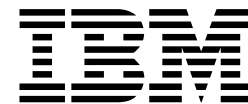




Analyzing Programs



IBM Asset Transformation Workbench
v2.1



Analyzing Programs

Note:

Before using this information and the product it supports, read the information in “Notices.”

Third Edition (August 2006)

This edition applies to IBM Asset Transformation Workbench (product number 5724-L54) and to all subsequent releases and modifications until otherwise indicated in new editions.

For the latest information about this product, please refer to the Release Notes.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of IBM. Information in this document is subject to change without notice and is not guaranteed to be error-free.

You can order publications through your IBM representative or the IBM branch office serving your locality. When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Licensed Materials - Property of IBM.

Product Reference: IBM Asset Transformation Workbench v2.1

Document Reference: REL7.3DOC04.B

© 2005, 2006 Copyright International Business Machines Corporation. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© 2005, 2006 Relativity Technologies, Inc. All rights reserved.

RescueWare is a registered trademark of Relativity Technologies, Inc. All other brands mentioned in this document are trademarks or registered trademarks of their respective holders.

Contents

Preface

<i>Audience</i>	xi
<i>Organization</i>	xii
<i>Conventions</i>	xii
<i>Related Manuals</i>	xiii
<i>Online Help</i>	xiv

1 Introducing HyperView

<i>Understanding HyperView Parse Trees</i>	1-1
<i>Starting HyperView</i>	1-3
<i>Using the HyperView Main Window</i>	1-4
<i>Context Pane</i>	1-5
<i>Source Pane</i>	1-6
<i>Clipper Pane</i>	1-6
<i>Impact Pane</i>	1-7
<i>Callie Pane</i>	1-7
<i>Execution Path Pane</i>	1-8
<i>Flowchart Pane</i>	1-8
<i>Data Flow Pane</i>	1-9

<i>Data View Pane</i>	1-9
<i>PL/I Call Diagram Pane</i>	1-10
<i>Glossary Pane</i>	1-11
<i>Rules Pane</i>	1-12
<i>Components Pane</i>	1-13
<i>Bird's Eye Pane</i>	1-14
<i>Model Reference Pane</i>	1-14
<i>Using HyperView Auxiliary Windows</i>	1-15
<i>Impact Report Window</i>	1-15
<i>Animator Window</i>	1-16
<i>Advanced Search Window</i>	1-16
<i>Using Basic Navigation Controls</i>	1-17
<i>Using the Properties Window</i>	1-17
<i>Using the List Browser Window</i>	1-20
<i>Opening a List</i>	1-20
<i>Creating a List</i>	1-22
<i>Exporting a List</i>	1-22
<i>Deleting a List</i>	1-22
<i>What's Next?</i>	1-23

2 Viewing Program Source and Context

<i>Using the Source Pane</i>	2-1
<i>Selecting and Copying Code</i>	2-2
<i>Navigating to Related Constructs</i>	2-3
<i>Navigating to Multiple Occurrences of a Construct</i>	2-3
<i>Searching for Code</i>	2-4
<i>Specifying the Change Magnitude for a Source File</i>	2-4
<i>Setting Source Pane Options</i>	2-5
<i>Adjusting the Display</i>	2-6
<i>Using the Context Pane</i>	2-7
<i>What's Next?</i>	2-8

3 Performing Advanced Searches

<i>Understanding Advanced Searches</i>	3-1
<i>Searching for Similarly Structured Code</i>	3-6

	<i>Executing Advanced Searches</i>	3-8
	<i>Defining Advanced Search Criteria</i>	3-9
	<i>What's Next?</i>	3-10
4	Staging Program Analysis with Clipper	
	<i>Getting Started in Clipper</i>	4-1
	<i>Using the Clipper Pane</i>	4-6
	<i>Working with Categories</i>	4-7
	<i>Working with Lists</i>	4-7
	<i>Generating Business Rules</i>	4-8
	<i>Showing Impacts</i>	4-9
	<i>Executing Clipper Searches</i>	4-9
	<i>Using Construct Lists to Narrow a Clipper Search</i>	4-10
	<i>Creating Projects in Clipper</i>	4-12
	<i>Generating Clipper Reports</i>	4-13
	<i>Printing Reports</i>	4-14
	<i>Exporting Reports</i>	4-14
	<i>What's Next?</i>	4-14
5	Analyzing Impact Traces	
	<i>Using the Impact Report Tool</i>	5-2
	<i>Understanding Impact Report Depths</i>	5-2
	<i>Generating an Impact Trace in the Impact Report Tool</i>	5-3
	<i>Using the Impact Report Window</i>	5-5
	<i>Setting Impact Report Options</i>	5-6
	<i>Creating Projects in the Impact Report Tool</i>	5-11
	<i>Exporting Trace Reports from the Impact Report Tool</i>	5-12
	<i>Using the Impact Pane</i>	5-14
	<i>Understanding the Hierarchical View</i>	5-14
	<i>Understanding the Diagram View</i>	5-14
	<i>Generating an Impact Trace in the Impact Pane</i>	5-15
	<i>Setting Impact Pane Options</i>	5-16
	<i>Exporting Trace Reports from the Impact Pane</i>	5-18
	<i>What's Next?</i>	5-19

6	Analyzing Program Control Flows	
	<i>Using the Callie Pane</i>	6-2
	<i>Choosing the Diagram View</i>	6-3
	<i>Redrawing the Diagram</i>	6-4
	<i>Setting Callie Pane Options</i>	6-4
	<i>Using the Execution Path Pane</i>	6-8
	<i>Generating an Execution Path</i>	6-8
	<i>Using the Flowchart Pane</i>	6-10
	<i>Generating a Flowchart</i>	6-10
	<i>Setting PL/I Flowchart Pane Options</i>	6-11
	<i>Using the Animator</i>	6-12
	<i>What's Next?</i>	6-14
7	Setting Up a Glossary of Data Items	
	<i>Understanding the Glossary Pane</i>	7-1
	<i>Generating a Glossary</i>	7-3
	<i>Assigning Business Names to Identifiers</i>	7-4
	<i>Assigning Business Names Manually</i>	7-4
	<i>Assigning Business Names in Batch Mode</i>	7-4
	<i>Importing and Exporting Business Names</i>	7-6
	<i>Editing Business Names and Business Name Descriptions</i>	7-7
	<i>Deleting Business Names and Business Name Descriptions</i>	7-8
	<i>Setting Glossary Pane Options</i>	7-8
	<i>Refreshing a Glossary</i>	7-9
	<i>Generating Glossary Reports</i>	7-10
	<i>What's Next?</i>	7-10
8	Extracting Business Rules	
	<i>Understanding Business Rules</i>	8-2
	<i>How Business Rules Are Organized</i>	8-2
	<i>How to Assign Segments to Business Rules</i>	8-2
	<i>Understanding Business Rule Attributes</i>	8-3
	<i>Extracting Business Rules Manually</i>	8-4
	<i>Autodetecting Business Rules</i>	8-5
	<i>Extracting Business Rules with Clipper</i>	8-6

Performing Basic Tasks in the Rules Pane 8-7

Editing Rule Attributes 8-9

Editing System Attributes 8-10

Editing User-Defined Attributes 8-11

Editing the Triggers Attribute 8-13

Editing the I/O Data Elements Attribute 8-13

Editing the Control Conditions Attribute 8-16

Searching for Rules 8-17

Batch Processing Rules 8-18

Validating Rule Segments after Refreshing or Editing Code 8-20

Customizing Rule Attributes 8-21

Generating Business Rule Reports 8-23

Printing Reports 8-24

Exporting Reports 8-24

What's Next? 8-25

Glossary

Bibliography

Notices

Index

Preface

The IBM Asset Transformation Workbench (ATW) is a suite of PC-based software products for analyzing, re-architecting, and transforming legacy applications. The products are deployed in an integrated environment with access to a common repository of program objects. Repository models serve as the basis for a rich set of diagrams, reports, and other documentation.

The ATW suite consists of customizable modules that together address the needs of organizations at every stage of legacy application evolution — maintenance/enhancement, renovation, and modernization.

Audience

This guide assumes that you are a corporate Information Technology (IT) professional with a working knowledge of the legacy platforms you are using the product to analyze. If you are transforming a legacy application, you should also have a working knowledge of the target platform.

Organization

This guide contains the following chapters:

- Chapter 1, “Introducing HyperView,” provides an overview of the HyperView tool.
- Chapter 2, “Viewing Program Source and Context,” describes how to use the Source and Context panes to examine legacy programs interactively.
- Chapter 3, “Performing Advanced Searches,” describes how to use the HyperView advanced search facility to create sophisticated filters for construct searches.
- Chapter 4, “Staging Program Analysis with Clipper,” describes how to use the Clipper tool to create lists of candidates for business rule extraction, event injection, impact analysis, and other tasks.
- Chapter 5, “Analyzing Impact Traces,” describes how to use the Impact pane and Impact Report tool to perform impact analysis.
- Chapter 6, “Analyzing Program Control Flows,” describes how to use the Callie, Execution Path, Flowchart, and Animator panes to examine the procedure flow for a legacy program.
- Chapter 7, “Setting Up a Glossary of Data Items,” describes how to use the Glossary pane to create a glossary of data names your application uses to identify data elements.
- Chapter 8, “Extracting Business Rules,” describes how to use the Rules pane to extract business rules.
- The Glossary defines the names, acronyms, and special terminology used in this guide.

Conventions

This guide uses the following typographic conventions:

- **Bold type** — Indicates a specific area within the graphical user interface, such as a button on a screen, a window name, or a command or function.

- *Italic type* — Indicates a new term. Also indicates a document title. Occasionally, italic type is used for emphasis.
- `Monospace type` — Indicates computer programming code.
- **Bold monospace type** — Indicates input you type on the computer keyboard.
- **1A/1B, 2A/2B** — In task descriptions, indicates mutually exclusive steps: perform step A or step B, but not both.

Related Manuals

This document is part of a complete set of ATW manuals. Together they provide all the information you need to get the most out of the system.

- *Getting Started* introduces ATW. This guide provides an overview of the workbench tools and discusses basic concepts. It describes how to install the product and how to manage licenses. It also describes how to use common product features.
- *Preparing Projects* describes how to set up ATW projects. This guide describes how to load applications in the repository and how to use reports and other tools to ensure that the entire application is available for analysis.
- *Analyzing Projects* describes how to analyze applications at the project level. This guide describes how to create diagrams of applications and how to perform impact analysis across applications. It also describes how to estimate project complexity and effort.
- *Profiling Projects* describes how to create and browse Web-generated views of the repositories in your organization.
- *Creating Components* describes how to extract program components from a legacy application.
- *Parser Reference Manual* describes legacy constructions supported by Application Analyzer in reference format.
- *Architecture Reference Manual* describes legacy constructions supported by Application Architect in reference format.

- *Error Messages* lists the error messages issued by ATW, with a brief explanation of each and instructions on how to proceed.
- *Analyzing Projects in Eclipse* describes a streamlined version of ATW designed for the Eclipse Platform.

Online Help

In addition to the manuals provided with the system, you can learn about the product using the integrated online help. All GUI-based tools include a standard Windows **Help** menu.

You can display:

- The entire help system, with table of contents, index, and search tool, by selecting **Help:Help Topics**.
- Help about a particular ATW window by clicking the window and pressing the **F1** key.

Many ATW tools have *guides* that you can use to get started quickly in the tool. The guides are help-like systems with hyperlinks that you can use to access functions otherwise available only in menus and other program controls.

To open the guide for a tool, choose **Guide** from the **View** menu. Use the table of contents in the **Page** drop-down to navigate quickly to a topic.

Introducing HyperView



Much of the power of ATW resides in a set of program analysis tools collectively called *HyperView*. HyperView lets you analyze legacy programs interactively, by examining synchronized, complementary views of the same information — source, context, impacts, and so forth. You can use HyperView to analyze procedure and data flows, stage program analyses, create a project data glossary, and extract business rules.

HyperView is designed primarily for programs, but you can also use it to analyze JCL or ECL files, DDLs or Natural Data Definition Modules, PSB and DBD files, and IDMS schemas.

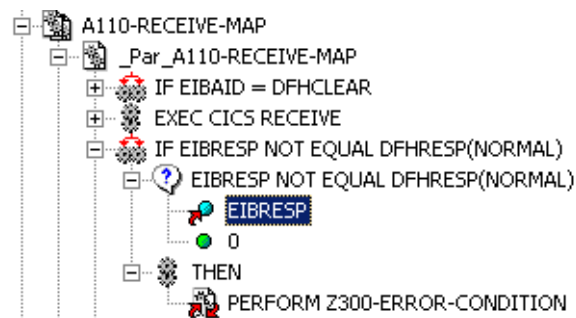
Understanding HyperView Parse Trees

When you verify a legacy source file, ATW creates a *parse tree* that defines the relationships between the constructs that comprise the file — its sections, paragraphs, statements, conditions, variables, and so forth. A variable, for example, can be related in the parse tree to any of three other constructs — a declaration, a dataport, or a condition. You view the

parse tree for a source file in the HyperView Context pane (see “[Context Pane](#)” on page 1-5).

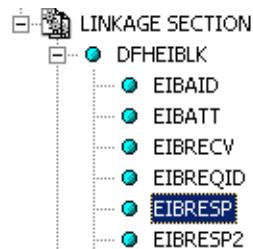
Figure 1-1 shows a portion of the parse tree for the GSS5.CBL program. The parse tree shows that the program executes a PERFORM statement if the value of the variable EIBRESP satisfies the condition EIBRESP NOT EQUAL DFHRESP(NORMAL).

Figure 1-1 Use of EIBRESP Variable



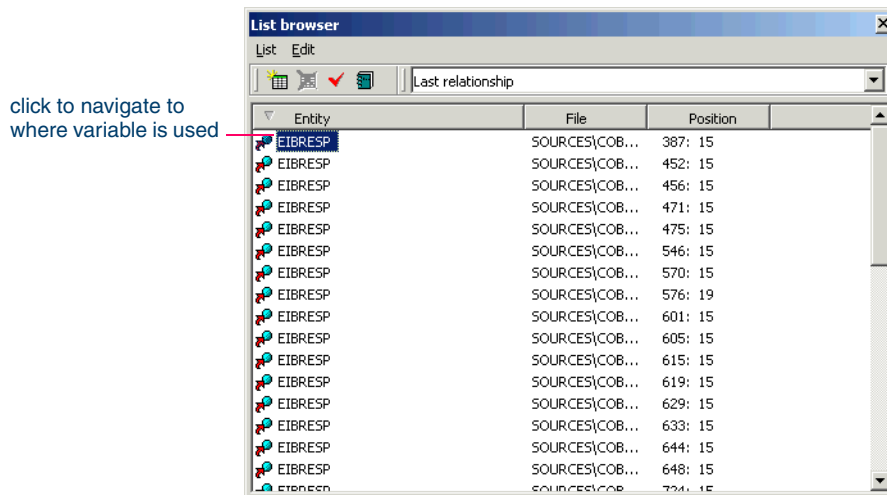
If you are interested in investigating other uses of EIBRESP in the program, you can navigate to the declaration of the variable in the parse tree (Figure 1-2). Select EIBRESP in the Context pane and choose **Declaration** in the **Edit** menu. The **Edit** menu displays choices for each type of construct to which the selection is related.

Figure 1-2 Declaration of EIBRESP Variable



From the declaration, you can generate a list of instances in which the variable is used (Figure 1-3 on page 1-3). Select the declaration in the Context pane and choose **Instances** in the **Edit** menu.

Figure 1-3 List of Instances of EIBRESP Variable



Select an instance in the list to navigate to that instance in the parse tree. You can also use the Properties window to navigate to related constructs, as described in [“Using the Properties Window” on page 1-17](#).

Starting HyperView

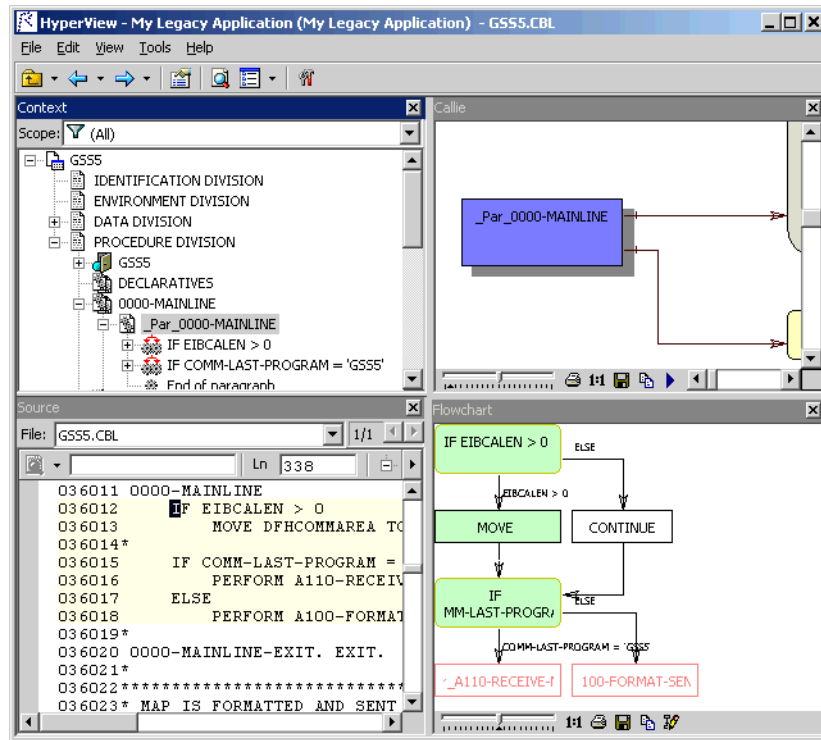
You can invoke HyperView directly by selecting a project or source file in the ATW Repository Browser and choosing **Interactive Analysis** in the **Analyze** menu. HyperView is also available within the project analysis tools — Diagrammer, Change Analyzer, and so forth — but “silently,” that is, without your ever actually invoking it as such. Tool usage is identical in either case.

To invoke HyperView directly:

- 1 In the Repository Browser, select the project or source file you want to analyze and choose **Interactive Analysis** in the **Analyze** menu. The HyperView window opens. Figure 1-4 on page 1-4 shows a typical HyperView configuration.

1-4 Introducing HyperView
Using the HyperView Main Window

Figure 1-4 *HyperView Window — Typical Configuration*



Using the HyperView Main Window

The HyperView main window consists of a series of panes that offer complementary views of code constructs in the selected source file. The display in each pane is synchronized with the others — selecting a construct in one pane automatically moves the cursor to the construct in the other panes.

The first time you open HyperView it displays the Source and Context panes. Select the appropriate choice in the **View** menu to show the other panes. Use the choices at the top of the **View** menu to configure the panes in logical groupings. Choose **Control Flow**, for example, to view the Source, Flowchart, Callie, and Animator panes. You can hide a pane by clicking the close box in the upper righthand corner.

A pane is available only if it has been configured for the workbench and if the view it offers is relevant for the selected source file — you will not see a Data Flow pane for a copybook, for example. **Edit** menu choices and pane-specific menu choices have equivalent choices in the right-click menu.

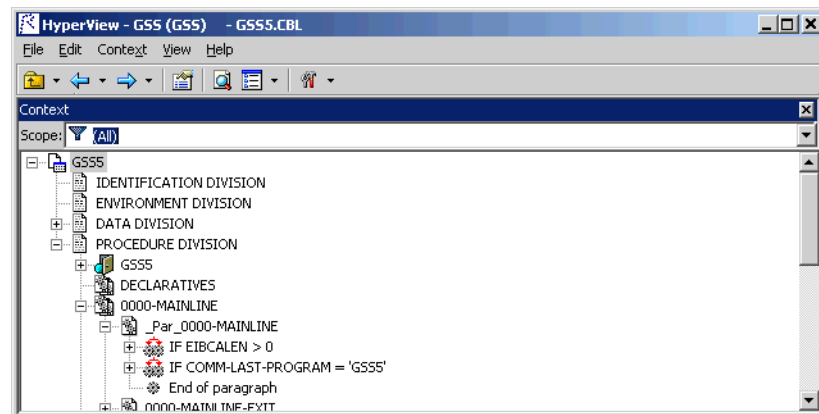
The HyperView window’s state — size, location, configuration, and option settings — is saved across sessions. For guidance on moving and resizing HyperView panes, and on configuring the workbench, see *Getting Started* in the ATW document set

In addition to the panes described below, HyperView provides access to the standard ATW Activity Log, User Preferences window, and Project Options window. To view the standard tools, select the appropriate choice in the **View** menu. For tool usage, see *Getting Started* in the ATW document set.

Context Pane

The Context pane (Figure 1-5) displays the *parse tree* for the selected source file. The parse tree displays source code constructs — sections, paragraphs, statements, conditions, variables, and so forth — in hierarchical form, making it easy to locate code constructs quickly. Constructs are displayed in the order they appear in your code.

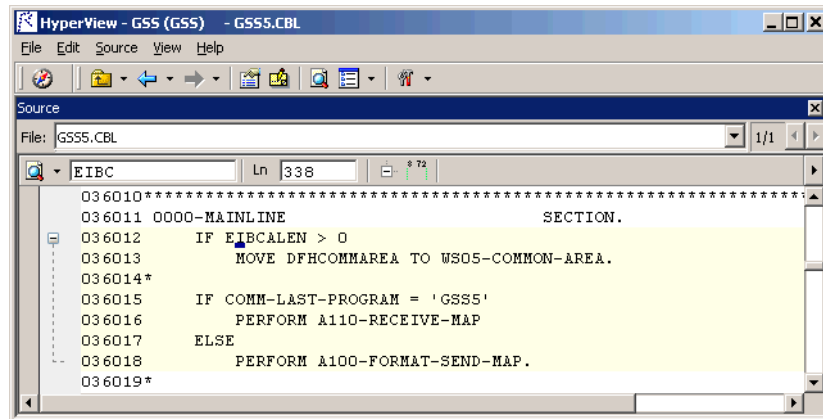
Figure 1-5 *Context Pane*



Source Pane

The Source pane (Figure 1-6) displays view-only source code for the selected file and included files. Sophisticated search facilities let you navigate to code constructs quickly.

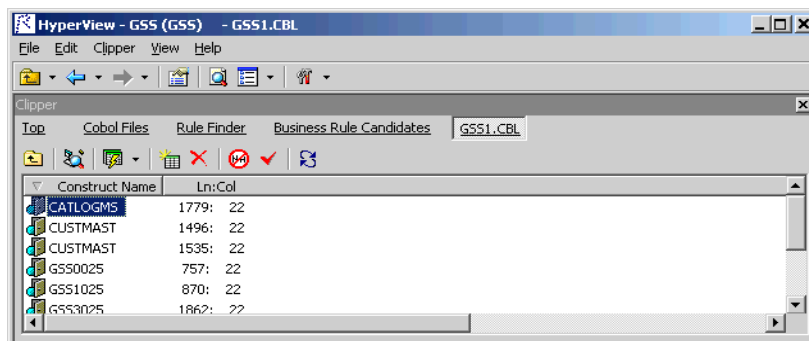
Figure 1-6 Source Pane



Clipper Pane

The Clipper pane (Figure 1-7) lets you perform batch searches for constructs of interest and stage the results as input for subsequent tasks — impact analysis, business rule extraction, event injection, and the like.

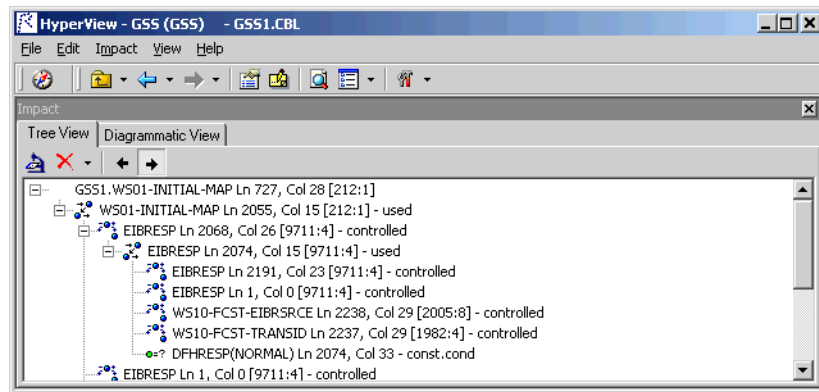
Figure 1-7 Clipper Pane



Impact Pane

The Impact pane (Figure 1-8) displays a hierarchical view and diagram of the *impact trace* for a program variable. An impact trace describes the flow of data within and between programs in an application: how data exchange values, use each other in computations, and so forth.

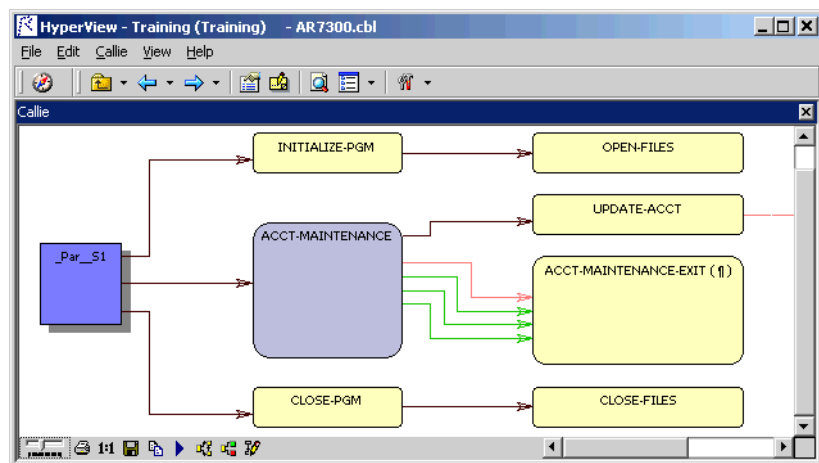
Figure 1-8 *Impact Pane (Hierarchy View)*



Callie Pane

The Callie pane (Figure 1-9) displays a diagram that shows the flow of control between paragraphs or procedures in a program.

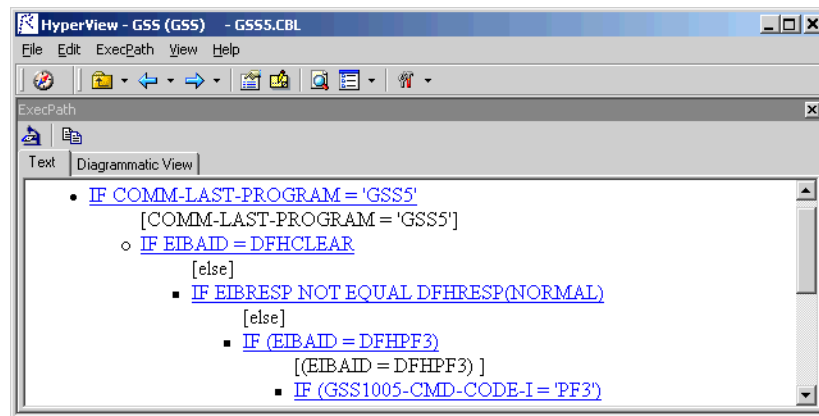
Figure 1-9 *Callie Pane*



Execution Path Pane

The Execution Path pane (Figure 1-10) displays a hierarchical view and diagram of the conditions that determine the flow of control in a program.

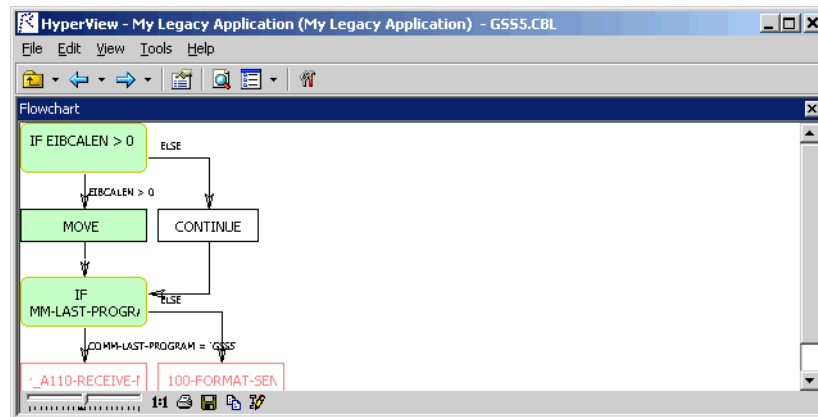
Figure 1-10 Execution Path Pane (Hierarchy View)



Flowchart Pane

The Flowchart pane (Figure 1-11) displays a diagram of the flow of control between statements in a paragraph or procedure.

Figure 1-11 Flowchart Pane

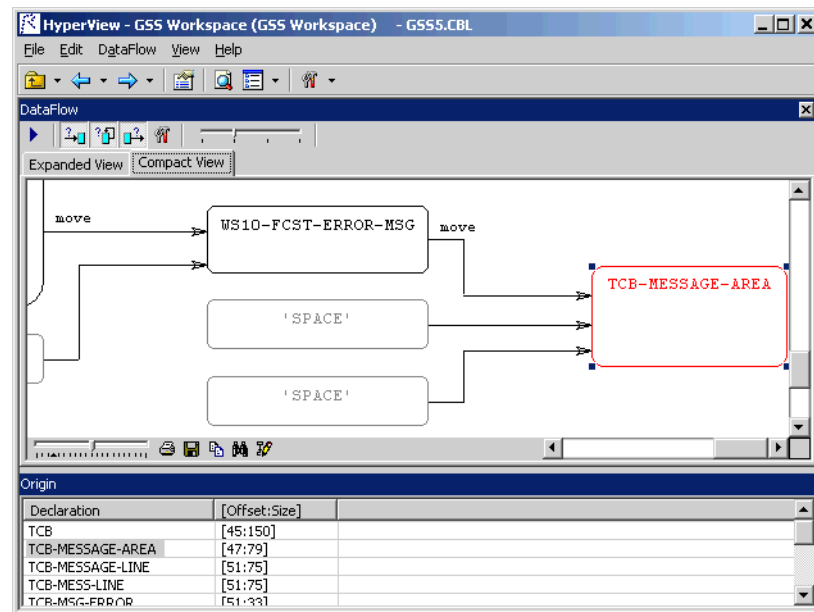


Data Flow Pane

The Data Flow pane (Figure 1-12) displays a diagram of the incoming and outgoing data flows for a program variable up to a *dataport* — an I/O statement or a call to or from another program. It also displays a list of variable offsets and memory allocations.

Note: Usage for the Data Flow and Data View panes (see below) is identical to that for the corresponding panes in the Global Data Flow tool. For more information, see *Analyzing Projects* in the ATW document set.

Figure 1-12 Data Flow Pane

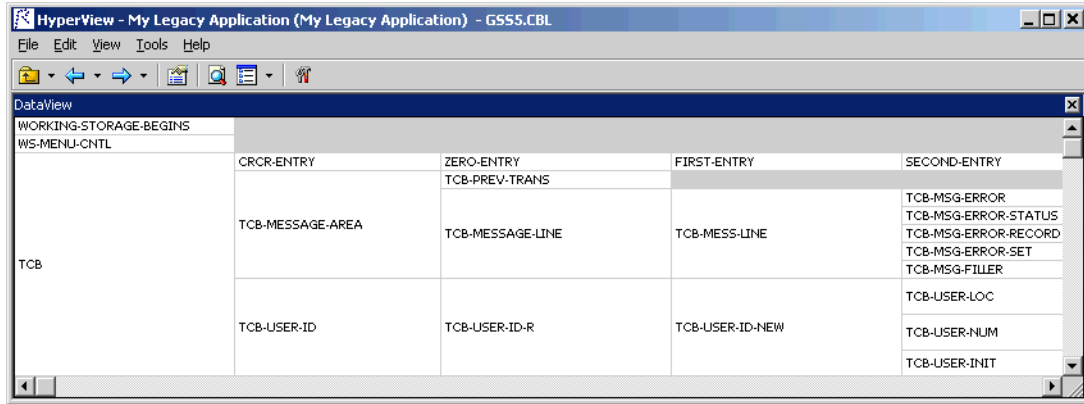


Data View Pane

The Data View pane (Figure 1-13 on page 1-10) displays program variable structures, substructures, and fields in a hierarchical grid that represents memory usage graphically.

1-10 Introducing HyperView
Using the HyperView Main Window

Figure 1-13 Data View Pane

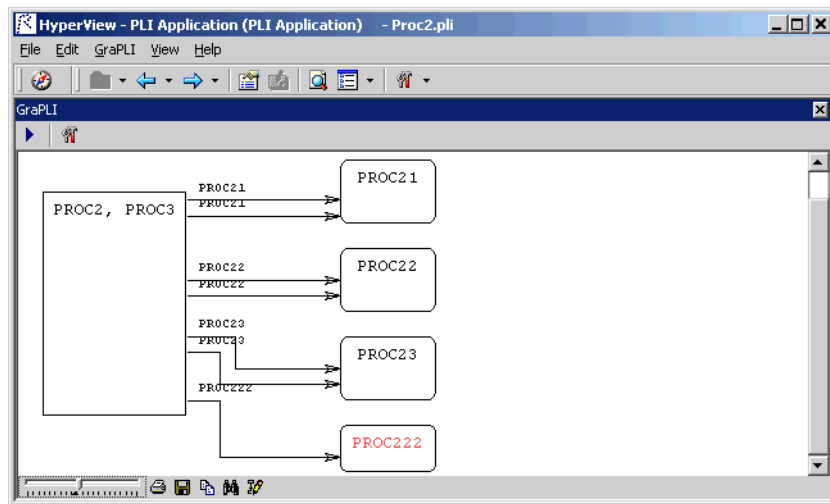


PL/I Call Diagram Pane

The PL/I Call Diagram pane (Figure 1-14) displays the call relationships between PL/I external and internal procedures. Use it to perform low-level analysis of PL/I programs.

Note: PL/I Call Diagram usage is described in *Analyzing Projects* in the ATW document set.

Figure 1-14 PL/I Call Diagram Pane

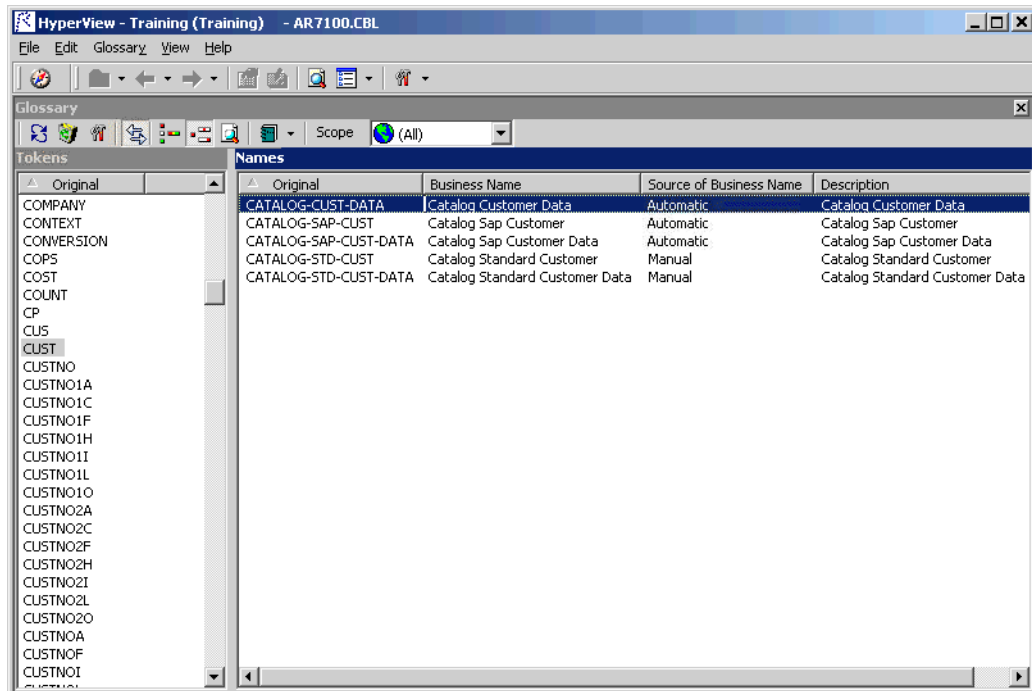


Glossary Pane

The Glossary pane (Figure 1-15) creates a conveniently organized dictionary, or *glossary*, of names your application uses to identify data elements. It's especially useful for assigning *business names* to identifiers — natural language names that make it easy to document and retrieve business rules. The field PLC_NUM, for example, might have the business name Policy Number.

The Glossary pane lets you assign business names to identifiers manually or in batch mode. You can also import business names into a glossary from a file. Use the Rules pane (Figure 1-16) to autoassign business names to business rules.

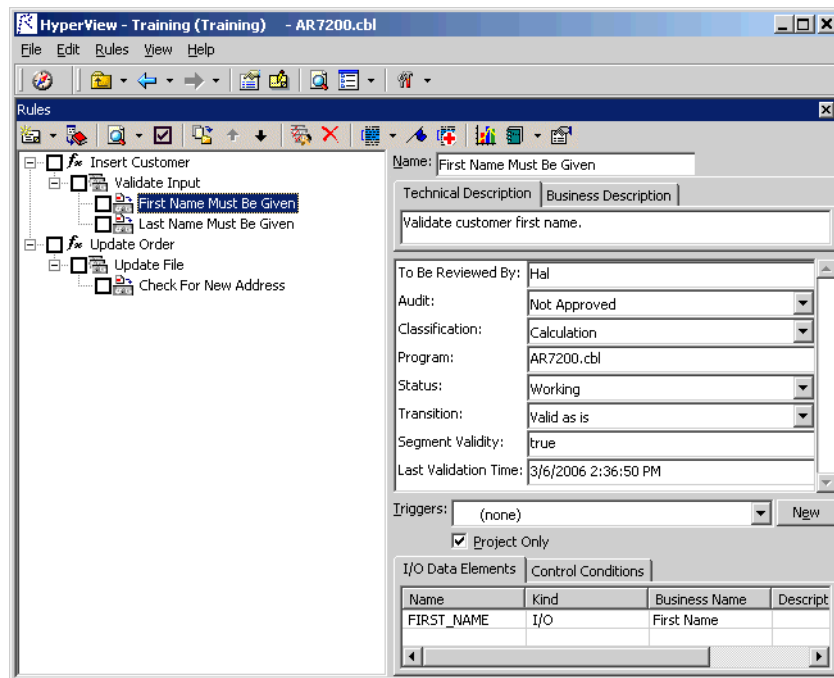
Figure 1-15 Glossary Pane



Rules Pane

The Rules pane (Figure 1-16) lets you create *business rules* from code segments extracted manually from source or autodetected. A business rule identifies a code segment according to its business function. You can batch edit rule attributes; create custom attributes suited to your particular needs; match input/output data elements with the business names you've assigned to them in your project glossary; and much more.

Figure 1-16 Rules Pane

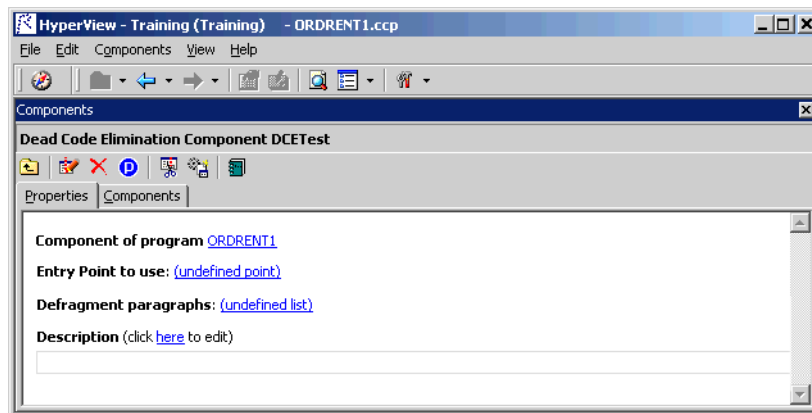
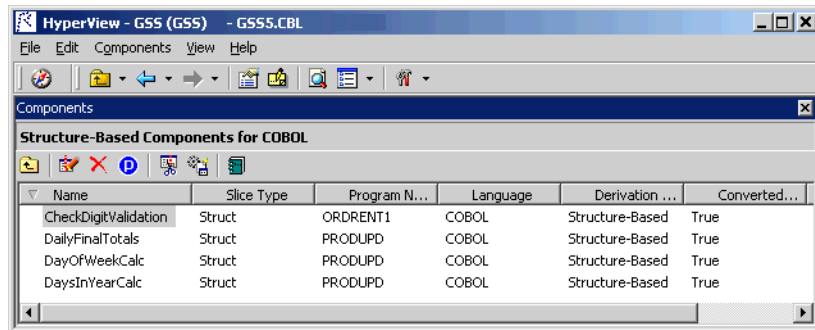


Components Pane

The Components pane (Figure 1-17) lets you extract *components* from legacy source. A component is a self-contained program that can be re-used with other programs in modular fashion.

Note: Components pane usage is described in *Creating Components* in the ATW document set.

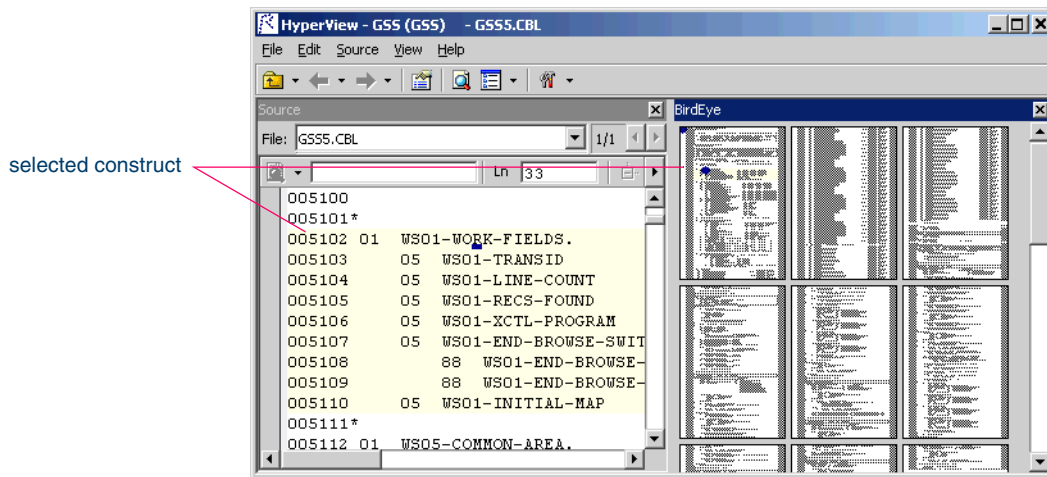
Figure 1-17 Components Pane Views



Bird's Eye Pane

The Bird's Eye pane (Figure 1-18) works with the Source pane to let you quickly identify the location of a code construct relative to the entire program and any included files.

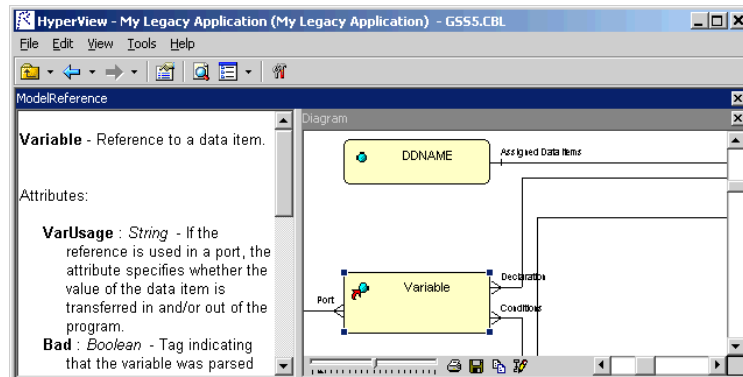
Figure 1-18 Bird's Eye Pane



Model Reference Pane

The Model Reference pane (Figure 1-19) displays the parse tree meta-model in text and diagram form.

Figure 1-19 Model Reference Pane



Using HyperView Auxiliary Windows

HyperView provides auxiliary windows for impact analysis, code animation, and construct searches. Displays are synchronized with the HyperView panes.

To open the impact analysis and code animation windows, select a construct in the Context or Source panes, then select **Impact Report** or **Animator** in the HyperView **View** menu. To open the search window, select **Find** in the **Edit** menu.

Impact Report Window







Like the Impact pane (see [“Impact Pane” on page 1-7](#)), the Impact Report window (Figure 1-20) displays an impact trace for a program variable. But where the Impact pane is better-suited for data flow analysis, the Impact Report window is a more traditional impact analysis tool — it lets you hone in on the data items in a project that may be impacted by changing another data item’s definition or usage.

Figure 1-20 *Impact Report Window with Impact Trace*

Depth	Item	Statement	Relationship	Object Name	File
0	WS01-LINE-C...	IF WS01-LINE-...	start	G555	SOURCES\COB
1	FA01-KEY	FCSTADJ	controlled	G555	SOURCES\COB
2	FA01-KEY	MOVE FA01-KE...	used	G555	SOURCES\COB
3	WS10-SCR-KEY	MOVE FA01-KE...	comp@	G555	SOURCES\COB
4	WS10-SCR-YY	MOVE WS10-S...	used	G555	SOURCES\COB
5	G551005-LN1-...	MOVE WS10-S...	comp@	G555	SOURCES\COB
6	G551005O	G551005	used	G555	SOURCES\COB
7	G551005I	G551005	used	G555	SOURCES\COB
8	G551005-LN1-...	MOVE G55100...	used	G555	SOURCES\COB
9	COMM-SELC-Y...	MOVE G55100...	move	G555	SOURCES\COB
10	WS05-COMMO...	83406288	used	G555	SOURCES\COB
11	WS05-COMMO...	MOVE DFHCO...	used	G555	SOURCES\COB
12	WS05-COMMO...	G554	used	G555	SOURCES\COB
13	DFHCOMMAREA	G554	calls	G554	SOURCES\COB
14	DFHCOMMAREA	G55	used	G554	SOURCES\COB
15	DFHCOMMAREA	G55	calls	G55	SOURCES\COB
16	DFHCOMMAREA	MOVE DFHCO...	used	G55	SOURCES\COB
17	WS05-COMMO...	MOVE DFHCO...	move	G55	SOURCES\COB
18	EIBCALEN	IF EIBCALEN = 0	controlled	G55	SOURCES\COB
19	EIBRESP	ASSIGN	controlled	G55	SOURCES\COB
20	EIBRESP	IF EIBRESP NO...	used	G55	SOURCES\COB

Using Basic Navigation Controls

HyperView offers Explorer-style navigation controls on the tool bar, with corresponding choices in the **Edit** menu:

- Click the  button on the tool bar to navigate backward in the history of your selections in the HyperView window (regardless of the pane in which they were made). Click the adjacent  button to display the selection history in a drop-down menu. Choose a selection by clicking it in the menu.
- Click the  button on the tool bar to navigate forward in the history of your selections in the HyperView window (regardless of the pane in which they were made). Click the adjacent  button to display the selection history in a drop-down menu. Choose a selection by clicking it in the menu.
- Click the  button on the tool bar to navigate to the parent of the selected construct in the parse tree. Click the adjacent  button to display all of the ancestors of the selected construct in a drop-down menu. Choose a selection by clicking it in the menu.

Using the Properties Window

HyperView Properties windows offer convenient ways to navigate program code and analyze program constructs. The Properties window for a construct identifies its:

- *Attributes* — the construct's type, name, location, and characteristics.
- *Relationships* — how the construct interacts with other constructs.
- *Structure* — the context of the construct in the parse tree.

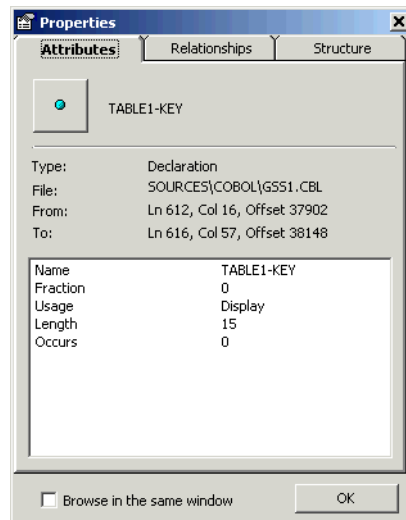
Choose a construct in a relationship list or structure tree to navigate to the construct in a HyperView pane. Properties windows are not modal, so you can leave the windows up throughout your HyperView session, and open them for as many constructs as necessary.

To view construct properties:

- 1 In any HyperView pane except the Model Reference pane, select the construct whose properties you want to view and choose **Properties**

in the **Edit** menu. The Properties window opens. Click the Attributes tab (Figure 1-23).

Figure 1-23 Properties Window — Attributes Tab



- 2 The Attributes tab displays the construct's type, name, location, and characteristics. Click the Relationships tab (Figure 1-24 on page 1-19).

Tip: Choose **Browse in the same window** only if you do *not* want to open Properties windows for other constructs concurrently.

- 3 The Relationships tab lists the types of relationships available for the selected construct. If relationships of that type exist, they are listed below the relationship type. Click the plus sign (+) next to a relationship type to expand the list. Click the minus sign (-) to collapse the list. Click a construct in the list to navigate to it in an open HyperView pane.


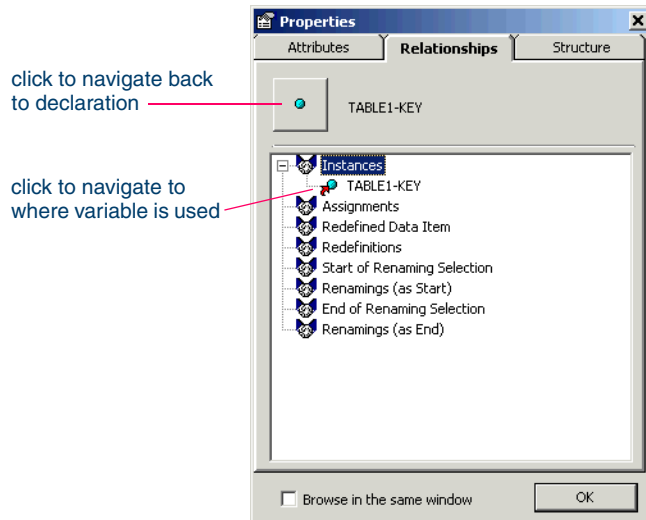
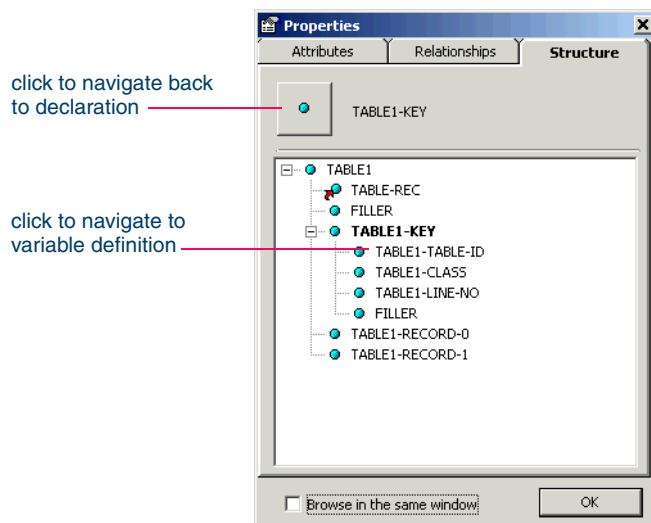
Tip: Click the  button below the Attributes label to navigate back to the construct described in the Properties window.

Figure 1-24 Properties Window — Relationships Tab



4 Click the Structure tab. The Structure tab opens (Figure 1-25).

Figure 1-25 Properties Window — Structure Tab



- 5 The Structure tab displays the context of the selected construct in the parse tree, from its parent to its children. Click a construct in the structure tree to navigate to it in an open HyperView pane.
- 6 Click **OK** to dismiss the window.



Using the List Browser Window

HyperView lists offer convenient ways to navigate program code and record the results of program analyses. HyperView generates lists whenever you perform advanced searches or navigate to a related construct. You can export HyperView lists to a variety of standard formats, and create customized lists.

Note: Some HyperView lists are generated by external tools, such as Component Maker.

Opening a List

Each time you perform an advanced search or navigate to a related construct, HyperView generates a list of target constructs. These lists are predefined and cannot be modified or deleted.

- If more than one target construct is listed, HyperView automatically opens the list in the List Browser (Figure 1-3 on page 1-3) and moves the cursor to the first item in the list in every open HyperView pane.
- If only one target construct is listed, HyperView simply moves the cursor to that construct in every open HyperView pane. To view the list, click the  button next to the  button on the tool bar and choose the list from the drop-down menu of available lists.

Because you can view a Clipper list directly in the Clipper window, HyperView does not automatically open it in the List Browser. Open these lists in the List Browser as described below. Clipper lists can be deleted but not modified.

To open a list:



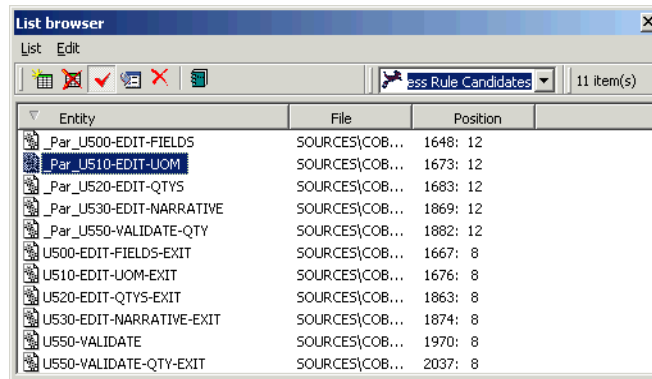
- 1 Click the  button next to the  button on the HyperView tool bar and choose the list you want to open from the drop-down menu of available lists. The list opens in the List Browser (Figure 1-26).

Figure 1-26 List Browser




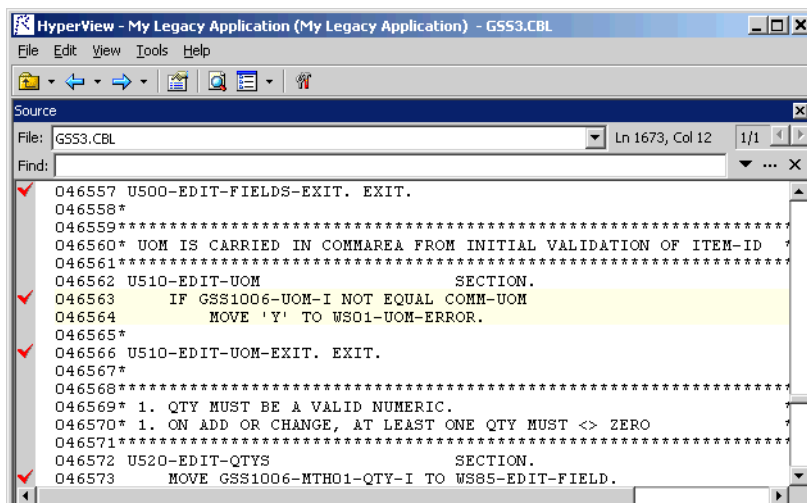
Tip: To place a check mark in the HyperView Source pane display for each item in a list (Figure 1-27), click the  button on the List Browser tool bar.



Figure 1-27 Source Pane with Checked List Items



Creating a List

When you create a list, you can copy constructs to it from another list or directly from the HyperView main window.

To create a list:

- 1 Click the  button next to the  button on the HyperView tool bar and choose **New list** from the drop-down menu of available lists. The New List dialog opens. Enter the name of the new list in the text field and click **OK**. The new list opens in the List Browser (Figure 1-26 on page 1-21).
- 2A To add constructs from another list, choose the list in the drop-down on the List Browser tool bar. When the list opens, select the constructs you want to add and choose **Copy to:New List** in the **Edit** menu. Use the drop-down on the tool bar to return to the new list.
- 2B To add a construct from the HyperView main window, select the construct and choose **Add Selected Construct** in the **Edit** menu.

Tip: Click a column heading in the List Browser to sort the entries by that column. Grab-and-drag the border of a column heading to increase or decrease the width of the column.

To delete an entry in a user-defined list, select it and choose **Delete** in the **Edit** menu. To move an entry from one user-defined list to another user-defined list, select the entry and choose **Move to:New List** in the **Edit** menu.

Exporting a List

You can export lists to a variety of standard formats. Choose **Save** in the **File** menu to export a list to HTML, Excel, RTF, Word, or formatted text.

Deleting a List

You can delete a list by opening it in the List Browser and choosing **Delete** in the **List** menu. You cannot delete the predefined search or relationship lists.

What's Next?

That completes your introduction to HyperView! Now let's take a closer look at how you use HyperView to:

- Analyze program source and context.
- Perform advanced searches.
- Stage program analysis.
- Analyze impacts.
- Analyze program control flows.
- Extract business rules.

Where to Look for Documentation on Other HyperView Tasks

Two other tasks you can perform with HyperView — analyzing program data flows and PL/I call flows — are described in *Analyzing Projects* in the ATW document set. A third task — creating logical components — is described in *Creating Components*.

1-24 Introducing HyperView
What's Next?

Viewing Program Source and Context



The HyperView Source and Context panes offer complementary views of legacy source code. The Source pane displays view-only source for the selected file. The Context pane displays the same code in hierarchical form, in a *parse tree* that defines the relationships between the code constructs that comprise the source.

Using the Source Pane

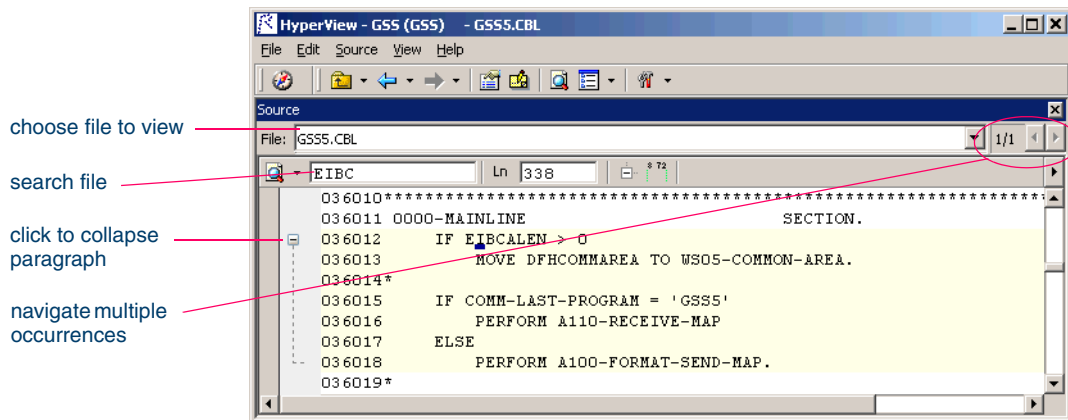
The Source pane displays view-only source for the selected legacy file (Figure 2-1 on page 2-2). The name of the file appears in the **File** drop-down. You can display the source code for an included file by choosing the file in the **File** drop-down.

Click in the Source pane to navigate to a construct. Enter a line number in the **Ln** field above the source code and press Enter to navigate to the line. Tool tips show line numbers in the source file when you scroll the Source pane vertically.

Note: Use the Editor in the ATW main window to modify program source code. For Editor usage, see *Getting Started* in the workbench documentation set.

2-2 Viewing Program Source and Context Using the Source Pane

Figure 2-1 Source Pane



Selecting and Copying Code

There are two ways to select and copy code in the Source pane — as *construct* or as *text*. Which you choose depends on the task you want to perform.

Selecting and Copying Constructs

Click inside a code construct in the Source pane to select it. The selected construct is highlighted in yellow. The number of the first line of the selection is displayed in the **Ln** field above the source code. To copy the construct to the clipboard, choose **Copy Selected Construct** in the **Source** menu.

Selecting and Copying Text

Copy code as text when you are assigning code segments to business rules manually and want to select either more or less code than a construct contains. To select a code segment as text, click-and-drag from the first line of the segment to the last line of the segment. The selected text is highlighted in blue. To copy the selected segment to the clipboard, choose **Copy Selected Text** in the **Source** menu.

Tip: Click the minus sign (-) next to a paragraph, procedure, or data definition to collapse its contents. Click the plus sign (+) to expand its contents.

Navigating to Related Constructs

The **Edit** menu lists choices corresponding to each type of relationship the selected construct has with other constructs in the parse tree. If you select a variable in the Source pane, for example, the **Edit** menu shows **Conditions**, **Port**, and **Declaration** choices. The choices are grayed-out if no relationships of the given type exist for the selected construct.

To view all the constructs in the source file that have a given relationship with a construct, select the construct in the Source pane and choose the appropriate relationship in the **Edit** menu.

- If only one construct has the specified relationship, HyperView simply moves the cursor to that construct in every open HyperView pane.
- If more than one construct has the specified relationship, HyperView automatically opens a list of related constructs in the List Browser and moves the cursor to the first item in the list in every open HyperView pane. To navigate to another item in the list, choose it in the List Browser. For List Browser usage, see [“Using the List Browser Window”](#) on page 1-20.

Tip: For a practical example of how you can use relationships to locate constructs of interest, see [“Understanding HyperView Parse Trees”](#) on page 1-1.



Navigating to Multiple Occurrences of a Construct


If a Cobol copybook, PL/I Include file, or Natural Include file is referenced multiple times in a program — in different structures, for example — you can use the arrows in the upper righthand corner of the Source pane (Figure 2-1 on page 2-2) to navigate between each occurrence of an included construct in the Context pane.

In the Source pane, click on the construct in the included file. The numbers in the upper righthand corner show the sequence number of the current construct versus the total number of constructs. The notation “2/3,” for example, identifies the second occurrence of a construct that occurs three times. Use the arrows to navigate between each occurrence. For Context pane usage, see [“Using the Context Pane”](#) on page 2-7.

Searching for Code

HyperView provides simple and advanced search facilities for source code. For advanced search usage, see [Chapter 3, “Performing Advanced Searches.”](#)

To use the simple search facility, enter the text for the search in the field next to the  button on the tool bar. HyperView locates text matches as you type. Click the  button or press Enter to navigate to the next matching construct.

To view matching constructs in a list, click the adjacent  button. From the drop-down menu, choose:

- **Find All** to display pattern matches in a list.
- **Wildcard Find All** to display wildcard pattern matches in a list. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).
- **Recent Search List** to display the results of the last simple search.

Double-click an item in a list to navigate to it in the Source pane.

Specifying the Change Magnitude for a Source File

Suppose you are planning to implement a change request and want to know how long it will take to complete the change. In this situation, you will typically run an *effort estimation* for your project based on weighted values for selected complexity metrics.

But what if your own analysis of the project shows that a given program will actually take much less time to change than the weighted calculation would suggest. The program might have thousands of source lines, for example, increasing its calculated complexity, while actually being very easy to modify.

A *change magnitude* is a way of overriding the calculated value for a source file. Your “subjective” estimate of the effort involved — Small, Medium, Large, Extra Large — becomes an input to the effort calculation, along with the weighted values.

To specify a change magnitude for a source file, right-click the file in the Source or Context window. In the right-click menu, set the change mag-

nititude to S for Small, M for Medium, L for Large, or XL for Extra Large. For more information on effort estimation, see *Analyzing Projects* in the ATW document set.

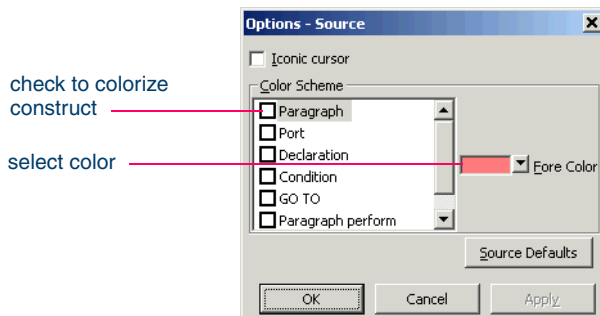
Setting Source Pane Options



In very complex programs, it may be helpful to color-code source constructs. And it's almost always a good idea to specify that the icon for a selected construct replace the standard cursor. Use the Options window for the Source pane to color-code source constructs and "iconize" the cursor.

To change display options:

- 1 In the **Source** menu, choose **Options**. The Options window opens (Figure 2-2).

Figure 2-2 *Options Window*



- 2 In the Color Scheme pane, place a check mark next to the construct type you want to display in color. The current color of the type (if any) is displayed in the **Fore Color** drop-down.
- 3 Click the adjacent  button to edit the color of the construct type. A standard Windows color control is displayed. Use the Palette tab to select the color from the Windows palette. Use the System tab to match the color with the color of standard Windows elements.
- 4 Select **Iconic cursor** to specify that the icon for a selected construct ( for a data port, for example) replace the standard cursor.

Tip: You can restore the default options settings by clicking **Source Defaults**, then choosing **Restore Source Defaults** in the drop-down menu. Choose **Save To** in the drop-down menu to save the option settings to a file. Choose **Load From** in the menu to restore the option settings from a file.

- 5A Click **Apply** if you want to save your settings without dismissing the Options window.
- 5B Click **OK** if you want to save your settings and dismiss the Options window.

Adjusting the Display

Use the choices in the **Source** menu to collapse and expand code, and demarcate code from preceding and trailing enumeration characters.

Collapsing and Expanding Paragraphs or Subprograms

In very long programs, it may be helpful to display the names of paragraphs or subprograms only and hide their source code. To collapse paragraphs or subprograms, choose **Collapse All** in the **Edit** menu. Click the plus sign (+) next to a collapsed paragraph or subprogram to show its source code again. To show the source code for all collapsed paragraphs or subprograms, choose **Expand All** in the **Edit** menu.

Showing Source Code Boundaries

To demarcate source code from leading and trailing enumeration characters in source lines, choose **Show Boundaries** in the **Edit** menu.

Using the Bird's Eye and Model Reference Panes

Use the Bird's Eye pane (Figure 1-18 on page 1-14) with the Source pane to quickly identify the location of a code construct relative to the entire program and any included files.

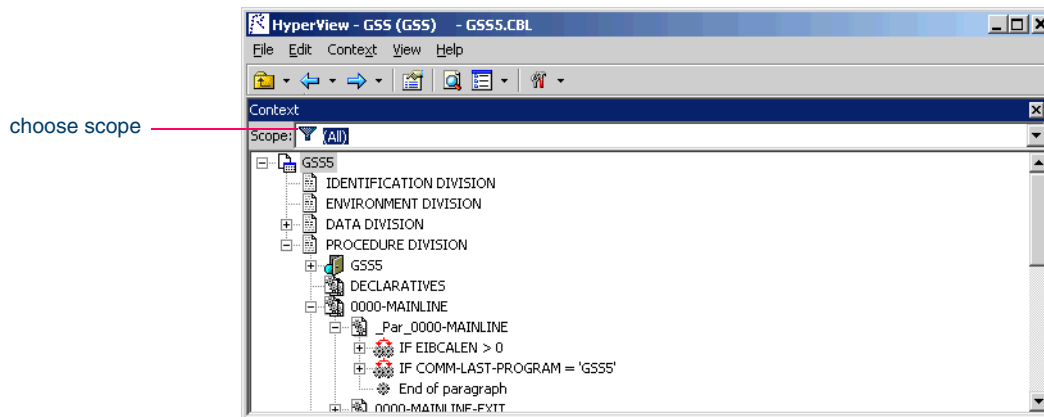
Use the Model Reference pane (Figure 1-19 on page 1-14) to view the parse tree metamodel in text and diagram form.

Using the Context Pane

The Context pane displays the *parse tree* for the selected source file (Figure 2-3). The parse tree displays source code constructs — sections, paragraphs, statements, conditions, variables, and so forth — in hierarchical form, making it easy to locate code constructs quickly. Constructs are displayed in the order they appear in your code. Refer to *Getting Started* in the workbench documentation set for a description of the icons used in the parse tree.

Note: HyperView adds artificial “owning” sections or paragraphs to the parse tree as necessary. An added, or *faked*, section or paragraph is denoted in the parse tree with a leading underscore: *_S1*, for example.

Figure 2-3 Context Pane



The *scope* of the parse tree determines the constructs and relationships it displays — all constructs, only control statements, only declarations, and so on. Scopes include the lists HyperView generates when you perform advanced searches, navigate to a related construct, or identify candidates in Clipper. Choose the scope of the parse tree in the **Scopes** drop-down on the Context pane tool bar.

Click the plus sign (+) next to a construct to expand its hierarchy. Click the minus sign (-) to collapse the hierarchy. The **Edit** menu lists every relationship type for the construct selected in the Context pane.

2-8 Viewing Program Source and Context *What's Next?*

Perform other tasks as you do in the Source pane:

- Navigate to related constructs as described in [“Navigating to Related Constructs” on page 2-3](#).
- Navigate to multiple occurrences of a construct as described in [“Navigating to Multiple Occurrences of a Construct” on page 2-3](#).
- Set change magnitudes as described in [“Specifying the Change Magnitude for a Source File” on page 2-4](#).

What's Next?

Now that you know how to view program source and context in HyperView, let's look at how use the HyperView advanced search facility to create sophisticated filters for construct searches. That's the subject of the next chapter.

Performing Advanced Searches



The HyperView advanced search facility distills the parse tree metamodel into a series of prompts that help you build complex filters for construct searches. It also offers predefined filters that you can use to search for dead code, GO TO statements, hard-coded constants, nested conditionals, file and screen reads and writes, program calls, SQL ports, and more.

Understanding Advanced Searches

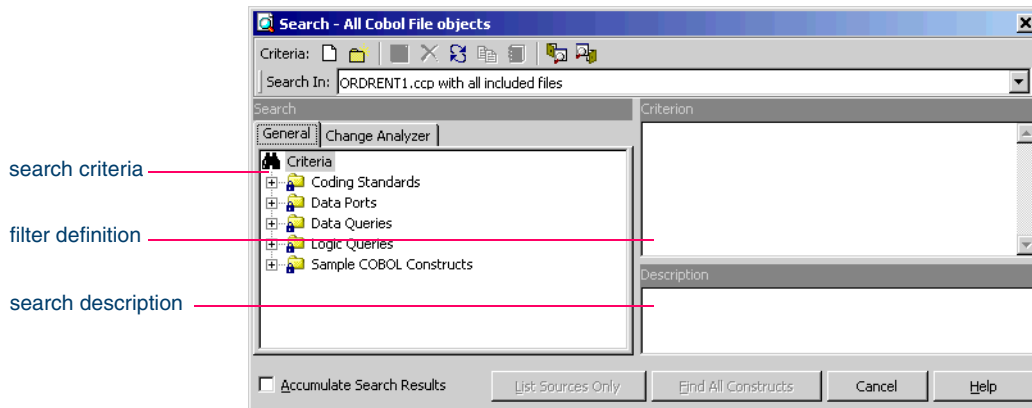
The advanced search facility uses the *attributes* and *relationships* of a construct to define the conditions you set in your search for the construct. As an example of how the facility works, let's look at how you might set up a search filter for the conditions a program uses to validate the variable EIBRESP.

To set up a search filter:

- 1 In the HyperView **Edit** menu, choose **Find**. The Search window opens. Click the **General** tab (Figure 3-1 on page 3-2).

3-2 Performing Advanced Searches
Understanding Advanced Searches

Figure 3-1 Search Window — General Tab




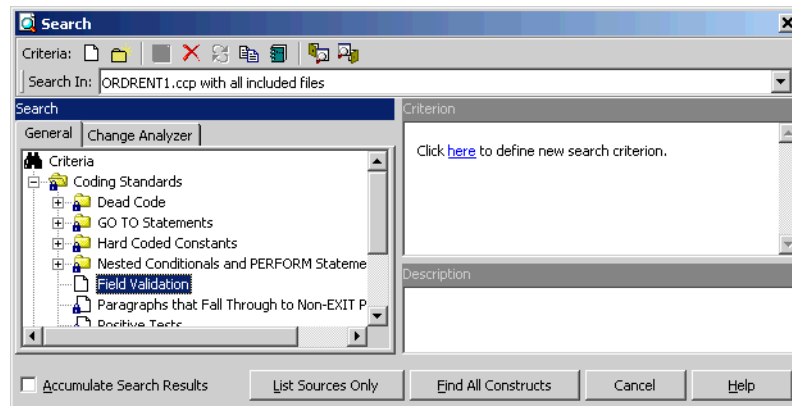
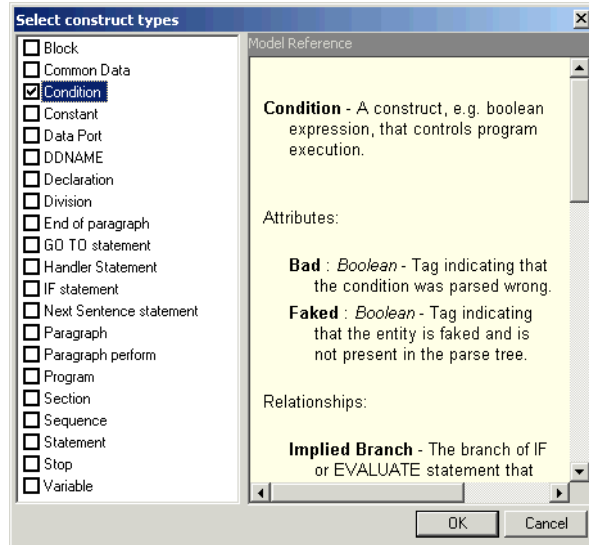
- 2 Select a folder for the new criterion in the General tab, then click the  button on the tool bar. The New Criterion dialog opens. Enter Field Validation in the text field and click **OK**. HyperView creates the Field Validation criterion in the selected folder (Figure 3-2).

Figure 3-2 Search Window — After Step 2



- 3 Click the [here](#) link in the righthand pane of the Search window. The Select Construct Types window opens (Figure 3-3 on page 3-3).

Figure 3-3 *Select Construct Types Window*

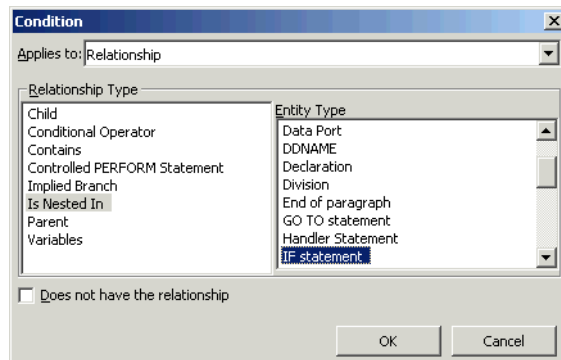


- 4 Select **Condition** in the list of constructs in the lefthand pane. In the righthand pane, review the definition of a condition in the parse tree metamodel, then click **OK**. HyperView adds the condition construct to the filter definition in the righthand pane of the Search window:

Find [All Condition](#)

- 5 Click the **All** link. The Condition window opens (Figure 3-4).

Figure 3-4 *Condition Window (After Editing)*



6 In the Condition window, choose:

- Relationship in the **Applies to** drop-down
- Is Nested In in the **Relationship Type** list box
- If Statement in the **Entity Type** list box

Click **OK**. HyperView adds the relationship to the filter definition in the righthand pane of the Search window:

Find **Condition**
which **is nested in any** IF statement

7 Click the **any** link. The Condition window opens. In the Condition window, choose:

- Relationship in the **Applies to** drop-down
- Contains in the **Relationship Type** list box
- Paragraph perform in the **Entity Type** list box

Click **OK**. HyperView adds the relationship to the filter definition in the righthand pane of the Search window:

Find **Condition**
which **is nested in** IF statement which **contains any**
Paragraph perform

8 Click the **any** link. The Condition window opens. In the Condition window, choose:

- Attribute in the **Applies to** drop-down
- Caption in the **Name** list box
- Like in the **Operations** drop-down

Enter *ERROR* in the **Values** field and click **OK**. HyperView adds the attribute to the filter definition in the righthand pane of the Search window:

Find **Condition**
which **is nested in** IF statement which **contains** Paragraph
perform such that **Caption** Like "*"ERROR*"

Note: Consult the Model Reference pane for attribute values.

9 Click the **is nested in** link. In the pop-up menu, choose **And**. The Condition window opens. In the Condition window, choose:

- Relationship in the **Applies to** drop-down
- Contains in the **Relationship Type** list box
- Variable in the **Entity Type** list box

Click **OK**. HyperView adds the relationship to the filter definition in the righthand pane of the Search window:

Find [Condition](#)
which [is nested in](#) IF statement which [contains](#) Paragraph
perform such that [Caption](#) Like **"*ERROR*"**
[and](#)
which [contains any](#) Variable


- 10 Click the **any** link. The Condition window opens. In the Condition window, choose:

- Attribute in the **Applies to** drop-down
- Caption in the **Name** list box
- Like in the **Operations** drop-down

Enter ***EIB*** in the **Values** field and click **OK**. HyperView adds the attribute to the filter definition in the righthand pane of the Search window:

Find [Condition](#)
which [is nested in](#) IF statement which [contains](#) Paragraph
perform such that [Caption](#) Like **"*ERROR*"**
[and](#)
which [contains](#) Variable such that [Caption](#) Like **"*EIB*"**

- Tip:** To edit an element in the filter definition, select its link and choose **Edit** in the pop-up menu. To delete an element from the filter definition, select its link and choose **Delete** in the pop-up menu.

- 11 Enter a description of the filter definition in the Description pane.
- 12 Click the  button on the tool bar to save the filter definition. To execute the search, follow the instructions in [“Searching for Similarly Structured Code”](#) on page 3-6.

Searching for Similarly Structured Code

Paragraphs that perform business logic often have identically or similarly structured code. The advanced search facility lets you use the *signature* of a paragraph to search for instances of similarly structured code. A signature identifies the structure of a paragraph, as in the following example:

```
BLOCK
IF
    THEN
        Perform_PROC
```

Note: This feature is not available for PL/I and Natural.

To search for similarly structured code:



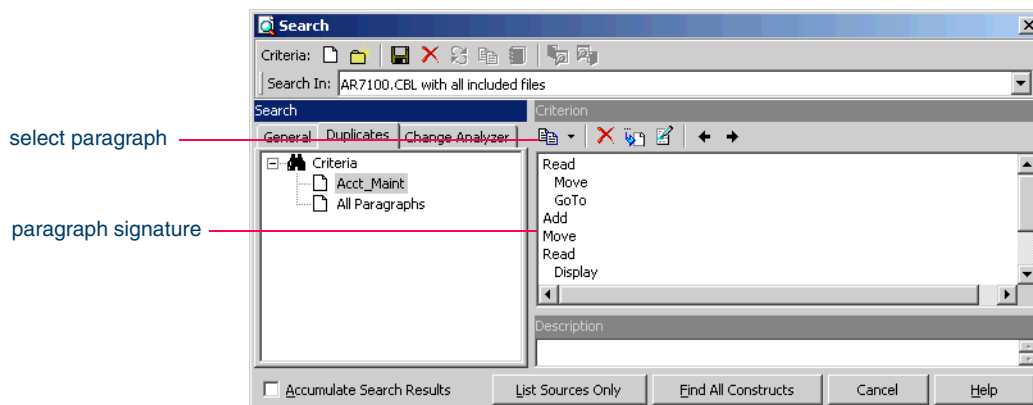



- 1 In the HyperView **Edit** menu, choose **Find**. The Search window opens. Click the **Duplicates** tab (Figure 3-5).
- 2 Define a new search criterion, as described in [“Understanding Advanced Searches”](#) on page 3-1.
- 3 Click the  button next to the  button on the tool bar and choose the paragraph whose signature you want to use from the drop-down menu of available paragraphs. HyperView displays the signature of the selected paragraph in the Search window (Figure 3-5).

Figure 3-5 Search Window — Duplicates Tab



Tip: Use the  and  buttons on the tool bar to adjust the nesting level of a line in the signature. Select a line in the signature and click the  button on the tool bar to delete the line.



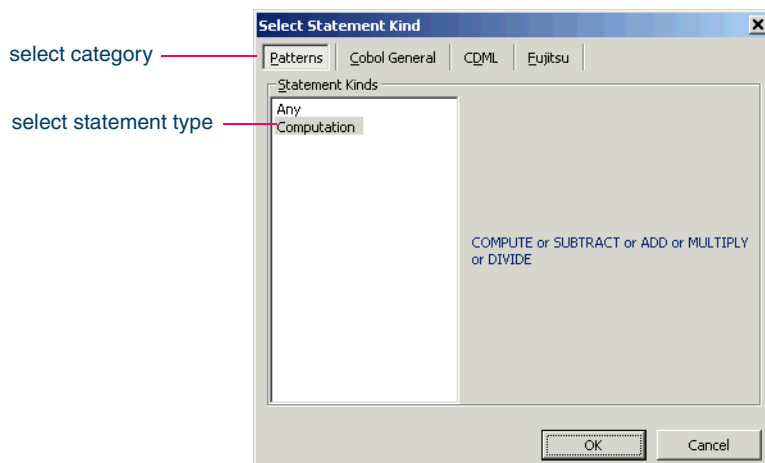

- 4 Insert statements in a signature by selecting a line in the signature and clicking the  button on the tool bar. HyperView inserts an ellipsis (...) above the selected line. Select the ellipsis and click the  button on the tool bar. The Select Statement Kind window opens (Figure 3-6).

Figure 3-6 *Select Statement Kind Window — Duplicates Tab*



- 5 Click the button for the category of statement you want to insert. In the Statement Kinds pane, choose the statement you want to insert and click **OK**. HyperView replaces the ellipsis with the selected statement.

Tip: You can also replace a line in the signature with a statement.

- 6 Click the  button on the tool bar to save the search criterion. To execute the search, follow the instructions in [“Searching for Similarly Structured Code”](#) on page 3-6.

Executing Advanced Searches

You can execute an advanced search in standalone mode, against the program or construct selected in the Source or Context panes; or you can execute the search in Clipper, in batch mode. For Clipper usage, see [Chapter 4, “Staging Program Analysis with Clipper.”](#)

Note: The Change Analyzer tab opens a scoped version of the advanced search facility for Change Analyzer. For Change Analyzer usage, see *Analyzing Projects* in the ATW document set.

To execute an advanced search in standalone mode:


- 1 In the HyperView **Edit** menu, choose **Find**. The Search window opens. Click the **General** tab (Figure 3-1 on page 3-2).
- 2 In the **Search In** drop-down on the tool bar, choose:
 - **Source file name with All Included Files** if you want to execute the search against the selected source file and any included files.
 - **Source file name Only** if you want to execute the search against the selected source file only.
 - **Selected Construct - Construct** if you want to execute the search against the construct selected in the Source or Context panes.
- 3 Choose the criterion for the search you want to execute from the list of criteria in the lefthand pane.
- 4 Select **Accumulate Search Results** if you want the results of the search to be added to the results of the previous search. Otherwise, the new results overwrite the previous results.
- 5A Select **Find All Constructs** if you want HyperView to create a list of all the constructs found by the search.
- 5B Select **List Sources Only** if you want HyperView simply to indicate whether the construct of interest exists in the searched files.
- 6 HyperView displays a dialog box with the results of the search. Click **OK**. The dialog and Search window are dismissed.


- If you selected **Find All Constructs** and only one construct was found, HyperView moves the cursor to that construct in every open HyperView pane.
- If you selected **Find All Constructs** and more than one construct was found, HyperView displays the results in the List Browser window (Figure 1-26 on page 1-21). For List Browser usage, see [“Using the List Browser Window” on page 1-20](#).


Defining Advanced Search Criteria


Define advanced search criteria as described in [“Understanding Advanced Searches” on page 3-1](#). You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).

Editing a Search Criterion To edit a search criterion, select the criterion, then select the link for the element you want to change in the filter definition. Choose **Edit** in the pop-up menu to edit the element, or **Delete** to delete the element.


Tip: To restore the original definition of a predefined search criterion, select the criterion and click the  button on the tool bar.


Copying a Search Criterion To copy a search criterion, select the criterion and click the  button on the tool bar. The New Criterion dialog opens. Enter the name of the new criterion in the text field and click **OK**.


Saving a Search Criterion To save a search criterion, select the criterion and click the  button on the tool bar.



Creating Folders for Search Criteria To create a folder for search criteria, click the  button on the tool bar. The New Folder dialog opens. Enter the name of the new folder in the text field and click **OK**. The new folder appears in alphabetical order in the tree in the lefthand pane of the window. Drag-and-drop search criteria to move them to the folder. You can create folders within folders.


3-10 Performing Advanced Searches *What's Next?*

To copy a folder and all its contents, select the folder and click the  button on the tool bar. The New Folder dialog opens. In the text field, enter text to be prepended to the folder name and to the names of each of its subfolders and search criteria, and click **OK**.

To modify a folder name, click in the name area for the folder to make the name editable, enter the new name, and press Enter. To delete a folder, select the folder and click the  button on the tool bar. You are prompted to confirm the deletion. Click **OK**.

Saving a Search Criterion as HTML To save a search criterion in an HTML file, select the criterion and click the  button on the tool bar. The Save Criteria dialog opens, where you can specify the name and folder for the HTML file.

Exporting and Importing a Search Criterion To export a search criterion to an XML file, select the criterion and click the  button on the tool bar. The Export Criteria dialog opens, where you can specify the name and folder for the XML file. To import a search criterion, click the  button on the tool bar. The Import Criteria dialog opens, where you can select the criterion you want to import.

Deleting a Search Criterion To delete a search criterion, select the criterion and click the  button on the tool bar. You are prompted to confirm the deletion. Click **OK**.

Note: You cannot delete a predefined search criterion.

What's Next?

That's all you need to know to perform an advanced search in HyperView. Now let's look at how you might use the search results to generate lists of candidates for business rule extraction, event injection, and impact analysis in Clipper.

Staging Program Analysis with Clipper



Clipper lets you perform batch searches for constructs of interest and stage the results as input for subsequent tasks — impact analysis, business rule extraction, event injection, and the like. A system analyst, for example, might use Clipper to create a list of programs impacted by a change request, then create a project that contains only the source files for those programs.

Getting Started in Clipper

Let's look at a sample use of Clipper that shows you how to create a set of business rules for segments of code that write to a file. The sample should help you get oriented in the tool. Usage for other tasks is similar.

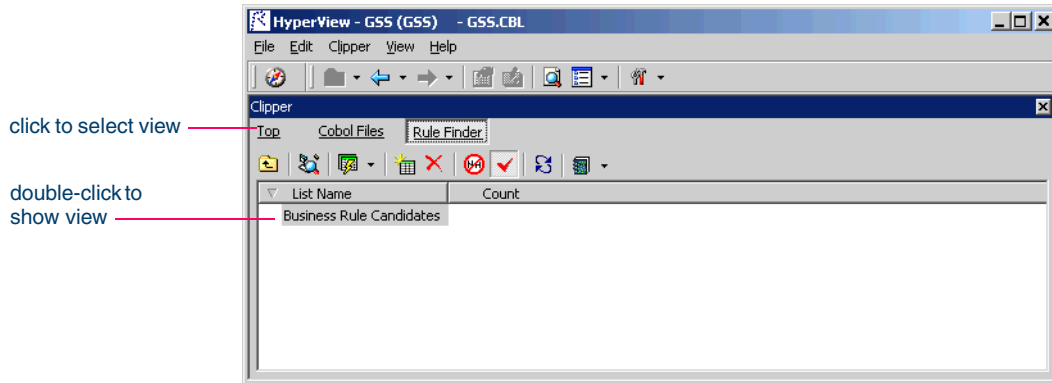
To create business rules with Clipper:

- 1 In the HyperView **View** menu, choose **Clipper**. The Clipper window opens. Click the **Top** view link.
- 2 The Top view opens. In the Model Name column, double-click Cobol File.

4-2 Staging Program Analysis with Clipper
Getting Started in Clipper

- 3 The Cobol File view opens. In the Category Name column, double-click Rule Finder. The Rule Finder view opens (Figure 4-1).

Figure 4-1 *Clipper Pane — Rule Finder View*




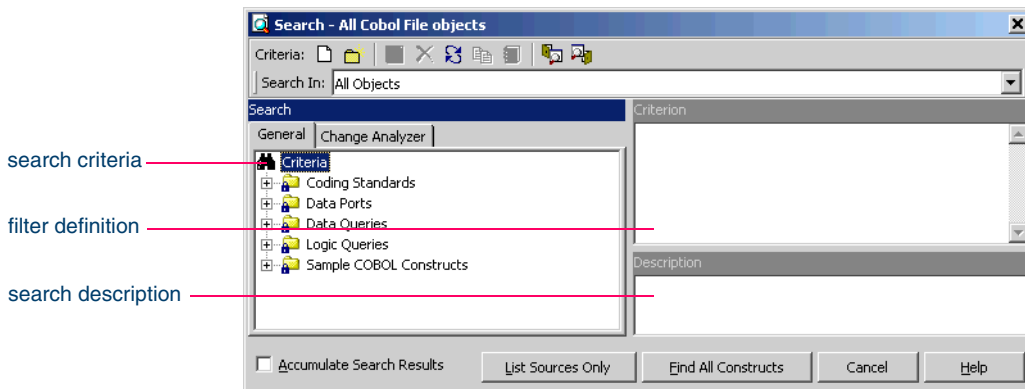
- 4 Click the  button on the tool bar. The Search window opens (Figure 4-2).

Figure 4-2 *Search Window for Clipper*

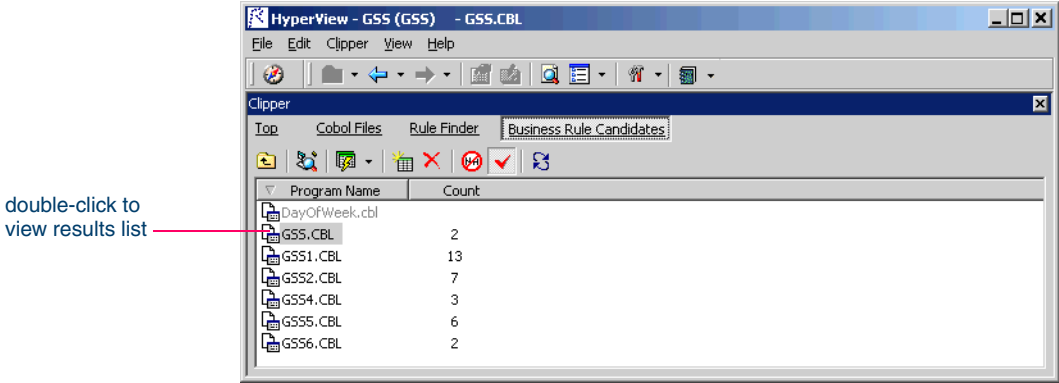



- 5 In the **Search In** drop-down, select **All Sources**.
- 6 Expand the Data Ports folder in the lefthand pane, then expand the Files folder. Select the predefined File Writes criterion and click **Find All Constructs**. HyperView displays a dialog box with the

results of the search operation. Click **OK**. The Search window is dismissed and the results of the search are displayed in the Clipper Business Rule Candidates view (Figure 4-3).

Note: You can modify the predefined criterion or create your own criterion as described in [“Defining Advanced Search Criteria”](#) on page 3-9. For Clipper search execution, see [“Executing Clipper Searches”](#) on page 4-9.

Figure 4-3 *Clipper Pane — Business Rule Candidates View*



Tip: Select a construct in the Source or Context panes and click the  button on the tool bar to add the construct to the list of candidates manually.



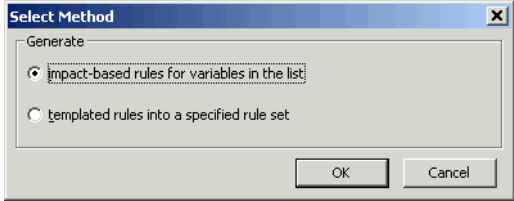
7 Click the  button next to the  button on the tool bar and choose **Create Rules** in the drop-down menu. The Select Method dialog opens (Figure 4-4).

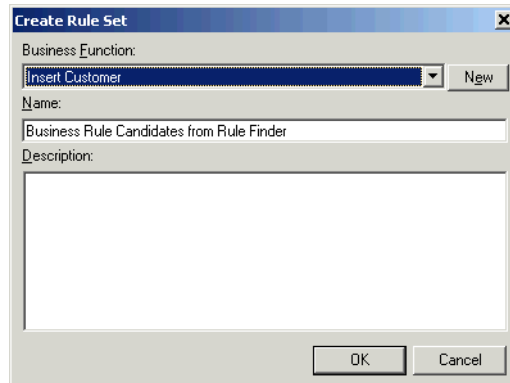
Figure 4-4 *Select Method Dialog*



- 8 In the Select Method dialog, select **templated rules into a specified rule set**. The Create Rule Set window opens (Figure 4-5).

Note: For information on impact-based rule autodetection, see [“Generating Business Rules” on page 4-8](#).

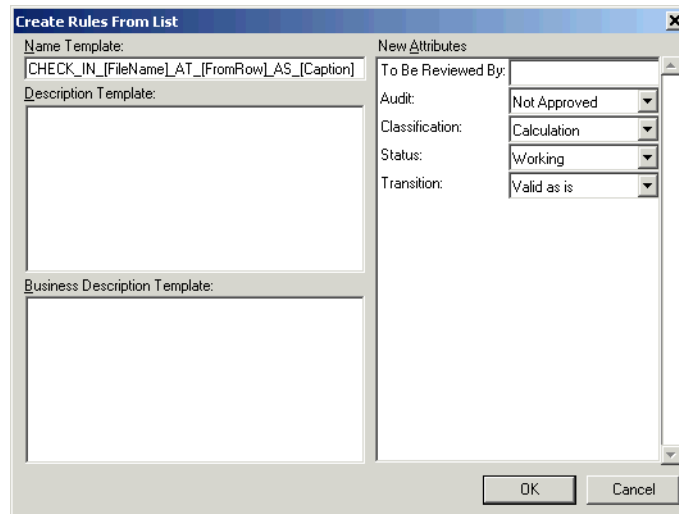
Figure 4-5 *Create Rule Set Window*



- 9 In the **Business Function** drop-down, select the business function for the set of rules you want to create. The current business function is displayed. To specify a new business function, click **New**. The Business Function Dialog opens. In the Business Function dialog, enter the name and description of the new business function and click **OK**.
- 10 In the **Name** field, enter the name of the rule set for the rules you want to create and click **OK**. The Create Rules From List window opens (Figure 4-6 on page 4-5).

Note: For background on business rules, see [Chapter 8, “Extracting Business Rules.”](#)

Figure 4-6 Create Rules From List Window



- 11 In the **Name Template** field, enter a template for the rule name attribute. You can use any combination of text and bracketed property names for the constructs returned by the search. So if you are batch processing data ports, the entry:

```
File Write [Caption]
```

might produce the following rule names:

```
File Write REWRITE CATALOG-MASTER
File Write REWRITE FCOST-MASTER-REC
File Write WRITE DATASET('INVOICE')
```

Valid properties are any property of a construct, plus:

- *Program* — the name of the program that contains the construct.
- *FileName* — the name of the source file that contains the construct.
- *FromRow* — the number of the first line in the source that contains the construct.
- *Caption* — the label of the construct in the parse tree.

You can substitute a business name for an I/O element in a property value by appending “/bus” to the property name. If the CATALOG-MASTER variable has the business name MASTER CATALOG:

```
File Write [Caption/bus]
```

produces the output:

```
File Write REWRITE MASTER CATALOG
```


Note: For background on business names, see *Analyzing Projects* in the ATW documentation set.

- 12 In the **Description Template** and **Business Description Template** fields, enter templates for the description and business description attributes, respectively. See [step 11 on page 4-5](#) for instructions and an example of the output you can expect.
- 13 In the New Attributes list box, specify the value for each attribute you want to set in the selected rules.
- 14 When you are satisfied with your entries in the Create Rules From List window, click **OK**. HyperView displays an informational message that tells you how many business rules it created. Click **OK**.

Using the Clipper Pane

The Clipper pane consists of a hierarchy of views that let you specify the candidate lists you want to display:

- The *Top* view displays the types of source files you can create lists for.
- The *Category* view displays the categories of lists you can create for the selected source file type.
- The *List* view displays the types of lists in the selected category.
- The *File* view displays the source files of the selected type in the current project.
- The *Construct* view displays each construct in the list for the selected source file.


Each view gives you access to the next-level view in the hierarchy. See Figure 4-1 on page 4-2 for basic navigation details. Click the  button on the tool bar to navigate to the parent of the current view.


Sorting Entries Click a column heading in a view to sort the view entries by that column.

Sizing Columns Grab-and-drag the border of a column heading to increase or decrease the width of the column.

Working with Categories


The Category view displays the categories of lists you can generate for a source file type. Generally speaking, categories correspond to tasks you perform with a list — event injection or business rule extraction for a Cobol source file, for example. HyperView supplies default categories for each source type. You can create your own categories as necessary.


Creating Categories In the Category view, click the  button on the tool bar to create a new category for the selected source type. The New Category dialog opens, where you can specify the name of the new category.


Deleting Categories In the Category view, select a category and click the  button on the tool bar to delete the category. You cannot delete a predefined category.


Working with Lists


The List view displays the lists in a category. HyperView supplies default lists for each category — calls and ports, for example, for the event injection category. You can create your own lists as necessary.


Creating Lists In the List view, click the  button on the tool bar to create a new list for the selected category. The New List dialog opens, where you can specify the name of the new list.


Deleting Lists In the List view, select a list and click the  button on the tool bar to delete the list.


Generating Lists Automatically In the List view, select a list and click the  button on the tool bar to open the Search window, where you can use the advanced search facility to generate list constructs automatically. For more information, see [“Executing Clipper Searches” on page 4-9](#).

Adding Constructs to a List Manually In the File view, select a source file, then choose a construct in the Source or Context panes. Click the  button on the tool bar to add the construct to the list of candidates for the selected file.



Deleting Constructs from a List In the Construct view, select a construct and click the  button on the tool bar to delete the construct from the selected list.

Hiding Files with Empty Lists In the File view, click the  button on the tool bar to hide source files with empty lists. Displayed source files with empty lists are grayed out.

Highlighting List Constructs in Source Code In the File view, select a source file and click the  button on the tool bar to place a check mark in the Source pane display for each item in the list of candidates for the selected file.

Refreshing Lists In the List view, select a list and click the  button on the tool bar to refresh the list.

Generating Business Rules

To generate business rules for all the constructs in a list, select the list in the List view for the business rule category, click the  button next to the  button on the tool bar, and choose **Create Rules** in the drop-down menu.



In the Select Method dialog (Figure 4-4 on page 4-3), you are prompted to autodetect segments that impact the constructs in the list, or generate business rules in a specified rule set:

- Select **impact-based rules for variables in the list** if you want Clipper to create rules for list constructs and any code segments that impact the constructs. The constructs must be variables. For more information, see [“Autodetecting Business Rules” on page 8-5](#).

- Select **templated rules into a specified rule set** if you want Clipper to create rules in a specified rule set. When you choose this option, you batch process rule attributes, including their names, in the generation step. For more information, see steps [9-14](#).

Tip: As long as the Rules pane is open, you can generate a rule for a single construct by selecting the construct in the Construct view and choosing **Create Rule** in the **Rules** menu. Follow the instructions for creating a rule manually in [“Extracting Business Rules Manually”](#) on page 8-4.


Showing Impacts

To show impacts for all the constructs in a list, select the list in the List view for the impact report category, click the  button next to the  button on the tool bar, and choose **Show Impacts** in the drop-down menu. HyperView displays the impacts for the constructs in the list in the Impact Report window. For usage, see [Chapter 5, “Analyzing Impact Traces.”](#)

Executing Clipper Searches

Use a Clipper search to generate Clipper lists automatically. You can modify the predefined search criteria or create your own criteria as necessary. For more information, see [“Defining Advanced Search Criteria” on page 3-9](#) and [“Using Construct Lists to Narrow a Clipper Search” on page 4-10](#).

To execute a Clipper search:

- 1 Click the  button on the tool bar. The Search window opens (Figure 4-2 on page 4-2).
- 2 In the **Search In** drop-down, select:
 - **All Sources** to execute the search against all the sources in the project.
 - **Sources in view** if you want to execute the search only against sources with non-zero counts in the selected file view. If you select **Sources in Impact Report.Roots**, for example, the search

4-10 Staging Program Analysis with Clipper
Using Construct Lists to Narrow a Clipper Search

will be executed only against sources with non-zero counts in the Roots file view in the impact report category for Cobol files.



- 3** Choose the criterion for the search you want to execute from the list of criteria in the lefthand pane.
- 4** Select **Accumulate Search Results** if you want the results of the search to be added to the results of the previous search. Otherwise, the new results overwrite the previous results.
- 5A** Select **Find All Constructs** if you want HyperView to create for each searched file a list of all the constructs found by the search.
- 5B** Select **List Sources Only** if you want HyperView simply to indicate for each searched file (with the notation >0) whether the construct of interest exists in the file.
- 6** HyperView displays a dialog box with the results of the search operation. Click **OK**. The Search window is dismissed and the results of the search are displayed in the appropriate Clipper file view.

Using Construct Lists to Narrow a Clipper Search

Once you have generated a construct list in Clipper, you can use the list itself as a condition of a search — you can use the search results to refine a subsequent search.

Suppose you have performed a Clipper search for duplicate paragraphs in Cobol programs, as described in [“Searching for Similarly Structured Code” on page 3-6](#). The next step might be to search for all PERFORM statements that reference the duplicate paragraphs. The example below shows how you might use a Clipper list of duplicate paragraphs as a condition of the search.

To narrow a Clipper search with a construct list:

- 1** Click the  button on the tool bar. The Search window opens (Figure 4-2 on page 4-2).
- 2** Select a folder for the new criterion in the General tab, then click the  button on the tool bar. The New Criterion dialog opens. Enter

Perform `Duplicates` in the text field and click **OK**. HyperView creates the `Perform Duplicates` criterion in the selected folder.

- 3 Click the [here](#) link in the righthand pane of the Search window. The `Select Construct Types` window opens (Figure 3-3 on page 3-3).
- 4 Select **Paragraph Perform** in the list of constructs in the lefthand pane, then click **OK**. HyperView adds the condition construct to the filter definition in the righthand pane of the Search window:

Find [All Paragraph Perform](#)

- 5 Click the **All** link. The `Condition` window opens (Figure 3-4 on page 3-3).
- 6 In the `Condition` window, choose:
 - Relationship in the **Applies to** drop-down
 - Called Paragraph in the **Relationship Type** list box

Click **OK**. HyperView adds the relationship to the filter definition in the righthand pane of the Search window:

Find [Paragraph Perform](#)
which [has related any](#) Paragraph (as Called Paragraph)

- 7 Click the **any** link. The `Condition` window opens (Figure 3-4 on page 3-3).
- 8 In the `Condition` window, choose:
 - List in the **Applies to** drop-down
 - `Duplicates.Matched Paragraphs` in the **Name** drop-down

Notes: `Duplicates.Matched Paragraphs` is the name of the Clipper list that HyperView created when you searched for duplicate paragraphs in the first place.

You can filter out list constructs by selecting **Does not belong to the list** in the `Condition` window.

Click **OK**. HyperView adds the list to the filter definition in the righthand pane of the Search window:

Find [Paragraph Perform](#)
which [has related](#) Paragraph (as Called Paragraph) which [is in list](#)

Duplicates.Matched Paragraphs





- 9 Execute the search as described in [“Executing Clipper Searches” on page 4-9](#). HyperView displays a dialog box with the results of the search operation. Click **OK**. The Search window is dismissed and the results of the search are displayed in the appropriate Clipper file view.

Tip: As a practical matter, you would probably also want to search for GO TO statements that reference the duplicate paragraphs. The results of this search you could simply add to the results of the search for PERFORM statements by using the **Accumulate Search Results** button described in [“Executing Clipper Searches” on page 4-9](#).

Creating Projects in Clipper

You can create a project directly in Clipper from the results of your analysis. The project contains only source files with non-empty lists of the selected type. You can also add source files to an existing project.

To create or add to a project:

- 1A To create a project that contains all source files with non-empty lists, double-click the list in the List view, click the  button next to the  button on the tool bar, and choose **Create Project** in the drop-down menu. The New Project dialog opens. Enter the name of the new project in the text field and click **OK**. The Change Magnitude window opens (Figure 4-7).
- 1B To add to an existing project all source files with non-empty lists, double-click the list in the List view, click the  button next to the  button on the tool bar, and choose **Add to Project Name project** in the drop-down menu. The Change Magnitude window opens (Figure 4-7 on page 4-13).
- 2 In the Change Magnitude window, select the **Automatically calculate change magnitude** check box if you want Clipper to set change magnitudes for the listed source files based on the ranges specified

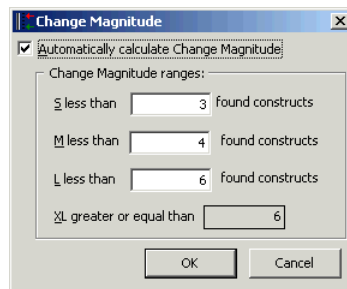
in the fields below the check box. The Clipper settings will override any existing change magnitudes for the files.

The fields below the check box contain default ranges for the available values: Small, Medium, Large, and Extra Large. Compare the number of constructs in each source file listed in the File view with the default ranges, then modify the ranges as necessary.

If you want Cobol source files with 6 to 10 constructs to have a Large change magnitude, for example, set the range for Medium to less than 6 and the range for Large to less than 11. When you are satisfied with your choices, click **OK**.


For more information on change magnitudes, see [“Specifying the Change Magnitude for a Source File” on page 2-4](#).

Figure 4-7 *Change Magnitude Window*




Generating Clipper Reports

Use Clipper reports to show the distribution of list instances across a project and view the details for each instance. The following reports are available:

- For each list in a category, the *Metrics Report* shows the number of list items in each program in the project. The report for the default event injection category, for example, shows the number of calls and ports. To generate a Metrics Report, select a category, then click the  button on the tool bar and choose the report type from the pull-down menu.

4-14 Staging Program Analysis with Clipper *What's Next?*

- For each instance in a list, the *Details Report* shows the program in which it occurs, its location in the program, and other attributes. You can customize the report to include any HyperView attribute related to the instance. To generate a Details Report, select a list, then click the  button on the tool bar and choose the report type from the pull-down menu. In the Select Attributes dialog, choose the attributes you want to view in the report.

Printing Reports

Click **Page Setup** in a report window to set up the page for a printed report. Click **Print** to print the report. The Print dialog opens, where you can set options for the print job. Click **OK**.

Exporting Reports

You can export Clipper reports to a variety of standard formats. Click **Save** in a report window to export a report to HTML, Excel, RTF, Word, or formatted text. A Save As dialog opens, where you can specify the name, location, and file type of the report.

Note: Excel is usually the best choice if you need to manipulate the report or perform computations.

What's Next?

Now that you know how to search programs in HyperView and stage program analysis in Clipper, you're ready to drill down deeper into programs. The next chapter looks at how you analyze impacts in HyperView.

Analyzing Impact Traces



An *impact trace* describes the flow of data within and between programs in an application: how data items exchange values, use each other in computations, and so forth. HyperView offers two tools for analyzing impact traces. Which one you use depends on whether you are interested in data flow analysis or traditional impact analysis:

- The Impact pane displays a hierarchical view and diagram of an impact trace. Use these views to analyze the data flows in a project.
- The Impact Report window displays a hierarchy of views that let you narrow an impact trace based on the depth of the affected item or the program that contains the affected item. Use these views to perform traditional impact analysis — to hone in on the data items in a project that may be impacted by changing another data item’s definition or usage.

Note: Set **Enable Impact Report** and **Enable Data Element Flow** in the Verification tab of the Project Options window to enable these tools. JCL files must have been verified according to the instructions in *Preparing Projects* in the ATW document set.

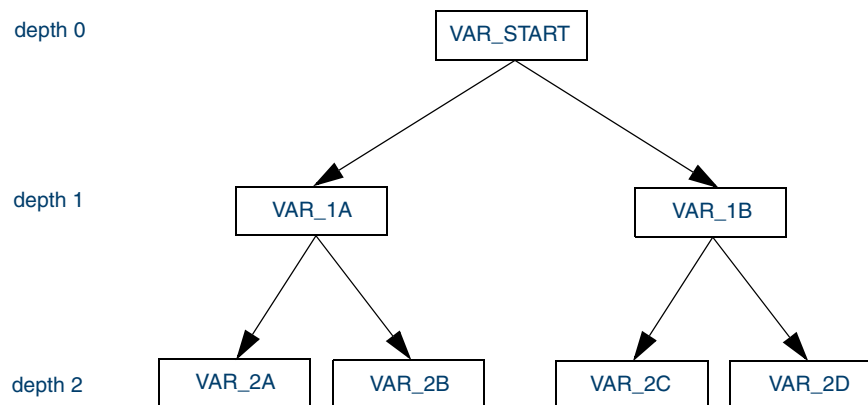
Using the Impact Report Tool

The Impact Report tool lets you answer the kinds of “What if?” questions posed in the recent past by the industry-wide changes for Y2K, Zip+4, and the Euro dollar: “What if I change the type of this variable, or the length of this field?”

Understanding Impact Report Depths

An impact report typically shows the flow of data from a *startup item* to every data item that would be affected by its modification. The report is organized in hierarchical form according to the *depth* of the affected item — items that would be directly affected by a change in the startup item are at depth 1, items that would be affected by a change in depth 1 items are at depth 2, and so on (Figure 5-1).

Figure 5-1 *Impact Report for Affected Data Items*



An impact report is organized in hierarchical form according to the depth of affected items.

An impact trace can also show the flow of data *to* a startup item from the data items that affect it — the *causes* of the startup item rather than its *consequences*. For complex tasks, such as changing the precision of a floating point variable, you can combine the two modes and show causes and consequences in the same trace.

Generating an Impact Trace in the Impact Report Tool

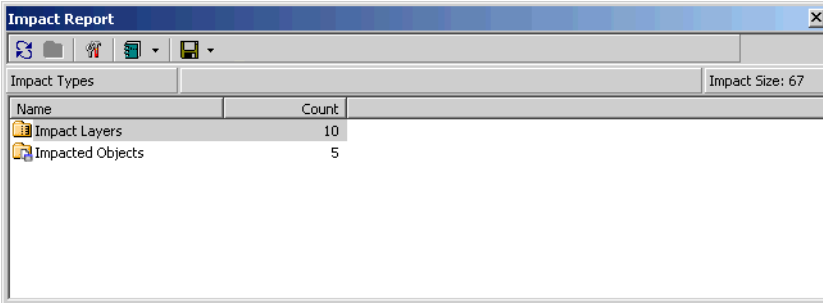
You can select the startup item for an impact trace manually in the Source or Context panes, or create a list of startup items in Clipper.

Note: For Clipper usage, see [Chapter 4, “Staging Program Analysis with Clipper.”](#)

To generate an impact trace in the Impact Report tool:

- 1 In the Source or Context pane, select the construct that contains the startup item or items and choose **Impact Report** in the **View** menu. The Impact Report window opens (Figure 5-2).

Figure 5-2 *Impact Report Window*



Name	Count
Impact Layers	10
Impacted Objects	5

- 2A To navigate to an impact trace based on the depth of the affected item (see [“Using the Impact Report Tool”](#) on page 5-2), double-click the Impact Layers folder. The Impact Layers view opens. Double-click the appropriate depth. The Impacted Data Items view opens (Figure 5-3). Continue to [step 3](#).
- 2B To navigate to an impact trace based on the object that contains the affected item, double-click the Impacted Objects folder. The Impacted Objects view opens. Double-click the appropriate object. The Impacted Data Items view opens (Figure 5-3 on page 5-4). Continue to [step 3](#).

5-4 Analyzing Impact Traces
Using the Impact Report Tool

Figure 5-3 Impacted Data Items View

Data Item Decl	Program	Impact Factor	File	Ln:Col
COMM-NEXT-BROWSE-KEY	GSS5	4	SOURCES\COBOL\GSS5.CBL	56: 12
DFHCOMMAREA	GSS4	1	SOURCES\COBOL\GSS4.CBL	197: 8
DFHCOMMAREA	GSS5	1	SOURCES\COBOL\GSS5.CBL	329: 8
GSS10050	GSS5	2	SOURCES\COBOL\GSS5.CBL	212: 8
TCB-USER-ID	GSS5	1	SOURCES\COPYBOOK\ON...	25: 12
WS05-COMMON-AREA	GSS	1	SOURCES\COBOL\GSS.CBL	37: 8
WS05-COMMON-AREA	GSS4	1	SOURCES\COBOL\GSS4.CBL	50: 8
WS10-SCR-ADJ-NO	GSS5	3	SOURCES\COBOL\GSS5.CBL	106: 20
WS10-SCR-ATTRB	GSS5	8	SOURCES\COBOL\GSS5.CBL	101: 16
WS10-SCREEN-LINES	GSS5	25	SOURCES\COBOL\GSS5.CBL	100: 12
WS10-SCREEN-LINES-TABLE	GSS5	2	SOURCES\COBOL\GSS5.CBL	99: 8

- 3 The Impacted Data Items view lists the declaration of each affected data item and the number of impacts on the data item (its *impact factor*). Double-click the declaration for the item whose trace you want to display. The Impacted Instances view opens (Figure 5-4).

Figure 5-4 Impacted Instances View

Data Item	Offset:Size	Program	File	Ln:Col
COMM-NEXT-BROWSE-KEY	397:10	GSS5	SOURCES\COBOL\GSS5.CBL	793: 47
COMM-NEXT-BROWSE-KEY	397:10	GSS5	SOURCES\COBOL\GSS5.CBL	803: 47
COMM-NEXT-BROWSE-KEY	397:10	GSS5	SOURCES\COBOL\GSS5.CBL	813: 47
COMM-NEXT-BROWSE-KEY	397:10	GSS5	SOURCES\COBOL\GSS5.CBL	823: 47

- 4 The Impacted Instances view lists each impact on the affected data item. Double-click the impact whose trace you want to display. The Impact Sequence view opens (Figure 5-5 on page 5-5).

Tip: Although a given use of a data item may be impacted multiple times, only one trace is shown.

Figure 5-5 Impact Sequence View

Depth	Item	Statement	Relationship	Offset:Size	Object Name	File	Ln:Col
0	COMM-NEXT-BROW...	MOVE CO...	start	224:512	G555	SOURCES\C...	435: 17
➔ 1	FA01-KEY	MOVE CO...	move	4248:10	G555	SOURCES\C...	435: 43
← 2	COMM-NEXT-BROW...	MOVE CO...	move	397:10	G555	SOURCES\C...	435: 17
← 3	WS05-COMMON-AR...	MOVE DF...	used	397:10	G555	SOURCES\C...	339: 36
➔ 4	WS05-COMMON-AR...	G555.1.8...	used	397:10	G555	SOURCES\C...	861: 27
← 5	COMM-NEXT-BROW...	MOVE CO...	used	397:10	G555	SOURCES\C...	793: 47


- 5 The Impact Sequence view lists the data items at each depth in the impact trace, the statement in which each data item is impacted, and the relationship of the data item to the previous item.

Note: For a description of the relationships, see Table 5-1 on page 5-9.


- A ➔ symbol next to an item means that it is a consequence of the previous item. A ← symbol means that it is a cause of the previous item.
- A ↔ symbol means that the item is both a cause and a consequence.

Using the Impact Report Window

The Impact Report window lets you navigate to an impact trace based on the depth of the affected item or the object that contains the affected item (see steps [2A-2B](#) on page 5-3). Each navigation path consists of a hierarchy of views that let you specify the data impacts you want to trace:

- The *Impact Layers* view lists the depths of affected items. The *Impacted Objects* view lists the objects that contain affected items and the number of impacts on the object.
- The *Impacted Data Items* view (Figure 5-3 on page 5-4) lists the declaration of each affected data item and the number of impacts on the data item.
- The *Impacted Instances* view (Figure 5-4 on page 5-4) lists each impact on the affected data item. Click the  button on the tool bar to save the instances to a list you can view in the List Browser.

- The *Impact Sequence* view (Figure 5-5 on page 5-5) lists the data items at each depth in the impact trace, the statement in which each data item is impacted, and the relationship of the data item to the previous item. For more information, see [step 5 on page 5-5](#).


Double-click an item in a view to navigate to the next-level view in the hierarchy. Click the  button on the tool bar to navigate to the parent of the current view.

Sorting Entries Click a column heading in a view to sort the view entries by that column.


Sizing Columns Grab-and-drag the border of a column heading to increase or decrease the width of the column.

Setting Impact Report Options

Use the Impact Report Options window to hide causes or consequences, control the depth of the impact trace, filter out unwanted relationships or variables, and limit the number of startup items. As a rule, limiting the scope of the report improves performance.

Tip: After setting impact report options, click the  button on the tool bar to refresh the impact report.

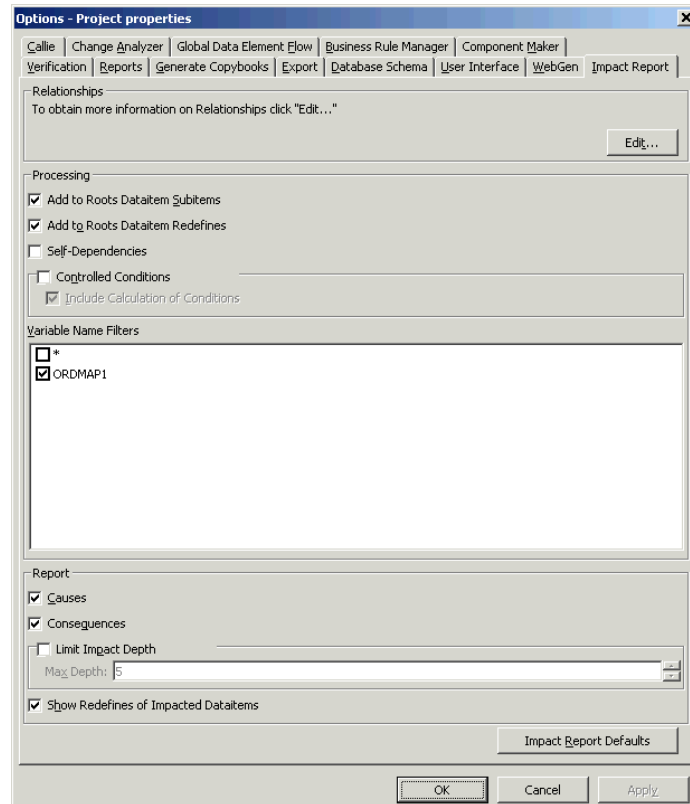
To set Impact Report options:

- 1 Click the  button on the tool bar. The Impact Report Options window opens (Figure 5-6 on page 5-7).
- 2 In the Relationships pane, click the **Edit** button. The Relationships window opens.
 - In the Intraprogram pane, place a check mark next to each intraprogram impact relationship you want the trace to include.
 - In the Interprogram pane, place a check mark next to each interprogram impact relationship you want the trace to include.

Note: For a description of intraprogram and interprogram relationships, see Table 5-1 on page 5-9

Click **OK** to dismiss the Relationships window and return to the main Impact Report Options window.

Figure 5-6 *Impact Report Options Window*



- 3 In the Processing pane, choose any combination of:
 - **Add to Roots Data Item Subitems** if you want the trace to include, as startup items, data items in nested declarations of the selected startup items.
 - **Add to Roots Data Item Redefines** if you want the trace to include, as startup items, data items that redefine the selected startup items.
 - **Self-Dependencies** if you want the trace to include recursive impacts on data items.

- **Controlled Conditions** if you want the trace to include controlled dependency relationships for data items, such as that between A and B in IF ... A THEN MOVE... TO B. In this example, B depends on A, since B is assigned with a value under the control of a condition that depends on A.

Select **Include Calculation of Conditions** if you want the report to show how the controlling data item is calculated.

- 4 In the Variable Name Filters pane, select the patterns that match the names of the variables you want to filter out of the impact trace. Recognized patterns are listed in the pane.

To add a pattern to the list, right-click in the Variable Name Filters pane and choose **Add** in the pop-up menu. HyperView displays an empty text field next to a selected check box. Enter the name of the new pattern in the field and click outside the field.

To edit a pattern, right-click it in the list and choose **Edit** in the pop-up menu. To delete a pattern, right-click it in the list and choose **Delete** in the pop-up menu.





- 5 In the Impact Report pane, set options specific to the Impact Report tool:
 - **Causes** if you want the report to include data items that cause a previous item in the trace.
 - **Consequences** if you want the report to include data items that are a consequence of a previous item in the trace.
 - **Limit Impact Depth** if you want the report to include data items only up to a specified depth. Specify the depth in the **Max Depth** combo box.
 - **Show Redefines of Impacted Data Items** if you want the report to include, as affected items, data items that redefine affected items.

Tip: You can restore the default option settings by clicking **Impact Report Defaults**, then choosing **Restore Impact Report Defaults** in the drop-down menu. Choose **Save To** in the drop-

down menu to save the option settings to a file. Choose **Load From** in the menu to restore the option settings from a file.

- 6A** Click **Apply** if you want to save your settings without dismissing the Options window.
- 6B** Click **OK** if you want to save your settings and dismiss the Options window.

Table 5-1 *Impact Report Relationships*

Relationship	Definition	Type	Icon	Description
calls	N/A	interprogram	N/A	Denotes that a parameter is passed in a call to another program.
cast	MOVE A TO B with data conversion	intraprogram		Denotes that a data item is moved to a data item of a different type.
common area transitions	N/A	interprogram	N/A	For Unisys Cobol, denotes that a common-storage data area item is passed in a call to another program. The Perform Unisys Common-Storage Area Analysis verification option must be set.
comp	STRING A ... INTO B	intraprogram		Denotes arbitrary computation. The result is produced by applying complex rules to the argument, such as STRING.
comp+	ADD A TO B	intraprogram		Denotes addition-like operations: ADD, SUBTRACT, and corresponding parts of COMPUTE.
comp*	MULTIPLY A BY B	intraprogram		Denotes multiplication-like operations: MULTIPLY, DIVIDE, and corresponding parts of COMPUTE.

5-10 Analyzing Impact Traces
Using the Impact Report Tool

Table 5-1 *Impact Report Relationships* (continued)










Relationship	Definition	Type	Icon	Description
comp@	MOVE ARRAY(IDX) TO A	intraprogram		Denotes operations with array elements.
cond	IF A = B ...	intraprogram		Denotes comparison of data items with a symmetric relationship.
cond*	IF A * X = B ...	intraprogram		Denotes comparison of a multiple of a data item with another data item.
const cond	IF A = 1 ...	intraprogram		Denotes comparison of a data item with a constant.
const.move	MOVE 1 TO B	intraprogram		Denotes that a constant is moved into a data item.
const.comp	ADD 1 TO B	intraprogram		Denotes arithmetic operations with constants.
const.init	03 A ... VALUE 1	intraprogram		Denotes a data item initialized by a constant.
DMS records	N/A	interprogram	N/A	For Unisys Cobol, denotes communication via Unisys DMS database records.
files	N/A	interprogram	N/A	Denotes communication via files. Traced only when corresponding JCL, ECL, FCT, or CSD files are verified.
move	MOVE A TO B	intraprogram		Denotes that a data item is moved to a data item of the same type.
screens	N/A	interprogram	N/A	Denotes that data is sent to a screen by one program and received in a screen by another.

Table 5-1 *Impact Report Relationships* (continued)





Relationship	Definition	Type	Icon	Description
start	N/A	intraprogram	N/A	Denotes a startup item. (Not shown in Options window.)
used	MOVE ... TO A ... MOVE A TO ...	intraprogram		Denotes that a value assigned in a statement is used as an argument in another statement. (Not shown in Options window.)

Note: The icons shown in Table 5-1 appear in the Impact pane only.

Creating Projects in the Impact Report Tool

You can create a project directly in the Impact Report tool from the results of your analysis. The project contains only source files that contain affected data items. You can also add source files to an existing project.

To create or add to a project:

- 1A** To create a project that contains all source files with affected data items, double-click the Impacted Objects folder, click the  button next to the  button on the tool bar, and choose **Create Project** in the drop-down menu. The New Project dialog opens. Enter the name of the new project in the text field and click **OK**. The Change Magnitude window opens (Figure 5-7).
- 1B** To add to an existing project all source files with affected data items, double-click the Impacted Objects folder, click the  button next to the  button on the tool bar, and choose **Add to Project Name project** in the drop-down menu. The Change Magnitude window opens (Figure 5-7 on page 5-12).
- 2** In the Change Magnitude window, select the **Automatically calculate change magnitude** check box if you want Impact Report to set change magnitudes for the listed objects based on the ranges specified in the fields below the check box. The Impact Report settings will override any existing change magnitudes for the objects.

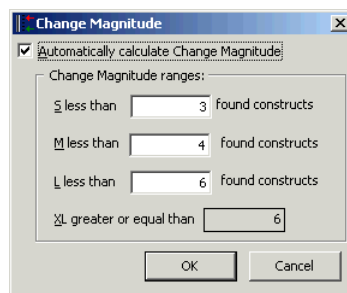
5-12 Analyzing Impact Traces
Using the Impact Report Tool

The fields below the check box contain default ranges for the available values: Small, Medium, Large, and Extra Large. Compare the number of affected items in each object listed in the Impacted Objects view with the default ranges, then modify the ranges as necessary.

If you want Cobol source files with 6 to 10 constructs to have a Large change magnitude, for example, set the range for Medium to less than 6 and the range for Large to less than 11. When you are satisfied with your choices, click **OK**.

For more information on change magnitudes, see [“Specifying the Change Magnitude for a Source File” on page 2-4](#).


Figure 5-7 *Change Magnitude Window*



Exporting Trace Reports from the Impact Report Tool

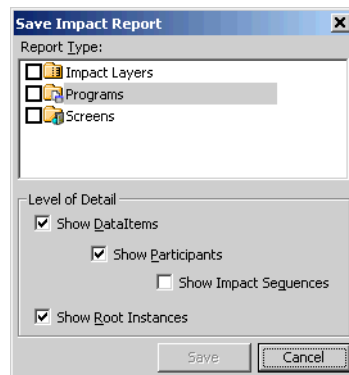
You can export an impact trace to an HTML file. You can also export a printable report for the selected impact layer to HTML, Excel, RTF, Word, or formatted text.

To export an HTML impact trace report:



- 1 Click the  button on the tool bar. The Save Impact Report window opens (Figure 5-8 on page 5-13).
- 2 In the Report Type pane, choose any combination of **Impact Layers** and the available check boxes for affected objects.
- 3 In the Level of Detail pane, choose any combination of:

- **Show Data Items** if you want the report to include affected data items. If you choose this option, choose **Show Participants** if you want the report to include each impact instance for the affected data items. If you choose **Show Participants**, choose **Show Impact Sequences** if you want the report to include the relationships between impact instances.
- **Show Root Instances** if you want the report to include startup items.

Figure 5-8 *Save Impact Report Window*



To export an impact layer report:

- 1 To export an impact layer report to HTML, Excel, RTF, Word, or formatted text, select an impact layer, then click the  button next to the  button on the tool bar and choose **Save Page As** in the drop-down menu. The printable report is displayed.
- 2 In the printable report window, click **Page Setup** to specify print job options. Click **Print** to send the job to the printer. Click **Save** to save the report to a variety of standard formats.

Using the Impact Pane

The Impact pane lets you analyze the data flows in a project. If you are extracting business rules, for example, you might want to view an impact trace to make sure that a segment encapsulates all of the business logic for a candidate rule. Or you might want to use a scoped trace to help locate candidates that exhibit a particular type of relationship.

The Impact pane offers both a hierarchical view and diagram of an impact trace. Both views show the flow of data to or from a *startup item*. Options let you filter out unwanted relationships or variables.

Important: The Impact pane shares options with the Impact Report tool. Use the Options described in steps [2-4](#) on pages 5-6 through 5-8 to filter out unwanted relationships or variables and limit the number of startup items.

Understanding the Hierarchical View

In the hierarchical view, each line in the trace contains a symbol indicating the type of relationship the data item has with the previous item, the data item name, and the name of the program it belongs to (unless the parent item belongs to the same program). Table 5-1 on page 5-9 defines the relationship symbols.

Options let you display the location of the data item in source, the name of the relationship, and its memory offset and size. Items that have not been visited by the trace are displayed in **bold** type. Repeated items are displayed in gray type. Repeated items at a different location are displayed in blue type. If a repeated item at a different location has not been visited, it is displayed in **bold blue** type. An item filtered out of the trace is displayed in red type.

Understanding the Diagram View

In the diagram view, each box in the trace contains the data item name and the name of the program it belongs to (unless the parent item belongs to the same program). Relationship lines contain the name of the relationship. Options let you display the memory offset and size of a data

item. Place your cursor over a variable for a moment to display the memory offset and size in a tool tip.

The startup item is displayed in a box with a red outline. An item that has not been visited by the trace is displayed in a blue box. An item filtered out of the trace is displayed in a box with a black outline. Usage is similar to that for the Diagrammer tool described in *Analyzing Projects*.

Generating an Impact Trace in the Impact Pane

Follow the steps below to generate an impact trace in the impact pane.

To generate an impact trace in the Impact pane:




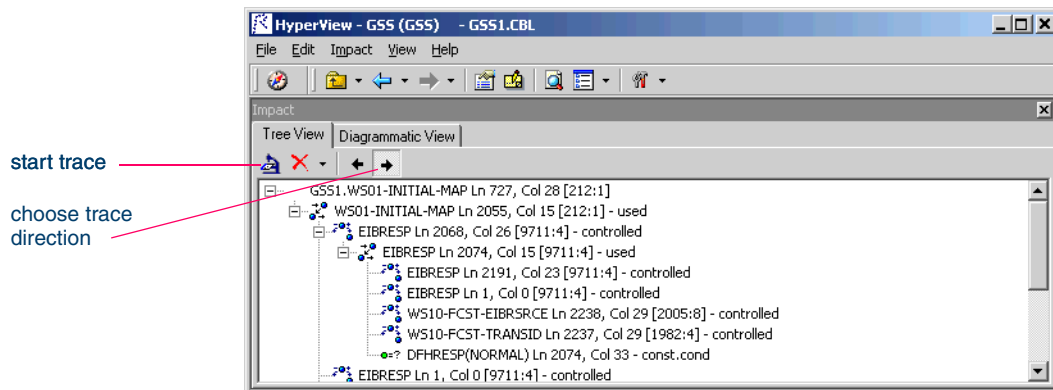
- 1 In the Source or Context pane, select the construct that contains the startup item or items. In the Impact pane, click the  button on the tool bar to view the flow of data into the startup items, click the  button to view the flow of data out of the startup items.
- 2 Click the  button on the tool bar to start the trace. HyperView displays the trace results in the Impact pane (Figure 5-9). Click the plus sign (+) next to a data item to expand its hierarchy. Click the minus sign (-) to collapse the hierarchy.

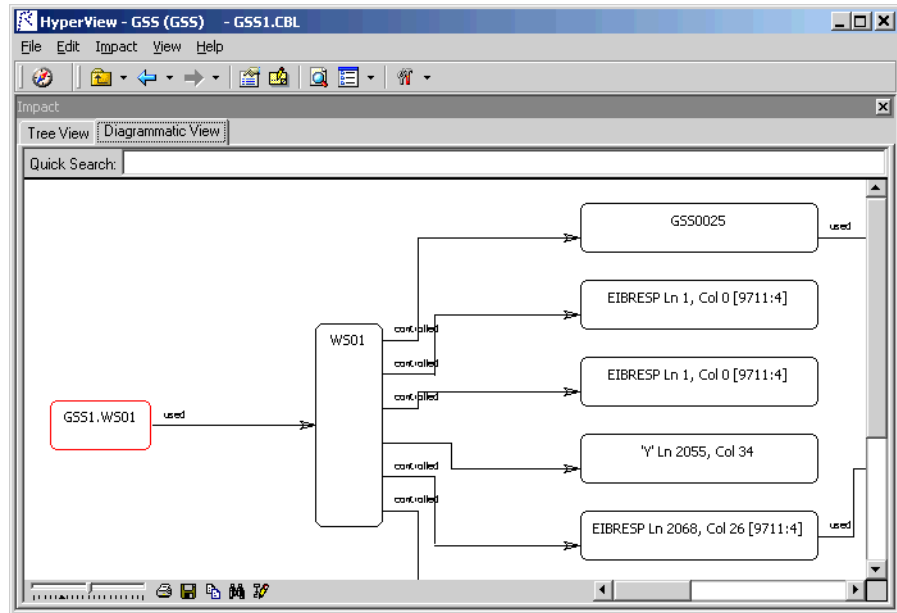
Figure 5-9 *Impact Pane with Trace Hierarchy*









- 3 Click the Diagrammatic View tab to display the impact trace in a diagram (Figure 5-10 on page 5-16).

5-16 Analyzing Impact Traces
Using the Impact Pane

Figure 5-10 *Impact Pane with Trace Diagram*



- 4 Select an item in the trace and click the  button to start a trace for that item. The new trace is displayed below the previous trace. To delete a trace, select its startup item and click the  button on the tool bar. To clear the pane, click the  button next to the  button and choose **Remove All** in the drop-down menu.
- 5 Click the  or  buttons on the tool bar to reverse the direction of the trace.

Setting Impact Pane Options

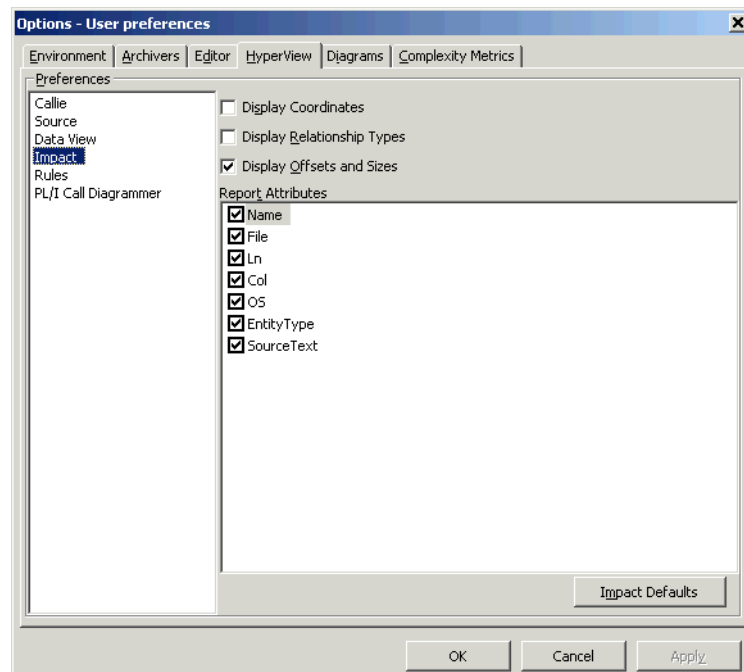
Use the Options window for the Impact pane to control the level of detail in the impact trace — whether to show or hide the locations of variables in source, their relationships, and their memory offsets and sizes. You can also use this window to specify the type of information to include in a trace report.

Important: Use the Options described in steps 2-4 on pages 5-6 through 5-8 to filter out unwanted relationships or variables and limit the number of startup items.

To set Impact Pane options:

- 1 In the **View** menu, choose **User Preferences**. The User Preferences window opens. Click the **HyperView** tab. In the Preferences pane, click **Impact**. The User Preferences window for the Impact pane opens (Figure 5-11).

Figure 5-11 *Impact Pane User Preferences Window*



- 2 Choose any combination of:
 - **Display Coordinates** if you want the hierarchical view to show line and column numbers for data items.
 - **Display Relationship Types** if you want the hierarchical view to show data item relationship names.



- **Display Offsets and Sizes** if you want both the hierarchical and diagram views to show the memory offsets and sizes for data items.
- 3 In the Report Attributes list box, place a check mark next to each type of information you want to be included in the trace report.
- Tip:** You can restore the default option settings by clicking **Impact Defaults**, then choosing **Restore Impact Defaults** in the drop-down menu. Choose **Save To** in the drop-down menu to save the option settings to a file. Choose **Load From** in the menu to restore the option settings from a file.
- 4A Click **Apply** if you want to save your settings without dismissing the Options window.
 - 4B Click **OK** if you want to save your settings and dismiss the Options window.

Exporting Trace Reports from the Impact Pane



Three types of report are available in the Impact pane:

- A *pane report* shows the entire tree for a trace in XML format.
- A *subtree report* shows the trace for a subtree in XML format or in a database.
- *Affected code reports* show code that would be impacted by changing a data item's definition or usage



To export a pane report:

- 1 To export a pane report, click the  button next to the  button on the tool bar and choose **Pane Report** in the drop-down menu.
- 2 A Save As dialog opens, where you can specify the name, location, and file type of the report.

To export a subtree report:

- 1 To export a subtree report, select the subtree in the Impact pane, then click the  button next to the  button on the tool bar and choose **SubTree Report** in the drop-down menu.
- 2 A Save As dialog opens, where you can specify the name, location, and file type of the report.

To export an affected code report:

- 1 To export an affected code report, select the item whose impacts you want to measure in the Impact pane, then click the  button next to the  button on the tool bar and choose **Affected Code Report** in the drop-down menu.
- 2 A Save As dialog opens, where you can specify the name, location, and file type of the report.

What's Next?

That's all you need to know to analyze impacts in HyperView. The next chapter looks at how you analyze program control flows.

5-20 Analyzing Impact Traces
What's Next?

Analyzing Program Control Flows



Control flows describe the processing paths in a program. Call flows, decision flows, and statement flows each offer a different way of understanding the procedures in a program. HyperView offers four related resources for analyzing program control flows:

- The Callie pane displays a diagram that shows the flow of control between paragraphs or procedures in a program.
- The Execution Path pane displays a hierarchical view and diagram that show the conditions that determine the flow of control in a program.
- The Flowchart pane displays a diagram that shows the flow of control between statements in a paragraph or procedure.
- The Animator lets you step through the code displayed in a HyperView pane.

Note: Flowchart, Execution Path, and Animator are not available for Natural programs. Callie and Animator are not available for PL/I programs. In this chapter, the term “paragraph” also refers to Natural subroutines and PL/I procedures.

Using the Callie Pane

The Callie pane displays a diagram that shows the flow of control between paragraphs in a program. Calls are traced to and from a *seed* paragraph — the paragraph selected in the Source or Context pane when the diagram is drawn or refreshed. Usage is similar to that for the Diagrammer tool described in *Analyzing Projects*.

The label of the box representing a paragraph contains the name of the paragraph. The selected paragraph is displayed with a background shadow. Call relationship lines are color-coded by default as follows:

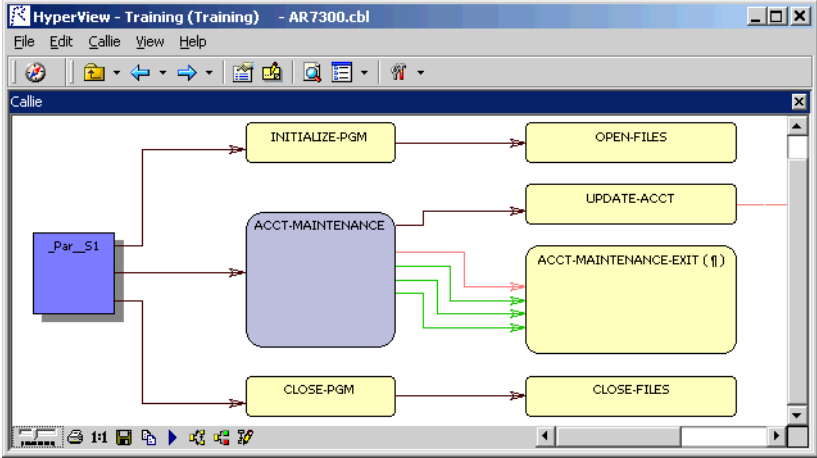
- Black = PERFORM
- Green = GO TO
- Red = Fall Thru

Options control the depth of the flow trace and the colors of display objects and relationship lines. For more information, see [“Setting Callie Pane Options” on page 6-4](#).

To generate a call trace diagram:

- 1A** Choose **SubGraph mode** in the **Callie** menu to select the subgraph view.
- 1B** Choose **SubTree mode** in the **Callie** menu to select the subtree view. For more information, see [“Choosing the Diagram View” on page 6-3](#).
- 2** In the Source or Context pane, select the seed paragraph. HyperView displays the call trace for the seed paragraph (Figure 6-1 on page 6-3).
- 3** Select a paragraph to navigate to it in the Source and Context panes. Select a relationship line to navigate to the corresponding statement in the Source and Context panes.
- 4** Refresh the display as described in [“Redrawing the Diagram” on page 6-4](#).

Figure 6-1 Callie Pane



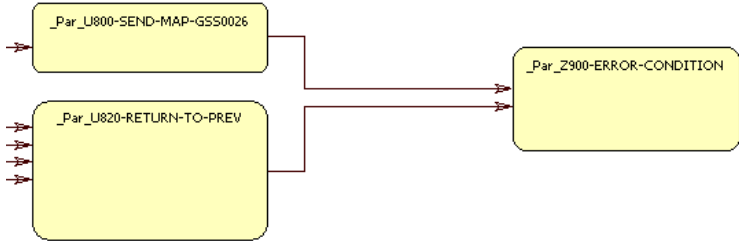
Choosing the Diagram View

The Callie pane offers two views of program control flows, a *subgraph view* and a *subtree view*. Figure 6-2 and Figure 6-3 on page 6-4 show the same paragraphs in both views.

Subgraph View

The subgraph view offers a cyclic representation of the information in the diagram. Paragraphs are drawn once. Relationship lines cross. Subgraph views are often easier to understand than subtree views. Choose **SubGraph mode** in the **Callie** menu to select the subgraph view.

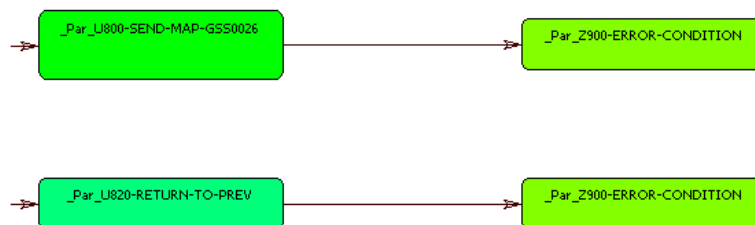
Figure 6-2 Subgraph View



Subtree View

The subtree view offers a linear representation of the information in the diagram. Paragraphs are drawn as many times as necessary. Relationship lines do not cross. Use this view if too many intersecting lines make a subgraph view hard to read. Choose **SubTree mode** in the **Callie** menu to select the subtree view.

Figure 6-3 Subtree View



Redrawing the Diagram

The diagram is automatically redrawn when you select a new seed paragraph in the Source or Context pane, unless that paragraph already appears in the current diagram. To redraw the diagram in that case, select the new seed paragraph in the diagram and choose **Reconstruct diagram** in the **Callie** menu.

Setting Callie Pane Options

Use the Callie User Preferences and Project Options windows to control the depth of the flow trace, the constructs displayed, and the colors of display objects. Callie User Preferences control the appearance and behavior of Callie across workspaces. Callie Project Options control the appearance and behavior of Callie in the selected project.

Setting Callie User Preferences (Subgraph View Only)

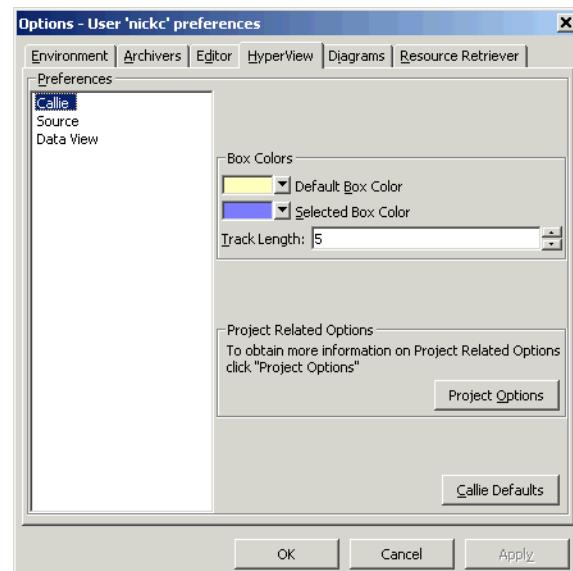
Callie makes it easy to track execution paths in subgraph views by displaying each node you visit in a different color. The colors range from the color specified for the selected paragraph to the color specified for unselected, or “default,” paragraphs — from violet to yellow in the pre-

set colors, for example. The *track length* determines the number of visited nodes that Callie displays in different colors.

To set Callie User Preferences:

- 1 In the **View** menu, choose **User Preferences**. The User Preferences window opens. Click the **HyperView** tab. In the Preferences pane, click **Callie**. The Callie User Preferences window opens (Figure 6-4).

Figure 6-4 *Callie User Preferences Window*



- 2 The current default background color of the box representing a paragraph is displayed in the **Default Box Color** drop-down. The current background color of the box representing the selected paragraph is displayed in the **Selected Box Color** drop-down.

Click the adjacent buttons to edit the color of the paragraphs. A standard Windows color control is displayed. Use the **Palette** tab to select the color from the Windows palette. Use the **System** tab to match the color with the color of standard Windows elements.

- 3 In the **Track Length** combo box, enter the number of visited nodes that Callie displays in different colors.

Tip: You can restore the default option settings by clicking **Callie Defaults**, then choosing **Restore Defaults** in the drop-down menu. Choose **Save To** in the drop-down menu to save the option settings to a file. Choose **Load From** in the menu to restore the option settings from a file.

- 4A Click **Apply** if you want to save your settings without dismissing the User Preferences window.
- 4B Click **OK** if you want to save your settings and dismiss the User Preferences window.

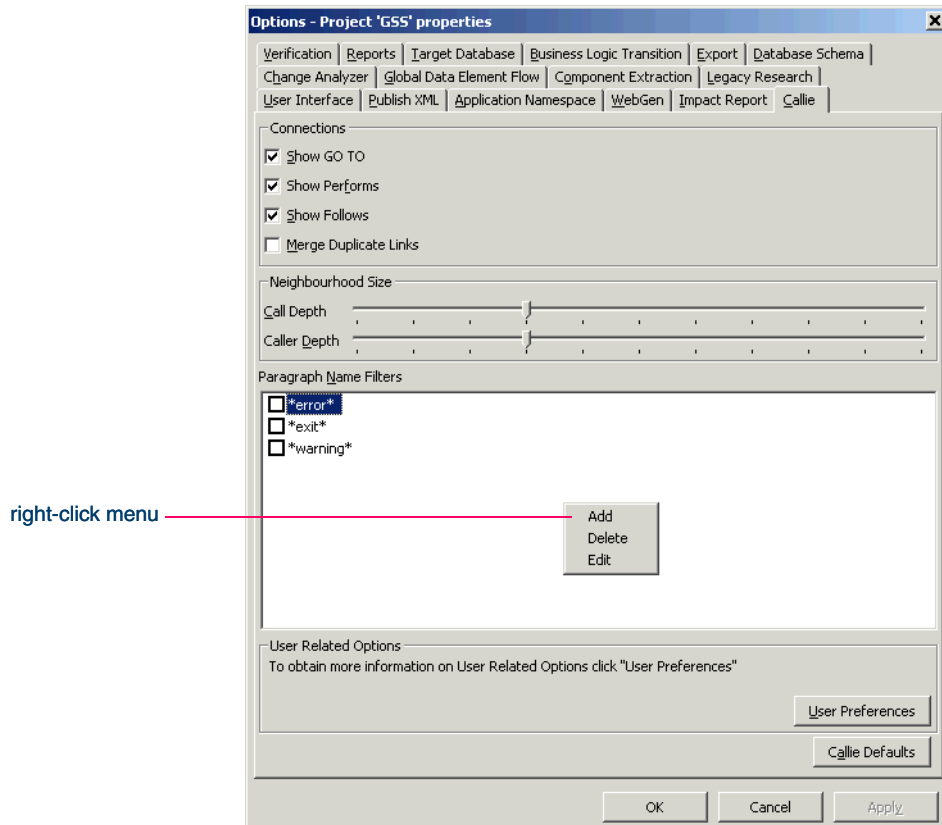
Setting Callie Project Options

Callie Project Options control the *depth* of the flow trace — how many calling and called nodes in a sequence the diagram displays to and from the seed paragraph. You can also set Project Options that limit the relationship types the diagram displays (PERFORMs, GO TOs, or Fall Thrus) and filter out paragraphs by name.

To set Callie Project Options:

- 1 In the **View** menu, choose **Project Options**. The Project Options window opens. Click the **Callie** tab. The Callie Project Options window opens (Figure 6-5 on page 6-7).
- 2 In the Connections pane, deselect relationship types you do not want to view in the diagram. Deselect:
 - **Show GO TO** to hide GO TO relationships.
 - **Show Performs** to hide PERFORMS relationships.
 - **Show Follows** to hide Fall Thru relationships.
- 3 In the Connections pane, select **Merge Duplicate Links** to display a single relationship line rather than multiple lines when constructs are connected by identical call relationships.
- 4 In the Neighborhood Size pane, use the **Call Depth** and **Caller Depth** sliders to specify the depth of the flow trace — how many calling and called nodes in a sequence, respectively, the diagram displays to and from the seed paragraph.

Figure 6-5 Callie Project Options Window



- 5 In the Paragraph Name Filters pane, select the pattern that matches the names of paragraphs you want to exclude from the diagram. The recognized patterns are listed in the Paragraph Name Filters pane.

Right-click in the Paragraph Name Filters pane and choose **Add** in the pop-up menu to add a pattern to the list. The system displays an empty text field next to a selected check box. Enter the name of the new pattern in the field and click outside the field.

You can edit a pattern by right-clicking it and choosing **Edit** in the pop-up menu. You can delete a pattern by right-clicking it and choosing **Delete** in the pop-up menu.

Tip: You can restore the default options settings by clicking **Callie Defaults**, then choosing **Restore Callie Defaults** in the drop-down menu. Choose **Save To** in the drop-down menu to save the option settings to a file. Choose **Load From** in the menu to restore the option settings from a file.

6A Click **Apply** if you want to save your settings without dismissing the Options window.

6B Click **OK** if you want to save your settings and dismiss the Options window.

Using the Execution Path Pane


The Execution Path pane displays a hierarchical view and diagram of the conditions that determine the control flow in a program. Each view traces the control flow from the first involved condition to the paragraph or statement selected in the Source or Context panes — the *seed* construct. Use an execution path to ensure that a code segment encapsulates all of the business logic for a candidate business rule.

Note: The Execution Path tool may give incorrect results if the program contains a paragraph that is used in multiple PERFORM statements or in some combination of PERFORM and GOTO statements.

Generating an Execution Path

Follow the steps below to generate an execution path.

To generate an execution path:

1 In the Source or Context pane, select the seed construct. In the Execution Path pane, click the  button on the tool bar. HyperView displays the execution path for the seed construct (Figure 6-6 on page 6-9).


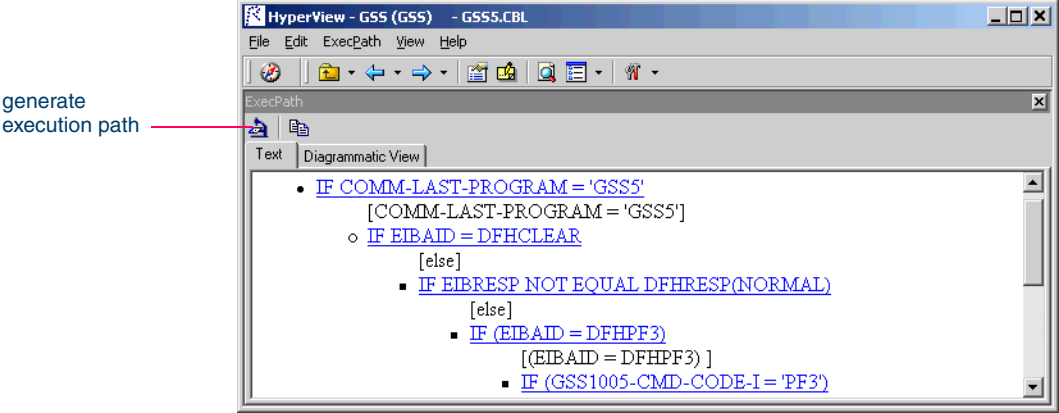
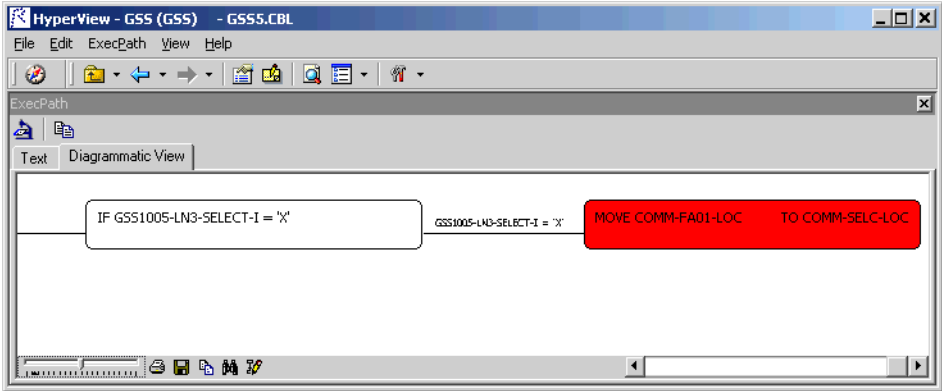
Tip: Click the  button on the tool bar to copy the execution path to the clipboard with all formatting preserved.


Figure 6-6 Execution Path Pane with Hierarchy



- 2 Click the Diagrammatic View tab to display the execution path in a diagram (Figure 6-7). The seed item is displayed in red. Usage is similar to that for the Diagrammer tool described in *Analyzing Projects*.

Figure 6-7 Execution Path Pane with Diagram



- 3 Select a statement in the execution path and click the  button to generate the execution path for that statement. The new execution path replaces the previous path.

Using the Flowchart Pane

The Flowchart pane displays a statement flow diagram for the paragraph selected in the Source or Context panes. Use the right-click menu to navigate to related constructs in the parse tree. Otherwise, usage is similar to that for the Diagrammer tool described in *Analyzing Projects*.

The label of the box representing a statement contains the abbreviated text of the statement. Place your cursor over a label for a moment to display a tool tip with the full text of the statement. Captions along relationship lines show the conditions or blocks that determine the flow.

For Cobol programs, ordinary statements are displayed in boxes with a green background. Paragraph PERFORM statements are displayed in boxes with a pink border and white background. The selected statement is displayed with a background shadow.

Two artificial constructs — the CONTINUE used to represent an empty IF, THEN, or ELSE branch and the PARAGRAPH that represents the object of a GO TO or PERFORM statement — are displayed in boxes with a black border and white background and cannot be selected.

Double-click a Paragraph PERFORM statement to redraw the diagram with the paragraph flow for the called paragraph. Double-click the paragraph for a GO TO statement to redraw the diagram with the paragraph flow for that paragraph.

For PL/I programs, the right-click menu prompts you to switch to the targets of CALL, RETURN, and GO TO statements. If no targets are available, the statement is displayed with a black border. Set PL/I Flowchart options as described in [“Setting PL/I Flowchart Pane Options” on page 6-11](#).

Generating a Flowchart

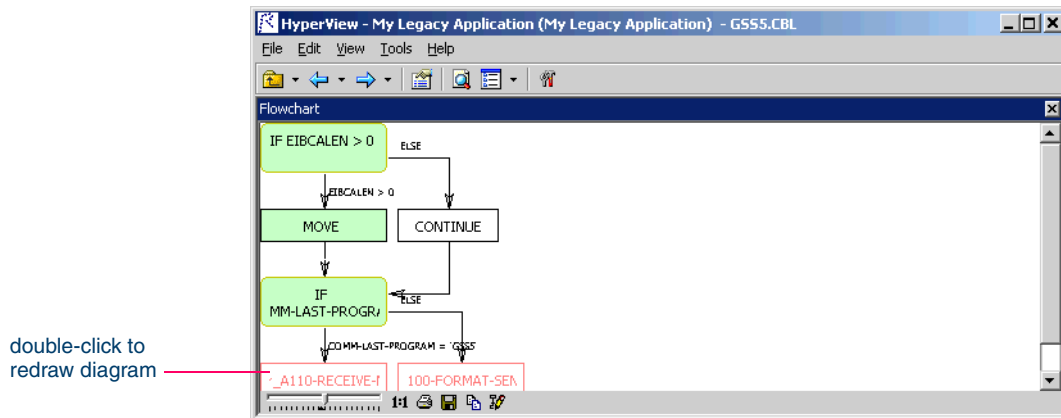
Follow the steps below to generate a flowchart.

To generate a flowchart:

- 1 In the Source or Context pane, select the paragraph whose flow of control you want to view. HyperView displays the flowchart for the selected paragraph (Figure 6-6 on page 6-9).

- 2 To redraw the diagram, select another paragraph in the Source or Context panes.

Figure 6-8 *Flowchart Pane*



Setting PL/I Flowchart Pane Options

Use the PL/I Flowchart User Preferences window to set the maximum number of characters to display before the tool truncates a label in the flowchart.

To set PL/I Flowchart User Preferences:

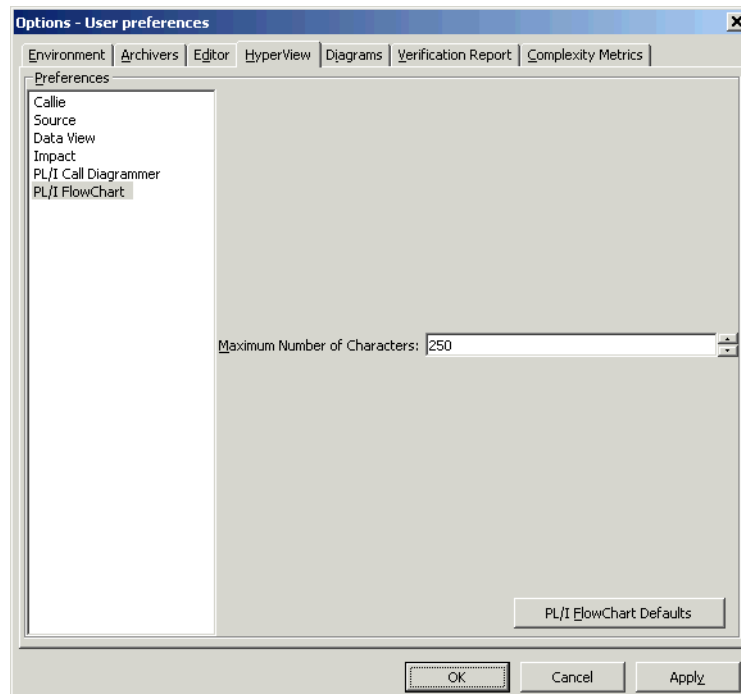
- 1 In the **View** menu, choose **User Preferences**. The User Preferences window opens. Click the **HyperView** tab. In the Preferences pane, click **PL/I Flowchart**. The PL/I Flowchart User Preferences window opens (Figure 6-9 on page 6-12).
- 2 In the **Maximum Number of Characters** field, enter the maximum number of characters to display before the tool truncates a label in the flowchart.

Tip: You can restore the default option settings by clicking **PL/I Flowchart Defaults**, then choosing **Restore Defaults** in the drop-down menu. Choose **Save To** in the drop-down menu to save the option settings to a file. Choose **Load From** in the menu to restore the option settings from a file.

6-12 Analyzing Program Control Flows
Using the Animator

- 3A** Click **Apply** if you want to save your settings without dismissing the User Preferences window.
- 3B** Click **OK** if you want to save your settings and dismiss the User Preferences window.

Figure 6-9 *PL/I Flowchart User Preferences Window*



Using the Animator

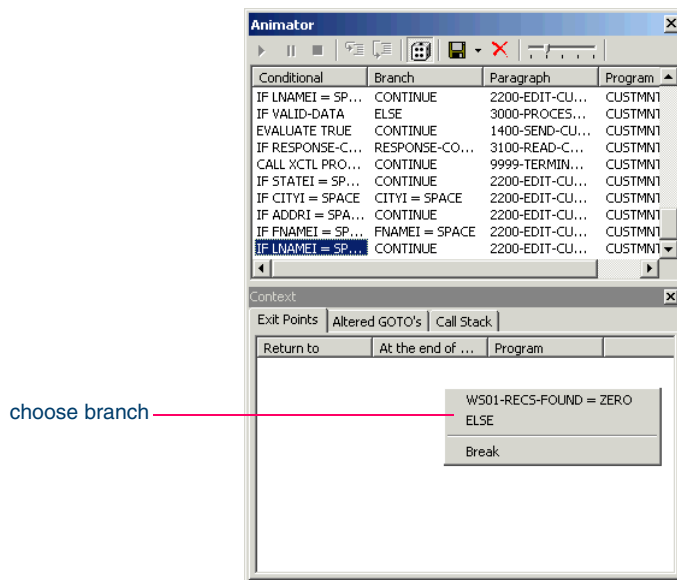
The Animator lets you step through the code displayed in a HyperView pane. You can choose program branches yourself, or have the animator choose them randomly.

Tip: Animator is especially useful in tracking execution paths in Callie subgraph views.

To step through code with the Code Animator:

- 1 In the Source or Context pane, select the construct you want to start the animation from, then choose **Animator** in the **View** menu. The Animator window opens (Figure 6-10).

Figure 6-10 *Animator Window with Context Pane*













- 2 Click the button on the tool bar if you want Animator to choose logical branches randomly.

Tip: Use the slider at the right of the tool bar to set the speed of the animation.

- 3 Click the button on the tool bar to start the animation. If you are choosing program branches yourself, Animator stops at each condition it encounters and displays a pop-up menu that prompts you to choose a logical branch. Click the branch you want to go down, or click **Break** to pause the animation.

HyperView populates the top part of the Animator window with each condition it encounters, the logical branch that you or the Animator chose, and the paragraph that contains the condition.

In the Context pane at the bottom of the window, click the appropriate tab to view exit points, altered GoTos, or the call stack for the paragraph currently being executed. You can hide the Context pane by deselecting the **Show Context** choice in the **Animator** menu.

- Tip:** Click the  button on the tool bar to step through the code manually. Each time you click the button the Animator steps into the next piece of program code. Click the  button if you want to step through the code manually but step over Paragraph PERFORM statements.
- 4 Click the  button on the tool bar to pause the animation. Click the  button to restart the animation. Click the  button to stop the animation.
 - 5A To save the animation results to a HyperView list, click the  button next to the  button on the tool bar and choose **Save Trace As List** in the drop-down menu. You can view the list in the HyperView List Browser, as described in [“Using the List Browser Window” on page 1-20](#).
 - 5B To save the animation results to XML, click the  button next to the  button on the tool bar and choose **Save Trace As XML** in the drop-down menu. A Save As dialog opens, where you can specify the name and location of the results report.
 - 6 Click the  to delete the results from the Animator window.

What's Next?

Now that you know how to analyze procedure flows in HyperView, let's look at how you use the Rules pane to create business rules. That's the subject of the final chapter.

Setting Up a Glossary of Data Items



The Glossary pane creates a conveniently organized dictionary, or *glossary*, of names your application uses to identify data elements. It's especially useful for assigning *business names* to identifiers — natural language names that make it easy to document and retrieve business rules. The field PLC_NUM, for example, might have the business name Policy Number.



The Glossary pane lets you assign business names to identifiers manually or in batch mode. You can also import business names into a glossary from a file. For information on how to autoassign business names to business rules, see [Chapter 8, “Extracting Business Rules.”](#)

Understanding the Glossary Pane

The lefthand pane of the Glossary tool (Figure 7-1 on page 7-2) displays a list of every *token* in the current scope. A token is an element in a program identifier delimited by a hyphen (-) or underscore (_). In the identifier WS01-CUST-FIELD, for example, there are three tokens: WS01,

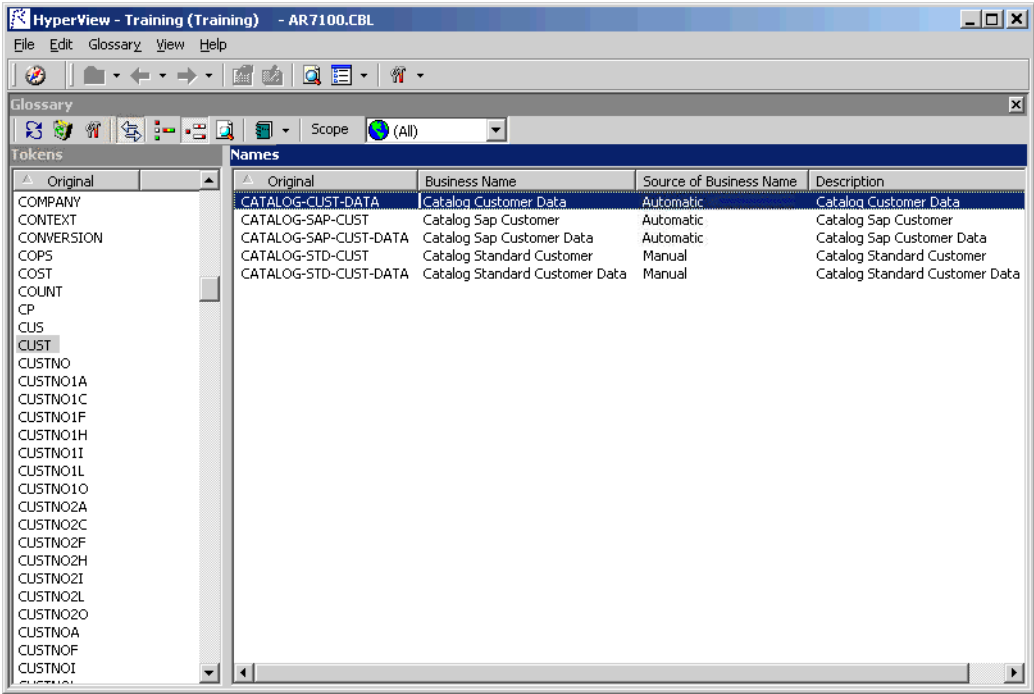
7-2 Setting Up a Glossary of Data Items
Understanding the Glossary Pane

CUST, and FIELD. The righthand pane displays a list of every identifier in the current scope.

- When the Link Names mode is selected — when you click the  icon on the tool bar or choose **Link Names** in the **Glossary** menu — the righthand pane lists every identifier that contains the token selected in the lefthand pane.
- When the Link Tokens mode is selected — when you click the  icon on the tool bar or choose **Link Tokens** in the **Glossary** menu — the lefthand pane lists every token contained in the identifier selected in the righthand pane.

You can select the modes in any combination — both on, both off, or one on, one off.

Figure 7-1 *Glossary Window*



Generating a Glossary

You can generate a glossary for a project or a single program. Once you have generated the glossary, you can assign business names and generate reports.

To generate a glossary:

- 1** In the HyperView **View** menu, choose **Glossary**. The Glossary window opens (Figure 7-1 on page 7-2).
- 2** In the **Scope** drop-down, choose the scope of the glossary.

Tip: Deselect **Synchronize with other views** in the **Glossary** menu to turn off synchronization with other HyperView panes. That typically improves Glossary tool performance.

- 3** In the **Glossary** menu, choose **Refresh Glossary**.
 - If neither the Link Tokens or Link Names mode is selected when you perform the refresh, the lefthand pane displays a list of every token in the selected scope, while the righthand pane displays a list of every identifier in the scope. Select the mode you want to use, as described in [“Understanding the Glossary Pane” on page 7-1](#).
 - If only the Link Names mode is selected when you perform the refresh (or if both modes are selected), the lefthand pane displays a list of every token in the selected scope, while the righthand pane displays every identifier that contains the first token in the list. Select another token to display the identifiers that contain it.
 - If only the Link Tokens mode is selected when you perform the refresh, the righthand pane displays a list of every identifier in the selected scope, while the lefthand pane displays every token contained in the first identifier in the list. Select another identifier to display the tokens it contains.

Tip: Click a column heading in the Glossary pane to sort the entries by that column. Grab-and-drag the border of a column heading to increase or decrease the width of the column.

Assigning Business Names to Identifiers

You can assign business names to identifiers manually or in batch mode. You can also import business names into a glossary from a file.

Assigning Business Names Manually

Use the procedure described below to assign business names manually, or to edit existing business names.

To assign business names manually:

- 1 In the Names pane, right-click the identifier you want to create a business name for in the Original column and choose **Business Name** in the pop-up menu.
- 2 The Glossary tool displays an edit box for the identifier in the Business Name column. Enter the business name in the edit box and click outside the box.
- 3 To enter a description of the business name, right-click the identifier and choose **Edit Description** in the pop-up menu. A dialog box opens where you can enter the description. Click **OK** to dismiss the dialog and return to the Glossary pane.

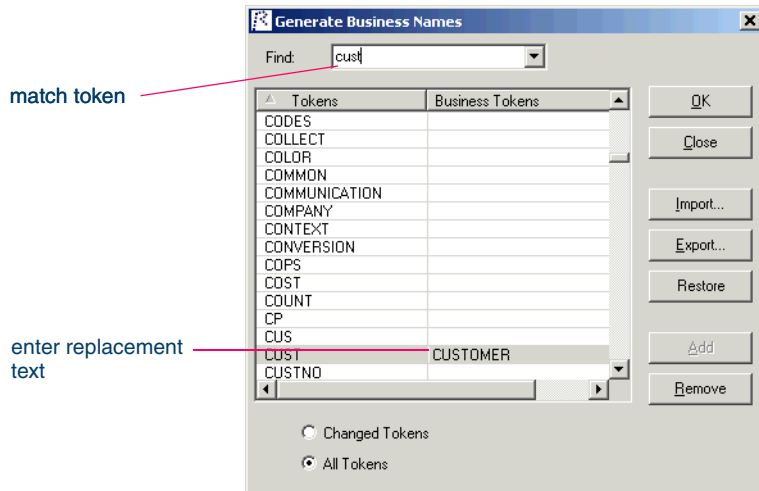
Assigning Business Names in Batch Mode

Ordinarily, you will want to assign business names in batch mode rather than manually. In the batch interface, you enter natural language replacement text for each token you want to change in an identifier — for the token NUM, for example, you would enter the replacement text Number.

To assign business names in batch mode:

- 1 In the **Glossary** menu, choose **Generate Business Names**. The Generate Business Names window opens (Figure 7-2 on page 7-5).
- 2 In the **Find** combo box, enter matching text for the token you want to replace. The Glossary tool locates the text match in the Tokens column as you type. You can also select an item to match in the drop-down for the combo box.

Figure 7-2 Generate Business Names Window



- 3 Click in the Business Tokens column for the token you want to replace. The Glossary tool displays an edit box for the token in the Business Tokens column. Enter the replacement text for the token in the edit box and click outside the box.

Tip: Select a token and click **Remove** to delete the token and its replacement text from the window. Click **Restore** to return the display to its state when you opened the window.

- 4 Repeat [step 3](#) for each token you want to replace.

Tip: Select the **Changed Tokens** radio button to display only tokens for which you have defined replacement text. Select the **All Tokens** radio button to display all the tokens in the glossary.

When **Changed Tokens** is selected, you can add a token to the display by entering its name in the **Find** combo box and clicking **Add**.

- 5 When you are satisfied with the replacement text for each token you want to change, click **OK**. The Glossary tool automatically gener-

ates business names for each identifier that contains the tokens, replacing them with the text you specified.

Note: You cannot edit the Description column in the Generate Business Names window. That column is used only to display imported descriptions.

Importing and Exporting Business Names

If you have propagated business names programmatically or by export from the Glossary pane, you can import them into your glossary as described below. You can replace both token and full identifier names. If you provide descriptions, the Glossary tool also imports them. The tool concatenates descriptions of replaced tokens to form full business name descriptions in the Names pane of the main window.

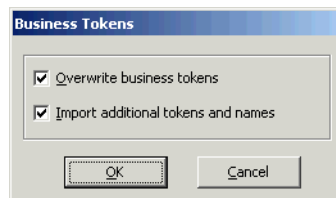
Source names, replacement text, and descriptions must appear in XML file format as follows:

```
<glossary>
  <item source="CUST" replacement="CUSTOMER"
    description="Replaces token"/>
  <item source="CUST_INFO" replacement="CUSTOMER
    INFORMATION" description="Replaces identifier"/>
  ...
</glossary>
```

To import business names into a glossary:

- 1 In the **Glossary** menu, choose **Generate Business Names**. The Generate Business Names window opens (Figure 7-2 on page 7-5).
- 2 Click **Import**. The Business Tokens dialog opens (Figure 7-3).

Figure 7-3 *Business Tokens Dialog*



- 3 In the Business Tokens Dialog, select:
 - **Overwrite business tokens** if you want the replacement text and description for imported tokens to overwrite the replacement text and description for existing tokens in the glossary. Import file tokens that do not exist in the glossary are not added.
 - **Import additional tokens** if you want import file tokens that do not exist in the glossary to be added.
- 4 When you are satisfied with your choices, click **OK**. An open dialog appears where you can select the file you want to import.

To export business names from a glossary:

- 1 In the **Glossary** menu, choose **Generate Business Names**. The Generate Business Names window opens (Figure 7-2 on page 7-5).
- 2 Click **Export**. A save as dialog appears where you can specify the name and location of the file you want to export.

Editing Business Names and Business Name Descriptions

Use the procedure described below to edit business names and business name descriptions.

To edit business names and business name descriptions:

- 1 In the Names pane, right-click the business name you want to edit in the Business Name column and choose **Business Name** in the pop-up menu.
- 2 The Glossary tool displays an edit box for the identifier in the Business Name column. Modify the business name in the edit box and click outside the box.
- 3 To edit a business name description, right-click the business name and choose **Edit Description** in the pop-up menu. A dialog box opens where you can enter the description. Click **OK** to dismiss the dialog and return to the Glossary pane.

Deleting Business Names and Business Name Descriptions

To delete a business name, right-click it in the Names pane and choose **Delete Business Name** in the pop-up menu. To delete all business names, only autogenerated business names, or only manually entered business names, right-click any business name and select the appropriate choice in the pop-up menu. In all cases, you are prompted to confirm the deletion. Click **Yes**. To delete business name descriptions as well, select **Delete Description(s)** in the confirmation dialog.

To delete all business name descriptions, right-click any business name and select **Delete All Business Descriptions** in the pop-up menu.

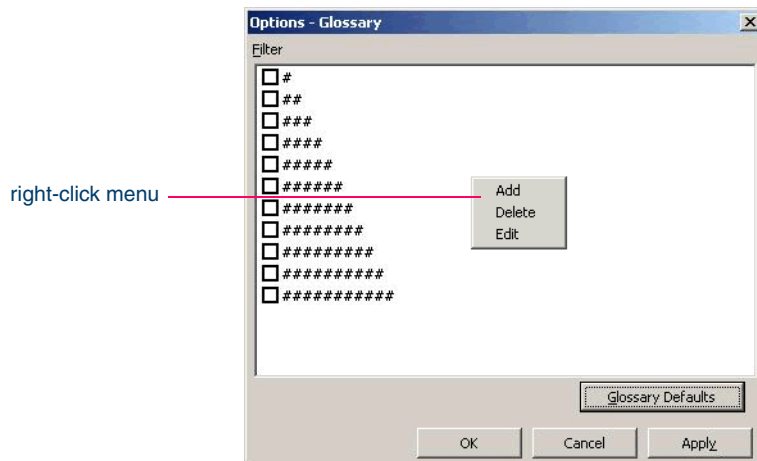
Setting Glossary Pane Options

Use Glossary tool options to define a filter that excludes unwanted tokens from the glossary display (but not the glossary database). By default, the tool defines filters for numeric tokens.

To set Glossary pane options:

- 1 In the **Glossary** menu, choose **Options**. The Glossary Options window opens (Figure 7-4).

Figure 7-4 *Glossary Options Window*



- 2** In the Filter pane, select the pattern that matches the names of tokens you want to exclude from the glossary display. The recognized patterns are listed in the Filter pane.

Right-click in the Filter pane and choose **Add** in the pop-up menu to add a pattern to the list. The system displays an empty text field next to a selected check box. Enter the name of the new pattern in the field and click outside the field. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).

You can edit a pattern by right-clicking it and choosing **Edit** in the pop-up menu. You can delete a pattern by right-clicking it and choosing **Delete** in the pop-up menu.

- 3A** Click **Apply** if you want to save your settings without dismissing the Options window.
- 3B** Click **OK** if you want to save your settings and dismiss the Options window.
- 4** To apply the filter, regenerate the glossary as described in [“Generating a Glossary” on page 7-3](#). To restore a filtered item, deselect the matching pattern in the Glossary Options window and regenerate the glossary.

Refreshing a Glossary

If you add a source file to a project and want its contents to be reflected in the glossary, you simply regenerate the glossary for the project. By contrast, if you delete a source file from a project, its tokens and identifiers and their related business names are *not* deleted from the glossary. This is to prevent you from deleting business names inadvertently.

If you decide that you do not want items from deleted source files to appear in the glossary, you need to *clean up* the glossary, as described below.

To clean up a glossary:

- 1 In the **Glossary** menu, choose **Clean Up**. A dialog box shows the number of items removed from the glossary as a result of the clean up. Click **OK** to dismiss the dialog and return to the Glossary pane.

Generating Glossary Reports

The Glossary tool offers two reports:

- The Tokens report lists the tokens in the selected scope, their replacement text, any descriptions for imported tokens, and the number of declarations in which the tokens appear. Choose **Tokens** in the **Glossary** menu to generate the Tokens report.
- The Names report lists the identifiers in the selected scope, any business names you have defined for the identifiers, the method of defining the business names, business name descriptions, and the number of declarations in which the identifiers appear. Choose **Names** in the **Glossary** menu to generate the Names report.

You can export Glossary reports to a variety of standard formats. Choose **Save** on the tool bar for the report to export the report to HTML, Excel, RTF, Word, or formatted text.

Note: Excel is usually the best option if you need to manipulate the report or perform computations. For more information on exporting reports, see *Getting Started* in the ATW document set.

What's Next?

That's all you need to know to set up a project glossary and assign business names. The final chapter shows you how to use the Rules pane to extract business rules.

Extracting Business Rules



Much of the code in a legacy application — by some estimates as much as 80% — is devoted to tasks with only a marginal relationship to the business logic of the application. *Business rule extraction* lets you separate the application's business logic from these other tasks — data transfer, in particular.

That makes the application easier to understand, document, maintain, and modernize. It also makes it easier to determine the overlap between legacy applications on the one hand, and any gaps in their functionality on the other.

The HyperView Rules pane offers a powerful set of tools for autodetecting, and autodocumenting, business rules. You can batch edit rule attributes; create custom attributes suited to your particular needs; match input/output data elements with the business names you've assigned to them in your project glossary; and much more.

Understanding Business Rules

A *business rule* is a named container that identifies and documents a *code segment* according to the task it performs in a business process. A business rule named Calculate Date Difference, for example, might consist of this segment:

```
COMPUTE WS-DATE-VARIANCE =  
WS-C-CARD-DATE-CCYYMM - WS-TODAYS-DATE-CCYYMM.
```

A business rule can contain one code segment only. The same segment can be assigned to different rules.

How Business Rules Are Organized

You organize business rules in *rule sets*. A rule set typically consists of rules executed together, such that if one rule is executed, all are executed — the statements in a paragraph, for example. You can think of a rule set as a business process, like Validate Input or Update File.

You organize rule sets, in turn, in *business functions*. A business function typically defines the order in which processes are triggered. Think of a business function as a use case, like Insert Customer or Update Order. Figure 8-1 on page 8-3 shows the HyperView Rules pane with a typical hierarchy of business functions, rule sets, and business rules.

You can list business rules in any order. For reporting purposes, order of execution is recommended. Autodetected rules are always initially listed in order of execution.

Business function and rule set names must be unique in the workspace. A rule set can exist in multiple business functions; a rule can exist in multiple rule sets.

How to Assign Segments to Business Rules

You can create business rules from code segments identified manually in source or autodetected. Use a Clipper list to run the autodetect method on multiple constructs, or to generate business rules in a specified rule set with templated attributes.

Understanding Business Rule Attributes

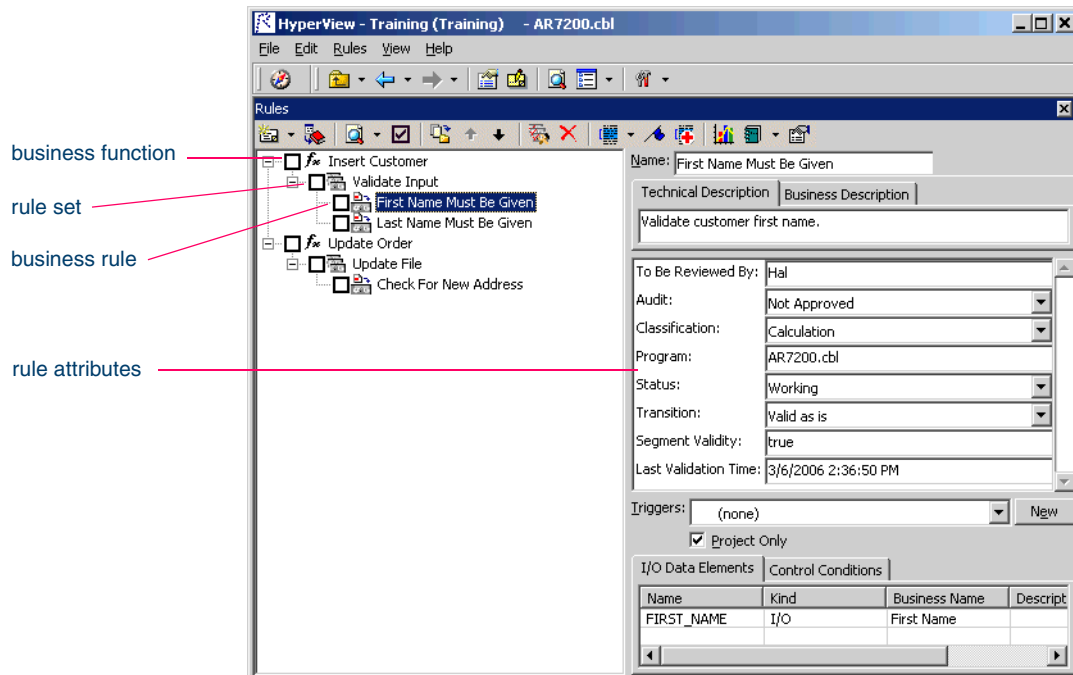
Business rule *attributes* define the characteristics of the rule: its name, classification, status, business description, and the like. You edit rule attributes in the righthand pane of the HyperView Rules window, as shown in Figure 8-1 on page 8-3.

You can edit rule attributes before or after you assign code segments, and you can use a variety of methods for batch editing rule attributes:

- The autodetection methods in both the Rules and Clipper panes let you specify the attributes for every rule autodetected by the method.
- The search facility in the Rules pane lets you locate existing rules with common characteristics, which you can then modify in bulk with the *Rules Changer*.

You can modify attribute characteristics and valid values, and you can create custom attributes suited to your particular needs.

Figure 8-1 *Business Functions, Rule Sets, and Business Rules*



Extracting Business Rules Manually

Even if you plan to autodetect business rules, it's important to know how to assign segments to rules manually, because in some cases it may be easier to adjust rules by hand than it is to run a full autodetect. The following description shows how to assign a rule segment at the same time as you create a rule. You can also assign a segment after creating a rule, as described in [“Assigning Segments to Existing Rules” on page 8-7](#).

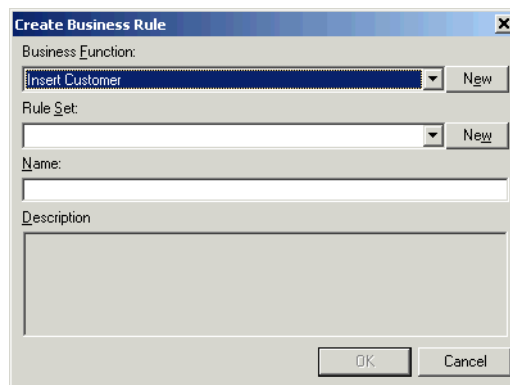
To extract business rules manually:

- 1 In the Source pane, select the segment you want to assign to a business rule. In the Rules pane, select the rule (if any) after which you want the new rule to be listed, then choose **Create Rule** in the **Rules** menu. The Create Business Rule dialog opens (Figure 8-2).

Tip: You can select a code segment as *construct* or *text*. Select a segment as text when you want to select either more or less code than a construct contains.

To select a segment as text, click-and-drag from the first line of the segment to the last line of the segment. The segment is highlighted in blue. To select a segment as construct, click inside the construct in the Source pane. The segment is highlighted in yellow.


Figure 8-2 *Create Business Rule Dialog*



- 2 In the **Business Function** drop-down, select the business function for the rule. The current business function is displayed. To specify a new business function, click **New**. The Business Function Dialog opens. In the Business Function dialog, enter the name and description of the new business function and click **OK**.
- 3 In the **Rule Set** drop-down, select the rule set for the rule. The current rule set is displayed. To specify a new rule set, click **New**. The Rule Set Dialog opens. In the Rule Set dialog, select the business function for the rule set in the **Business Function** drop-down, or create a new business function as described in [step 1 on page 8-4](#). Enter the name and description of the new rule set and click **OK**.
- 4 In the **Name** field, enter the name of the business rule.
- 5 In the **Description** field, enter a technical description of the business rule.
- 6 Click **OK**. As shown in Figure 8-1 on page 8-3, HyperView:
 - Lists the new rule under the specified business function and rule set in the lefthand pane of the window. Change the order in which rules are listed and perform other tasks as described in [“Performing Basic Tasks in the Rules Pane” on page 8-7](#).
 - Displays the attributes of the new rule in the righthand pane. Edit the new rule’s attributes as described in [“Editing Rule Attributes” on page 8-9](#).

Autodetecting Business Rules

The autodetect method identifies statements containing fields that impact a *seed variable*. The method traces the flow of data from the variable to its source, assigning each statement (including the statement that contains the variable) to a new rule. The method detects the input/output data elements for each rule, and any business names you have assigned to them in your project glossary.

You can invoke the autodetect method manually, by selecting the seed variable and clicking the  button on the Rules pane tool bar. You can also invoke the method for a dataport or I/O statement with nested vari-

ables, or for every variable in a Clipper list, as described in [“Generating Business Rules” on page 4-8](#). Cancel autodetection at any time during the process by clicking the **Cancel** button in the upper righthand corner of the HyperView window.

Note: Specify the impact relationships you want to autodetect in the Impact Report Options window, as described in [step 2 on page 5-6](#).

Set **Enable Impact Report** and **Enable Data Element Flow** in the Verification tab of the Project Options window to enable the autodetect method.

Rules in the same paragraph are assigned to a rule set with the name of the paragraph. Rule sets in the same program are assigned to a business function with the name of the program. Rules themselves have the name of the statement assigned to them.

Extracting Business Rules with Clipper

Clipper is a staging tool that lets you create lists of candidates for business rule extraction and other tasks. Use Clipper to:

- Autodetect each code segment that impacts a listed construct. When you choose this option, you edit rule attributes in batch mode in the Rules pane. For more information, see [“Generating Business Rules” on page 4-8](#).
- Create rules in a specified rule set. When you choose this option, you batch process rule attributes in the generation step. For more information, see steps [9-14](#) on pages [4-4-4-6](#).

Performing Basic Tasks in the Rules Pane

The lefthand pane of the Rules window lists every business rule in the current project, organized by business function and rule set. Select:

- A business function to view its rule sets in the righthand pane of the window.
- A rule set to view its rules in the righthand pane of the window. If a rule set is triggered by a rule, the trigger is listed in the lower portion of the righthand pane.
- A rule to view its attributes in the righthand pane of the window and its segment in the HyperView Source and Context panes.

For more information on rule attributes, including triggers, see [“Editing Rule Attributes” on page 8-9](#).

Viewing the Activity Log To view a chronological record of your activities with respect to a rule, select the rule and choose **Activity Log** in the **Rules** menu.

Assigning Segments to Existing Rules To assign a segment to an existing rule (whether or not a segment has already been assigned to it), select the rule, then select the segment in the Source pane. Choose **Assign segment** in the **Rules** menu. For instructions on how to select a segment in the Source pane, see the tip on [page 8-4](#).

Deleting a Segment from a Rule To delete a segment from a rule, select the rule and choose **Delete segment** in the **Rules** menu.

Selecting Rules for Batch Processing To make a rule a candidate for batch processing, click the check box next to the rule. To check all rules, choose **Select All** in the **Rules** menu. Choose **Unselect All** to uncheck the rules. To check all the rules in a rule set, click the check box for the rule set. To check all the rules in a business function, click the check box for the business function.

Tip: Choose **Selected Only** in the **Rules** menu to display only the checked business functions, rule sets, or rules (and their parents or children) in the Rules pane.

Moving Rules To move a rule to a different rule set, select it and choose **Copy/Move** in the **Rules** menu. The Move Business Rule to Another Rule Set dialog opens, where you can specify the new rule set for the rule.

Moving or Copying Rule Sets To move or copy a rule set to a different business function, select it and choose **Copy/Move** in the **Rules** menu. The Copy/Move Rule Set dialog opens, where you can specify the new business function for the rule set and whether you want to move or copy it.

Documenting Rules Sets and Business Functions To edit the description of a rule set or business function, select it, then select the Description tab in the righthand pane of the window. In the Description tab, enter the text of the description. To edit the business area of a rule set or business function, select it, then select the Business Area tab in the righthand pane of the window. In the Business Area tab, enter the text for the business area.

Deleting Rules, Rules Sets, and Business Functions To delete a rule, rule set, or business function, select it and choose **Delete** in the **Rules** menu. You are prompted to confirm the deletion. Click **OK**. You cannot delete a non-empty rule set or business function.

Changing the Order of Rules and Rule Sets To move a rule up in the sequence for its rule set, select it and choose **Move Up in Sequence** in the **Rules** menu. To move a rule down in the sequence for its rule set, select it and choose **Move Down in Sequence** in the **Rules** menu. You can also use these menu choices to move a selected rule set up or down in the sequence for its business function.

Tip: For reporting purposes, it's usually best to list rules in order of execution. Autodetected rules are always initially listed in order of execution.

Exporting Rules, Rule Sets, and Business Functions To export rules, rule sets, or business functions to an ERD file — for import to another workspace or a third-party tool — select the items you want to export, then choose **Rule Export** in the **Rules** menu. You are prompted to con-



tinue. Click **Yes**. The Export *Objects* dialog opens, where you can specify the name and folder for the ERD file.

Note: Only the parents of exported items are exported. Children are not. That is, if you export a business function, only the business function is exported. If you export a rule set, only the rule set and its business function are exported.

Importing Rules, Rule Sets, and Business Functions To import rules, rule sets, or business functions from an ERD file, choose **Rule Import** in the **Rules** menu. The Import Rules dialog opens, where you can specify the name and folder for the ERD file.

Parents are also imported. If the parent for an imported item already exists in the workspace and has different children, the item is added to the existing children.

Note: If the segment for an imported rule does not exist in the workspace, the segment validity attribute for the rule is set to False.

Flagging Segments in Source To place a  symbol in the HyperView Source pane display next to each line in the assigned segment, click the  button on the tool bar.

Editing Rule Attributes

A business rule is initially just a shell. Before the rule will be useful to other members of your organization, you need to define its attributes: what task the rule performs, whether it has been approved, whether it triggers another rule set, and so forth. You edit rule attributes in the right-hand pane of the HyperView Rules window, as shown in Figure 8-1 on page 8-3.

You can edit rule attributes before or after you assign code segments, and you can use a variety of methods for batch editing rule attributes:

- The autodetection methods in both the Rules and Clipper panes let you specify the attributes for every rule autodetected by the method.
- The search facility in the Rules pane lets you locate existing rules with common characteristics, which you can then modify in bulk

with the Rules Changer. For more information, see [“Searching for Rules” on page 8-17](#) and [“Rule Search Window” on page 8-18](#).

You can modify attribute characteristics and valid values, and you can create custom attributes suited to your particular needs. For more information, see [“Customizing Rule Attributes” on page 8-21](#).

Editing System Attributes

System attributes cannot be modified. The values for many of these attributes are set programmatically.

Editing the Name Attribute

You specify the name of a rule when you create it. To edit the Name attribute, enter a new name in the **Name** field.

Editing the Technical Description Attribute

The Technical Description attribute identifies the role of a rule in program logic. To edit the Technical Description attribute, enter appropriate text in the **Technical Description** tab. You can also specify the Technical Description when you create a rule.

Editing the Business Description Attribute

The Business Description attribute identifies the business role of a rule. To edit the Business Description attribute, enter appropriate text in the **Business Description** tab.

Editing the To Be Reviewed By Attribute

The To Be Reviewed By attribute identifies the business analyst responsible for reviewing and approving a rule. To edit the attribute, enter appropriate text in the **To Be Reviewed By** field.

Editing the Program Attribute

The Program attribute identifies the program that contains the rule segment. The value is supplied when you assign a segment to a rule.

Editing the Segment Validity Attribute

Refreshing or editing source code may result in a mismatch between rules and segments. Lines of code may have been added or deleted during the refresh or edit, causing a rule no longer to be synchronized with its segment after re-verification.

The Segment Validity attribute specifies whether a rule segment is valid. The value True is supplied when a segment is validated. The value False is supplied when a segment becomes invalid. For more information, see [“Validating Rule Segments after Refreshing or Editing Code” on page 8-20.](#)

Editing the Last Validation Time Attribute

The Last Validation Time attribute identifies the date and time a rule segment was last validated. The value is supplied when a segment is validated. For more information, see [“Editing the Segment Validity Attribute.”](#)

Editing User-Defined Attributes

User-defined attributes can be renamed, defined with different values, or deleted altogether, as described in [“Customizing Rule Attributes” on page 8-21.](#) Their default names and values are described below.

Editing the Audit Attribute

Newly created rules are considered unapproved, pending a decision that the rule represents valid business logic. The Audit attribute identifies whether a rule has been audited and approved. To edit the Audit attribute, choose one of the following from the **Audit** drop-down:

- **Not Approved** if the rule has not been accepted as valid business logic.
- **Approved** if the rule has been accepted as valid business logic.

Editing the Classification Attribute

The Classification attribute identifies the programming task a rule performs. To edit the classification attribute, choose one of the following from the **Classification** drop-down:

- **I/O** if the rule performs an input or output function.
- **Calculation** if the rule performs a calculation.
- **Security** if the rule performs a security function.
- **Decision** if the rule resolves a decision.
- **Validation** if the rule performs validation.
- **Flow** if the rule performs a data flow function.

Editing the Status Attribute

The Status attribute identifies the status of a rule in a code extraction or other project. To edit the Status attribute, choose one of the following from the **Status** drop-down:

- **(none)** if the rule has no extraction status.
- **Extracted** if the rule has been extracted, but not accepted or rejected.
- **Working** if the rule is still being extracted.
- **Accepted** if the rule has been accepted.
- **Rejected** if the rule has been rejected.


Editing the Transition Attribute

The Transition attribute identifies the status of a rule in an application redevelopment project. To edit the Transition attribute, choose one of the following from the **Transition** drop-down:

- **Valid as is** if the rule is valid for redevelopment.
- **Obsolete** if the rule is obsolete.
- **Requires modification** if the rule requires modification for redevelopment.
- **Duplicate** if the rule is a duplicate of a rule being used for redevelopment.

Editing the Triggers Attribute

When you organize rules in rule sets, you are typically identifying them as members of a business process. That means you will usually want to indicate when a rule's execution completes one business process and triggers another. The Valid Phone Number rule in the rule set Validate Input might trigger the rule set Update File, for example.

The Triggers attribute identifies the rule set a rule triggers. To edit the Triggers attribute, click  button for the **Triggers** field. The Select Triggered Rule Sets window opens.

Tip: Select the **Project Only** check box in the Select Triggered Rule Sets window to display only the rule sets in the selected project, rather than all the rulesets in the workspace.

In the Rule Set pane, select the rule set for the rule to trigger. To create a new rule set for the rule to trigger, click **New**, then follow the instructions for creating a rule set in [step 3 on page 8-5](#).

Note: A rule can trigger the rule set it belongs to. A rule set can be triggered by multiple rules.

Editing the I/O Data Elements Attribute

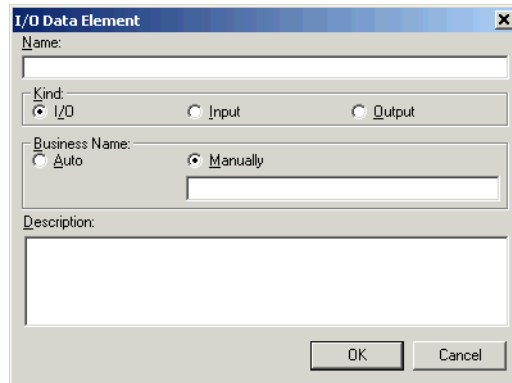
The I/O Data Elements attribute identifies the input, output, and input/output fields in the code for a rule. You can specify the fields manually or autodetect them.

Use the autoretrieve feature to match I/O elements with the business names you've assigned to them in your project glossary, as described in [Chapter 7, "Setting Up a Glossary of Data Items."](#) A business name identifies the business function of an I/O element in natural language, and is invaluable in documenting and retrieving business rules. The field PLC_NUM, for example, might have the business name Policy Number.

To specify I/O data elements manually:

- 1 Select a rule and choose **Create I/O Data Element** in the **Rules** menu. The I/O Data Element dialog opens (Figure 8-3 on page 8-14).

Figure 8-3 I/O Data Element Dialog



- 2 In the **Name** field, enter the name of the data element as it appears in the code.
- 3 In the Kind group box, choose:
 - **I/O** for an input and output field.
 - **Input** for an input field.
 - **Output** for an output field.

Note: If a data element already exists for a rule, and a data element with the same name but a different type is autodetected for the rule, the type of the original data element is defined as I/O (input/output).

- 4 In the Business Name area, select:
 - **Auto** if you want to automatically retrieve the business name for the I/O element from your project glossary.
 - **Manually** if you want to enter the business name for the I/O element by hand. Enter the name in the field below the radio button
- 5 In the **Description** field, enter a description of the I/O element.
- 6 Click **OK** to dismiss the dialog and return to the Rules window. The specified field is listed in the I/O Data Elements tab in the righthand pane of the window.

- 7 Repeat steps [1-6](#) for each field you want to add to the rule.

Tip: To edit a field in the table, right-click the field and choose **Edit** in the pop-up menu. The I/O Data Element dialog opens (Figure 8-3 on page 8-14), where you can make the necessary changes. To delete a field, right-click it and choose **Delete** in the pop-up menu.

To autodetect I/O data elements:

- 1 Select a rule and choose **Detect I/O Data Elements** in the **Rules** menu.
- 2 When the autodetect process is complete, a dialog box displays the number of autodetected I/O elements. Click **OK**. Click the I/O Data Elements tab in the righthand pane of the Rules window to view the autodetected I/O elements.

Note: Set **Enable Impact Report** and **Enable Data Element Flow** in the Verification tab of the Project Options window to enable the autodetect method for I/O data elements.

The method is based on a set of heuristics designed to produce reasonable output. Results may need to be verified by a detailed impact analysis. For more information on the heuristics, contact support services.

To retrieve I/O data element business names:

- 1 Select a rule and choose **Populate I/O Business Names** in the **Rules** menu to automatically retrieve business names for rule I/O elements from your project glossary. You are prompted to continue. Click **Yes**.
- 2 When the autoretrieve process is complete, a dialog box displays the number of autoretrieved I/O business names. Click **OK**. Click the I/O Data Elements tab in the righthand pane of the Rules window to view the autoretrieved business names.

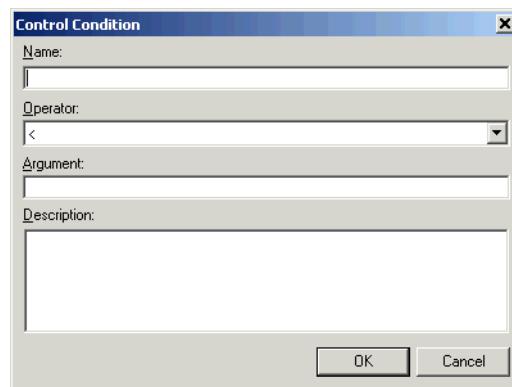
Editing the Control Conditions Attribute

The Control Conditions attribute identifies the conditions that govern the execution of a rule. The rule may be executed, for example, only when the value of the State Field is California.

To specify control conditions:

- 1 Select a rule and choose **Create Control Condition** in the **Rules** menu. The Control Condition dialog opens (Figure 8-4).

Figure 8-4 Control Condition Dialog



- 2 In the **Name** field, enter the name of the data item evaluated by the condition as it appears in the code.
- 3 In the **Operator** drop-down, choose the operator for the comparison that determines whether the condition is met.
- 4 In the **Argument** field, enter the constant value to which the data item is compared. The value may be the name of another field.
- 5 In the **Description** field, enter a description of the control condition.
- 6 Click **OK** to dismiss the dialog and return to the Rules window. The specified condition is listed in the Control Conditions tab in the righthand pane of the window.

Tip: To edit a field in the table, right-click the field and choose **Edit** in the pop-up menu. The Control Condition dialog opens

(Figure 8-4), where you can make the necessary changes. To delete a field, right-click it and choose **Delete** in the pop-up menu.

- 7 Repeat steps [1-6](#) for each control condition you want to add to the rule.

Searching for Rules

Use the search facility in the Rules pane to find rules based on segment location and/or attribute values. Use the Rules Changer to batch-edit the rules returned by the search. For more information, see [“Rule Search Window” on page 8-18](#).

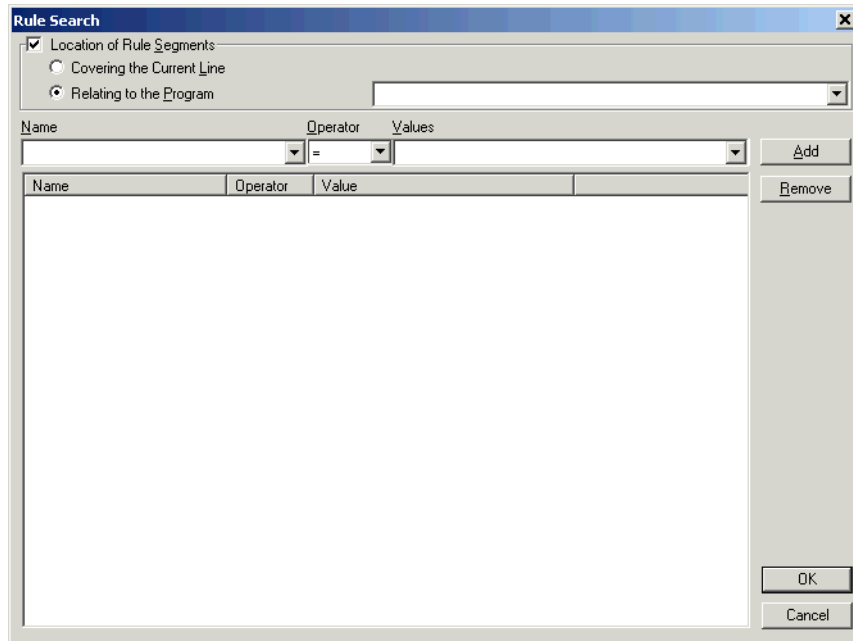
To search for rules:

- 1 In the **Rules** menu, choose **Start Search**. The Rule Search window opens (Figure 8-5 on page 8-18).
- 2 To search for rules by segment location, select **Location of Rule Segments**, then choose:
 - **Covering the Current Line** if you want to find rules with a segment that contains the line of code selected in the Source pane.
 - **Relating to the Program** if you want to find rules related to a given program. Select the program in the adjacent drop-down.
- 3 To search for rules by attribute value, select:
 - The attribute in the **Name** drop-down.
 - The operator for the attribute value comparison in the **Operator** drop-down.
 - The value of the attribute in the **Value** drop-down.

Click **Add**. The search criterion is added to the list box below the drop-downs. Repeat this step for each attribute you want to search by. Select a criterion and click **Remove** to delete the criterion from the list box.

- 4 When you are satisfied with your entries in the Rule Search window, click **OK**. The search facility automatically selects the check box for each rule returned by the search.

Figure 8-5 *Rule Search Window*



Batch Processing Rules

Use the Rules Changer to batch process rules. You can select batch-processing candidates manually, or use the search facility to select candidates automatically.

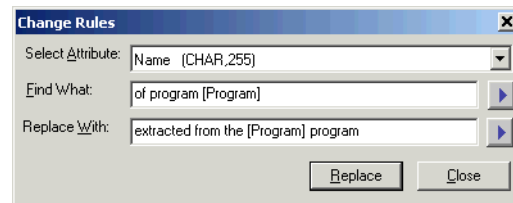
To batch process rules:

- 1 Select the check box for each rule you want to batch process. If you are batch processing rules returned by a search, the check boxes are already selected.

Tip: To select all the rules returned by a previous search, select **Previous Selected** in the **Rules** menu. To return to the rules selected in the next most recent search, select **Next Selected** in the **Rules** menu.


- 2 In the **Rules** menu, choose **Rules Changer**. You are prompted to continue. Click **Yes**. The Change Rules window opens (Figure 8-6).

Figure 8-6 *Change Rules Window (After Step 3)*



- 3 In the **Attribute** drop-down, select the attribute you want to batch edit. In the **Find What** field, enter a pattern for the search text. In the **Replace With** field, enter a pattern for the replacement text. You can use any combination of text and bracketed attribute names.

Tip: If you select an attribute with enumerated values, the values are displayed in drop-downs for the **Find What** and **Replace With** fields. Use the drop-downs to populate the fields.

If you select an attribute without enumerated values, click the  button for the **Find What** and **Replace With** fields to select the attributes you want to use in the find and replace.

So if you wanted to rename two rules:

```
Rule 1 of program ABC
Rule 2 of program XYZ
```

as follows:

```
Rule 1 extracted from the ABC program
Rule 2 extracted from the XYZ program
```

you would select the Name attribute in the **Select Attribute** field and enter in the **Find What** field:

```
of program [Program]
```

and in the **Replace With** field:

```
extracted from the [Program] program
```

You can substitute a business name for an I/O element in an attribute value by appending “/bus” to the attribute name in either the **Find What** or **Replace With** fields.

Suppose the Catalog-Master variable has the business name Master Catalog. Now you want to replace the existing Business Description for the Rewrite Catalog-Master rule with the text “Rewrite Master Catalog.” You would select the Business Description attribute in the **Select Attribute** field and enter in the **Replace With** field:

```
[Name/bus]
```

- 4 When you are satisfied with your entries in the Change Rules window, click **Replace**. The Rules Changer modifies the selected rules. Click **Close** to dismiss the window and return to the Rules pane.

Validating Rule Segments after Refreshing or Editing Code

Refreshing or editing source code may result in a mismatch between a rule and segment. Lines of code may have been added or deleted during the refresh or edit, causing a rule no longer to be synchronized with its segment after re-verification.

The Segment Validity attribute of a rule indicates whether a rule segment no longer is valid. You can re-synchronize the segment manually, as described in [“Extracting Business Rules Manually” on page 8-4](#), or automatically, as described below.

To validate rules automatically:

- 1 Select the rules you want to validate and choose **Validate** in the **Rules** menu. You are prompted to continue. Click **Yes**.

- 2 When the validation process is complete, a dialog box displays the number of validated rules. Click **OK**. The Segment Validity attribute for the rules is updated.

Some Limitations on Automatic Rule Validation

Automatic rule validation will fail to re-synchronize a segment if the segment has been modified in any way during the refresh or edit — if a line has been inserted in the segment code, for example, or if the text of a line has been changed. You must re-synchronize these segments manually. Blank lines are OK.

Note, too, that if the code for a segment is duplicated in a program, the autovalidation method will synchronize the rule with the instance of the duplicated code nearest to its previous location, whether or not that instance was originally associated with the rule. The rule will be the same, but the program context may be different from the one you intended. Here, too, you will have to recreate the rule manually.

Customizing Rule Attributes

Use the workbench Administration tool to modify characteristics and valid values of user-defined attributes, and to create custom attributes suited to your particular needs. Custom attributes are available to each workspace you create in the workbench.

The attribute definitions are stored in a file with a name of the form *.Repstry.xml. You can create multiple custom attribute files if you want, and switch back and forth between them as needed. ATW applies the most recently saved definitions to your business rules.

Tip: Choose **Extended Rule Attributes** in the HyperView **Rules** menu to open a limited version of this tool in which you can modify or delete the values for character-type attributes. If you delete values in use in existing business rules, the default value of the attribute is restored.

To customize rule attributes:

- 1 In the Windows **Start** menu, go to the **Programs** folder. Choose **IBM**, then **ATW Administration**. The ATW Administration tool window opens.
- 2 In the **Administer** menu, choose **Edit Rule Attributes**.
 - If you have never customized rule attributes, the window opens automatically with the default business rule attributes.
 - If you have already customized rule attributes, and the **Use Latest File** choice is not selected in the **Administer** menu, an Open dialog appears where you can select the *.Repstry.xml file that contains the customized attributes you want to modify. The Edit Rule Attributes window opens (Figure 8-7 on page 8-23).
 - If you have already customized rule attributes, and the **Use Latest File** choice is selected in the **Administer** menu, the Edit Rule Attributes window opens with the customized attributes for the *.Repstry.xml file you last saved in the Edit Rule Attributes window.

Tip: Select **Open as read-only** in the Open dialog if you want to open a read-only version of the *.Repstry.xml file.

- 3 To modify an existing attribute, select the attribute in the lefthand pane, then edit the attributes you want to modify in the righthand pane.

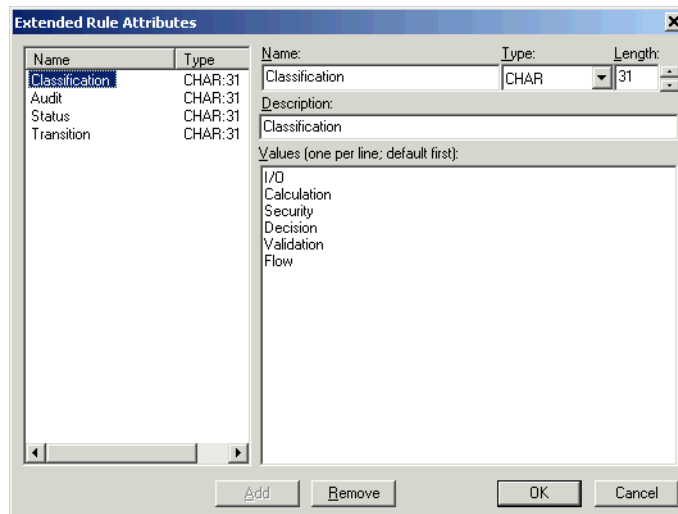
Tip: To add a value in the Values pane, go to the end of any line containing a value, press Enter to create a new line, and type in the new value. To remove an attribute, select it and click **Remove**.

- 4 To create a new attribute, select an existing attribute in the lefthand pane, then overwrite its name with the name of the new attribute in the righthand pane. Edit the remaining characteristics of the new attribute as necessary, then click **Add**. The new attribute is added to the list in the lefthand pane.


- 5 When you are satisfied with your entries, click **OK**. A Save As dialog opens, where you can specify the name and location of the *.Repstry.xml file. ATW applies the definitions to your business rules.

Tip: To switch to the definitions in a different *.Repstry.xml file, deselect **Use Latest File** in the **Administer** menu, open the file in the Edit Rule Attributes window, and click **OK** (whether or not you make changes to the file). Then save the file in the Save As dialog.

Figure 8-7 *Edit Rule Attributes Window*




Generating Business Rule Reports

Business rule reports make it easy to develop the kind of detailed specification you'll need for a code extraction or redevelopment project. To generate a report, click the  button on the tool bar and choose the report type from the pull-down menu.

The following reports are available:

- The *Business Rules Report* summarizes the rules in a project. The report lists the attributes, data elements, and segments of each rule in the project.
- The *Process Outline Report* shows the context in which rules are executed. The report lists every paragraph in the selected program, the process outline of the paragraph, every rule with a segment that starts in the paragraph, and the segment itself. A process outline describes the call flow of a paragraph. If you have annotated the paragraph (see [“Annotating a Paragraph”](#) on page 8-24), the annotation appears in the paragraph column below the paragraph name.
- The *Coverage Report* shows the percentage of program logic (including blank lines and comments) that contains rule segments. The report lists the programs in the project, the total number of lines of code they contain, the number of lines of code with program logic, the number of business rule segments the program contains, and the percentage of program logic that contains business rule segments.

Annotating a Paragraph

Select a paragraph in a HyperView pane and click the  button on the tool bar to open the Annotate dialog, where you can enter a business name and description of the paragraph. The annotation appears in the Paragraph column of the Process Outline Report under the paragraph name.

Printing Reports

Click **Page Setup** in a report window to set up the page for a printed report. Click **Print** to print the report. The Print dialog opens, where you can set options for the print job. Click **OK**.

Exporting Reports

You can export business rule reports to a variety of standard formats. Click **Save** in a report window to export a report to HTML, Excel, RTF, Word, or formatted text. A Save As dialog opens, where you can specify the name, location, and file type of the report.

Note: Excel is usually the best choice if you need to manipulate the report or perform computations.

What's Next?

That completes your tour of HyperView! Now you're ready to begin extracting components from your legacy application — self-contained programs that can be reused with other programs in modular fashion. *Creating Components* in the workbench documentation set describes how to extract components.

8-26 Extracting Business Rules
What's Next?

Glossary

ADABAS

ADABAS is a Software AG relational [DBMS](#) for large, mission-critical applications.

API

API stands for application programming interface, a set of routines, protocols, and tools for building software applications.

applet

See [Java applet](#).

AS/400

The AS/400 is a midrange server designed for small businesses and departments in large enterprises.

BMS

BMS stands for Basic Mapping Support, an interface between application formats and [CICS](#) that formats input and output display data.

BSTR

BSTR is a Microsoft format for transferring binary strings.

CDML

CDML stands for Cobol Data Manipulation Language, an extension of the [Cobol](#) programming language that enables applications programmers to code special instructions to manipulate data in a [DMS](#) database and to compile those instructions for execution.

CICS

CICS stands for Customer Information Control System, a program that allows concurrent processing of [transactions](#) from multiple terminals.

Cobol

Cobol stands for Common Business-Oriented Language, a high-level programming language used for business applications.

COM

COM stands for Component Object Model, a software architecture developed by Microsoft to build [component](#)-based applications. COM objects are discrete components, each with a unique identity, which expose interfaces that allow applications and other components to access their features.

complexity

An application's complexity is an estimate of how difficult it is to maintain, analyze, transform, and so forth.

component

A component is a self-contained program that can be reused with other programs in modular fashion.

construct

A construct is an item in the [parse tree](#) for a source file — a section, statement, condition, variable, or the like. A variable, for example, can be related in the parse tree to any of three other constructs — a declaration, a dataport, or a condition.

copybook

A copybook is a common piece of source code to be copied into many [Cobol](#) source programs. Copybooks are functionally equivalent to C and C++ include files.

CORBA

CORBA stands for Common Object Request Broker Architecture, an architecture that enables distributed objects to communicate with one another regardless of the programming language they were written in or the operating system they are running on.

CSD file

CSD stands for [CICS](#) System Definition. A CSD file is a [VSAM](#) data set containing a resource definition record for every resource defined to [CICS](#).

database schema

A database schema is the structure of a database system, described in a formal language supported by the [DBMS](#). In a relational database, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

dataport

A dataport is an input/output statement or a call to or from another program.

DB/2

DB/2 stands for Database 2, an IBM system for managing relational databases.

DBCS

DBCS stands for double-byte character string, a character set that uses two-byte (16-bit) characters rather than one-byte (8-bit) characters.

DBMS

DBMS stands for database management system, a collection of programs that enable you to store, modify, and extract information from a database.

DDL

DDL stands for Data Description Language (DDL), a language that describes the structure of data in a database.

decision resolution

Decision resolution lets you identify and resolve dynamic calls and other relationships that the [parser](#) cannot resolve from static sources.

DMS

DMS stands for Data Management System, a Unisys database management software product that conforms to the CODASYL (network) data model and enables data definition, manipulation, and maintenance in mass storage database files.

DPS

DPS stands for Display Processing System, a Unisys product that enables users to define forms on a terminal.

ECL

ECL stands for Executive Control Language, the operating system language for Unisys OS 2200 systems.

effort

Effort is an estimate of the time it will take to complete a task related to an application, based on weighted values for selected [complexity](#) metrics.

EJB

EJB stands for Enterprise JavaBeans, a [Java API](#) developed by Sun Microsystems that defines a [component](#) architecture for multi-tier client/server systems.

EMF

EMF stands for Enhanced MetaFile, a Windows format for graphic images.

entity

An entity is an object in the [repository](#) model for a legacy application. The relationships between entities describe the ways in which the elements of the application interact.

FCT

FCT stands for File Control Table (FCT), a [CICS](#) table that contains processing requirements for output data streams received via a remote job entry session from a host system. Compare [PCT](#).

HTML

HTML stands for HyperText Markup Language, the authoring language used to create documents on the World Wide Web.

IDL

IDL stands for Interface Definition Language (IDL), a generic term for a language that lets a program or object written in one language communicate with another program written in an unknown language.

IDMS

IDMS stands for Integrated Database Management System, a Computer Associates database management system for the IBM mainframe and compatible environments.

IMS

IMS stands for Information Management System, an IBM program product that provides transaction management and database management functions for large commercial application systems.

Java

Java is a high-level [object-oriented programming](#) language developed by Sun Microsystems.

Java applet

A [Java](#) applet is a program that can be sent with a Web page. Java applets perform interactive animations, immediate calculations, and other simple tasks without having to send a user request back to the server.

JavaBeans

JavaBeans is a specification developed by Sun Microsystems that defines how [Java](#) objects interact. An object that conforms to this specification is called a JavaBean.

JCL

JCL stands for Job Control Language, a language for identifying a [job](#) to OS/390 and for describing the job's requirements.

JDBC

JDBC stands for Java Database Connectivity, a standard for accessing diverse database systems using the [Java](#) programming language.

job

A job is the unit of work that a computer operator or a program called a *job scheduler* gives to the operating system. In IBM mainframe operating systems, a job is described with job control language ([JCL](#)).

logical component

A logical component is an abstract [repository](#) object that gives you access to the source files that comprise a [component](#).

MFS

MFS stands for Message Format Service, a method of processing [IMS](#) input and output messages.

Natural

Natural is a programming language developed and marketed by Software AG for the enterprise environment.

object model

An object model is a representation of an application and its encapsulated data.

object-oriented programming

Object-oriented programming organizes programs in terms of objects rather than actions, and data rather than logic.

ODBC

ODBC stands for Open Database Connectivity, a standard for accessing diverse database systems.

orphan

An orphan is an object that does not exist in the reference tree for any startup object. Orphans can be removed from a system without altering its behavior.

parser

The ATW parser defines the [object model](#) and [parse tree](#) for a legacy application.

parse tree

A parse tree defines the relationships between the constructs that comprise a source file — its sections, paragraphs, statements, conditions, variables, and so forth.

PCT

PCT stands for Program Control Table, a [CICS](#) table that defines the transactions that the CICS system can process. Compare [FCT](#).

PL/I

PL/I stands for Programming Language One, a third-generation programming language developed in the early 1960s as an alternative to assembler language, [Cobol](#), and FORTRAN.

profile

Profiles are HTML views into a [repository](#) that show all of the analysis you have done on an application. Profiles are convenient ways to share information about legacy applications across your organization.

QSAM

QSAM stands for Queued Sequential Access Method, a type of processing that uses a queue of data records—either input records awaiting processing or output records that have been processed and are ready for transfer to storage or an output device.

relationship

The relationships between entities in the [repository](#) model for a legacy application describe the ways in which the elements of the application interact.

relaxed parsing

Relaxed parsing lets you verify a source file despite errors. Ordinarily, the parser stops at a statement when it encounters an error. Relaxed parsing tells the parser to continue to the next statement.

repository

A repository is a database of program objects that comprise the model for an application.

schema

See [database schema](#).

SQL

SQL stands for Structured Query Language, a standard language for relational database operations

system program

A system program is a generic program — a mainframe sort utility, for example — provided by the underlying system and used in unmodified form in the legacy application.

TIP

TIP stands for Transaction Processing, the Unisys real-time system for processing transactions under Exec control.

transaction

A transaction is a sequence of information exchange and related work (such as database updating) that is treated as a unit for the purposes of satisfying a request and for ensuring database integrity.

VALTAB

VALTAB stands for Validation Table, which contains the information the system needs to locate, load, and execute transaction programs. See also [TIP](#).

VSAM

VSAM stands for Virtual Storage Access Method, an IBM program that controls communication and the flow of data in a Systems Network Architecture network.

XML

XML stands for Extensible Markup Language, a specification for creating common information formats.

Bibliography

- *IBM Asset Transformation Workbench v2.1 Getting Started (SC31-6877-01)*
- *IBM Asset Transformation Workbench v2.1 Preparing Projects (SC31-6879-01)*
- *IBM Asset Transformation Workbench v2.1 Analyzing Projects (SC31-6880-01)*
- *IBM Asset Transformation Workbench v2.1 Analyzing Programs (SC31-6878-02)*
- *IBM Asset Transformation Workbench v2.1 Profiling Projects (SC31-6881-01)*
- *IBM Asset Transformation Workbench v2.1 Creating Components (SC31-6876-01)*
- *IBM Asset Transformation Workbench v2.1 Parser Reference (SC31-6882-01)*
- *IBM Asset Transformation Workbench v2.1 Architecture Reference (SC31-6898-01)*
- *IBM Asset Transformation Workbench v2.1 Error Messages (SC31-6956-00)*
- *IBM Asset Transformation Workbench v2.1 Analyzing Projects in Eclipse (SC23-5820-00)*

Notices

This information was developed for products and services offered in the U.S.A. IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION •AS IS• WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
P.O. Box 12195, Dept. TL3B/B503/B313
3039 Cornwallis Rd.
Research Triangle Park, NC 27709-2195
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

The following terms are trademarks of the IBM Corporation or its subsidiaries in the United States or other countries or both:

Table 1. Trademarks

IBM	MVS
AS400	CICS
IMS	DB/2
Database 2	OS/390
S/390	z/OS

The following terms are trademarks of other companies:

Java and JavaScript are registered trademarks and Sun Solaris and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

Index

A

- advanced search facility
 - defining search criteria 3-9, 4-9, 4-10
 - executing advanced searches 3-8, 4-10
 - performing advanced searches 3-1
 - searching for duplicated code 3-6
 - use with Clipper 4-9, 4-10

- Animator, using 6-12

- annotations 8-24

- Audit attribute 8-11

- autodetecting business rules 8-5

B

- Bird's Eye pane 1-14

- Business Description attribute 8-10

- business functions

 - creating 8-5

 - deleting 8-8

 - documenting 8-8

 - exporting 8-8

 - importing 8-9

 - overview 8-2

 - business names

 - assigning in batch mode 7-4

 - assigning manually 7-4

 - deleting 7-8

 - editing 7-7

 - exporting and importing 7-6

 - overview 8-13

 - retrieving 8-14, 8-15

 - business rules

 - assigning segments 8-4, 8-7

 - autodetecting 8-5

 - batch processing 8-18

 - customizing attributes 8-21

 - deleting 8-8

 - deleting segments 8-7

 - editing rule attributes 8-9

- exporting 8-8
- extracting in Clipper pane 4-1, 8-6
- extracting manually 8-4
- flagging segments in source 8-9
- generating reports 8-23
- importing 8-9
- moving in rule set 8-8
- moving to a different rule set 8-8
- overview 8-1
- printing reports 4-14
- searching for 8-17
- selecting for batch processing 8-7
- selecting segments 8-4
- validating 8-20

Business Rules Report 8-24

C

Callie pane

- choosing the diagram view 6-3
- overview 6-2
- setting project options 6-6
- setting user preferences 6-4

Callie view

- refreshing the display 6-4

change magnitude, specifying 2-4

Classification attribute 8-12

Clipper pane

- creating a list 4-1
- creating or adding to a project 4-12
- extracting business rules 4-1
- generating reports 4-13
- injecting events 4-9
- showing impacts 4-9
- working with categories 4-7
- working with lists 4-7

Components pane 1-13

Context pane 2-7

Control Conditions attribute 8-16

Coverage Report 8-24

D

Data Flow pane 1-9

Data View pane 1-9

Details Report 4-14

duplicated code, searching for 3-6

E

Execution Path pane, using 6-8

exporting business names 7-6

F

Flowchart pane, using 6-10

G

Glossary pane

- assigning business names in batch mode 7-4

- assigning business names manually 7-4

- choosing a scope 7-3

- cleaning up a glossary 7-9

- deleting business names 7-8

- desynchronizing with other panes 7-3

- editing business names and descriptions 7-7

- generating a glossary 7-3

- generating reports 7-10

- importing and exporting business names 7-6

- overview 7-1

- setting options 7-8

H

HyperView 1-3
 advanced search facility 3-1
 Animator 6-12
 Callie pane 6-2
 Clipper pane 4-1
 Context pane 2-7
 Execution Path pane 6-8
 Flowchart pane 6-10
 Impact pane 5-15
 Impact Report tool 5-2
 List Browser window 1-20
 overview 1-1
 parse trees 1-1
 Properties window 1-17
 Rules pane 8-1
 Source pane 2-1
 starting 1-3

I

I/O Data Elements attribute 8-13
 Impact pane
 generating an impact trace 5-15
 relationships 5-9
 setting options 5-16
 Impact Report tool
 creating or adding to a project 5-11
 generating an impact trace 5-3
 relationships 5-9
 report depths 5-2
 setting options 5-6
 importing business names 7-6

L

Last Validation Time attribute 8-11
 Link Names mode 7-2
 Link Tokens mode 7-2
 List Browser window
 creating a list 1-22

deleting a list 1-22
 exporting a list 1-22
 opening a list 1-20
 overview 1-20

M

Metrics Report 4-13
 Model Reference pane 1-14

N

Name attribute 8-10

P

PL/I Call Diagram pane 1-10
 Process Outline Report 8-24
 process outlines 8-24
 Program attribute 8-10
 program control flows
 analyzing with Callie 6-2
 analyzing with the Animator 6-12
 analyzing with the Execution Path
 pane 6-8
 analyzing with the Flowchart pane
 6-10
 overview 6-1
 project
 creating or adding to in Clipper 4-12
 creating or adding to in Impact Report
 5-11
 Properties window 1-17

R

rule sets
 creating 8-5
 deleting 8-8
 documenting 8-8
 exporting 8-8
 importing 8-9
 moving in business function 8-8

- moving or copying 8-8
- overview 8-2
- triggering 8-13

Rules pane

- batch processing rules 8-18
- customizing rule attributes 8-21
- editing rule attributes 8-9
- extracting business rules manually 8-4
- generating business rule reports 8-23
- overview 8-1
- performing basic tasks 8-7
- printing business rule reports 4-14
- searching for rules 8-17
- validating rules 8-20

S

search

- advanced search facility 3-1
- Clipper search facility 4-9
- simple search facility 2-4

Segment Validity attribute 8-11

segments

- assigning to business rules 8-4, 8-7
- flagging in source 8-9
- selecting 8-4

signatures 3-6

Source pane

- collapsing paragraphs or subprograms 2-6
- flagging segments 8-9
- marking list items 1-21
- navigating to multiple occurrences of a construct 2-3
- navigating to related constructs 2-3
- overview 2-1
- searching for code 2-4
- selecting and copying code 2-2
- setting options 2-5

- showing source code boundaries 2-6
- specifying the change magnitude 2-4

Status attribute 8-12

T

Technical Description attribute 8-10

To Be Reviewed By attribute 8-10

tokens 7-1

Transition attribute 8-12

Triggers attribute 8-13