (NAME (XX)  -        XX  AKR  NAME  HERE  * *
          XX (XX)              XX  ALLOC  UNITS  AND  QUANTITY  HERE  *
*
          VOLUME (SRT801)
          CONTROLINTERVALSIZE (4096)
          NUMBERED  -
          RECORDSIZE (4089  4089)
          RECOVERY  -
          ERASE  -
          UNIQUE  -
          SHAREOPTIONS (3  3) )  -
       DATA  -
          (NAME (XX.DATA) )    XX  AKR  NAME  HERE  *  *
//*
//INITAKR.VIASYSIN  DD  *
 INIT  DSNAME  (XX)            XX  AKR  NAME  HERE  * *
//*
//DELETE.SYSIN  DD  *
 DELETE  XX                    XX  AKR  NAME  HERE  * *
//*
```

To allocate an AKR

1 Replace *xx* in the NAME(*XX*), NAME(*XX*.DATA), and the DSNAME(*XX*) parameters with the name of the AKR to be allocated.

2 Replace XX(*XX*) with the allocation units values and quantities for your site.

Figure 91 • VIASAKRX JCL for a VSAM AKR (1 of 3)

```
// ASG  JOB  (  ), 'EXPAND AKR'
//*    INSERT  '/*ROUTE PRINT NODE.USER'  HERE  IF  NEEDED.
//*
//* ********************************************************************
//*   *  ASG,  INC.               ASG-CENTER  RX.X                MMM, YYYY*
//*   *                                                                    *
//*   *                                                                    *
//*   *  JCL  PROCEDURE TO EXPAND AN EXISTING ASG                          *
//*   *  APPLICATION  KNOWLEDGE  REPOSITORY  (AKR).                        *
//*************************************************** ********************
//*
//VIASAKXP  PROC  SYSOUT='*'                PRINTED OUTPUT MESSAGE CLASS
//              ASG='ASG'                   HIGH LEVEL NODE FOR ASG DATA SETS
//              CENTER='VIACEN50'           MIDDLE NODE FOR ASG DATA SETS
//*
//*     ****************************************************
//*     *  DEFINE  A  NEW  ASG  AKR  FILE                  *
//*     ****************************************************
//*
//DEFAKR      EXEC PGM=IDCAMS,REGION=512K
//SYSPRINT      DD SYSOUT=&SYSOUT
//SYSOUT        DD SYSOUT=&SYSOUT
//SYSIN         DD DDNAME=SYSIN
//*
//*     ****************************************************
//*     *  INITIALIZE NEW  DATA  SET  AS  ASG  AKR         *
//*     ****************************************************
//*
//INITAKR     EXEC PGM=VIASAKRU,REGION=2048K,COND=(0,LT,)
//STEPLIB       DD DSN=&ASG..&CENTER..LOADLIB,DISP=SHR
//VIASYSIN      DD DDNAME=SYSIN
//VIALOG        DD SYSOUT=&SYSOUT
//SYSUDUMP      DD SYSOUT=&SYSOUT
//*
//*     ****************************************************
//*     *  COPY  OLD  AKR  TO  NEW  AKR                    *
//*     ****************************************************
//*
//REPRO       EXEC PGM=IDCAMS,REGION=512K,COND=(0,LT)
//SYSPRINT      DD SYSOUT=&SYSOUT
//SYSOUT        DD SYSOUT=&SYSOUT
//SYSIN         DD DDNAME=SYSIN
//*
//*
//*     ****************************************************
//*     *  UPDATE  INTERNAL  SIZE  OF  ASG  AKR            *
//*     ****************************************************
//*
//RESIZE      EXEC PGM=VIASAKRU,REGION=2048K,COND=(0,LT,)
//STEPLIB       DD DSN=&ASG..&CENTER..LOADLIB,DISP=SHR
//VIASYSIN      DD DDNAME=SYSIN
//VIALOG        DD SYSOUT=&SYSOUT
//SYSUDUMP      DD SYSOUT=&SYSOUT
//*
//*     ****************************************************
//*     *  DELETE  OLD  AKR  AND  RENAME  NEW  AKR  TO     *
//*     *       OLD  AKR  NAME                             *
//*     ****************************************************
```

Figure 92 • VIASAKRX JCL for a VSAM AKR (2 of 3)

```
//*     ************************************************
//*     *  RUN  THIS  STEP  ONLY  IF  ALL  ABOVE  STEPS     *
//*     *       RUN  SUCCESSFULLY                          *
//*     ************************************************
//*
//TESTCODE    EXEC PGM=IEFBR14, COND=(0,LT,)
//SYSIN       DD DDNAME=SYSIN
//*
//*     ************************************************
//*     *  DELETE  OLD  AKR  AND  RENAME  NEW  AKR  TO      *
//*     *       OLD  AKR  NAME                             *
//*     ************************************************
//*
//RENAME      EXEC PGM=IDCAMS,REGION=512K,COND=(0,LT,)
//SYSPRINT    DD SYSOUT=&SYSOUT
//SYSOUT      DD SYSOUT=&SYSOUT
//SYSIN       DD DDNAME=SYSIN
//*
//*     ************************************************
//*     *  RUN  THIS  SETP  ONLY  IF  ALL  ABOVE  STEPS     *
//*     *       RUN SUCCESSFULLY                           *
//*     ************************************************
//*
//TESTCODE    EXEC PGM=IEFBR14, COND=(0,LT,)
//SYSIN       DD DDNAME=SYSIN
//*
//*
//*     ************************************************
//*     *  DELETE  NEW  AKR  ONLY  IF  EXPAND  IS  NOT      *
//*     *       SUCCESSFUL                                 *
//*     ************************************************
//*
//DELETE      EXEC PGM=IDCAMS,REGION=512K,
//                 COND=(EVEN, (0,LT,DEFAKR), (0, EQ,TESTCODE))
//SYSPRINT    DD   SYSOUT=&SYSOUT
//SYSOUT      DD   SYSOUT=&SYSOUT
//SYSIN       DD   DDNAME=SYSIN
//*
//            PEND
//*
//VIASAKRX    EXEC VIASAKXP
//*
//DEFAKR.SYSIN  DD  *
 DEFINE  CLUSTER
              (NAME (ASG.VIACEN50.AKR.EX)                  /*  AKR  NAME  */  -
              RECORDS (6000)      /*  NEW  ALLOC  UNITS  AND  QUANTITY        */  -
              VOLUME (XXXXXX)          /*  AKR  VOLSER  */  -
              CONTROLINTERVALSIZE (4096)  -
              NUMBERED  -
              RECORDSIZE (4089  4089)
              RECOVERY  -
              ERASE  -
              UNIQUE  -
              SHAREOPTIONS (3  3) )  -
         DATA  -
              (NAME (ASG.VIACEN50.AKR.EX) ) )              /*  AKR  NAME  * * -
//*
```

Figure 93 • VIASAKRX JCL for a VSAM AKR (3 of 3)

```
//INITAKR.VIASYSIN  DD  *
INIT  DSNAME  (ASG.VIACEN50.AKR.EX)                        /*  AKR  NAME  */  -
//*
//*
//REPRO.SYSIN  DD  *
REPRO  INDATASET (ASG.VIACEN50.AKR)                        /*  AKR  NAME  */  -
       OUTDATASET (ASG.VIACEN50.AKR.EX)                    /*  AKR NAME  */  -
       REPLACE
//*
//*
//RESIZE.VIASYSIN  DD  *
RESIZE  DSNAME (ASG.VIACEN50.AKR.EX)                       /*  AKR  NAME  */  -
//*
//*
//RENAME.SYSIN  DD  *
DELETE  ASG.VIACEN50.AKR                                   /* *  AKR  NAME  HERE  * * /
ALTER   ASG.VIACEN50.AKR.EX                                /*  AKR  NAME  */  -
        NEWNAME (ASG.VIACEN50.AKR)   /*  AKR  NAME  */  -
ALTER   ASG.VIACEN00.AKR.DATA                              /*  AKR  NAME  */  -
        NEWNAME (ASG.VIACEN00.AKR.DATA)                    /*  AKR  NAME  */  -
//*
//*
//DELETE.SYSIN  DD  *
 DELETE  ASG.VIACEN50.AKR.EX                               /*  AKR  NAME  HERE  * /
//*
```

To expand an AKR

1. Replace *xx* in the NAME(*XX*.EX), NAME(*XX*.EX.DATA), DSNAME(*XX*.EX), INDATASET(*XX*), and the OUTDATASET(*XX*.EX) parameters with the name of the AKR to be expanded.

2. Replace *XX(XX)* with the allocation units values and quantities for your site.

To rename an AKR

1. Replace the XX in the DELETE statement with the AKR to be renamed.

2. Enter the new AKR name in the NEWNAME(*XX*) parameter.

14 Online Component Commands

This chapter describes the SmartDoc online component commands and contains these sections:

Introduction

SmartDoc is primarily a batch-oriented product; however, the online component offers commands as well. Enter these commands on SmartDoc screens in the command input area on line four. This chapter describes all SmartDoc commands.

Command Processing

Enter commands in the command input area on SmartDoc screens. SmartDoc handles the commands that you press a PF key to enter in the same manner as those you enter in the command input area. If you press a PF key to enter a command and a command is already in the command input area, SmartDoc appends the contents of the command input area to the PF key command. The combined commands are then executed as a whole. Separate multiple commands entered in the command input area by a semicolon (;). Multiple commands process from left to right.

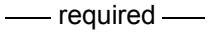
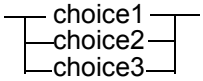
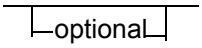
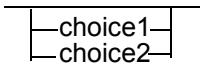
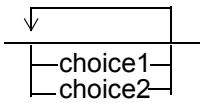
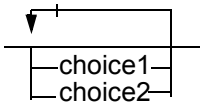
Note:

The SmartDoc command delimiter is the same character defined for ISPF in the COMMAND DELIMITER field on the ISPF Terminal Characteristics screen. Typically, this character is the semicolon.

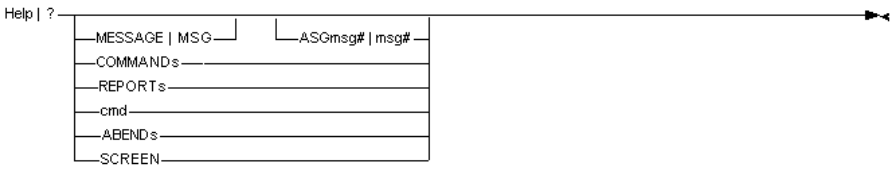
Command Diagrams

These notational conventions are used to describe command syntax:

Item	Description
ABBREViations	Command abbreviations are shown in uppercase letters; lowercase letters in the command are optional.
lowercase	Lowercase values indicate user-supplied variable information.
UPPERCASE	Uppercase words indicate either commands or keywords.
Bold	Bold operands are available only if Insight is installed and a Insight analysis has been run on the COBOL program being tested.
Underline	The default value of an operand is underlined.
|	A vertical bar separates synonymous commands or operands.
	A right ending arrow indicates that the command syntax is continued on the next line.
	A right beginning arrow indicates the command syntax is continued from the previous line.
	Right and left ending arrows indicate the end of the command syntax.\

Item	Description
	An operand or keyword appearing on the main command line is required.
	Stacked operands on the main line indicate a choice of one required item.
	An operand or keyword appearing below the main command line is optional.
	Stacked operands below the main line show a choice of one optional item.
	A returning arrow indicates that more than one operand can be chosen.
	A returning arrow with a + (plus sign) indicates that operands can be concatenated by placing a + between them.

This is a syntax diagram for the HELP command. Since H is shown in uppercase letters, H is the minimum you can enter. You can enter a question mark (?) instead of H or HELP. You do not need to enter an operand with the HELP command since all are shown below the path of the main line. If the path is followed down the first line, any operand shown can be specified. Since the MESSAGE operand is separated with a vertical bar (|), you can enter MESSAGE or MSG.



ANALYZE Command

The ANALYZE command displays the Analyze Submit pop-up. Use this pop-up to submit a compile/analyze job without ending the current SmartDoc function. This is the syntax diagram for the ANALYZE command:

Function

Note:

A program must be analyzed before it can be used in SmartDoc. See Chapter 11, "Analyze," on page 157 for additional information.

Operands

None.

Usage Notes

To display the Analyze Submit pop-up, follow this step:

- Select Analyze ▸ File and press Enter.

See Chapter 6, "File," on page 111 for additional information.

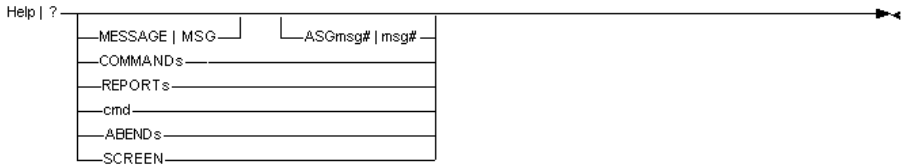
The ANALYZE command can be entered on any SmartDoc screen.

HELP Command

Use the HELP command to display information about these items:

- the current SmartDoc screen
- pop-ups
- reports
- commands
- messages
- abend codes

This is a syntax diagram for the HELP command:



See Chapter 9, "Help," on page 147 for more information.

Operand

These are the Help command operands:

Operand	Description
blank	If no message displays on the screen, enter `HELP` without operands to display the Help Tutorial for the current screen or pop-up. Here you'll find a description of all fields on the screen and any special processing considerations. If a message displays on the screen or pop-up, enter `HELP` without operands to display the Help Explanation and Action screen for the current message.
MESSAGE | MSG	Use to display the Help Explanation and Action screen, that shows the specified short and long message, an explanation of the message, and any actions to be performed.
ASGmsg# | msg#	Use to define the ESW message number for the Help Explanation and Action screen to be displayed. This number consists of 1 through 4 digits. You do not need to enter leading zeros.
COMMANDS	Use to display a list of all SmartDoc primary commands. Information for a particular command is displayed by selecting the appropriate number.

Operand	Description
REPORTS	Use to display a list of all SmartDoc reports. Information for a particular report will display when you select the appropriate number.
cmd	This operand is a SmartDoc primary command. When you request help for a command, a long message displays giving a brief description of that command. Requesting help again displays more detailed information about the command.
ABENDS	Use to display the Abends screen that lists the types of abend codes.
SCREEN	Use to display the Help Tutorial for the current screen or pop-up, that describes all fields on the screen or pop-up and any special processing considerations.

Usage Notes

Note:

You can also obtain Help information by selecting an action on the Help pull-down.

See Chapter 9, "Help," on page 147 for more information on the Help pull-down and its actions, and Chapter 15, "Help Facility," on page 225 for more information on the Help facility and the Help Tutorial.

KEYS Command

The KEYS command displays the Options - PF Key Definition pop-up. Use this pop-up to display and/or modify the current SmartDoc PF key assignments. This is the syntax for the KEYS command:

Operands

None.

Usage Notes

To display the Options - PF Key Definition pop-up, follow this step:

- Select PF keys ▸ Options and press Enter.

To display and/or modify the current PF key assignments, enter the KEYS command on any SmartDoc screen. Values assigned to the SmartDoc PF keys have no affect on other ISPF applications.

See Chapter 8, "Options," on page 137 for additional information on PF key assignments and the Options - PF Key Definition pop-up.

LOCATE Command

Use the LOCATE command on the File - AKR Directory pop-up to locate a particular program. To display an item that matches a specified string, enter the LOCATE command with the string. This is the syntax for the LOCATE command:

Operands

string. Either an alphanumeric or DBCS character string to locate a particular item.

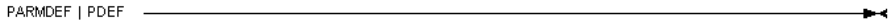
Usage Notes

When you do not know the exact name of either a program or a list item, specify a character string. The LOCATE command displays the item that most closely matches the character string you specified. Matching is done alphabetically.

See Chapter 6, "File," on page 111 for more information on the File - AKR Directory pop-up.

PARMDEF Command

The PARMDEF command displays the Options - Product Parameters pop-up. Use this pop-up to set parameters that affect the online operation of SmartDoc. This is the syntax for the PARMDEF command



Operands

None.

Usage Notes

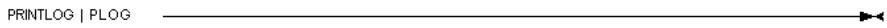
To display the Options - Product Parameters pop-up, follow this step:

- Select Options ▸ Product Parameters and press Enter.

See Chapter 8, "Options," on page 137 for more information on the Options - Product Parameters pop-up.

PRINTLOG Command

The PRINTLOG command displays the Options - Log File Definition pop-up. Use this pop-up to print the Log file. This is the PRINTLOG command syntax:



Operands

None.

Usage Notes

To display the Options - Log File Definition pop-up, follow this step:

- Selecting Options ▸ Log file and press Enter.

The PRINTLOG command can be issued on any SmartDoc screen.

See Chapter 8, "Options," on page 137 for additional information on the Options - Log File Definition pop-up.

PRODLVL Command

Use the PRODLVL command to display the current SmartDoc and Center product levels. This is the syntax for the PRODLVL command:

Operands

None.

Usage Notes

The PRODLVL command displays the product name, operating system, product release number, and release level on the message line, for example:

```
ASG1554I ASG-SMARTDOC-OS (XA) Rn.n AT Lnnn, CENTER Rn.n AT Lnnn
```

where:

Rn.n is the release number

Lnnn is the release level.

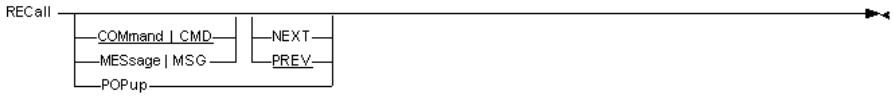
Note:

This information is requested when the ASG Service Desk group is contacted for assistance. To display this information select Help ▸ About and press Enter.

See the online help for more information on the ASG Service Desk and the Help - About pop-up.

RECALL Command

The RECALL command displays the previous primary or internal command, message, or pop-up. The last 20 executed commands and the last 20 displayed messages are stacked. To redisplay these commands or messages, use the RECALL command. This is the syntax for the RECALL command:



Operands

These are the RECALL command operands:

Operand	Description
blank	Use to display the last primary command that was stacked. After you enter the RECALL command with operands, subsequently entering RECALL with no operands automatically uses the operands that were last entered.
COMmand | CMD	Use to display a stacked primary command. This is the default.
MESsage | MSG	Use to display a stacked message.
NEXT	Use to display the next command or message in the stack.
PREV	Use to display the previous command or message in the stack. This is the default value.
POPup	Use to display the pop-up that was most recently requested from a pull-down.

Usage Notes

To display any of the 20 stacked commands or messages, type the `RECALL` command repeatedly. Use the NEXT and PREV operands to either move forward or backward through the stacked commands or messages. After the desired command displays, press Enter to execute it again. You can change any recalled command before executing it.

ESW products issue internal commands when you make certain selections on pull-downs and pop-ups. The operands you specified for the RECALL command remain in effect until one of these conditions occur:

- You specify a different operand.
- You execute a different primary command. When this occurs, the RECALL command default operands automatically set. A message displays that indicates all stacked commands or messages have been shown and the stack is displaying again.

15 Help Facility

This chapter discusses the online Help facilities and contains these sections:

Introduction

SmartDoc provides comprehensive and context sensitive online Help facilities to answer most of your questions. The Help Tutorial contains help information on several subjects, such as screens, pop-ups, reports, commands, messages and abends. The Help Tutorial also includes a Table of Contents that describes each major SmartDoc function, and a comprehensive Index where you can view specific information.

These are the ways to request SmartDoc online help:

- Select Help on the action bar and press Enter.
- Press PF1/13.
- Type HELP in the command input area on any screen and press Enter.
- Type question mark (?) in the command input area on any screen and press Enter.

Help Navigational Commands

Use the Table of Contents or the Index to access each online help subject from anywhere within the Help Tutorial. After you access the Help Tutorial, it provides commands for navigating within it. These commands are listed in this table:

Help Command	Purpose
BACK	Use to redisplay the previous Help Tutorial screen.
END	Use to exit the Help Tutorial.
Enter	Use to display the next screen in a continuation series.
INDEX	Use to display the first screen of the Help Index.
SKIP	Use to go directly to the next subject.
TOC	Use to display the Help Table of Contents.
UP	Use to display the next higher-level subject.
alpha character	Enter an alphabetic character on an Index screen, displays the Index screen corresponding to that character.

Screen Help

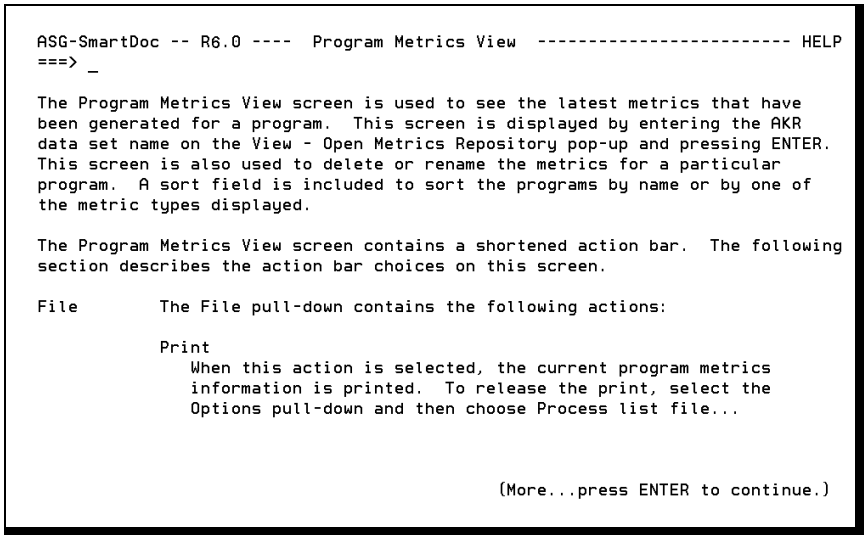
The Help Tutorial for each screen or pop-up describes all the options available on that screen, lists descriptions of all the screen fields, and notes any special processing considerations.

These are the ways to request Help for the current screen or pop-up:

- Enter the HELP primary command and press Enter.
- Press PF1/13 while there are no screen messages
- Enter the HELP SCREEN command and press Enter.
- Select Help ▸ Current Screen and press Enter.

The Help Tutorial for the current screen displays (see Figure 94).

Figure 94 • Screen Help Example



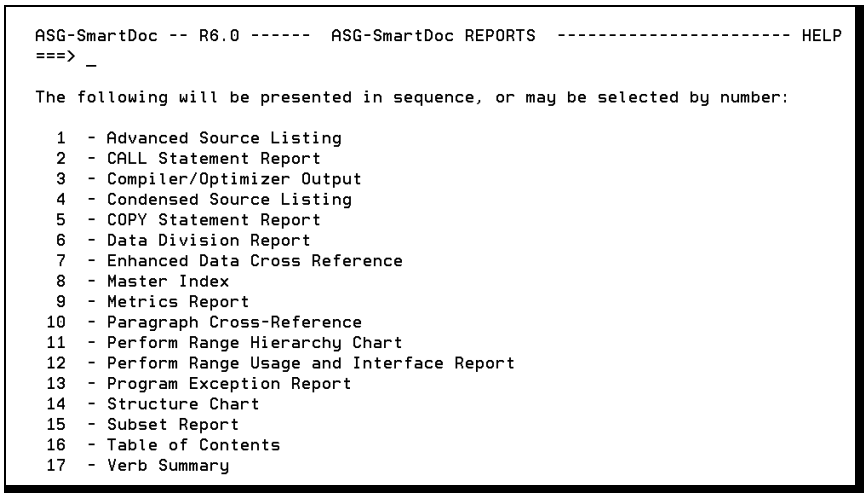
Report Help

The SmartDoc Reports Help Tutorial screen (see Figure 95), lists all SmartDoc reports. Select any report for further information. Use the Help Index to request help for specific reports.

These are the ways to display the SmartDoc Reports Help Tutorial screen:

- Enter HELP REPORTS and press Enter.
- Enter REPORTS and press PF1/13.
- Select Help ▸ All reports and press Enter.

Figure 95 • SmartDoc Reports Help Tutorial Screen



```
ASG-SmartDoc -- R6.0 ------  ASG-SmartDoc REPORTS  ---------------------- HELP
===> _

The following will be presented in sequence, or may be selected by number:

  1  - Advanced Source Listing
  2  - CALL Statement Report
  3  - Compiler/Optimizer Output
  4  - Condensed Source Listing
  5  - COPY Statement Report
  6  - Data Division Report
  7  - Enhanced Data Cross Reference
  8  - Master Index
  9  - Metrics Report
 10  - Paragraph Cross-Reference
 11  - Perform Range Hierarchy Chart
 12  - Perform Range Usage and Interface Report
 13  - Program Exception Report
 14  - Structure Chart
 15  - Subset Report
 16  - Table of Contents
 17  - Verb Summary
```

Command Help

The Help Tutorial (see Figure 96 on page 229) for each command displays the command syntax diagram, and each operand in the command.

To display a list of all SmartDoc commands, enter the UP command on a command help screen.

Figure 96 • Command Help Example

```
ASG-SmartDoc -- R6.0 -------------  RECALL  ------------------------------ HELP
===> _

The RECALL command displays the previous ASG primary or internal command,
message, or pop-up.  The last twenty commands that have been executed and the
last twenty messages that have been displayed are stacked.  These commands and
messages can be displayed using RECALL.  Once the desired command is displayed,
it can be executed again by pressing ENTER or changed prior to execution.

The RECALL command syntax is:

RECall ------------------------------------------------------------------------><
          |-COMmand|CMD-|     |-NEXT-|  |
          |-MESsage|MSG-|     |-PREV-|  |
          |-POPup----------------------|
                                    Minimum Abbreviations are in CAPS
                                    Default operands are highlighted
                                    LEGEND: ---required------------------><
                                                          |-optional-|

The following topic will be presented only if explicitly selected by number:

 1 - Operand Descriptions
```

To request help for a specific command

1 Perform one of these actions to display a message describing a specific command:

- Type the command in the input area and press PF1/13.
- Type HELP followed by the desired command name and press Enter.
- Select Help ▸ Specific Command and press Enter.

2 Press PF1/13 again to display the Help Tutorial screen for that command.

To request help on all SmartDoc commands

1 Use the HELP COMMANDS command.

Or

Select Help ▸ All Commands and press Enter.

A list displays containing most of the primary commands.

2 After this message displays, press PF1/13 for a complete list of all SmartDoc commands. From this list, select the appropriate number to display command information.

General Information

To request general help information, follow this step:

- Enter the Help Tutorial and issue the TOC command.

  Or

  Select Help ▸ Table of contents and press Enter.

  The Help Table of Contents Screen displays (see Figure 97).

Figure 97 • Help Table of Contents

```
ASG-SmartDoc -- R6.0 -----  HELP TABLE OF CONTENTS  ---------------------- HELP
===> _

The topics below represent general categories of information about the
ASG-ESW Program Documentation component, ASG-SmartDoc.  To get help for
a pull-down, select the Action Bar topic.  This Help Table of Contents
may be redisplayed from any Help screen by entering the TOC command.

The following topics will be presented only if explicitly selected by number:

  1  Overview of ASG-SmartDoc

  2  Introduction to CUA

  3  The Action Bar

  4  Customer Support

  5  Release 6.0 Summary of Revisions

  6  Index for ASG-SmartDoc Help
```

Specific Information

To request help for specific subjects

1 To request help for specific subjects, perform one of these actions:

- Enter the Help Tutorial, then enter the INDEX command.
- From the Help Table of Contents select option 6 and press Enter.
- Select Help ▸ Index and press Enter.

2 The Help Index displays (see Figure 98 on page 231). Select the appropriate Index entry to view the Help for a specific subject.

Note:

Enter an alphabetic character on any Index screen to display the Index screen corresponding to that character.

Figure 98 • Help Index Example

```
ASG-SmartDoc -- R6.0 ---------  INDEX A - C  ----------------------------  HELP
===> _
To select a topic, enter the two- or three-character identifier.

A1  - Action Bar                       C1  - Call Statement Report
A2  - Advanced Source Listing          C2  - CALLREPT ASG-SmartDoc option
A3  - AKR description                  C3  - CALL subset
A4  - ALIAS data item                  C4  - CICS subset
A5  - ALLOCDEF command                 C5  - CMPOUT ASG-SmartDoc option
A6  - ANALYZE commands                 C6  - COBOL subsets
A7  - Analyze options                  C7  - CONDSRCLIST ASG-SmartDoc option
A8  - Analyze Submit pop-up            C8  - COBOLII subset
A9  - Analyze Submit Parameters        C9  - COLON ASG-SmartDoc option
        screen                         C10 - Commands
A10 - ASSIGNMENT subset                C11 - COMMENT subset
                                       C12 - Compiler/Optimizer Output
                                       C13 - Condensed Source Listing
                                       C14 - CONDITIONAL subset
B1  - BANNER ASG-SmartDoc option       C15 - Control flow
B2  - BIRDSEYE ASG-SmartDoc option     C16 - Copy Statement Report
                                       C17 - COPYREPT ASG-SmartDoc option
                                       C18 - CUA, Introduction
Another index page can be displayed by entering its letter.
```

Help Abends

To request Help for ASG user abends

1 To request Help for ASG user abends, perform one of these actions:

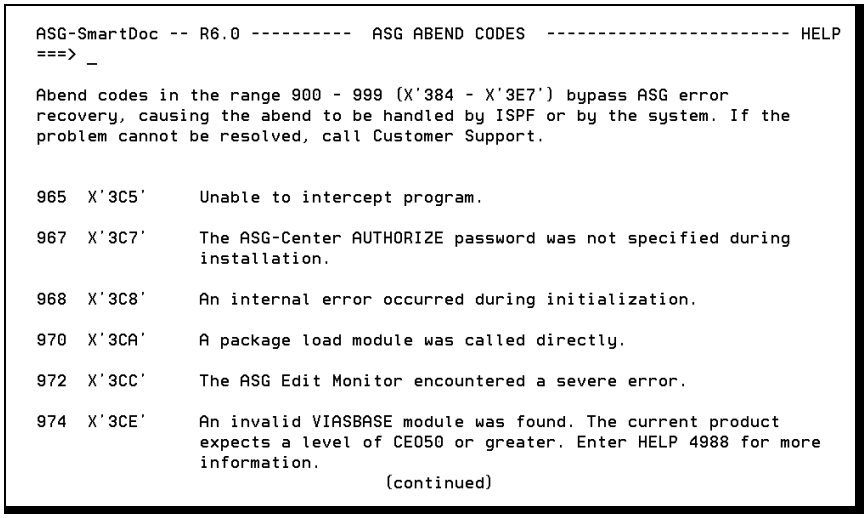
- Enter the HELP ABENDS command and press Enter.
- Type ABENDS in the command input area and press PF1/13.
- Select Help ▸ Common Abends and press Enter.

2 The ABENDS screen displays (see Figure 99 on page 232).

Note:

Select Topic 2 on this screen to display the ASG Abend Codes screen that lists all user abend messages and explanations for each message.

Figure 99 • ASG Abend Codes Screen



```
ASG-SmartDoc -- R6.0 ----------  ASG ABEND CODES  ------------------------ HELP
===> _

Abend codes in the range 900 - 999 (X'384 - X'3E7') bypass ASG error
recovery, causing the abend to be handled by ISPF or by the system. If the
problem cannot be resolved, call Customer Support.


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.

970  X'3CA'     A package load module was called directly.

972  X'3CC'     The ASG Edit Monitor encountered a severe error.

974  X'3CE'     An invalid VIASBASE module was found. The current product
                expects a level of CE050 or greater. Enter HELP 4988 for more
                information.
                                (continued)
```

Help Messages

SmartDoc messages display in the long message area, which is located on the bottom of the screen. The message format is:

`ASG`*`nnnnx text`*

where:

`nnnn` is the message number

x is one of the severity levels listed below

`text` is the text of either the long or the short message.

Severity Levels	
I	Informational - no required action.
W	Warning - an error condition exists that is not critical.
E	Error - a critical error condition exists.
D	Disaster - serious error condition exists and the product is unable to continue.
T	Termination - product terminated with the specified error.

Short messages display when available. Long messages display when short message does not exist, or when help is requested immediately following a displayed short message.

To display Help for a specific message, follow this step:

- Enter the HELP primary command, followed by the message number.

The Help Explanation and Action Panel for that message displays (see Figure 100).

Figure 100 • Help Explanation and Action Panel

```
                    HELP Explanation and Action Panel
Command ===> _____________________________________________ Scroll ===> CSR

Additional support may be found at our Web Site: support.asg.com

ASG1317  ENTER YES OR NO AT THE CURSOR LOCATION OR BLANK FOR THE DEFAULT.

EXPLANATION:
    The input field at the cursor location is expected to be a YES
    or NO.

ACTION:
    Correct the value or blank it out for the default.

****************************** BOTTOM OF DATA ********************************
```

Printing Messages

Use the VIASMPRT program to print either all or a range of SmartDoc messages. The VIASMPRT program produces a list of specified messages that includes this data:

- Message number
- Short message (if available)
- Long message
- Explanation of the message
- Action (if any)

The JCL to execute the VIASMPRT program resides in ASG.VIACEN*xx*.CNTL(VIASMPRT). The entire message file prints unless you specify a range in the PRM parameter. This syntax would print messages 300 through 499, for example:

```
PRM='START=300,END=499'
```

To print all messages, specify the ALL keywords in the PRM parameter.

The default value for START is 1. The default value for END is 9999. If you only enter the START value, messages print starting at the specified message number and ending with 9999.

If you only enter the END value, the messages print starting with 1 and ending with the message number that you specified.

The NOTES keyword specifies that any notes associated with a message are printed. The default is NONOTES. Typically, notes are provided to illustrate Center primary commands.

Figure 101 and Figure 102 on page 235 illustrate the VIASMPRT JCL and the output from the job.

Figure 101 • VIASMPRT JCL

```
// ASG  JOB  (  ),ASG-CENTER  VIASMPRT'
//*        INSERT  '/*ROUTE PRINT NODE.USER'  HERE  IF  NEEDED.
//*
//* ************************************************************
//*   *  ASG,  INC.         ASG-CENTER  RX.X                     *
//*   *                                                          *
//*   *                                                          *
//*   *              UTILITY TO PRINT ASG MESSAGES               *
//*   *                                                          *
//************************************************************
//*
//VIASMPRT  PROC  ASG='ASG',  HIGH LEVEL NODE OF ASG DATA SETS
//              CENTER='VIACEN50',  MIDDLE NODE OF ASG DATA SETS
//              SYSOUT='*',       PRINT OUTPUT MESSAGE CLASS
//              PRM='  '          PARM FOR MESSAGES TO BE PRINTED
//*
//*******************************************************************
//*   *                 MESSAGE    PRINT    UTILITY                *
//*   *                                                            *
//*   * THIS PROGRAM WILL PRINT ALL OF THE MESSAGES IN THE ASG     *
//*   * MESSAGE FILE AND THE HELP TEXT ASSOCIATED WITH EACH        *
//*   * MESSAGE IT WILL PRINT THE ENTIRE FILE BY DEFAULT. YOU MAY* *
//*   * SELECT A GIVEN RANGE OF MESSAGES BY SPECIFYING THE        *
//*   * OPTIONAL PARAMATER KEYWORDS: START AND END. FOR EXAMPLE:  *
//*   * PRM='START=300,END=499'                                    *
//*   * WILL PRINT MESSAGES NUMBER 300 THROUGH 499, INCLUSIVE.     *
//*   * THE DEFAULT VALUES FOR START AND END ARE 1 AND 99999       *
//*   * RESPECTIVELY. CONSEQUENTLY THE PRM VALUE 'END=300' WILL    *
//*   * PRINT MESSAGES 1 THROUGH 300, AND THE PRM VALUE            *
//*   * 'START=4000' WILL PRINT MESSAGES 4000 THROUGH 99999.       *
//*   *                                                            *
//*   * AN OPTIONAL KEYWORD, NOTES, WILL ALSO PRINT ANY NOTES      *
//*   * ASSOCIATED WITH A MESSAGE.                                 *
//*   *                                                            *
//*   * ADDITIONALLY, THE KEYWORD 'ALL' WILL EXPLICITLY PRINT ALL* *
//*   * MESSAGES.                                                  *
//*   *                                                            *
//*******************************************************************
//*
//VIASMPRT  EXEC  PGM=VIASMPRT,REGION=4096K,
//          PARM='&PRM'
//SYSLIB   DD DSN=&ASG..&CENTER..LOADLIB,DISP=SHR
//VIAMSGS  DD DSN=&ASG..&CENTER..VIAMSGS,DISP=SHR
//SYSPRINT DD SYSOUT=&SYSOUT
//VIAPRINT DD SYSOUT=&SYSOUT
//VIALOG   DD SYSOUT=&SYSOUT
//SYSUDUMP DD SYSOUT=&SYSOUT
//*
//       PEND
//*
//VIASMPRT  EXEC  VIASMPRT    PRINT MESSAGES
//*
```

Figure 102 • VIASMPRT Output

```
PRINTING  MESSAGES  FROM  1222  TO  1223.
MESSAGES  PRINTED.
END  OF  MESSAGE  PRINT  PROCESSING.
ASG1222  ENTER  THE  REQUIRED  FIELD  AT  THE  CURSOR  LOCATION.

 EXPLANATION:
 The field at the cursor location is a required field that must
 be Entered for processing to continue.

 ACTION:
 Enter the information required to continue processing or exit
 from The panel.


ASG1223  VALID  UNIT  TYPES:  CYLINDERS/COL  OR  TRACKS/TRK  OR
RECORDS/REC.

 EXPLANATION:
 The space unit allocation type is invalid.

 ACTION:
 Reenter the field with one of the following:
  CYLINDERS,  CYL,  TRACKS,  TRK,  RECORDS,  REC.
```

Glossary

action bar
The line of keywords at the top of a screen. Each keyword represents a category of actions that may be performed on that screen. Select an action by moving the cursor to the desired keyword and pressing Enter. See Chapter 1, "Introduction," on page 1 for more information.

alias of
A field on a pop-up listing entries in the AKR. If the analyzed program contains an ENTRY point, Alias Of is the name of the program containing the ENTRY point. If the name in the PROGRAM-ID statement was overridden at the time the analyze job was submitted, Alias Of is the name that was entered in the AKR program name field on the Analyze Submit pop-up.

analyze
The process used by SmartDoc to prepare a COBOL program for reporting.See Chapter 11, "Analyze," on page 157 for more information.

analyze options
Run-time options that control the Analyzer processing. Many of these options are similar to the COBOL compiler options. Default values are established at installation time and can be overridden by editing the Analyzer JCL or by using the Analyze screens. The Chapter 11, "Analyze," on page 157 contains a complete description of each Analyze option.

Analyze Summary report
A summary of the run-time statistics and diagnostic messages produced when an Analyze job completes.

AKR
Either a BDAM or VSAM file organization that contains all analysis information produced by the Analyze job. Multiple AKRs can be defined. See Chapter 13, "AKR Utilities," on page 193 for more information.

browser
An application that displays World Wide Web content.

COBOL subset
COBOL verbs of a similar nature that have been grouped together. For example, READ, WRITE, OPEN, and CLOSE are grouped into the IO subset. See Chapter 2, "Concepts," on page 7 for more information.

command input area
The field on SmartDoc screens where primary commands are entered, indicated by ===> on the fourth line of the screen.

dataname
A standard COBOL term for fields defined in the DATA DIVISION of a COBOL program. Variable names, files, groups, array elements, and fully qualified datanames.

data usage
Defines how a data item is used: DEF indicates the statements in the DATA DIVISION where the data item is defined; USE indicates the statements where the value is used or is tested; MOD indicates the statements where the value is set or is modified; REF indicates any of the above conditions.

DBCS
See "double byte character set (DBCS)" on page 236.

DDL
DB2 SQL Data Definition Language, a subset of SQL.

diagnostic message
An informational or an error message generated by the online and batch components. Online - A short message displays in the upper right corner of the screen (if available). A long message displays on line three when PF1/PF13 is pressed or when `HELP` is entered for a short message. Batch - Messages are included in the Analyzer Summary.

DL/I | DL/1
The database (DB) portion of the IMS system.

DML
DB2 SQL Data Manipulation Language, a subset of SQL.

double byte character set (DBCS)
A character set that uses two bytes to represent each character. Various Double Byte Character Sets are used with languages such as Chinese and Japanese that cannot be represented with single byte codes.

file transfer protocol (FTP)
A protocol that defines how to transfer files from one computer to another.

help
SmartDoc has three levels of Help: Long messages, notes, and tutorial screens. Specific command information is available by entering a command, then pressing PF1/13. The Help facility can also be accessed from the either the Help pull-down or any SmartDoc screen. See Chapter 9, "Help," on page 147 and Chapter 15, "Help Facility," on page 225 for more information.

label name
Any PROCEDURE DIVISION paragraph or section name and the PROCEDURE and PROC literals.

live exit
An abnormality in program control caused by out of perform range GO TOs and overlapping perform ranges.

log file
A file allocated bySmartDoc and used for error messages and log commands.

logical unit
Performed paragraphs (or sections) and CALLed programs reported on the Structure Chart.

long message
Either a diagnostic or an error message that displays on line five of SmartDoc screens. Long messages are sometimes preceded by short messages displayed in the upper right corner of the screen. Pressing PF1/PF13 after receiving a short message displays the corresponding long message.

member
A member in a PDS or a source manager such as Panvalet or Librarian. This can be the alias name found in the AKR.

metrics
A measure of either program quality or complexity.

page mode
A formatting option for the Structure Chart. Page Mode indicates the Structure Chart is printed on pages that can be placed in a notebook.

perform range
A perform range consists of the source code contained in a PERFORM statement, and includes all code that is or could be executed as a result of GO TOs, PERFORMs, etc., within that PERFORM.

pop-up
A window displayed by selecting an item on a pull-down or a pop-up, or by entering certain commands. It is superimposed on the screen to allow entry of information for the requested action. See Chapter 1, "Introduction," on page 1 for more information.

primary command
An instruction entered in the command input area of the screen.

program
Program source member name, the name specified in the IDENTIFICATION DIVISION of a COBOL program, or the CSECT name of a program that is not COBOL.

pull-down
The list that displays when an action is selected on the action bar. On a pull-down, actions followed by ... display a pop-up when selected. Actions not followed by ... immediately activate internal commands.

recursion
Either a perform range or paragraph that performs itself.

SBCS
See "single byte character set (SBCS)" on page 238.

screen
A full-width display of information containing an action bar as the first line. SmartDoc screens are modeled after TSO/ISPF screens.

short message
Either a diagnostic or an error message that displays in the upper right corner of SmartDoc screens. Pressing PF1/13 after receiving a short message displays the corresponding long message.

single byte character set (SBCS)
A character set that uses one byte to represent each character. Single Byte Character Sets are used with languages such as English where the characters can be represented with a one-byte code.

SQL
DB2 Structured Query Language, including DML and DDL.

storage management subsystem (SMS)
An operating environment that automates and centralizes the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

subset
See "COBOL subset" on page 236.

tile mode
A formatting option for the Structure Chart. Tile Mode indicates the Structure Chart is printed on pages that can be pasted together to produce a single chart of the entire program.

VIASUB
An edit macro included with the SmartDoc product used to submit an Analyze job.

VIASUBDS
A CLIST included with the SmartDoc product used to submit an Analyze job.

Index

D

N

O

P

R

V

X