Micro Focus®

Modernization Workbench™

Analyzing Programs

Copyright © 2009 Micro Focus (IP) Ltd. All rights reserved.

Micro Focus (IP) Ltd. has made every effort to ensure that this book is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time. The software described in this document is supplied under a license and may be used or copied only in accordance with the terms of such license, and in particular any warranty of fitness of Micro Focus software products for any particular purpose is expressly excluded and in no event will Micro Focus be liable for any consequential loss.

Micro Focus, the Micro Focus Logo, Micro Focus Server, Micro Focus Studio, Net Express, Net Express Academic Edition, Net Express Personal Edition, Server Express, Mainframe Express, Animator, Application Server, AppMaster Builder, APS, Data Express, Enterprise Server, Enterprise View, EnterpriseLink, Object COBOL Developer Suite, Revolve, Revolve Enterprise Edition, SOA Express, Unlocking the Value of Legacy, and XDB are trademarks or registered trademarks of Micro Focus (IP) Limited in the United Kingdom, the United States and other countries.

IBM®, CICS® and RACF® are registered trademarks, and IMS™ is a trademark, of International Business Machines Corporation.

Copyrights for third party software used in the product:

- The YGrep Search Engine is Copyright (c) 1992-2004 Yves Roumazeilles
- Apache web site (http://www.microfocus.com/docs/links.asp?mfe=apache)
- Eclipse (http://www.microfocus.com/docs/links.asp?nx=eclp)
- Cyrus SASL license
- Open LDAP license

All other trademarks are the property of their respective owners.

No part of this publication, with the exception of the software product user documentation contained on a CD-ROM, may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine-readable form without prior written consent of Micro Focus (IP) Ltd. Contact your Micro Focus representative if you require access to the modified Apache Software Foundation source files.

Licensees may duplicate the software product user documentation contained on a CD-ROM, but only to the extent necessary to support the users authorized access to the software under the license agreement. Any reproduction of the documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

U.S. GOVERNMENT RESTRICTED RIGHTS. It is acknowledged that the Software and the Documentation were developed at private expense, that no part is in the public domain, and that the Software and Documentation are Commercial Computer Software provided with RESTRICTED RIGHTS under Federal Acquisition Regulations and agency supplements to them. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 et. seq. or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at FAR 52.227-19, as applicable. Contractor is Micro Focus (IP) Ltd, 9420 Key West Avenue, Rockville, Maryland 20850. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.

Contents

Preface

1 Introducing HyperView

2 Understanding Program Context

3 Performing Advanced Searches

4 Staging Program Analysis with Clipper

5 Analyzing Impact Traces

6 Analyzing Program Control Flows

7 Setting Up a Glossary

8 Extracting Business Rules

Preface

The Modernization Workbench 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. Language-specific parsers generate repository models that serve as the basis for a rich set of diagrams, reports, and other documentation.

The Modernization Workbench 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, “Understanding Program 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 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,” 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.
- Appendix A, “Using the Batch Duplicate Finder,” describes how to use the Batch Duplicate Finder to search for similarly coded paragraphs and programs.
- 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 Modernization Workbench manuals. Together they provide all the information you need to get the most out of the system.

- *Getting Started* introduces the Modernization Workbench. This guide provides an overview of the workbench tools, discusses basic concepts, and describes how to use common product features.
- *Preparing Projects* describes how to set up Modernization Workbench 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.
- *Managing Application Portfolios* describes how to build enterprise dashboards that track survey-based metrics for applications in your

portfolio. It also describes how to use Enterprise View Express to browse Web-generated views of application repositories.

- *Creating Components* describes how to use Application Architect to extract program components from legacy applications.
- *Batch Refresh Process* describes how to use the Modernization Workbench Batch Refresh Process utility to batch-synchronize workbench sources with sources at their original location.
- *Transforming Applications* describes how to redevelop legacy applications in modern languages.
- *Error Messages* lists the error messages issued by Modernization Workbench, with a brief explanation of each and instructions on how to proceed.

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 Modernization Workbench window by clicking the window and pressing the **F1** key.

Many Modernization Workbench 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

1

Much of the power of Modernization Workbench 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 glossary, and extract business rules.

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

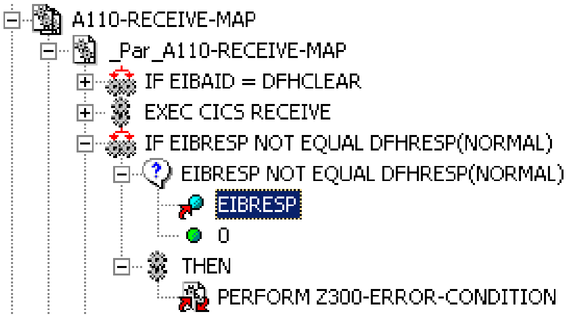
Understanding HyperView Models

For each object that comprises an application, HyperView generates a *construct model* that defines its syntax. The construct model shows in abstract form how the syntactical constructs that comprise the object (its sections, paragraphs, statements, conditions, variables, and so forth) are related.

A variable, for example, can be related in the construct model to its declaration, a dataport (if it is used in an I/O statement), or a condition (if the condition uses an expression of which the variable forms a part). You view the construct model for a source file in the HyperView Context pane (see “Context Pane” on page 1-6).

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

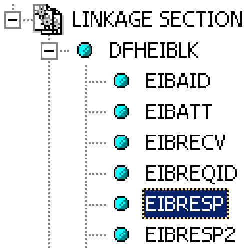
Figure 1-1 *Use of EIBRESP Variable*



```
A110-RECEIVE-MAP
  _Par_A110-RECEIVE-MAP
    IF EIBAID = DFHCLEAR
    EXEC CICS RECEIVE
    IF EIBRESP NOT EQUAL DFHRESP(NORMAL)
      EIBRESP NOT EQUAL DFHRESP(NORMAL)
        EIBRESP
        0
      THEN
        PERFORM Z300-ERROR-CONDITION
```

If you are interested in investigating other uses of EIBRESP in the program, you can navigate to the declaration of the variable in the construct model (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.

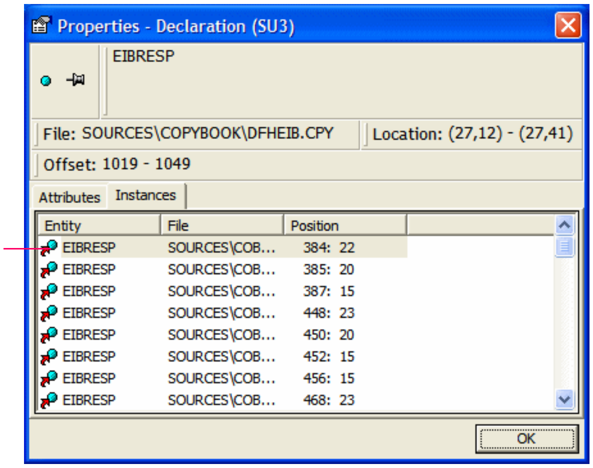
Figure 1-2 *Declaration of EIBRESP Variable*



```
LINKAGE SECTION
    DFHEIBLK
        EIBAID
        EIBATT
        EIBRECV
        EIBREQID
        EIBRESP
        EIBRESP2
```

From the declaration, you can generate a list of instances in which the variable is used (Figure 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*



click to navigate to where variable is used

EIBRESP

File: SOURCES\COPYBOOK\DFHEIB.CPY

Location: (27,12) - (27,41)

Offset: 1019 - 1049

Entity	File	Position
EIBRESP	SOURCES\COB...	384: 22
EIBRESP	SOURCES\COB...	385: 20
EIBRESP	SOURCES\COB...	387: 15
EIBRESP	SOURCES\COB...	448: 23
EIBRESP	SOURCES\COB...	450: 20
EIBRESP	SOURCES\COB...	452: 15
EIBRESP	SOURCES\COB...	456: 15
EIBRESP	SOURCES\COB...	468: 23

Select an instance in the list to navigate to that instance in the construct model.

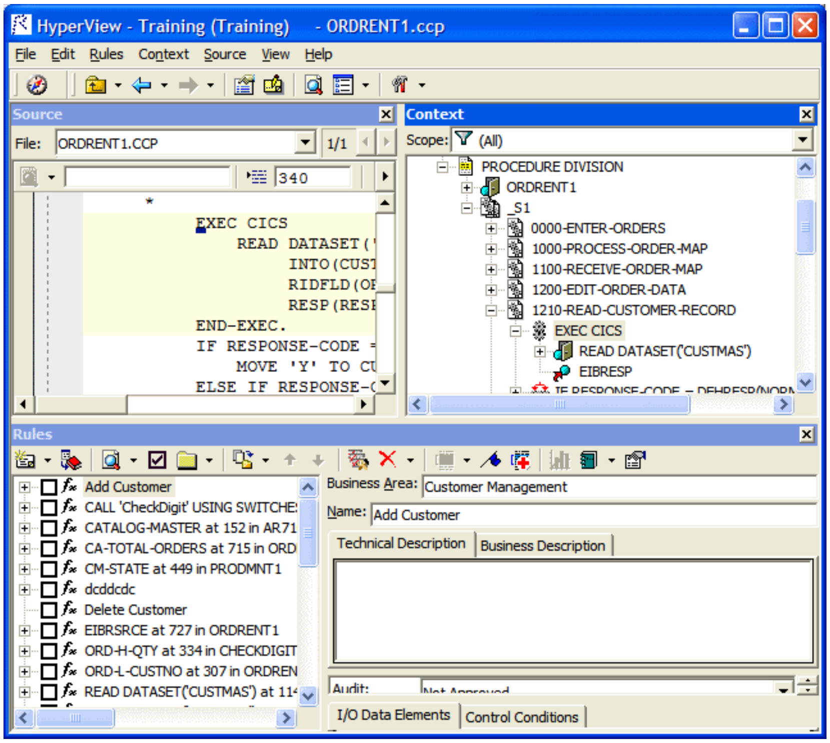
Starting HyperView

You can invoke HyperView directly by selecting a project or source file in the Modernization Workbench 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 shows a typical HyperView configuration. For instructions on how to use the Object pane to select a file in HyperView, see "Using the Objects Pane" on page 2-9.

Figure 1-4 *HyperView Window: Typical Configuration*



File: ORDRENT1.CCP 1/1

340

```
      *
                EXEC CICS
                    READ DATASET('
                         INTO(CUST
                         RIDFLD(OF
                         RESP(RESF
                END-EXEC.
                IF RESPONSE-CODE =
                    MOVE 'Y' TO CU
                ELSE IF RESPONSE-(
```

Scope: (All)

□ Add Customer
□ CALL 'CheckDigit' USING SWITCHE:
□ CATALOG-MASTER at 152 in AR71
□ CA-TOTAL-ORDERS at 715 in ORD
□ CM-STATE at 449 in PRODMNT1
□ dcddcdc
□ Delete Customer
□ EIBRSRCE at 727 in ORDRENT1
□ ORD-H-QTY at 334 in CHECKDIGIT
□ ORD-L-CUSTNO at 307 in ORDREN
□ READ DATASET('CUSTMAS') at 11

Business Area: Customer Management

Name: Add Customer

Audit: Not Approved

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.

Tip: Double-click the title bar of a pane to maximize the pane in HyperView, hiding any other open panes. Double-click the title bar again to restore the previous view.

A pane is available only if the view it offers is relevant for the source file type selected in the Objects pane. You will not see a Rules pane, for example, if a BMS file is selected in Objects. **Edit** menu choices and pane-specific menu choices have equivalent choices in the right-click menu.

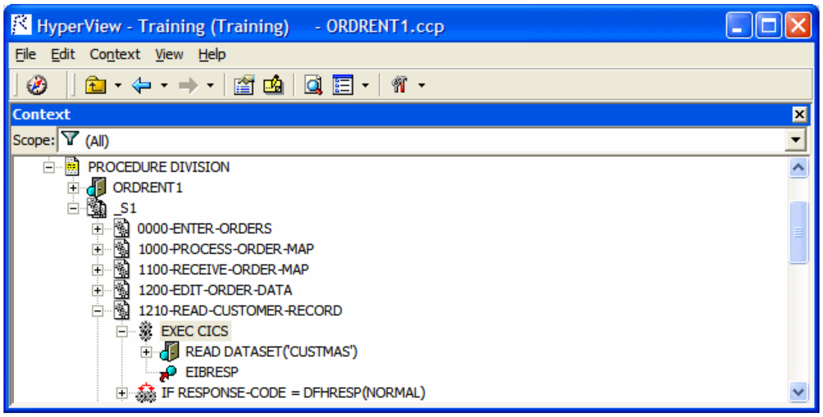
The HyperView window's state (size, location, configuration, and option settings) is saved across sessions. For guidance on moving and resizing HyperView panes, see *Getting Started* in the workbench documentation set.

In addition to the panes described below, HyperView provides access to the Activity Log and relevant options windows. To view these items, select the appropriate choice in the **View** menu. For usage information, see *Getting Started* in the workbench documentation 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*

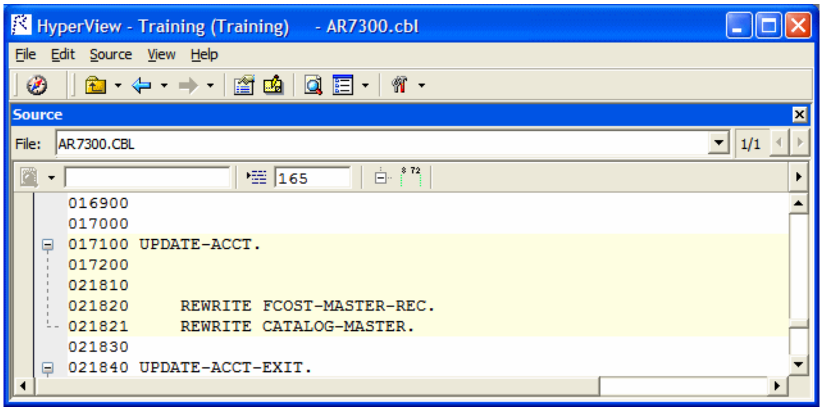


Scope: (All)

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*

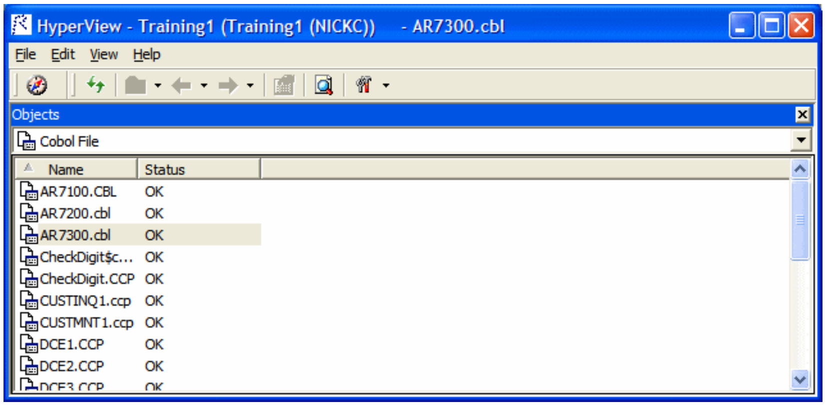

HyperView - Training (Training) - AR7300.cbl
File Edit Source View Help
Source
File: AR7300.CBL
1/1
165

```
016900
017000
017100 UPDATE-ACCT.
017200
021810
021820      REWRITE FCOST-MASTER-REC.
021821      REWRITE CATALOG-MASTER.
021830
021840 UPDATE-ACCT-EXIT.
```

Objects Pane

The Objects pane (Figure 1-7) lists the files in the project. Select a file to open it in the Source or Context panes. Choose the type of file to display in the drop-down under the title bar.

Figure 1-7 *Objects Pane*



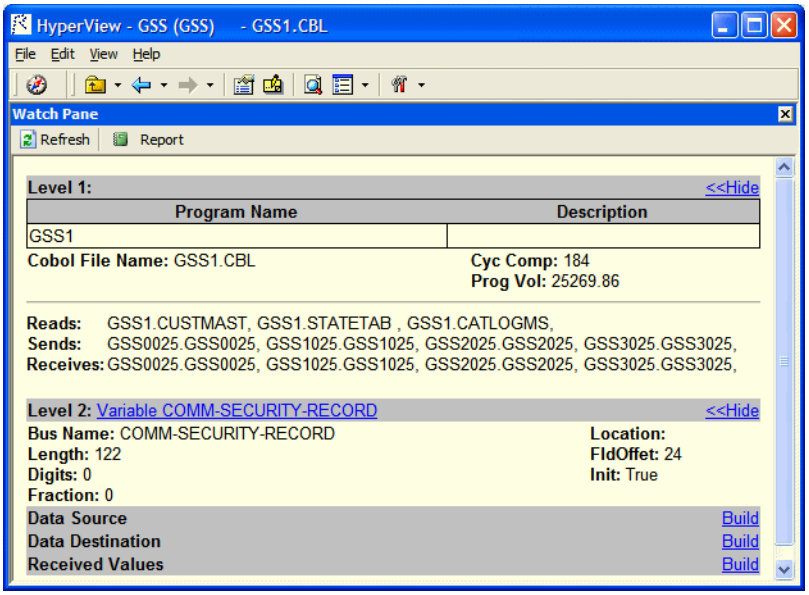
Cobol File

Name	Status
AR7100.CBL	OK
AR7200.cbl	OK
AR7300.cbl	OK
CheckDigit$c...	OK
CheckDigit.CCP	OK
CUSTINQ1.ccp	OK
CUSTMNT1.ccp	OK
DCE1.CCP	OK
DCE2.CCP	OK
DCE3.CCP	OK

Watch Pane

The Watch pane (Figure 1-8) displays current program context in summary form, showing, in a single view, object model context (Level 1) for the selected source file and construct model context (Level 2) for the selected construct.

Figure 1-8 *Watch Pane*



Level 1: <<Hide

Program Name	Description
GSS1	

Cobol File Name: GSS1.CBL
Cyc Comp: 184
Prog Vol: 25269.86

Reads: GSS1.CUSTMAST, GSS1.STATETAB , GSS1.CATLOGMS,
Sends: GSS0025.GSS0025, GSS1025.GSS1025, GSS2025.GSS2025, GSS3025.GSS3025,
Receives: GSS0025.GSS0025, GSS1025.GSS1025, GSS2025.GSS2025, GSS3025.GSS3025,

Level 2: Variable COMM-SECURITY-RECORD <<Hide

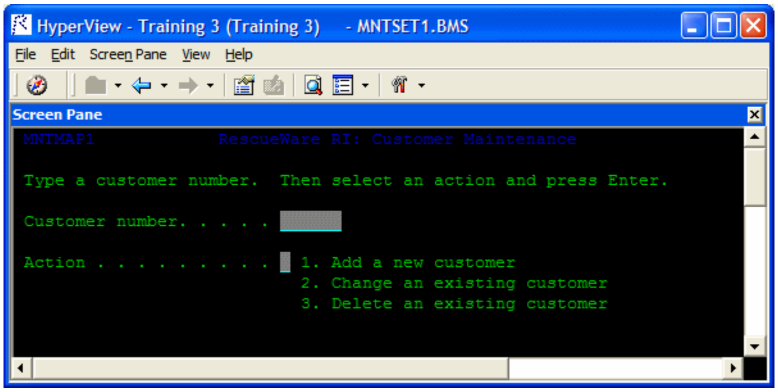
Bus Name: COMM-SECURITY-RECORD
Length: 122
Digits: 0
Fraction: 0
Location:
FldOffet: 24
Init: True

Data Source Build
Data Destination Build
Received Values Build

Screen Pane

The Screen pane displays the screen object defined in the selected map file (Figure 1-9). It complements the “bottom-up” analysis of application business processes in the Source and Context panes with a “top-down” view that lets you navigate quickly to data items of interest in your application.

Figure 1-9 *Screen Pane*

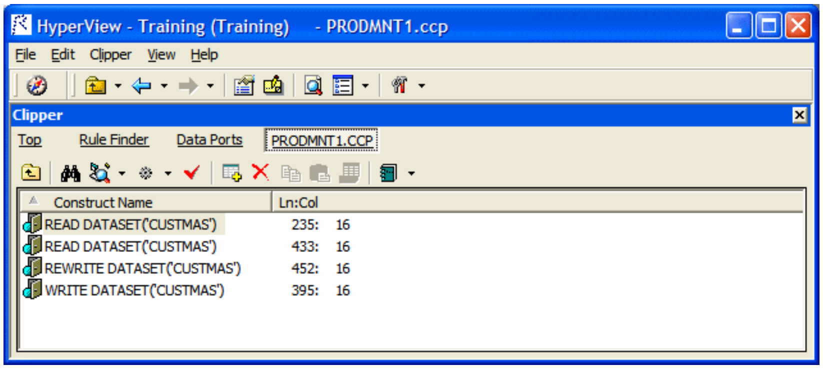

HyperView - Training 3 (Training 3) - MNTSET1.BMS
File Edit Screen Pane View Help
Screen Pane

```
MNTMAP1              RescueWare RI: Customer Maintenance

Type a customer number.  Then select an action and press Enter.

Customer number. . . . .

Action . . . . . . . . .   1. Add a new customer
                            2. Change an existing customer
                            3. Delete an existing customer
```

Clipper Pane

The Clipper pane (Figure 1-10) lets you perform batch searches of project source files, then stage the results as input for subsequent tasks: viewing source and context, impact analysis, business rule extraction, and the like.

Figure 1-10 *Clipper Pane*

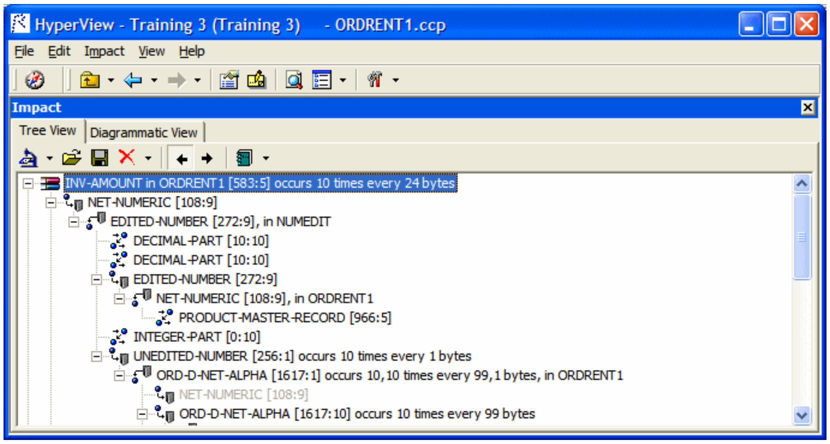

HyperView - Training (Training) - PRODMNT1.ccp
File Edit Clipper View Help
Clipper
Top
Rule Finder
Data Ports
PRODMNT1.CCP

Construct Name	Ln:Col
READ DATASET('CUSTMAS')	235: 16
READ DATASET('CUSTMAS')	433: 16
REWRITE DATASET('CUSTMAS')	452: 16
WRITE DATASET('CUSTMAS')	395: 16

Impact Pane

The Impact pane (Figure 1-11) 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 1-11 *Impact Pane (Hierarchy View)*

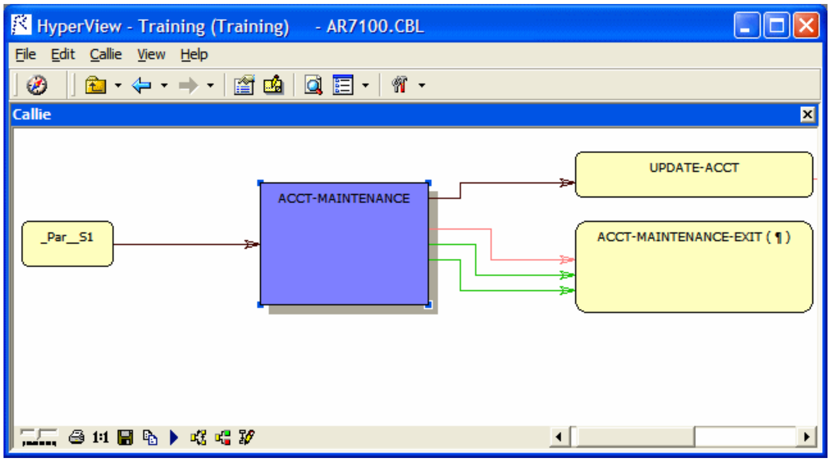


```
INV-AMOUNT in ORDRENT1 [583:5] occurs 10 times every 24 bytes
  NET-NUMERIC [108:9]
    EDITED-NUMBER [272:9], in NUMEDIT
      DECIMAL-PART [10:10]
      DECIMAL-PART [10:10]
      EDITED-NUMBER [272:9]
        NET-NUMERIC [108:9], in ORDRENT1
          PRODUCT-MASTER-RECORD [966:5]
      INTEGER-PART [0:10]
      UNEDITED-NUMBER [256:1] occurs 10 times every 1 bytes
        ORD-D-NET-ALPHA [1617:1] occurs 10,10 times every 99,1 bytes, in ORDRENT1
          NET-NUMERIC [108:9]
          ORD-D-NET-ALPHA [1617:10] occurs 10 times every 99 bytes
```

Callie Pane

The Callie pane (Figure 1-12) displays a diagram that shows the call flow for paragraphs in a Cobol program or for procedures in a PL/I or Natural program.

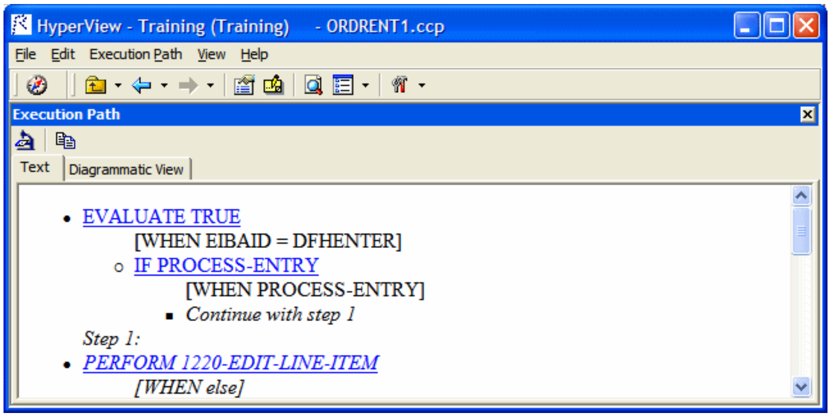
Figure 1-12 *Callie Pane*


HyperView - Training (Training) - AR7100.CBL
File Edit Callie View Help
Callie
_Par__S1
ACCT-MAINTENANCE
UPDATE-ACCT
ACCT-MAINTENANCE-EXIT (¶)
1:1

Execution Path Pane

The Execution Path pane (Figure 1-13) displays a hierarchical view and diagram of the conditions that determine the flow of control in a Cobol or PL/I program.

Figure 1-13 *Execution Path Pane (Hierarchy View)*



- EVALUATE TRUE
  [WHEN EIBAID = DFHENTER]
  - IF PROCESS-ENTRY
    [WHEN PROCESS-ENTRY]
    - *Continue with step 1*

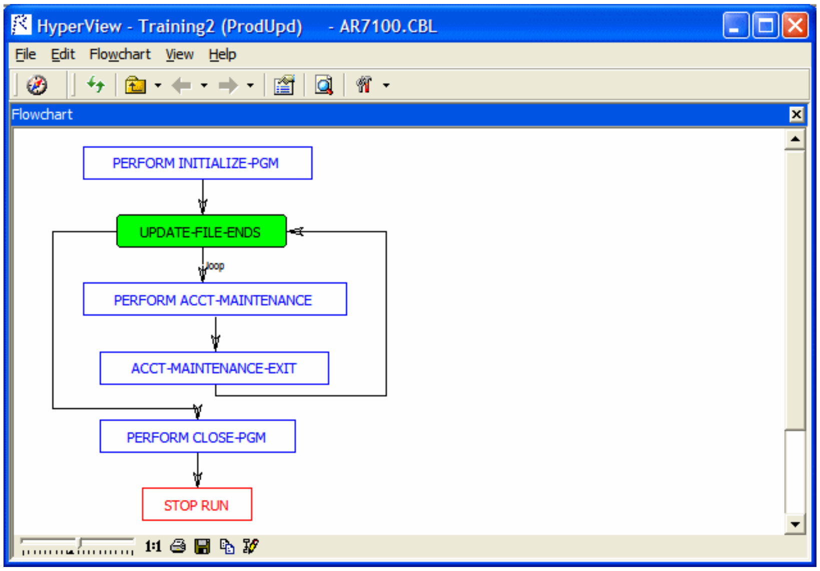
Step 1:

- *PERFORM 1220-EDIT-LINE-ITEM*
  [WHEN else]

Flowchart Pane

The Flowchart pane (Figure 1-14) displays a diagram of the flow of control between statements in a Cobol paragraph or PL/I procedure, or between steps in a job or JCL procedure.

Figure 1-14 *Flowchart Pane*

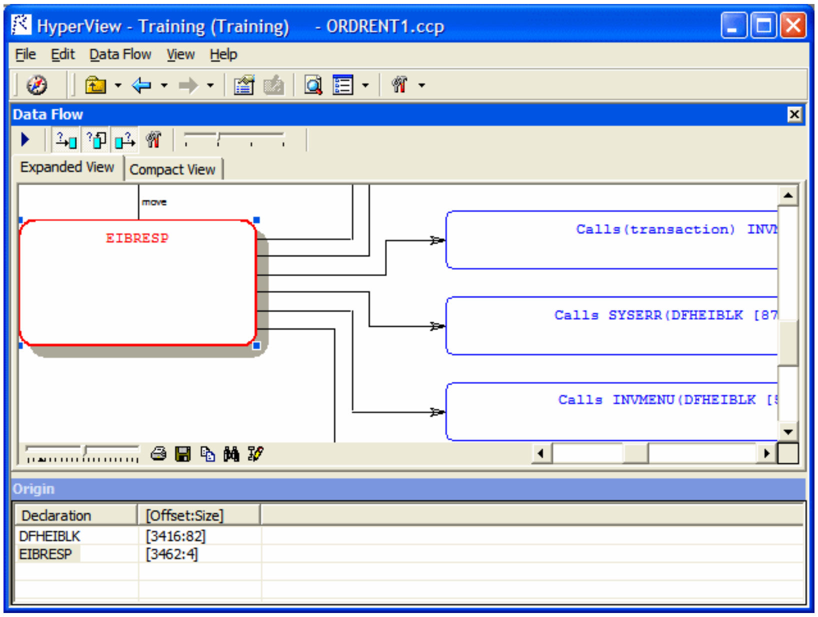


Data Flow Pane

The Data Flow pane (Figure 1-15) 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 pane is identical to that for the corresponding pane in the Global Data Flow tool. For more information, see *Analyzing Projects* in the workbench documentation set.

Figure 1-15 *Data Flow Pane*



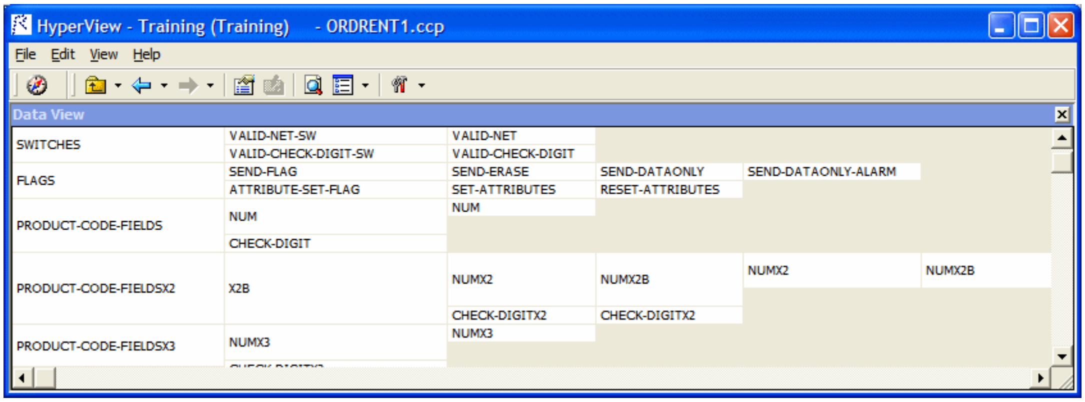
Declaration	[Offset:Size]
DFHEIBLK	[3416:82]
EIBRESP	[3462:4]

Data View Pane

The Data View pane (Figure 1-16) displays program variable structures, substructures, and fields in a hierarchical grid that represents memory usage graphically.

Note: Usage for the Data View pane is identical to that for the corresponding pane in the Global Data Flow tool. For more information, see *Analyzing Projects* in the workbench documentation set.

Figure 1-16 *Data View Pane*



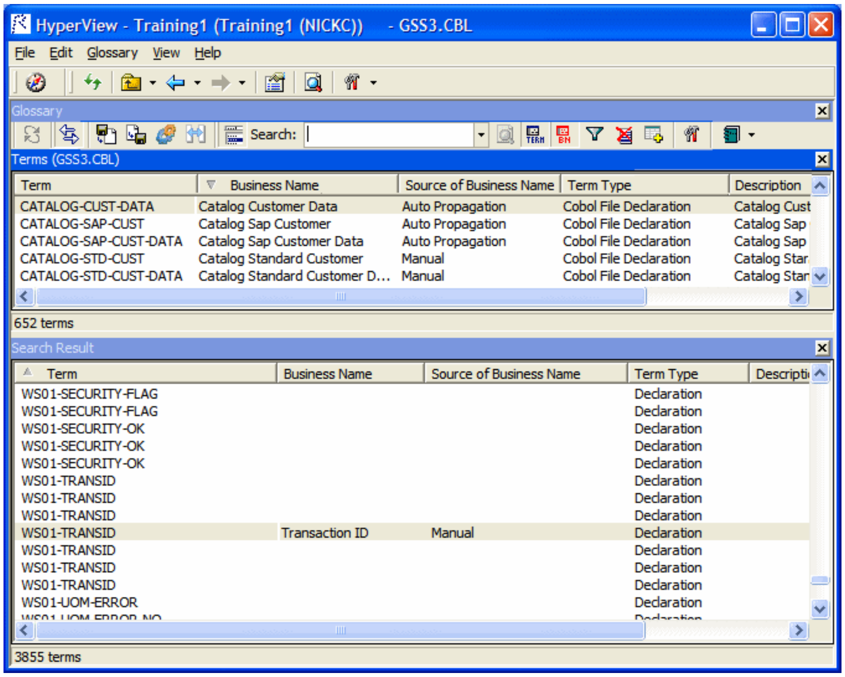
SWITCHES	VALID-NET-SW	VALID-NET			
	VALID-CHECK-DIGIT-SW	VALID-CHECK-DIGIT			
FLAGS	SEND-FLAG	SEND-ERASE	SEND-DATAONLY	SEND-DATAONLY-ALARM	
	ATTRIBUTE-SET-FLAG	SET-ATTRIBUTES	RESET-ATTRIBUTES		
PRODUCT-CODE-FIELDS	NUM	NUM			
	CHECK-DIGIT				
PRODUCT-CODE-FIELDSX2	X2B	NUMX2	NUMX2B	NUMX2	NUMX2B
		CHECK-DIGITX2	CHECK-DIGITX2		
PRODUCT-CODE-FIELDSX3	NUMX3	NUMX3			

Glossary Pane

The Glossary pane (Figure 1-17) creates a conveniently organized dictionary, or *glossary,* of meaningful names for items of interest in your application: data elements, paragraphs, programs, screens, tables, and the like. 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 1-18) to autoassign business names to business rules.

Figure 1-17 *Glossary Pane*



Term	Business Name	Source of Business Name	Term Type	Description
CATALOG-CUST-DATA	Catalog Customer Data	Auto Propagation	Cobol File Declaration	Catalog Cust
CATALOG-SAP-CUST	Catalog Sap Customer	Auto Propagation	Cobol File Declaration	Catalog Sap
CATALOG-SAP-CUST-DATA	Catalog Sap Customer Data	Auto Propagation	Cobol File Declaration	Catalog Sap
CATALOG-STD-CUST	Catalog Standard Customer	Manual	Cobol File Declaration	Catalog Star
CATALOG-STD-CUST-DATA	Catalog Standard Customer D...	Manual	Cobol File Declaration	Catalog Star

652 terms

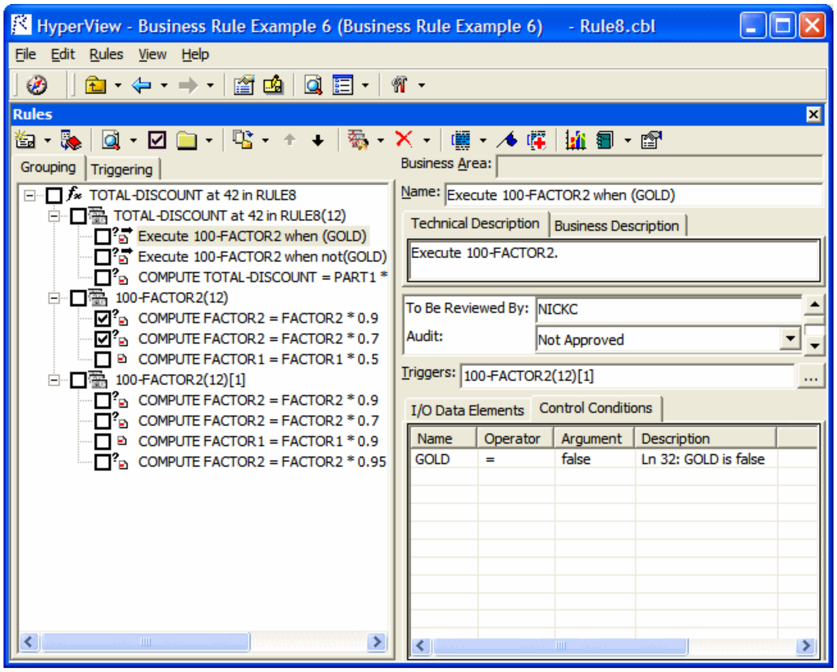
Term	Business Name	Source of Business Name	Term Type	Descripti
WS01-SECURITY-FLAG			Declaration	
WS01-SECURITY-FLAG			Declaration	
WS01-SECURITY-OK			Declaration	
WS01-SECURITY-OK			Declaration	
WS01-SECURITY-OK			Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID	Transaction ID	Manual	Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID			Declaration	
WS01-UOM-ERROR			Declaration	
WS01-UOM-ERROR-NO			Declaration	

3855 terms

Rules Pane

The Rules pane (Figure 1-18) 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 1-18 *Rules Pane*



- □ TOTAL-DISCOUNT at 42 in RULE8
  - □ TOTAL-DISCOUNT at 42 in RULE8(12)
    - □ Execute 100-FACTOR2 when (GOLD)
    - □ Execute 100-FACTOR2 when not(GOLD)
    - □ COMPUTE TOTAL-DISCOUNT = PART1 *
  - □ 100-FACTOR2(12)
    - ☑ COMPUTE FACTOR2 = FACTOR2 * 0.9
    - ☑ COMPUTE FACTOR2 = FACTOR2 * 0.7
    - □ COMPUTE FACTOR1 = FACTOR1 * 0.5
  - □ 100-FACTOR2(12)[1]
    - □ COMPUTE FACTOR2 = FACTOR2 * 0.9
    - □ COMPUTE FACTOR2 = FACTOR2 * 0.7
    - □ COMPUTE FACTOR1 = FACTOR1 * 0.9
    - □ COMPUTE FACTOR2 = FACTOR2 * 0.95

Name: Execute 100-FACTOR2 when (GOLD)

Execute 100-FACTOR2.

To Be Reviewed By: NICKC

Audit: Not Approved

Triggers: 100-FACTOR2(12)[1]

Name	Operator	Argument	Description
GOLD	=	false	Ln 32: GOLD is false

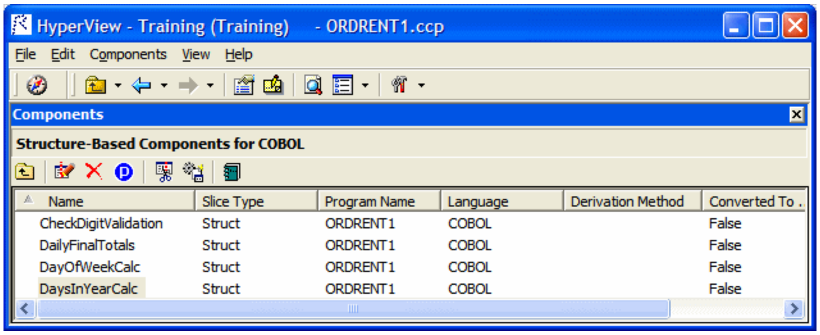
Components Pane

The Components pane (Figure 1-19) offers a variety of advanced algorithms for *slicing* logic from legacy source: all the code needed for a computation, for example, or to "specialize" a program based on the value of a variable. You can create a self-contained program, called a *component,* from the sliced code or simply generate a HyperView list of sliced constructs for further analysis. You can mark and colorize the list constructs in the HyperView Source pane.

Both the component generation and list functions are supported in the full version of the Components pane available to users of Application Architect. The list function only is supported in the restricted version of the Components pane, called Logic Analyzer, available to users of Application Analyzer.

The methods for slicing code are described in *Creating Components* in the workbench documentation set. For instructions on how to use Clipper to view listed results, see "Marking and Colorizing Slice Results" on page 4-10.

Figure 1-19 *Components Pane Views*



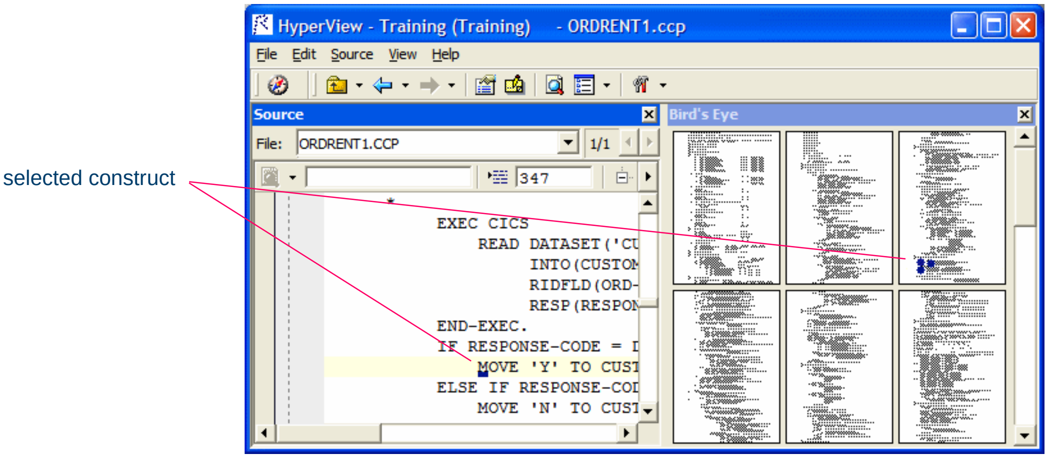
Structure-Based Components for COBOL

Name	Slice Type	Program Name	Language	Derivation Method	Converted To .
CheckDigitValidation	Struct	ORDRENT1	COBOL		False
DailyFinalTotals	Struct	ORDRENT1	COBOL		False
DayOfWeekCalc	Struct	ORDRENT1	COBOL		False
DaysInYearCalc	Struct	ORDRENT1	COBOL		False

Bird’s Eye Pane

The Bird’s Eye pane (Figure 1-20) lets you quickly identify the location of a code construct relative to the entire program and any included files.

Figure 1-20 *Bird’s Eye Pane*

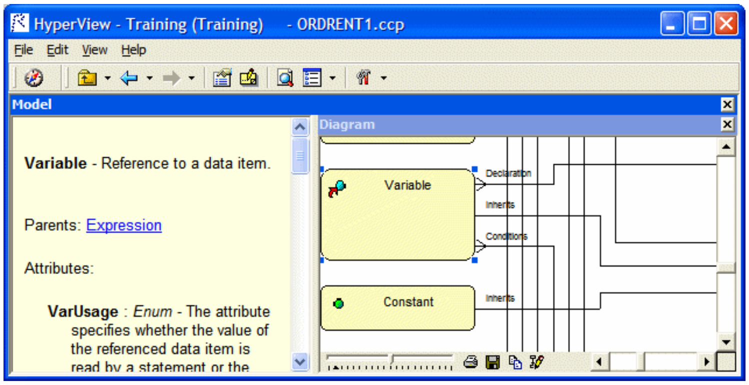


```
    *
        EXEC CICS
            READ DATASET('CU
                 INTO(CUSTOM
                 RIDFLD(ORD-
                 RESP(RESPON
        END-EXEC.
        IF RESPONSE-CODE = I
            MOVE 'Y' TO CUST
        ELSE IF RESPONSE-COD
            MOVE 'N' TO CUST
```

Model Reference Pane

The Model Reference pane (Figure 1-21) displays the HyperView metamodel in text and diagram form. Check the model reference embedded in the advanced search utility for the data types and definitions of construct attributes you use in your searches.

Figure 1-21 *Model Reference Pane*



Variable - Reference to a data item.

Parents: Expression

Attributes:

VarUsage : *Enum* - The attribute specifies whether the value of the referenced data item is read by a statement or the

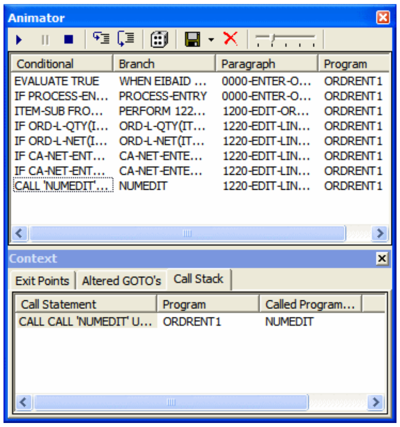
Using HyperView Auxiliary Windows

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

Animator Window

The Animator (Figure 1-22) lets you step through the code displayed in a HyperView pane. You can choose program branches yourself, or have the Animator choose them randomly.

Figure 1-22 *Code Animator Window*



Conditional	Branch	Paragraph	Program
EVALUATE TRUE	WHEN EIBAID ...	0000-ENTER-O...	ORDRENT1
IF PROCESS-EN...	PROCESS-ENTRY	0000-ENTER-O...	ORDRENT1
ITEM-SUB FRO...	PERFORM 122...	1200-EDIT-OR...	ORDRENT1
IF ORD-L-QTY(I...	ORD-L-QTY(IT...	1220-EDIT-LIN...	ORDRENT1
IF ORD-L-NET(I...	ORD-L-NET(IT...	1220-EDIT-LIN...	ORDRENT1
IF CA-NET-ENT...	CA-NET-ENTE...	1220-EDIT-LIN...	ORDRENT1
IF CA-NET-ENT...	CA-NET-ENTE...	1220-EDIT-LIN...	ORDRENT1
CALL 'NUMEDIT'...	NUMEDIT	1220-EDIT-LIN...	ORDRENT1

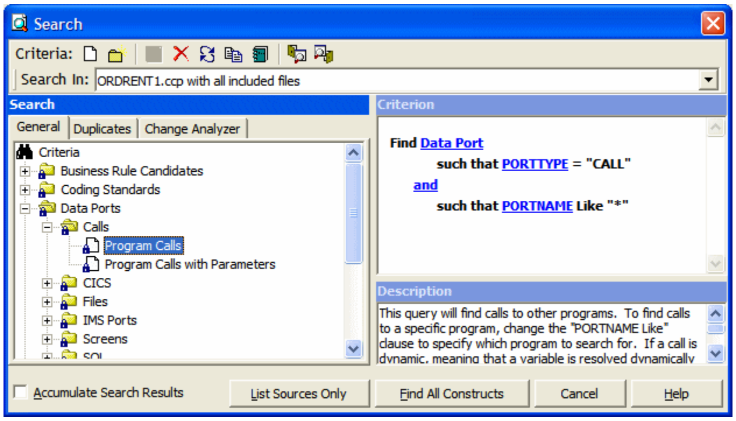
Call Statement	Program	Called Program...
CALL CALL 'NUMEDIT' U...	ORDRENT1	NUMEDIT

Advanced Search Window

HyperView provides simple and advanced search facilities. Use the Advanced Search pane (Figure 1-23) to create sophisticated filters for construct searches. A rich set of predefined filters let you search for dead code, hard-coded constants, nested conditionals, file and screen reads and writes, program calls, SQL ports, and more.

You typically use the *batch advanced search* facility, embedded in the Clipper pane, to find constructs in all files of a given type in your project. But you can also use *standalone advanced search* to find constructs in a single file.

Figure 1-23 *Advanced Search Window*



Find Data Port
such that PORTTYPE = "CALL"
and
such that PORTNAME Like "*"

This query will find calls to other programs. To find calls to a specific program, change the "PORTNAME Like" clause to specify which program to search for. If a call is dynamic, meaning that a variable is resolved dynamically

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

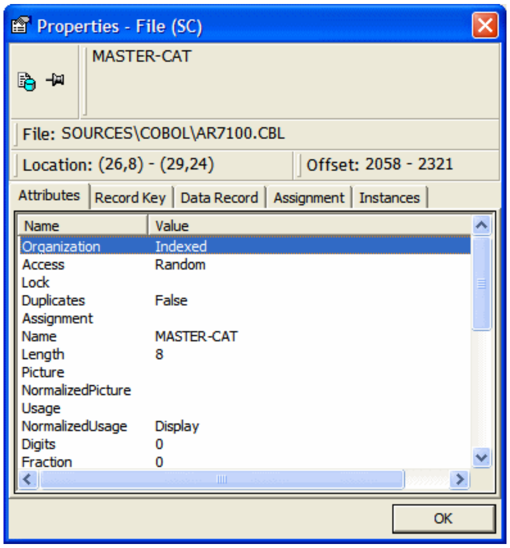
HyperView Properties windows offer convenient ways to navigate program code and analyze program constructs. The Properties window for a construct identifies its:

- *Attributes*: the construct's type, name, location, and characteristics.
- *Related constructs*: the constructs with which the construct interacts.

A Properties window is not modal, so you can leave it up throughout your HyperView session.

Opening a Properties Window In any HyperView pane except the Model Reference pane, select the construct whose properties you want to view and choose **Properties** in the right-click menu. The Properties window opens (Figure 1-24).

Figure 1-24 *Properties Window (Attributes Tab)*



Properties - File (SC)

MASTER-CAT

File: SOURCES\COBOL\AR7100.CBL

Location: (26,8) - (29,24) | Offset: 2058 - 2321

Name	Value
Organization	Indexed
Access	Random
Lock	
Duplicates	False
Assignment	
Name	MASTER-CAT
Length	8
Picture	
NormalizedPicture	
Usage	
NormalizedUsage	Display
Digits	0
Fraction	0

Viewing and Navigating to Related Constructs Click the tab for the type of related constructs you want to view. Click a related construct to navigate to it in the Source or Context panes.

Tip: Click the button in the Properties window to navigate back to the original construct.

Opening a Properties Window for a Related Construct Select a related construct in a Properties window and choose **Properties** in the right-click menu to open the Properties window for that construct.

Tip: Click the button to “pin” a Properties window, so that it remains open when you open another Properties window. Click the button again to unpin the window.

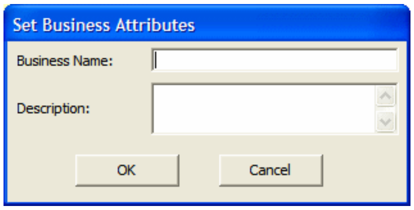
Assigning Business Names and Descriptions

In most HyperView panes, you can assign a business name and business description to an eligible object. For more information on business names, see Chapter 7, "Setting Up a Glossary."

To assign business names and descriptions:

1. Select an object in a HyperView pane and choose **Set Business Attributes** in the right-click menu. The Set Business Attributes dialog opens (Figure 1-25).
2. In the Set Business Attributes dialog, enter the business name for the object in the **Business Name** field and the business description in the **Description** field. Click **OK** to dismiss the dialog and return to the current pane.

Figure 1-25 *Set Business Attributes Dialog*



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 and program control flows, set up a glossary, and extract business rules.

Where to Look for Documentation on Other HyperView Tasks

Analyzing program data flows is described in *Analyzing Projects* in the documentation set. Methods for slicing code are described in *Creating Components*.

Understanding Program Context

2

The HyperView panes described in this chapter offer a set of complementary views of legacy source code. Together these views provide all the information you need to understand the context of a program item:

- 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.
- The Objects pane lists the files in the project. Select a file to open it in the Source or Context panes.
- The Watch pane displays program context in summary fashion, showing, in a single view, object model context for the selected source file and construct model context for the selected construct.
- The Screen pane displays the screen object defined in the selected map file, complementing the “bottom-up” analysis of application

business processes in the Source and Context panes with a “top-down” view that lets you navigate quickly to an item of interest.

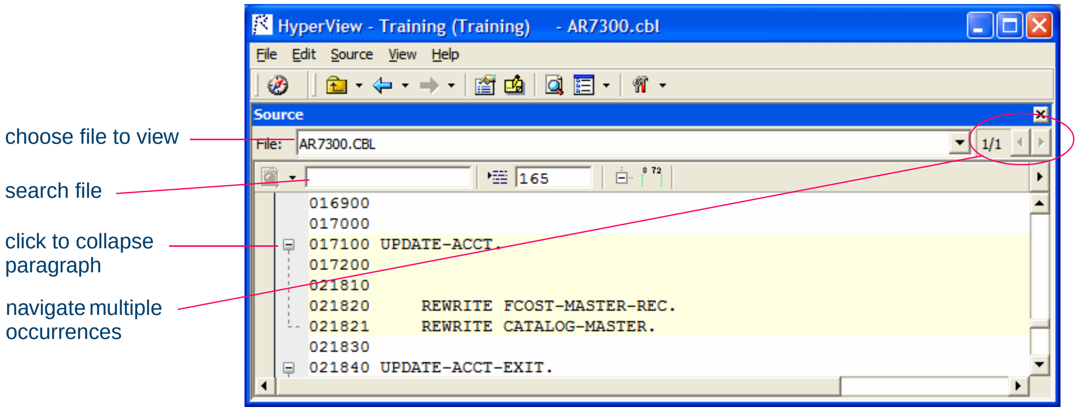
Using the Source Pane

The Source pane (Figure 2-1) displays view-only source for the selected file. Choose **Source** in the HyperView **View** menu to open the Source pane.

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. Use the Objects pane to select a different file, as described in “Using the Objects Pane” on page 2-9.

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

Figure 2-1 *Source Pane*


HyperView - Training (Training) - AR7300.cbl
File
Edit
Source
View
Help
Source
File: AR7300.CBL
1/1
165
choose file to view
search file
click to collapse paragraph
navigate multiple occurrences

```
016900
017000
017100 UPDATE-ACCT.
017200
021810
021820     REWRITE FCOST-MASTER-REC.
021821     REWRITE CATALOG-MASTER.
021830
021840 UPDATE-ACCT-EXIT.
```

Navigating and Searching for Source

Click in the Source pane to navigate to a construct. Enter a line number in the field next to the button and click the button or press Enter to navigate to the line. Tool tips show line numbers in the source file when you scroll the Source pane vertically.

Searching for Constructs

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.

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 opens a list of related constructs in the Properties window (Figure 1-24) 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 Properties window. For Properties window usage, see “Using the Properties Window” on page 1-25.

Tip: For a practical example of how you can use relationships to locate constructs of interest, see “Understanding HyperView Models” on page 1-1.

Navigating to Multiple Occurrences of a Construct

If an included 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) 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 2-8.

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 construct in the Source pane to select it. The selected construct is highlighted. 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.

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 that situation, you typically run an effort estimation report 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, select the file in the Source or Context pane. In the right-click menu, set the change magnitude 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 workbench documentation set.

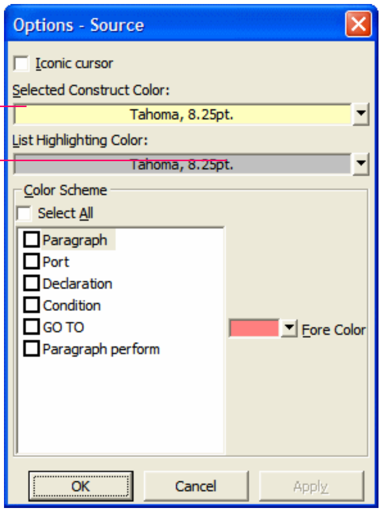
Setting Source Pane User Preferences

Use the Options window for the Source pane to specify the color used to highlight selected constructs, the color of the text for construct types, and so forth.

To set Source pane user preferences:

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

Figure 2-2 *Options Window*



check to colorize construct

select color

Options - Source

□ Iconic cursor

Selected Construct Color:

Tahoma, 8.25pt. ▾

List Highlighting Color:

Tahoma, 8.25pt. ▾

Color Scheme

□ Select All

□ Paragraph
□ Port
□ Declaration
□ Condition
□ GO TO
□ Paragraph perform

🟥 ▾ Fore Color

OK Cancel Apply

2. Select **Iconic cursor** to specify that the icon for a selected construct (for a data port, for example) replace the standard cursor.
3. In the **Selected Construct Color** field, click the adjacent ▾ button to edit the color used to highlight the selected construct. 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. In the **List Highlighting Color** field, click the adjacent ▾ button to edit the color used to highlight lists. 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.
5. In the Color Scheme pane, select the construct type whose text you want to display in color. The current color of the type (if any) is displayed in the **Fore Color** drop-down.

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.

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 1-20) 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 1-21) to view the HyperView metamodel in text and diagram form. Make sure to check the model reference embedded in the Advanced Search pane for the data types and definitions of construct attributes you use in your searches.

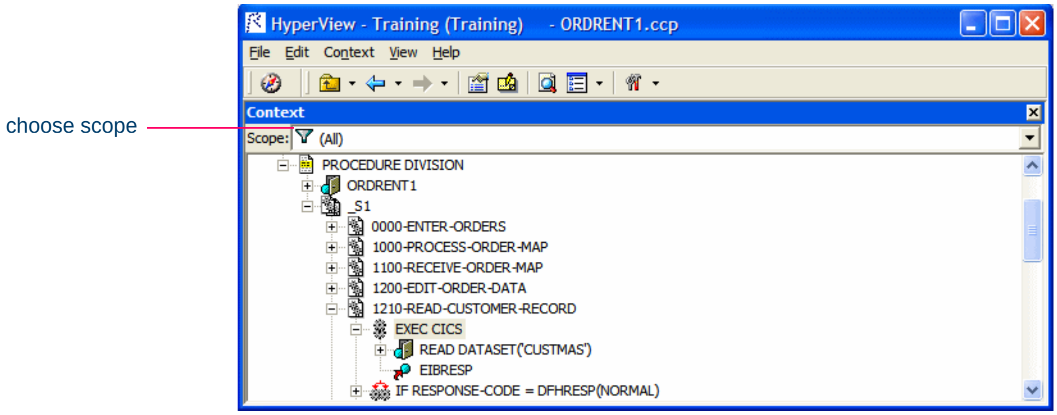
Using the Context Pane

The Context pane (Figure 2-3) displays the *parse tree* for the selected source file. Choose **Context** in the HyperView **View** menu to open the Context pane.

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. *Getting Started* in the workbench documentation set describes 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*



Scope: (All)

```
PROCEDURE DIVISION
  ORDRENT1
  _S1
    0000-ENTER-ORDERS
    1000-PROCESS-ORDER-MAP
    1100-RECEIVE-ORDER-MAP
    1200-EDIT-ORDER-DATA
    1210-READ-CUSTOMER-RECORD
      EXEC CICS
        READ DATASET('CUSTMAS')
        EIBRESP
      IF RESPONSE-CODE = DFHRESP(NORMAL)
```

The *scope* of the parse tree determines the constructs and relationships it displays: all constructs, only control statements, only declarations, and so forth. 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.

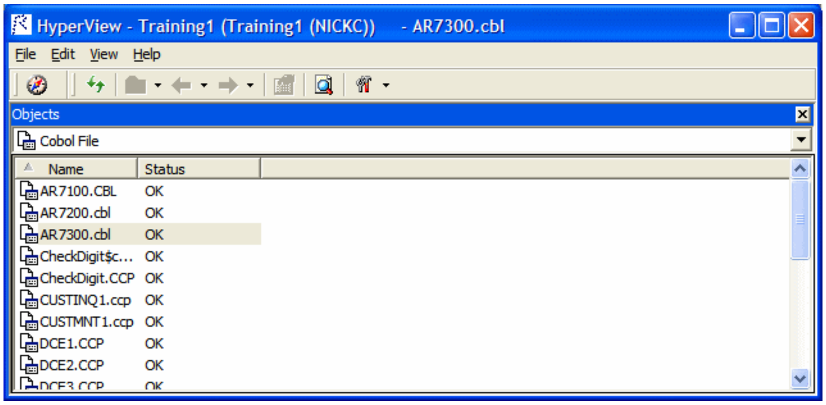
Perform other tasks as you do in the Source pane:

- Select a source file as described in "Using the Objects Pane" on page 2-9.
- Navigate to related constructs as described in "Navigating and Searching for Source" on page 2-2.
- Navigate to multiple occurrences of a construct as described in "Navigating to Multiple Occurrences of a Construct" on page 2-4.
- Set change magnitudes as described in "Specifying the Change Magnitude for a Source File" on page 2-5.

Using the Objects Pane

The Objects pane (Figure 2-4) lists the files in the project. Select a file to open it in the Source or Context panes. Choose the type of file to display in the drop-down under the title bar.

Figure 2-4 *Objects Pane*



Cobol File

Name	Status
AR7100.CBL	OK
AR7200.cbl	OK
AR7300.cbl	OK
CheckDigit$c...	OK
CheckDigit.CCP	OK
CUSTINQ1.ccp	OK
CUSTMNT1.ccp	OK
DCE1.CCP	OK
DCE2.CCP	OK
DCE3.CCP	OK

Using the Watch Pane

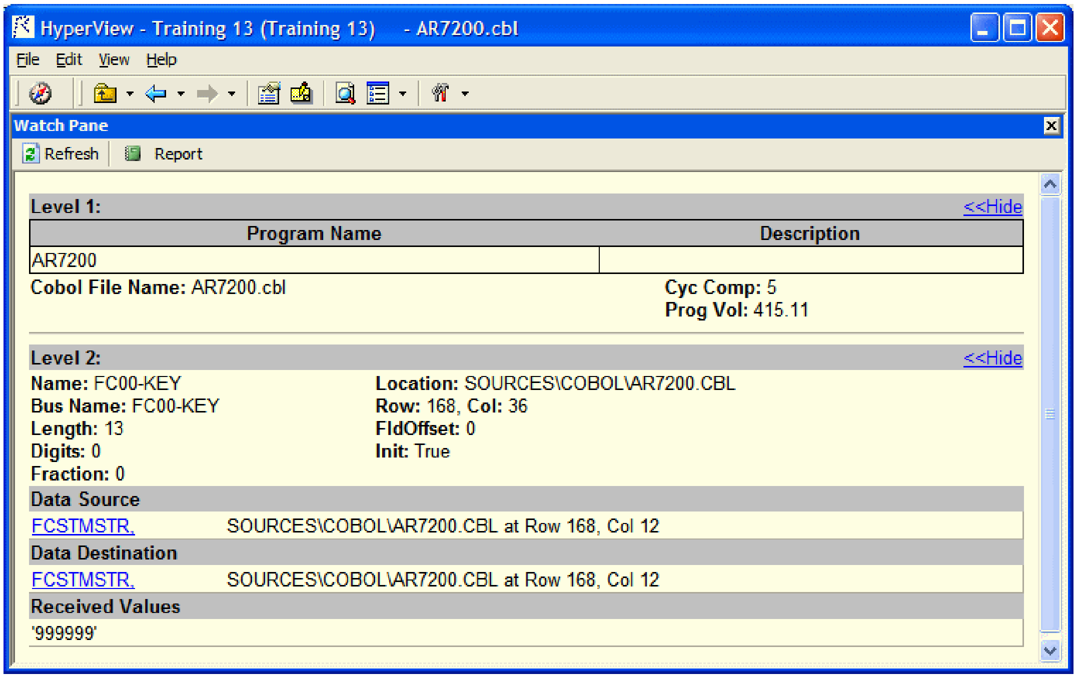
The Watch pane displays current program context in summary form, showing, in a single view, object model context (Level 1) for the selected source file and construct model context (Level 2) for the selected construct. Both the content of the Watch pane and its layout are customizable with the externalized XML queries and CSS stylesheets provided with the workbench. For more information, contact support services.

To view current context summaries in the Watch pane:

1. Select the source file you want to view, as described in "Using the Objects Pane" on page 2-9.
2. Click the construct you want to view in the Source or Context pane.
3. In the HyperView **View** menu, choose **Watch Pane**. The Watch pane opens (Figure 2-5). Click the 🔄 button to refresh the display.
   - The object model context (Level 1) displays the logical object extracted from the selected source file and its key metrics and relationships. For background on the object model, see *Getting Started* in the workbench documentation set.
   - The construct model context (Level 2) displays the selected construct, its business name, its location in source, and other characteristics. For a variable, click the **Build** link to view a summary of impact trace results:
     - Data Source displays the flow of data through each port in the trace to the originating port. Click the link for a port to navigate to it in the Source or Context pane.
     - Data Destination displays the flow of data through each port in the trace to the destination port. Click the link for a port to navigate to it in the Source or Context pane.
     - Received Values displays the value of the watched variable at each location in the impact trace. Click the link for a location to navigate to it in the Source or Context pane.
4. Click the **Hide** link to hide a summary. Click the **Show** link to restore a summary.

5 Click the ▤ button to save the contents of the Watch pane to an HTML file. The Save Watch View dialog opens, where you can specify the name and folder of the HTML file. The report opens when the save operation is complete.

Figure 2-5 *Watch Pane*



Level 1:

Program Name	Description
AR7200	

Cobol File Name: AR7200.cbl

Cyc Comp: 5
Prog Vol: 415.11

Level 2:

Name: FC00-KEY
Bus Name: FC00-KEY
Length: 13
Digits: 0
Fraction: 0

Location: SOURCES\COBOL\AR7200.CBL
Row: 168, **Col:** 36
FldOffset: 0
Init: True

Data Source

FCSTMSTR. SOURCES\COBOL\AR7200.CBL at Row 168, Col 12

Data Destination

FCSTMSTR. SOURCES\COBOL\AR7200.CBL at Row 168, Col 12

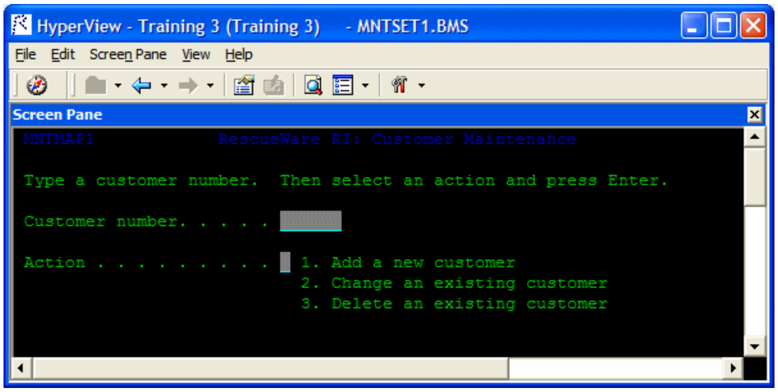
Received Values

'999999'

Using the Screen Pane

The Screen pane displays the screen object defined in the selected map file (Figure 2-6). It complements the “bottom-up” analysis of application business processes in the Source and Context panes with a “top-down” view that lets you navigate quickly to data items of interest in your application.

Figure 2-6 *Screen Pane*



HyperView - Training 3 (Training 3) - MNTSET1.BMS
File Edit Screen Pane View Help
Screen Pane

```
MNTMAP1              RescueWare RI: Customer Maintenance

Type a customer number.  Then select an action and press Enter.

Customer number. . . . .

Action . . . . . . . . .   1. Add a new customer
                           2. Change an existing customer
                           3. Delete an existing customer
```

Select the map file you want to view as described in "Using the Objects Pane" on page 2-9. Click on a screen in the Source or Context pane to view its rendering in the Screen pane.

In the Screen pane, click on the data entry area for a field to navigate to it in the Source or Context pane. Select the data entry area and choose **Data Items** in the right-click menu to view the declarations for the field in the Properties window (Figure 1-24). Click a declaration to navigate to it in the Source or Context panes. For Properties window usage, see "Using the Properties Window" on page 1-25.

Set change magnitudes as described in "Specifying the Change Magnitude for a Source File" on page 2-5. Choose **Fit to Window** in the **Screen** menu to fit the screen to the Screen pane. Choose **100%** in the **Screen** menu to restore the screen to its actual size.

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

3

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, hard-coded constants, nested conditionals, file and screen reads and writes, program calls, SQL ports, and more.

You typically use the batch advanced search facility, embedded in the Clipper pane, to find constructs in all files of a given type in your project. But you can also use standalone advanced search to find constructs in a single file.

Note: Certain performance-intensive searches, with criteria such as Contains or Is Nested In, may be restricted to the master user.

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.

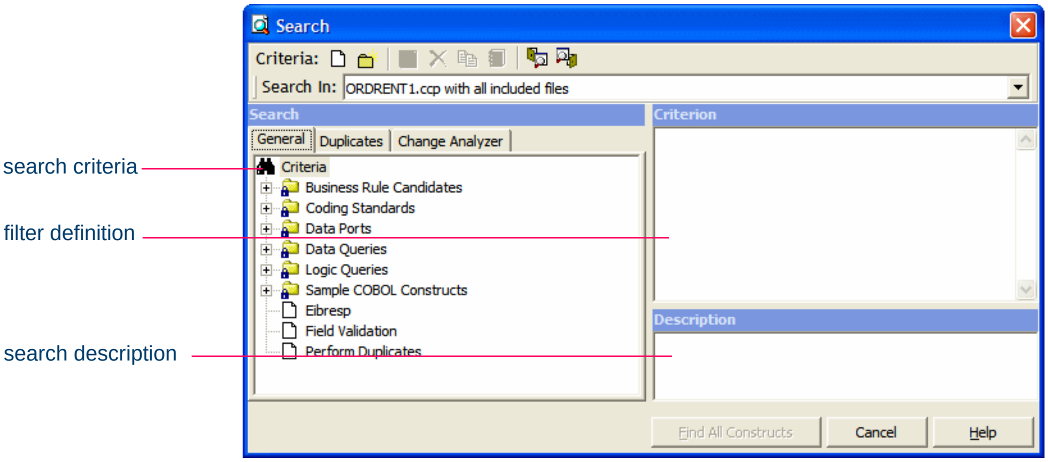
As an example of how the facility works, let's look at how you might define a search filter in standalone mode for the conditions a program uses to validate the variable EIBRESP. For batch search usage, see Chapter 4, "Staging Program Analysis with Clipper."

To define a search filter:

1 In the Objects pane, select the file you want to search. For more information, see "Using the Objects Pane" on page 2-9.

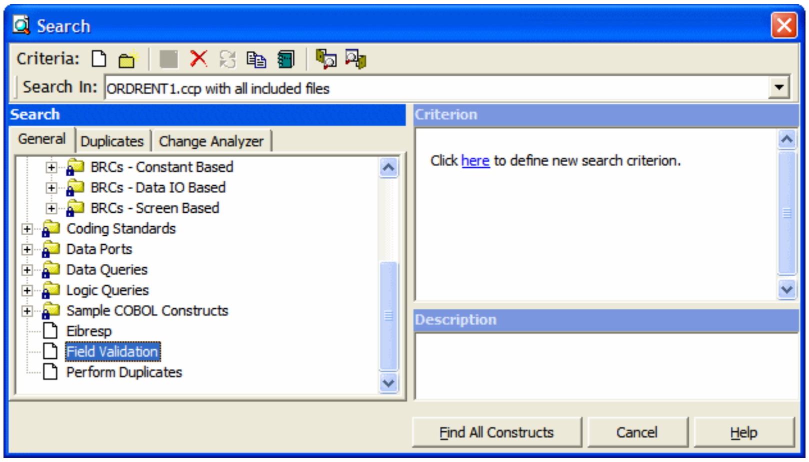
2 In the HyperView **Edit** menu, choose **Find**. The Search window opens. Click the **General** tab (Figure 3-1).

Figure 3-1 *Search Window (General Tab)*



3 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 3)*



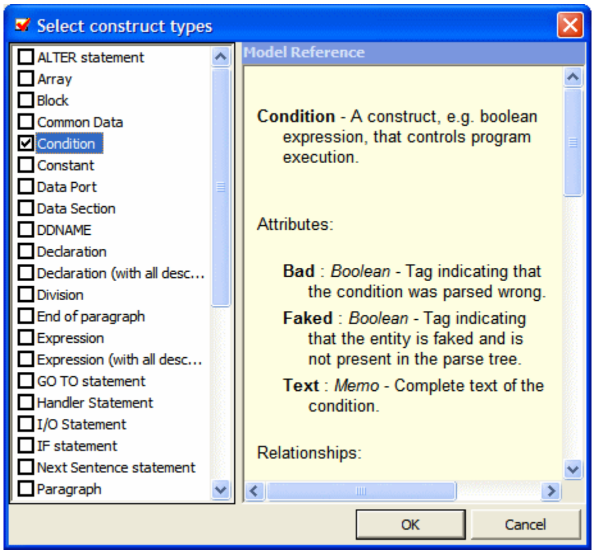
Search In: ORDRENT1.ccp with all included files

- BRCs - Constant Based
- BRCs - Data IO Based
- BRCs - Screen Based
- Coding Standards
- Data Ports
- Data Queries
- Logic Queries
- Sample COBOL Constructs
- Eibresp
- Field Validation
- Perform Duplicates

Click here to define new search criterion.

4 Click the **here** link in the righthand pane of the Search window. The Select Construct Types window opens (Figure 3-3).

Figure 3-3 *Select Construct Types Window*



☐ ALTER statement
☐ Array
☐ Block
☐ Common Data
☑ Condition
☐ Constant
☐ Data Port
☐ Data Section
☐ DDNAME
☐ Declaration
☐ Declaration (with all desc…
☐ Division
☐ End of paragraph
☐ Expression
☐ Expression (with all desc…
☐ GO TO statement
☐ Handler Statement
☐ I/O Statement
☐ IF statement
☐ Next Sentence statement
☐ Paragraph

Condition - A construct, e.g. boolean expression, that controls program execution.

Attributes:

Bad : *Boolean* - Tag indicating that the condition was parsed wrong.

Faked : *Boolean* - Tag indicating that the entity is faked and is not present in the parse tree.

Text : *Memo* - Complete text of the condition.

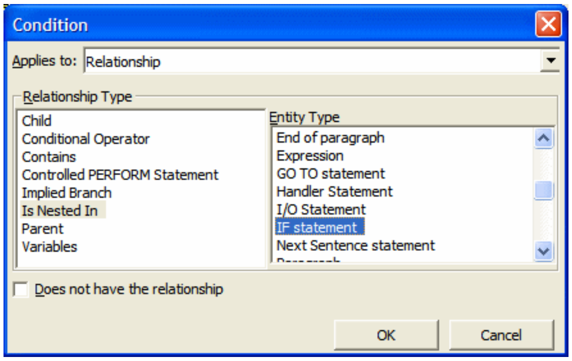
Relationships:

5 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
```

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

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



Condition

Applies to: Relationship

Relationship Type

- Child
- Conditional Operator
- Contains
- Controlled PERFORM Statement
- Implied Branch
- Is Nested In
- Parent
- Variables

Entity Type

- End of paragraph
- Expression
- GO TO statement
- Handler Statement
- I/O Statement
- IF statement
- Next Sentence statement

☐ Does not have the relationship

7 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
```

8 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
- 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
    Perform
```

9 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
Perform such that Caption Like "*ERROR*"
```

Note: You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA). Consult the Model Reference pane for attribute values.

10 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
Perform such that Caption Like "*ERROR*"
and
    which contains any Variable
```

11 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
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.

12. Enter a description of the filter definition in the Description pane.
13. Click the 💾 button on the tool bar to save the filter definition. To execute the search, follow the instructions in “Executing Advanced Searches” on page 3-8.

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: Use the Batch Duplicate Finder (BDF) to perform more sophisticated comparisons of similarly structured paragraphs, and to compare PL/I or RPG procedures. For more information, see Appendix A, “Using the Batch Duplicate Finder.”

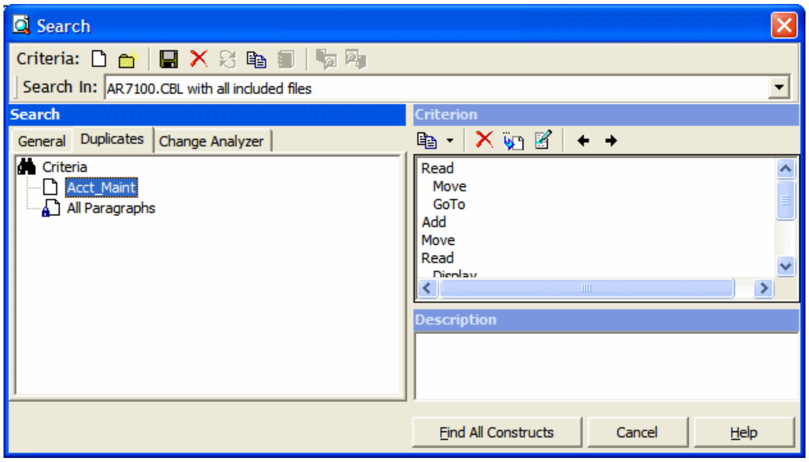
To search for similarly structured code:

1 In the HyperView **Edit** menu, choose **Find**. The Search window opens. Click the **Duplicates** tab (Figure 3-5).

2 Create a new search criterion, as described in “Creating a Search Criterion” on page 3-9.

3 Right-click in the Criterion pane and choose **Use Paragraph:*Paragraph*** in the pop-up menu. HyperView displays the signature of the selected paragraph.

Figure 3-5 *Search Window (Duplicates Tab)*



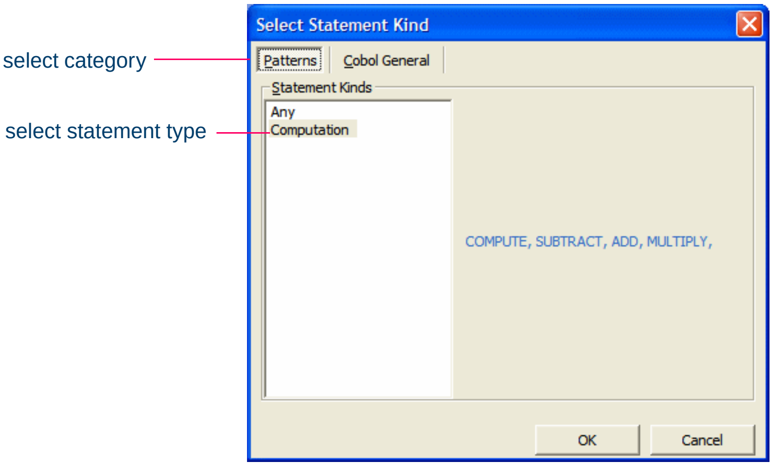
Search In: AR7100.CBL with all included files

```
Read
  Move
  GoTo
Add
Move
Read
  Display
```

4 To insert a statement in the signature, select a line in the signature and choose **Insert:Before/After** in the right-click menu. HyperView inserts an ellipsis (...) at the specified location. Select the ellipsis and choose **Change** in the right-click menu. The Select Statement Kind window opens (Figure 3-6).

Tip: Replace a statement in the signature by selecting the statement and choosing **Change** in the right-click menu.

Figure 3-6 *Select Statement Kind Window (Duplicates Tab)*



5 Click the tab for the category of statement you want to insert. Choose the Patterns tab if you want to insert a pattern rather than a specific statement. In the Statement Kinds pane, choose the pattern or statement you want to insert and click **OK**. HyperView replaces the ellipsis with the selected pattern or statement.

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.

6 Click the 💾 button on the tool bar to save the search criterion. To execute the search, follow the instructions in "Executing Advanced Searches" on page 3-8.

Executing Advanced Searches

You can execute an advanced search in standalone mode, against the file or construct selected in the Source or Context pane, or in batch mode, in the embedded search pane in Clipper. 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 workbench documentation set.

To execute an advanced search in standalone mode:

1. In the HyperView **Edit** menu, choose **Find**. The Search window opens. Click the tab for the type of search you want to perform.
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 **Find All Constructs**. HyperView displays a dialog box with the results of the search. Click **OK**. The dialog and Search window are dismissed. HyperView displays the results in a list named Last Search Results in the Internal category in Clipper. For list usage, see "Using the Clipper Pane" on page 4-5.

Working with Advanced Search Criteria

Define advanced search criteria as described in "Understanding Advanced Searches" on page 3-1 and "Searching for Similarly Structured Code" on page 3-6. This section describes common tasks you perform with advanced search criteria.

Creating a Search Criterion To create a search criterion, select the folder for the new criterion in the General tab, then click the 🗋 button on the tool bar. The New Criterion dialog opens. Enter the name of the criterion in the text field and click **OK**.

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. The criterion is available to every project in the workspace.

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.

Creating a Folder 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.

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.

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, impact analysis, and similar tasks in Clipper.

Staging Program Analysis with Clipper

4

Use Clipper to perform a batch search of project source files, then stage the results as input for subsequent tasks: viewing source and context, colorizing sliced code, extracting business rules, 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.

Clipper searches are executed against the current project. You can also execute searches against the results of a previous search.

Note: Certain performance-intensive Clipper searches, with criteria such as Contains or Is Nested In, may be restricted to the master user.

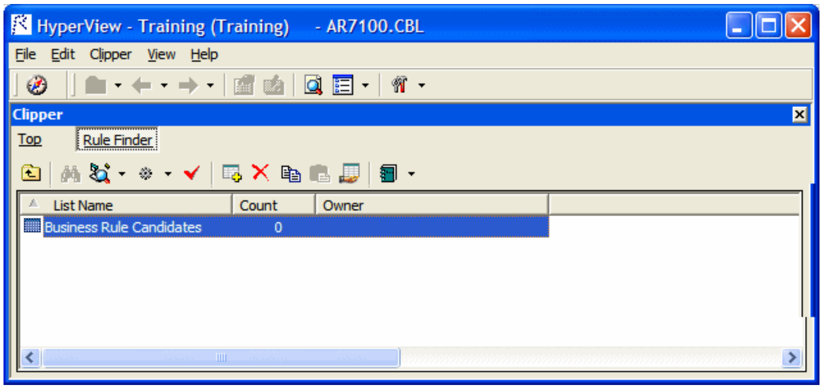
Getting Started in Clipper

Let's look at a sample use of Clipper that shows you how to create business rules for segments of code that write to a file. Working through 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 pane opens. Click the **Top** view link.
2. The Top view opens. In the Category Name column, double-click Rule Finder. The Rule Finder view opens (Figure 4-1).

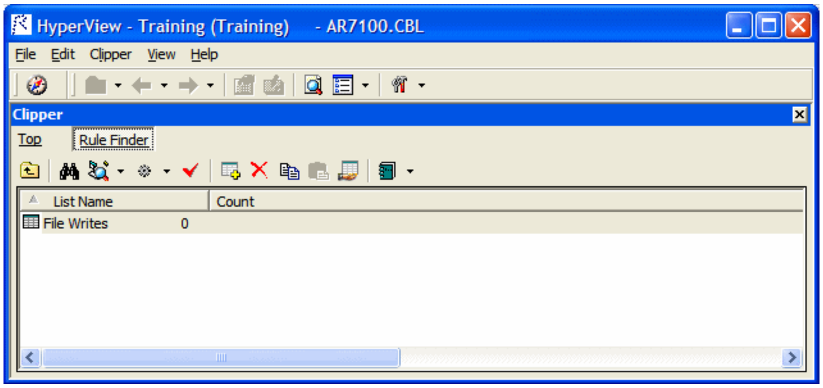
Figure 4-1 *Clipper Pane (Rule Finder View)*


HyperView - Training (Training) - AR7100.CBL
File Edit Clipper View Help
Clipper
Top
Rule Finder

List Name	Count	Owner
Business Rule Candidates	0	

3. Click the button on the tool bar. The New List dialog opens. Enter “File Writes” and click **OK**. An empty list named File Writes is added to the Rule Finder view.

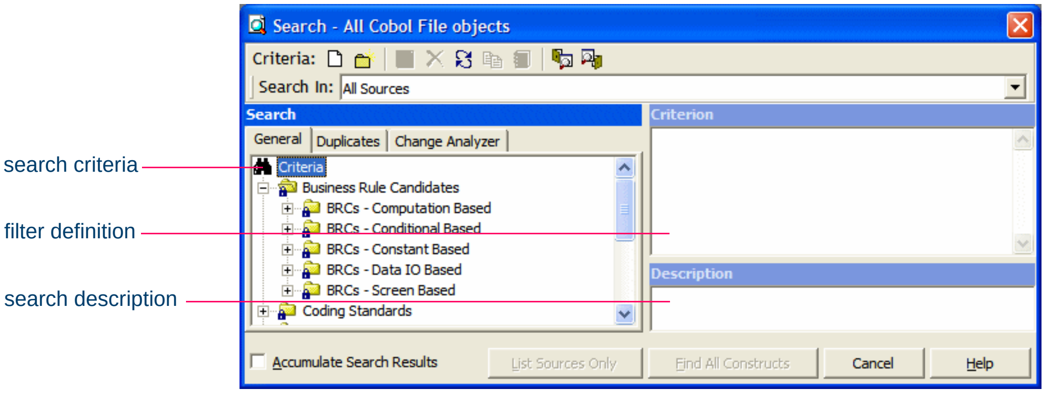
Figure 4-2 *Clipper Pane (Rule Finder View with New List)*


HyperView - Training (Training) - AR7100.CBL
File Edit Clipper View Help
Clipper
Top
Rule Finder

List Name	Count
File Writes	0

4 Select the File Writes list, then click the ▾ button next to the 🔍 button on the tool bar. In the drop-down, choose the type of object you want to execute the search against, Cobol File in our example. The Search window opens (Figure 4-3).

Figure 4-3 *Search Window for Clipper*

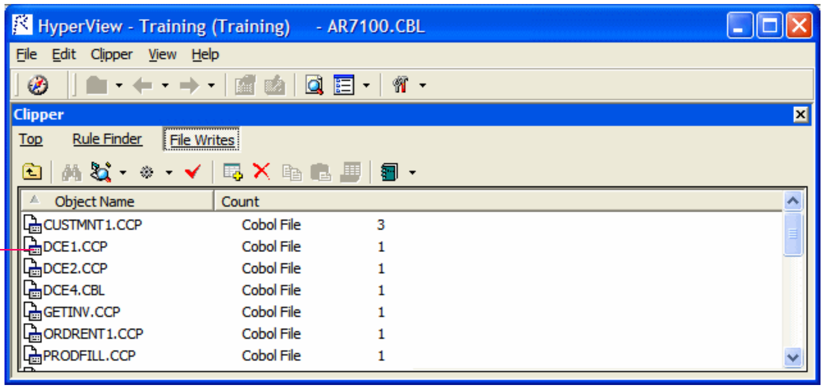


5 In the **Search In** drop-down, select **All Sources**.

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 a Clipper view named for the File Writes list (Figure 4-4).

Note: You can modify the predefined criterion or create your own criterion as described in "Working with Advanced Search Criteria" on page 3-9. For Clipper search execution, see "Executing Clipper Searches" on page 4-11.

Figure 4-4 *Clipper Pane (File Writes View)*


HyperView - Training (Training) - AR7100.CBL
File Edit Clipper View Help
Clipper
Top Rule Finder File Writes

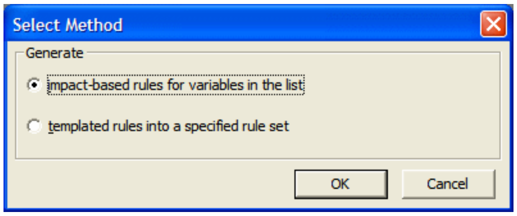
Object Name	Count	
CUSTMNT1.CCP	Cobol File	3
DCE1.CCP	Cobol File	1
DCE2.CCP	Cobol File	1
DCE4.CBL	Cobol File	1
GETINV.CCP	Cobol File	1
ORDRENT1.CCP	Cobol File	1
PRODFILL.CCP	Cobol File	1

double-click to view results list

Tip: To add a construct manually to the list for a source file, select the source file in the File Writes view, select the construct in the Source or Context panes, then click the ⊞ button on the tool bar.

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 4-5).

Figure 4-5 *Select Method Dialog*



● impact-based rules for variables in the list

○ templated rules into a specified rule set

8 In the Select Method dialog, select **impact-based rules for variables in the list** or **templated rules into a specified rule set**, then follow the instructions in “Generating Business Rules” on page 4-7.

Using the Clipper Pane

HyperView lists offer convenient ways to navigate program code and record the results of program analyses. Use the hierarchy of views in the Clipper pane to view HyperView lists and perform other list-related tasks:

- The *Top* view displays the categories of lists you can create for the selected source file type.
- The *Category* view displays the lists in the selected category.
- The *List* view displays source files with non-zero counts in the list.
- The *File* view displays each construct in the list for the selected source file. Click a construct in the list to navigate to it in the HyperView Source and Context panes.

Each view gives you access to the next-level view in the hierarchy. Click the label for a view to navigate to it. Double-click an item in the view to drill down to the next 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.

Working with Categories

The Top view displays the categories of lists you can generate for a source file type. Double-click a category to navigate to it in the Category view. Double-click a list in the Category view to navigate to it in the List view.

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 file type.

Use the General and Internal categories for unenumerated tasks. HyperView puts the results of standalone advanced searches in a predefined list called Last Search Results in the Internal category.

Working with Lists

The List view displays the lists in a category. Double-click a list in the Category view to navigate to it in the List view. Double-click a source file in the List view to navigate to its list of constructs in the File view. HyperView supplies default lists for each category: Calls and Ports, for example, for the Event Injector category.

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

Deleting Lists In the Category view, select a list and click the ✕ button on the tool bar to delete the list. You are prompted to confirm the deletion. Click **OK**.

Copying and Pasting Lists In the Category view, select a list and click the button on the tool bar to copy the list. In the Top view, select the category you want to copy the list into and click the button. The Copy List dialog opens, where you can enter the name of the new list. You can copy a list into the same category.

Sharing Lists In the Category view, select a list and click the button on the tool bar to share the list with other users in a multiuser environment. The list is marked with a similar icon to indicate that it is shared. Click the button again to turn sharing off.

Generating Lists Automatically In the Category view, select a list, then 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 4-11.

Generating a List of Similar Constructs In the Source or Context panes, select a construct, then select a list in the Category view. Click the button on the tool bar. The New Criterion dialog opens. Enter a criterion name or accept the default.

The Search window opens with a predefined search criterion that will generate a list of constructs similar to the selected construct. The new criterion is added to the General tab of the advanced search facility. Edit the criterion if necessary, then execute the search as described in "Executing Clipper Searches" on page 4-11.

Adding Constructs to a List Manually In the List 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 constructs for the selected file.

Deleting Constructs from a List In the File view, select a construct and click the ✗ button on the tool bar to delete the construct from the selected list. You are prompted to confirm the deletion. Click **OK**.

Marking and Colorizing Listed Constructs In the Category view, select a list and click the ✓ button on the tool bar to place a check mark in the HyperView Source pane next to each listed construct and colorize the construct. To mark and colorize listed constructs in a single file, select the file in the List view and click the ✓ button. To mark and colorize a single construct, select it in the File view and click the ✓ button. Click the ✓ button again to turn off marking and colorizing.

Generating Business Rules

Clipper offers two methods for generating business rules for multiple constructs. You can:

- Autodetect each code segment that impacts a listed construct. Clipper creates a rule for each autodetected segment. When you choose this option, you edit rule attributes in batch mode in the Rules pane.
- Generate rules for listed constructs in a specified rule set. When you choose this option, you batch process rule attributes in the generation step.

Tip: As long as the Rules pane is open, you can generate a rule for a single construct by selecting the construct in the File view and choosing **Create Rule** in the **Rules** menu. Follow the instructions for creating a rule manually in "Extracting Business Rules Manually" on page 8-7.

Autodetecting Business Rules in Clipper

The backward autodetect method traces the flow of data from a variable to its source, assigning each statement in the flow to a new rule. Follow the instructions below to backward-autodetect code segments that impact constructs in a Clipper list. Clipper creates a rule for each autodetected segment. The constructs must be variables. For more information on the autodetection method, see "Autodetecting Business Rules" on page 8-9.

To autodetect business rules in Clipper:

1 To autodetect rules for listed constructs, select a list in the Rule Finder category, 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 4-5).

2 In the Select Method dialog, select **impact-based rules for variables in the list** and click **OK**.

3 If the autodetect trace detects conditional logic, the Autodetected Condition Values dialog (Figure 8-4) prompts you to "prune out" logic branches not of interest to you, as described in "Pruning Autodetect Results" on page 8-12. If you set a condition to True, for example, the False branch (assuming one exists) is not explored by the trace. When you are satisfied with your choices in the Autodetected Condition Values dialog, click **OK**.

4 The autodetect method creates a rule for each code segment that impacts a listed construct. HyperView displays an informational message that tells you how many rules it created. Click **OK**.

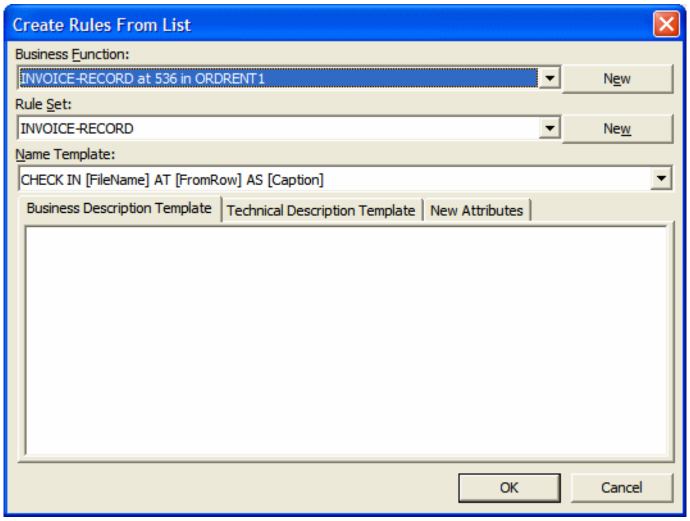
Generating Rules in a Specified Rule Set

Follow the instructions below to generate rules in a specified rule set with predefined attributes. Clipper creates a rule for each construct in the Clipper list. For more information on rule sets, see “Understanding Business Rules” on page 8-2.

To generate business rules in a specified rule set:

1. To generate rules in a specified rule set, select a list in the Rule Finder Category, 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 4-5).
2. In the Select Method dialog, select **templated rules into a specified rule set** and click **OK**. The Create Rule Set window opens.
3. In the **Business Function** drop-down, select the business function for the rule set. The current business function is displayed. To create a new business function, click **New**, then follow the instructions in “Creating Business Functions” on page 8-19.
4. 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).

Figure 4-6 *Create Rules From List Window*



Create Rules From List

Business Function: INVOICE-RECORD at 536 in ORDRENT1

Rule Set: INVOICE-RECORD

Name Template: CHECK IN [FileName] AT [FromRow] AS [Caption]

5 In the **Name Template** combo box, select a template for the rule name attribute from the drop-down, or enter a template manually. Templates derive from the list in the Rule Defaults tab in the Project options for Business Rule Manager. If you enter a template manually in Clipper, the new template is added to the list. For instructions on how to create a template manually, see “Setting Options on the Rule Defaults Tab” on page 8-41.

6 On the Business Description Template and Technical Description Template tabs, enter templates for the description and business description attributes, respectively. See “Setting Options on the Rule Defaults Tab” on page 8-41 for an example of the output you can expect.

7 On the New Attributes tab, specify the value for each attribute you want to set in the selected rules.

8 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 rules it created. Click **OK**.

Generating an Impact Subtree Report

To generate an impact subtree report for all the constructs in a list, select the list in the Impact Report category, click the ▾ button next to the ⚙ button on the tool bar, and choose **Create Impact SubTree Report** in the drop-down menu. A Save As dialog opens, where you can specify the name, location, and file type of the report. For more information, see “Exporting Trace Reports from the Impact Pane” on page 5-14.

Marking and Colorizing Slice Results

The Components pane (Figure 1-19) offers a variety of advanced algorithms for *slicing* logic from legacy source: all the code needed for a computation or to “specialize” a program based on the value of a variable. You can create a self-contained program, called a *component,* from the sliced code or simply generate a HyperView list of sliced constructs for further analysis. You can mark and colorize the list constructs in the HyperView Source pane.

Both the component generation and list functions are supported in the full version of the Components pane available to users of Application Architect. The list function only is supported in the restricted version of the Components pane, called Logic Analyzer, available to users of Application Analyzer.

The methods for slicing code are described in *Creating Components* in the workbench documentation set. Follow the instructions below to mark and colorize slice results.

To mark and colorize slice results:

1 In the Logic Analyzer category, select the list of generated architect results and click the ✓ button on the tool bar. To mark and colorize sliced constructs in a single file, select the file in the List view and click the ✓ button. To mark and colorize a single construct, select it in the File view and click the ✓ button. Click the ✓ button again to turn off marking and colorizing.

Note: For Logic Analyzer, sliced data declarations are not marked and colorized.

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. You can use a construct list to narrow the search. For more information, see “Working with Advanced Search Criteria” on page 3-9 and “Using Construct Lists to Narrow a Clipper Search” on page 4-12.

To execute a Clipper search:

1 In the Category view, select a list, then click the ▾ button next to the 🔍 button on the tool bar. In the drop-down, choose the type of object you want to execute the search against. The Search window opens (Figure 4-3).

2 In the **Search In** drop-down, select:

- **All Sources** to execute the search against all the sources of the specified type in the project.
- A list if you want to execute the search only against sources of the specified type with non-zero counts in the selected list. If you select **Rule Finder.File Writes**, for example, the search will be executed only against sources of the specified type with non-zero counts in the File Writes list in the Rule Finder category.

3 Choose the criterion for the search you want to execute from the list of criteria in the lefthand pane. Modify the predefined criteria or create your own criteria as described in "Working with Advanced Search Criteria" on page 3-9.

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 List 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. That is, 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 In the Category view, select a list, then click the ▾ button next to the 🔍 button on the tool bar. In the drop-down, choose the type of object you want to execute the search against. The Search window opens (Figure 4-3).

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).

4 Select **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 Perform
```

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

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 Perform
which has related any Paragraph (as Called Paragraph)
```

7 Click the **any** link. The Condition window opens (Figure 3-4).

8 In the Condition window, choose:

- List in the **Applies to** drop-down
- Duplicates.Matched Paragraphs in the **Name** drop-down

Note: 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 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 4-11. 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 List 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 4-11.

Creating Projects in Clipper

You can create a project directly in Clipper from the results of your analysis. 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 the source files in a list, select the list in the Category 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 the source files in a list, select the list in the Category 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).

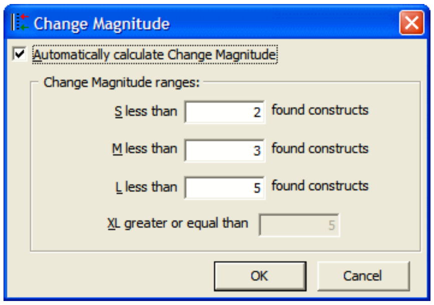
2 In the Change Magnitude window, select the **Automatically calculate change magnitude** check box if you want Clipper to set change magnitudes for the listed source files based on the ranges specified in the fields below the check box. The Clipper settings will override any existing change magnitudes 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 2-5.

Figure 4-7 *Change Magnitude Window*



☑ Automatically calculate Change Magnitude

Change Magnitude ranges:

S less than 2 found constructs

M less than 3 found constructs

L less than 5 found constructs

XL greater or equal than 5

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 the selected category, the *Metrics Report* shows the number of list items in each program. To generate a Metrics Report, select a category in the Top view, then click the ▾ button next to the ▤ button on the tool bar and choose **Metrics report** in the drop-down menu.

- For each source file in the selected list, the *Details Report* shows the constructs in the list, their type, and their location in the file. You can customize the report to include any HyperView attribute related to the constructs. To generate a Details Report, select a list in the Category view, then click the ▾ button next to the ▤ button on the tool bar and choose **Detailed report** in the drop-down menu. In the Select Attributes dialog, choose the construct 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 t**o 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.

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

5

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. 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 HyperView Impact pane displays a hierarchical view and diagram of an impact trace. Both views show the flow of data to or from a *startup item*. You can analyze the entire tree for a trace, or a consolidated tree that displays only the ports and leaf nodes traversed in the trace. Options let you filter out unwanted relationships or variables. You can save trace results and restore them in a later session.

Note: Set **Enable Impact Report** and **Enable Data Element Flow** in the project verification options to enable the Impact tool. For more information, see *Preparing Projects* in the workbench documentation set.

Understanding the Hierarchical View

In the hierarchical view (Figure 5-1), each line in the trace contains a symbol indicating the type of relationship the affected data item has with the previous item and a label showing the details of the relationship. Table 5-2 defines the relationship symbols.

Options give you complete control over the information displayed in the label. 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 (Figure 5-2), 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. For tool bar usage, see Appendix B, “Common Diagramming Features,” in *Analyzing Projects* in the workbench documentation set.

Generating an Impact Trace

Follow the steps below to generate an impact trace.

To generate an impact trace:

1 Set user preferences and project options for the Impact pane, as described in “Setting Impact Pane Options” on page 5-5.

2 In the Source or Context pane, select the construct that contains the startup item or items and choose **Impact** in the HyperView **View** menu. The Impact pane opens.

3 In the **Impact** menu, choose:

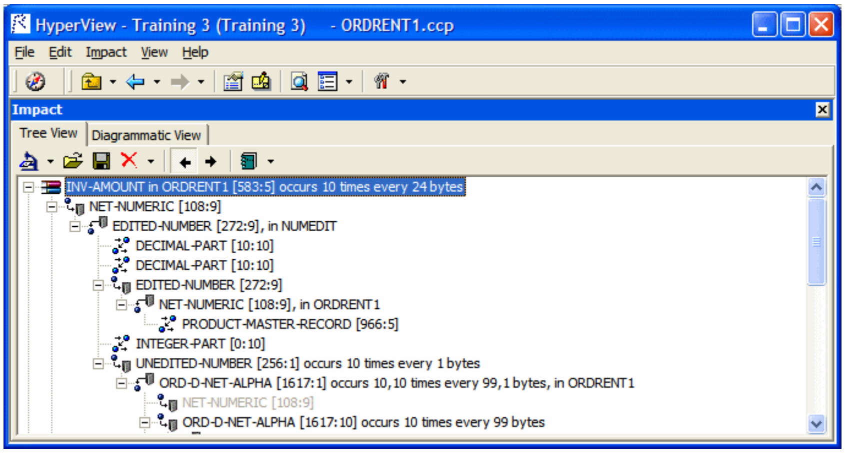
- **Backward Analysis** to view the flow of data into the startup items.
- **Forward Analysis** to view to view the flow of data out of the startup items.

4 In the **Impact** menu, choose:

- **Detailed Analysis** to generate the entire tree for a trace.
- **Consolidated Analysis** to generate a consolidated tree that displays only the ports and leaf nodes traversed in the trace. Only the subtrees for ports are displayed.

HyperView displays the trace results (Figure 5-1). Click the plus sign (+) next to an item to expand its hierarchy. Click the minus sign (-) to collapse the hierarchy.

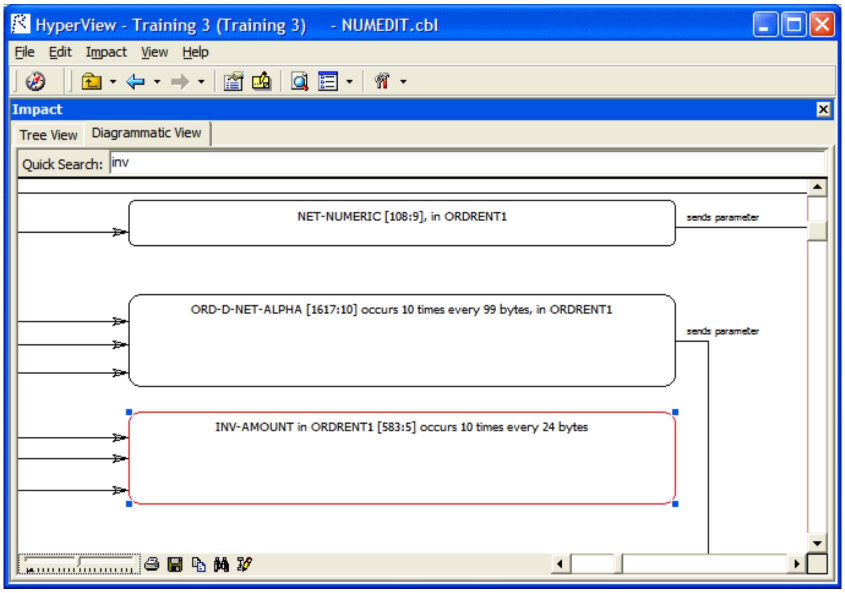
Figure 5-1 *Impact Pane with Consolidated Analysis*



```
INV-AMOUNT in ORDRENT1 [583:5] occurs 10 times every 24 bytes
  NET-NUMERIC [108:9]
    EDITED-NUMBER [272:9], in NUMEDIT
      DECIMAL-PART [10:10]
      DECIMAL-PART [10:10]
      EDITED-NUMBER [272:9]
        NET-NUMERIC [108:9], in ORDRENT1
          PRODUCT-MASTER-RECORD [966:5]
      INTEGER-PART [0:10]
      UNEDITED-NUMBER [256:1] occurs 10 times every 1 bytes
        ORD-D-NET-ALPHA [1617:1] occurs 10,10 times every 99,1 bytes, in ORDRENT1
          NET-NUMERIC [108:9]
          ORD-D-NET-ALPHA [1617:10] occurs 10 times every 99 bytes
```

5 Click the Diagrammatic View tab to display the impact trace in a diagram (Figure 5-2).

Figure 5-2 *Impact Pane with Trace Diagram*


HyperView - Training 3 (Training 3) - NUMEDIT.cbl
File Edit Impact View Help
Impact
Tree View
Diagrammatic View
Quick Search: inv
NET-NUMERIC [108:9], in ORDRENT1
sends parameter
ORD-D-NET-ALPHA [1617:10] occurs 10 times every 99 bytes, in ORDRENT1
sends parameter
INV-AMOUNT in ORDRENT1 [583:5] occurs 10 times every 24 bytes

6 Select an item in the trace and repeat steps step 3 and step 4 to start a trace for that item. The new trace is displayed below the previous trace.

7 To delete a trace, select its startup item and choose **Remove Root** in the **Impact** menu. To clear the pane, choose **Remove All** in the **Impact** menu.

Tip: Click the ← or → buttons on the tool bar to reverse the direction of a trace.

Setting Impact Pane Options

Use the Impact pane User Preferences and Project Options windows to control the level of detail provided in the label for each relationship type in the trace, the variables to filter out of the trace, and the impact relationships to show in the trace. Impact User Preferences control the appearance and behavior of the Impact pane across workspaces. Impact Project Options control the appearance and behavior of the Impact pane for the selected project.

Setting Impact Pane User Preferences

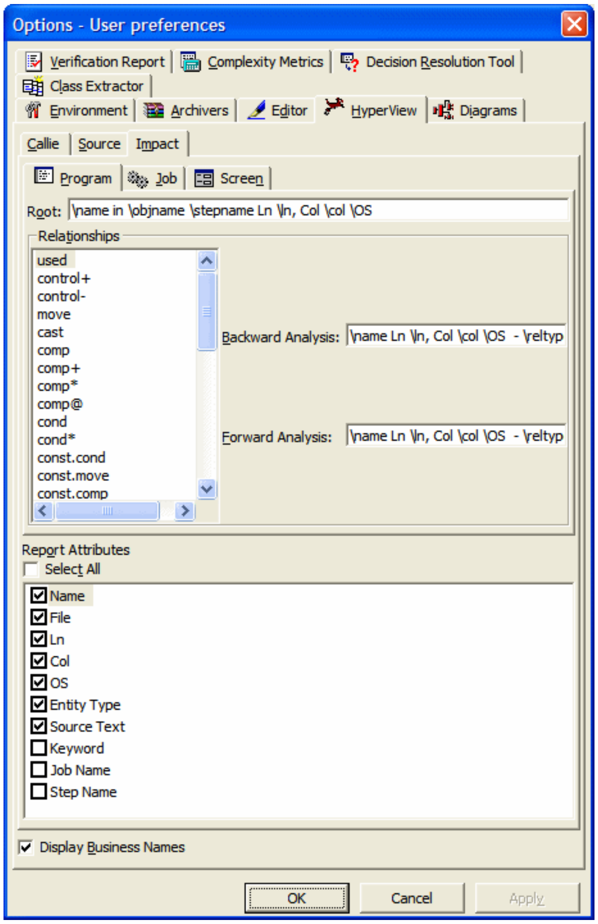
Use the Options window for the Impact pane to control the level of detail provided in the label for each relationship type in the trace: whether to show or hide the locations of variables in source, their memory offsets and sizes, and so forth. You can also use this window to specify the level of detail to include in a trace report.

To set Impact Pane user preferences:

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 5-3).
2. Choose the tab for the type of object whose relationship labels and report attributes you want to specify.
3. In the **Root** field, edit the template for the startup item label. Items preceded by a backslash are replaced with the values described in Table 5-1. You can modify unescaped text as necessary.
4. In the Relationship pane, select a relationship, then edit the template for its label in the **Backward Analysis** or **Forward Analysis** fields as appropriate. Items preceded by a backslash are replaced with the values described in Table 5-1. You can modify unescaped text as necessary.
5. In the Report Attributes pane, select the detail types you want to include in the trace report. The types are described in Table 5-1.

6 Choose **Display Business Names** if you want to display business names in the impact trace rather than original identifiers. For more information on business names, see Chapter 7, “Setting Up a Glossary.”

Figure 5-3 *Impact Pane User Preferences Window*



Root: \name in \objname \stepname Ln \ln, Col \col \OS

Relationships

used
control+
control-
move
cast
comp
comp+
comp*
comp@
cond
cond*
const.cond
const.move
const.comp

Backward Analysis: \name Ln \ln, Col \col \OS - \reltyp

Forward Analysis: \name Ln \ln, Col \col \OS - \reltyp

Report Attributes

☐ Select All

☑ Name
☑ File
☑ Ln
☑ Col
☑ OS
☑ Entity Type
☑ Source Text
☐ Keyword
☐ Job Name
☐ Step Name

☑ Display Business Names

Table 5-1 *Values for Label and Report Details*

Value	Description
Col	Number of the source column that references the affected data item.
EntityType	Entity type of the affected data item.
File	Name of the file that references the affected data item.
FromSel	Character offset for the end of the statement that references the affected data item.
JobName	Name of the job that references the affected data item.
Keyword	Keyword of the affected data item.
Ln	Number of the source line that references the affected data item.
Name	Name of the affected data item.
OS	Memory offset and size in bytes of the affected data item, separated by a colon (:).
RelType	Impact relationship type for the affected data item.
SourceText	Source text that references the affected data item.
StepName	Name of the job step that references the affected data item.
StepNum	Number of the job step that references the affected data item.
ToSel	Character offset for the beginning of the statement that references the affected data item.

Setting Impact Pane Project Options

Project options for the Impact pane determine:

- The variables to filter out of the trace, whether nested data items or redefines are treated as startup items, and whether recursive impacts or controlled dependency relationships are included in the trace.
- The impact relationships to include in the trace.

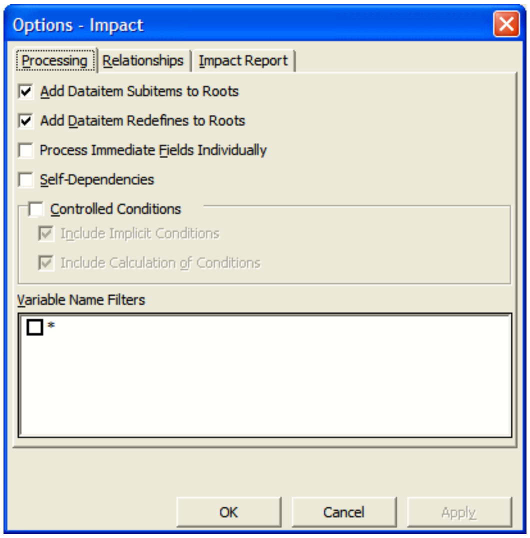
Setting Options on the Processing Tab

Use the Processing tab in the Project options for the Impact pane to filter variables out of the trace, control whether nested data items or redefines are treated as startup items, and determine whether recursive impacts or controlled dependency relationships are included in the trace.

To specify options on the Processing tab:

1 Click the button on the tool bar. The Impact Options window opens. Click the Processing tab (Figure 5-4).

Figure 5-4 *Processing Tab*



☑ Add Dataitem Subitems to Roots
☑ Add Dataitem Redefines to Roots
☐ Process Immediate Fields Individually
☐ Self-Dependencies
☐ Controlled Conditions
☑ Include Implicit Conditions
☑ Include Calculation of Conditions

Variable Name Filters
☐ *

2 In the Processing tab, 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 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.
- **Process Immediate Fields Individually** if you want the trace to treat each subitem of the startup item as a root. The startup item is not included in the trace.
- **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 a value under the control of a condition that depends on A. Choose:
  - **Include Implicit Conditions** if you want the trace to include relationships derived from implicit control conditions. In the following example:

    ```
    IF A > 0 THEN
      MOVE A TO B   (1)
    ELSE
      GOBACK.

    MOVE B TO C.    (2)
    ```

    statement (1) is explicitly controlled by the condition A > 0 because it is nested in the IF statement. Statement (2) is implicitly controlled by the condition because, on the one hand, there is no surrounding IF statement but, on the other hand, control may not reach the statement if the condition is not satisfied.
  - **Include Calculation of Conditions** if you want the trace to show how the controlling data item is calculated.

3. 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 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).

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

Setting Options on the Relationships Tab

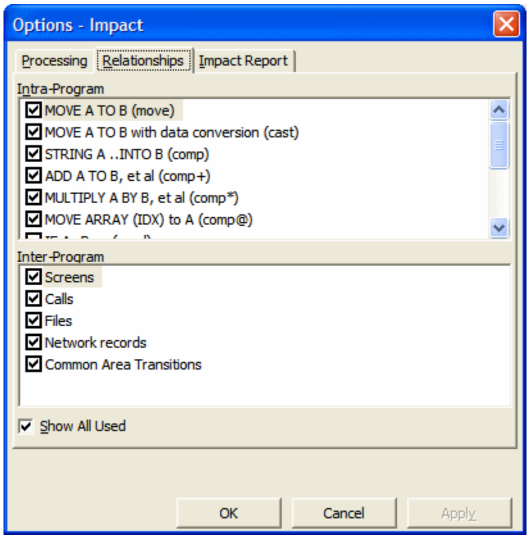
Use the Relationships tab in the Project options for the Impact pane to specify the intraprogram and interprogram impact relationship to include in the trace. For a description of the relationships, see “Impact Trace Relationships” on page 5-11.

To specify options on the Relationships tab:

1. Click the 🛠 button on the tool bar. The Impact Options window opens. Click the Relationships tab (Figure 5-5).
2. In the Relationships tab:
   - 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.

   For a description of intraprogram and interprogram relationships, see “Impact Trace Relationships” on page 5-11.
3. Remove the check mark next to **Show All Used** if you want the trace to include only used nodes that are necessary to its calculation.

Figure 5-5 *Relationships Tab*



Intra-Program

- ☑ MOVE A TO B (move)
- ☑ MOVE A TO B with data conversion (cast)
- ☑ STRING A ..INTO B (comp)
- ☑ ADD A TO B, et al (comp+)
- ☑ MULTIPLY A BY B, et al (comp*)
- ☑ MOVE ARRAY (IDX) to A (comp@)

Inter-Program

- ☑ Screens
- ☑ Calls
- ☑ Files
- ☑ Network records
- ☑ Common Area Transitions

☑ Show All Used

Impact Trace Relationships

The following table describes the intraprogram and interprogram relationships detected by the Impact tool.

Table 5-2 *Impact Trace Relationships*

Relationship	Definition	Type	Icon	Description
calls	N/A	interprogram	N/A	A parameter passed in a call to another program.
cast	MOVE A TO B with data conversion	intraprogram	●~+	A data item moved to a data item of a different type.

Table 5-2 *Impact Trace Relationships* (continued)

Relationship	Definition	Type	Icon	Description
common area transitions	N/A	interprogram	N/A	For Unisys 2200 Cobol, a common-storage data area item 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	(●)→	An arbitrary computation. The result is produced by applying complex rules to the argument, such as STRING.
comp+	ADD A TO B	intraprogram	●+→	An addition-like operation: ADD, SUBTRACT, or corresponding parts of COMPUTE.
comp*	MULTIPLY A BY B	intraprogram	●×→	A multiplication-like operation: MULTIPLY, DIVIDE, or corresponding parts of COMPUTE.
comp@	MOVE ARRAY(IDX) TO A	intraprogram	●(i)→	An operation with array elements.
cond	IF A = B ...	intraprogram	●=?	Comparison of data items with a symmetric relationship.
cond*	IF A * X = B ...	intraprogram	●×?	Comparison of a multiple of a data item with another data item.
const cond	IF A = 1 ...	intraprogram	●=?	Comparison of a data item with a constant.
const.move	MOVE 1 TO B	intraprogram	●→	A constant moved into a data item.
const.comp	ADD 1 TO B	intraprogram	●+→	An arithmetic operation with constants.
const.init	03 A ... VALUE 1	intraprogram	●=→	A data item initialized by a constant.

Table 5-2 *Impact Trace Relationships* (continued)

Relationship	Definition	Type	Icon	Description
DMS records	N/A	interprogram	N/A	For Unisys 2200 Cobol, data communication via Unisys DMS database records.
files	N/A	interprogram	N/A	Data communication via files. Traced only when corresponding JCL, ECL, FCT, or CSD files are verified.
files in jobs	N/A	interprogram	N/A	Data flow in JCL datasets when *files* is selected.
input port	N/A	intraprogram	↳	A data item in which data is received. (Not shown in Options window.)
move	MOVE A TO B	intraprogram	●→	A data item moved to a data item of the same type.
network records	N/A	interprogram	N/A	For Unisys 2200 Cobol, data communication via network records.
output port	N/A	intraprogram	↱	A data item from which data is sent. (Not shown in Options window.)
screens	N/A	interprogram	N/A	Data sent to a screen by one program and received in a screen by another.
screen definitions	N/A	interprogram	N/A	Data flow in screen fields when *screens* is selected.
start	N/A	intraprogram	☰	The startup item in an Impact pane consolidated analysis. (Not shown in Options window.)

Table 5-2 *Impact Trace Relationships* (continued)

Relationship	Definition	Type	Icon	Description
used	MOVE ... TO A ... MOVE A TO ...	intraprogram		A value assigned in a statement used as an argument in another statement. (Not shown in Options window.)

Saving and Restoring an Impact Trace

Save an impact trace in XML format by selecting its startup item and clicking the 💾 button on the tool bar. A Save As dialog opens, where you can specify the name and location of the impact trace.

Restore a saved impact trace by clicking the 📂 button on the tool bar. An Open dialog appears, where you can select the trace you want to restore.

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 (Cobol and APS only).

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 (Cobol and APS only):

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.

Analyzing Program Control Flows

6

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 these resources for analyzing program control flows:

- The Callie pane displays a diagram that shows the call flow for paragraphs in a Cobol program, subroutines in an RPG program, or procedures in a PL/I or Natural program.
- The Execution Path pane displays a hierarchical view and diagram that show the conditions that determine the flow of control in a Cobol or PL/I program.
- The Flowchart pane displays a diagram that shows the flow of control between statements in a Cobol paragraph or PL/I procedure, or between steps in a job or JCL procedure.
- The Animator lets you step through Cobol or Natural code displayed in a HyperView pane.

Using the Callie Pane

The Callie pane displays a diagram that shows the call flow for paragraphs in a Cobol program, subroutines in an RPG program, or procedures in a PL/I or Natural program. The flow is traced from a *seed* item to a called external program, subroutine, or procedure (if any).

The label of the box representing an item contains its name. If an item has a business name, the business name is displayed. Internal items are displayed with a black border. External items are displayed with a red border.The selected item is displayed with a background shadow.

Call relationship lines are color-coded as follows:

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

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 6-5. For tool bar usage, see Appendix B, "Common Diagramming Features," in *Analyzing Projects* in the workbench documentation set.

To generate a call trace diagram:

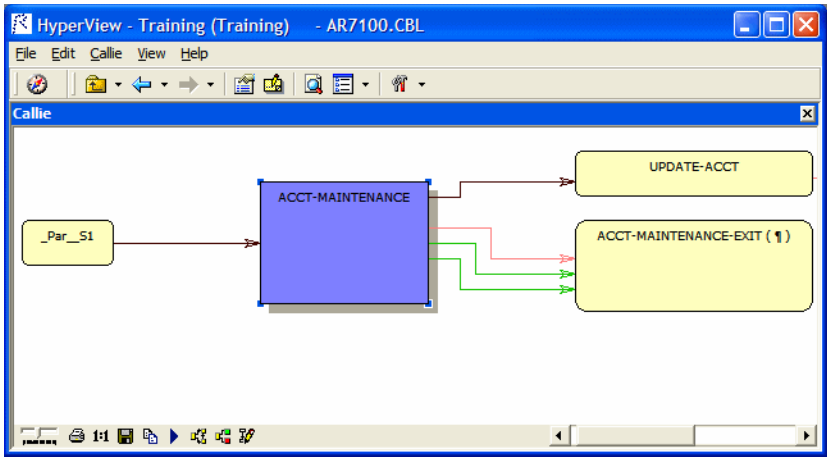
1. Set user preferences and project options for the Callie pane, as described in "Setting Callie Pane Options" on page 6-5.
2. In the Source or Context pane, select the seed item and choose **Callie** in the HyperView **View** menu. The Callie pane opens with the call trace for the seed item (Figure 6-1).
3. In the **Callie** menu, choose:
   - **SubGraph mode** to select the subgraph view.
   - **SubTree mode** to select the subtree view.

   For more information, see "Choosing the Diagram View" on page 6-3.

4 Select an internal item to navigate to it in the Source and Context panes. For Cobol, select an external program to display it in the Source and Context panes. For PL/I and Natural, select an external procedure and choose **Switch to *Procedure*** in the right-click menu to display it in the Source and Context panes. Select a relationship line to navigate to the call in the Source and Context panes.

5 Refresh the display, as described in “Redrawing the Diagram” on page 6-4.

Figure 6-1 *Callie Pane*


HyperView - Training (Training) - AR7100.CBL
File
Edit
Callie
View
Help
Callie
UPDATE-ACCT
ACCT-MAINTENANCE
_Par__S1
ACCT-MAINTENANCE-EXIT (¶)
1:1

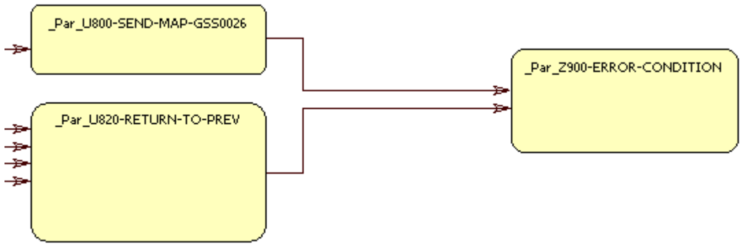
Choosing the Diagram View

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

Subgraph View

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

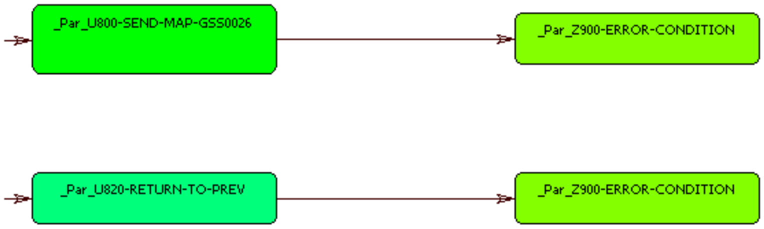
Figure 6-2 *Subgraph View*



Subtree View

The subtree view offers a linear representation of the information in the diagram. Items 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*


_Par_U800-SEND-MAP-GSS0026
_Par_Z900-ERROR-CONDITION
_Par_U820-RETURN-TO-PREV
_Par_Z900-ERROR-CONDITION

Redrawing the Diagram

The diagram is automatically redrawn when you select a new seed item in the Source or Context pane, unless that item already appears in the current diagram. To redraw the diagram in that case, select the new seed item 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, colors of display objects, and so forth. 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

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 item to the color specified for unselected, or “default,” items. 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).

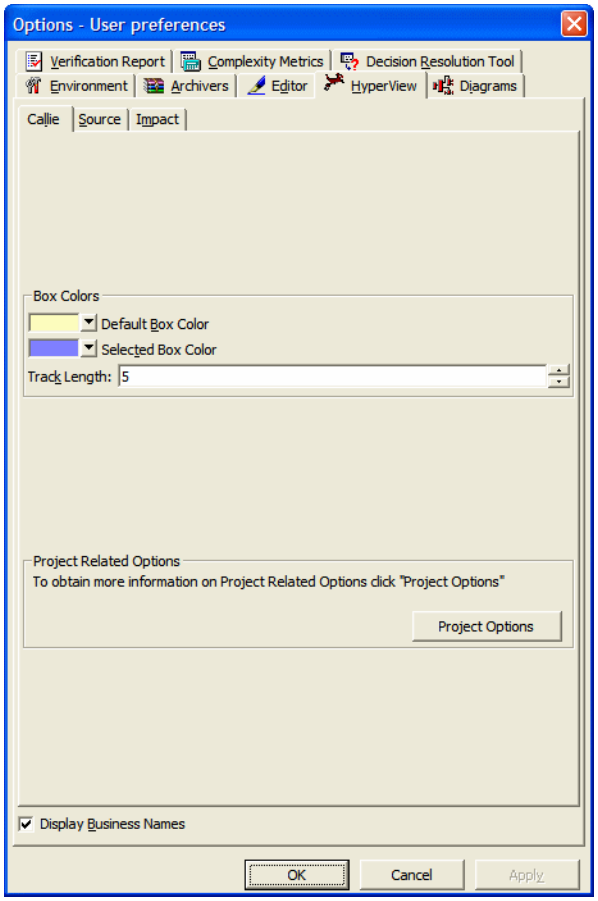
2. The current default background color of the box representing an item is displayed in the **Default Box Color** drop-down. The current background color of the box representing the selected item is displayed in the **Selected Box Color** drop-down.

   Click the adjacent ▾ buttons to edit the color of the items. 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.

4. Choose **Display Business Names** if you want to display business names in the flow trace rather than original identifiers. For more information on business names, see Chapter 7, “Setting Up a Glossary.”

Figure 6-4 *Callie User Preferences Window*



Default Box Color

Selected Box Color

Track Length: 5

To obtain more information on Project Related Options click "Project Options"

☑ Display Business Names

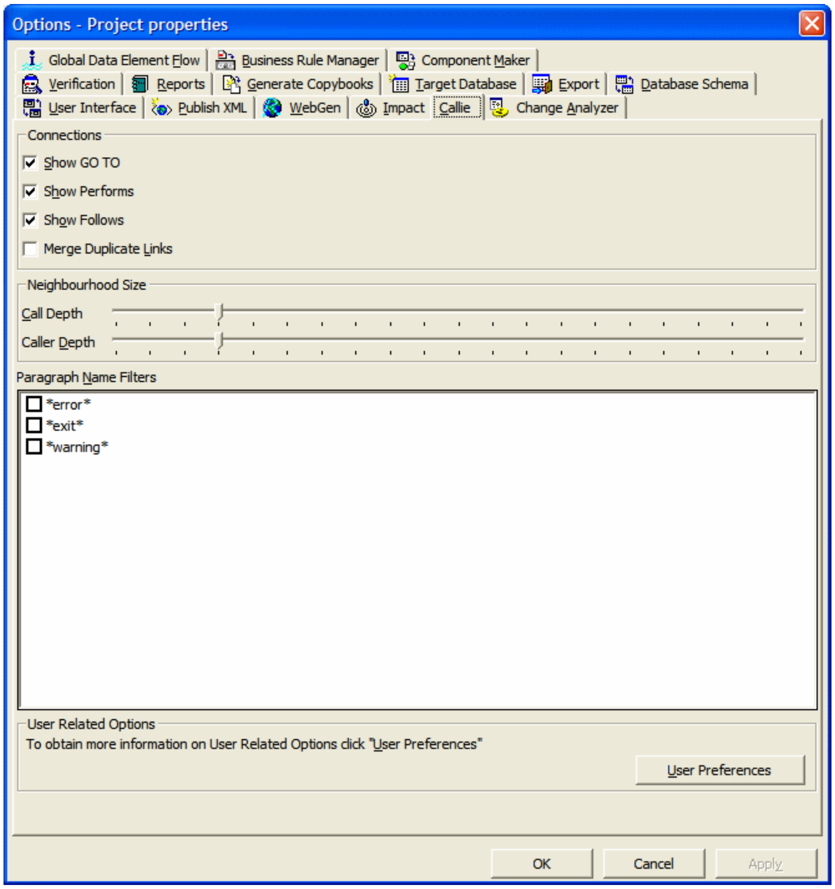
Setting Callie Project Options

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 item. You can also set Project Options that limit the relationship types the diagram displays (PERFORMs, GO TOs, or Fall Thrus) and filter out items 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).

Figure 6-5 *Callie Project Options Window*



Connections

☑ Show GO TO

☑ Show Performs

☑ Show Follows

☐ Merge Duplicate Links

Paragraph Name Filters

☐ *error*

☐ *exit*

☐ *warning*

User Related Options

To obtain more information on User Related Options click "User Preferences"

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 item.

Note: If Callie loads too slowly in subgraph mode, set the these options to values lower than 5.

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

To add a pattern to the list, right-click in the Paragraph Name Filters pane and choose **Add** in the pop-up menu. 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.

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

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 *seed construct* selected in the Source or Context pane. The seed construct can be either a paragraph or statement. Use an execution path to ensure that a code segment encapsulates all of the business logic for a candidate business rule.

Note: For a Cobol program, 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

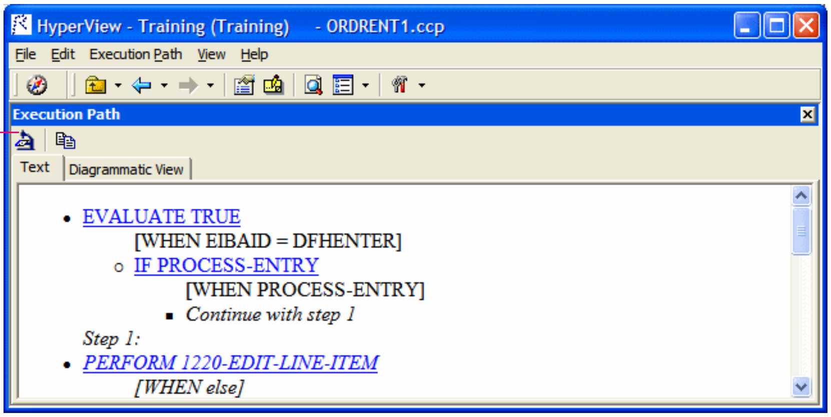
Follow the steps below to generate an execution path.

To generate an execution path:

1. In the Source or Context pane, select the seed construct and choose **Execution Path** in the HyperView **View** menu. The Execution Path pane opens.
2. In the **Execution Path** menu, choose **Analyze Selected Construct**. HyperView displays the execution path for the seed construct (Figure 6-6).

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*


generate
execution path
HyperView - Training (Training) - ORDRENT1.ccp
File Edit Execution Path View Help
Execution Path
Text
Diagrammatic View

- EVALUATE TRUE
  [WHEN EIBAID = DFHENTER]
  - IF PROCESS-ENTRY
    [WHEN PROCESS-ENTRY]
    - *Continue with step 1*

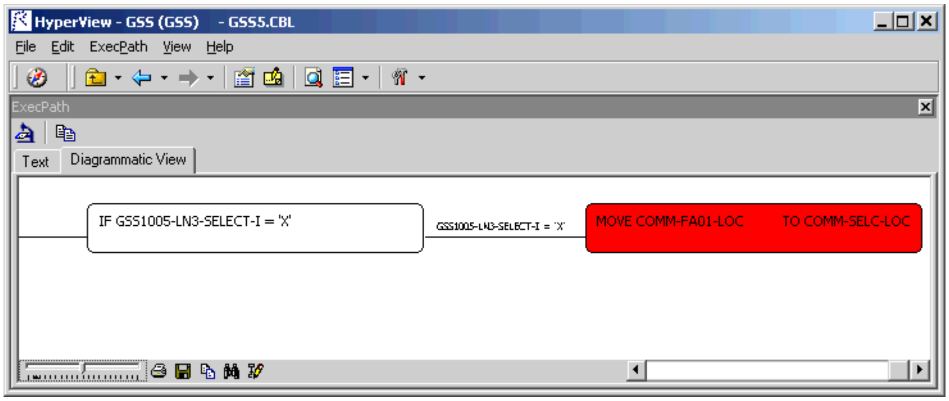
Step 1:

- *PERFORM 1220-EDIT-LINE-ITEM*
  [WHEN else]

3. Click the Diagrammatic View tab to display the execution path in a diagram (Figure 6-7). The seed construct is displayed in red. For tool

bar usage, see Appendix B, “Common Diagramming Features,” in *Analyzing Projects* in the workbench documentation set.

Figure 6-7 *Execution Path Pane with Diagram*


HyperView - GSS (GSS) - GSS5.CBL
File
Edit
ExecPath
View
Help
ExecPath
Text
Diagrammatic View
IF GSS1005-LN3-SELECT-I = 'X'
GSS1005-LN3-SELECT-I = 'X'
MOVE COMM-FA01-LOC TO COMM-SELC-LOC

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 diagram that shows the flow of control between statements in a Cobol paragraph or PL/I procedure, or between steps in a job or JCL procedure. The label of the box representing an item in the flow contains the text of the item.

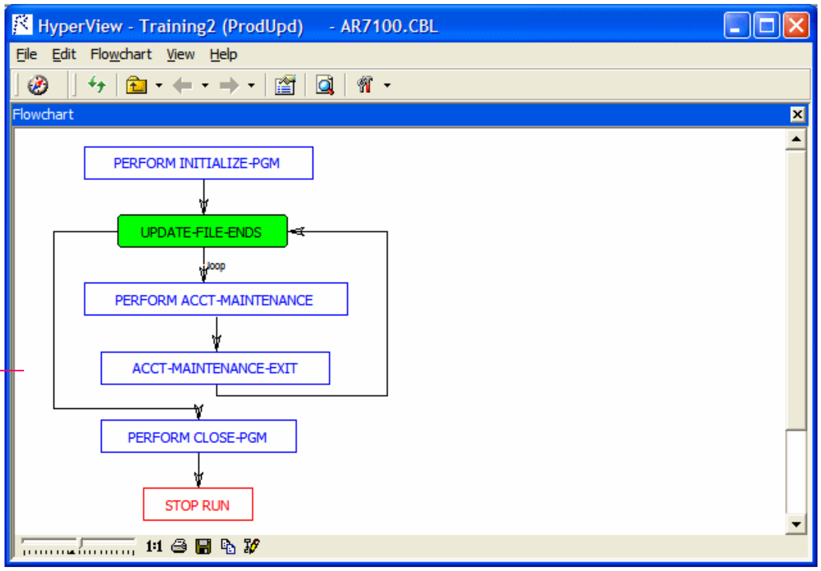
PERFORM, CALL, RETURN, and GO TO statements are displayed in boxes with a blue border and white background. Double-click the box for the statement, or select it and choose **Switch to *Statement*** in the right-click menu, to redraw the diagram with the paragraph flow for its target. Double-click the maximize box for a step to view a diagram of its relationships. Double-click the minimize box in the diagram to return to the flow chart.

Conditions or conditional statement are displayed in boxes with a green background. Captions along relationship lines show the conditions or blocks that determine the flow. The statement at which the flow in a program terminates is displayed in a box with a red border and white background. All other statements are displayed in boxes with a black border and white background. For tool bar usage, see Appendix B, "Common Diagramming Features," in *Analyzing Projects* in the workbench documentation set.

To generate a flowchart:

In the Source or Context pane, select the paragraph, procedure, job, or JCL procedure whose flow of control you want to view and choose **Flowchart** in the HyperView **View** menu. HyperView displays the flowchart for the selected item (Figure 6-6).

Figure 6-8 *Flowchart Pane*



double-click to redraw diagram

Using the Animator

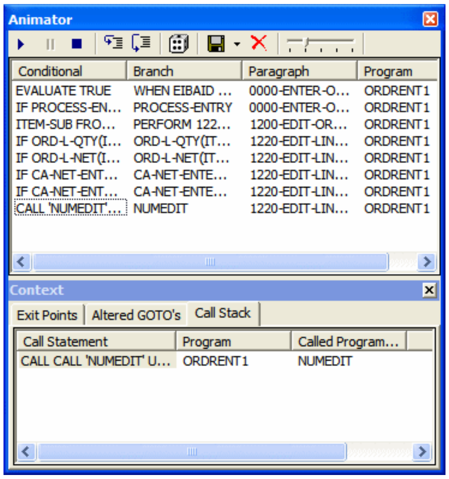
The Animator lets you step through the code displayed in a HyperView pane. You can choose program branches yourself, or have the animator choose them randomly.

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 and choose **Animator** in the HyperView **View** menu. The Animator window opens (Figure 6-9).

Figure 6-9 *Animator Window with Context Pane*



Conditional	Branch	Paragraph	Program
EVALUATE TRUE	WHEN EIBAID …	0000-ENTER-O…	ORDRENT1
IF PROCESS-EN…	PROCESS-ENTRY	0000-ENTER-O…	ORDRENT1
ITEM-SUB FRO…	PERFORM 122…	1200-EDIT-OR…	ORDRENT1
IF ORD-L-QTY(I…	ORD-L-QTY(IT…	1220-EDIT-LIN…	ORDRENT1
IF ORD-L-NET(I…	ORD-L-NET(IT…	1220-EDIT-LIN…	ORDRENT1
IF CA-NET-ENT…	CA-NET-ENTE…	1220-EDIT-LIN…	ORDRENT1
IF CA-NET-ENT…	CA-NET-ENTE…	1220-EDIT-LIN…	ORDRENT1
CALL 'NUMEDIT'…	NUMEDIT	1220-EDIT-LIN…	ORDRENT1

Call Statement	Program	Called Program…
CALL CALL 'NUMEDIT' U…	ORDRENT1	NUMEDIT

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 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 Internal category in Clipper. For list usage, see "Using the Clipper Pane" on page 4-5.

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 Glossary pane to create a project glossary and define business names. That's the subject of the next chapter.

Setting Up a Glossary

7

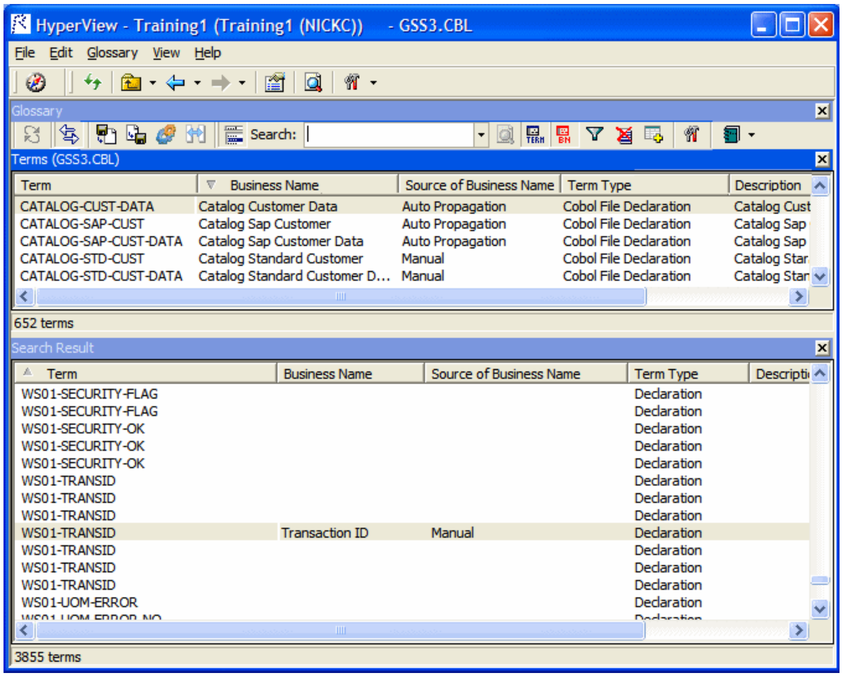
Use the Glossary pane to create a dictionary, or *glossary,* of meaningful names for items of interest in your workspace: data elements, paragraphs, programs, screens, tables, and the like. A glossary is especially useful for assigning *business names* to identifiers: natural language names that make 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

A *term* is the name of an item of interest in your application. The Glossary pane automatically generates a list of terms for the file selected in the Objects pane and any included files (Figure 7-1). The list is filtered by your choices in the workspace options for the Glossary pane, as described in “Setting Options on the Entities Tab” on page 7-13.

Figure 7-1 *Glossary Pane*



Term	Business Name	Source of Business Name	Term Type	Description
CATALOG-CUST-DATA	Catalog Customer Data	Auto Propagation	Cobol File Declaration	Catalog Cust
CATALOG-SAP-CUST	Catalog Sap Customer	Auto Propagation	Cobol File Declaration	Catalog Sap
CATALOG-SAP-CUST-DATA	Catalog Sap Customer Data	Auto Propagation	Cobol File Declaration	Catalog Sap
CATALOG-STD-CUST	Catalog Standard Customer	Manual	Cobol File Declaration	Catalog Star
CATALOG-STD-CUST-DATA	Catalog Standard Customer D…	Manual	Cobol File Declaration	Catalog Star

Term	Business Name	Source of Business Name	Term Type	Descripti
WS01-SECURITY-FLAG			Declaration	
WS01-SECURITY-FLAG			Declaration	
WS01-SECURITY-OK			Declaration	
WS01-SECURITY-OK			Declaration	
WS01-SECURITY-OK			Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID	Transaction ID	Manual	Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID			Declaration	
WS01-TRANSID			Declaration	
WS01-UOM-ERROR			Declaration	
WS01-UOM-ERROR-NO			Declaration	

The Glossary tool consists of two panes:

- The *Terms* pane displays the terms in the selected file.
- The *Search Result* pane displays the terms returned by the most recent Glossary search. The scope of the search is workspace-wide.

Each pane displays the term type, the business name and description assigned to the term, and how the business name and description was assigned. To hide a pane, choose **View Terms** or **View Search Result** in the **Glossary** menu, as appropriate. Deselect the menu item to show the pane again.

Choose **Synchronize with other views** in the **Glossary** menu to synch the Glossary with other HyperView panes. Select a term in the Terms or Search pane to navigate to the term in other HyperView panes. Deselect the **Synchronize with other views** menu item to turn off synchronization.

Tip: Turning off synchronization may improve Glossary performance for very large workspaces.

Searching for Terms

The Glossary pane offers simple and advanced search facilities. The scope of all searches is workspace-wide.

Tip: Choose **Clear Search Result** in the **Glossary** menu to clear the Search Result pane.

To use the simple search facility:

1. Enter the text for the search (or select the text of a recent search) in the **Search** combo box on the tool bar. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA), as described in *Getting Started* in the workbench documentation set.

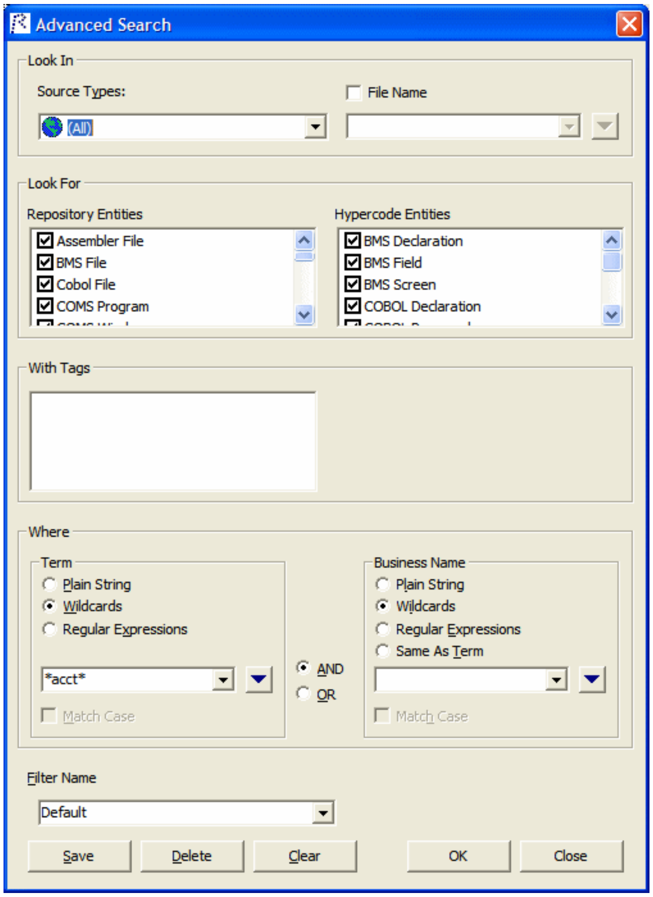
Tip: Select a column heading and click the ▤ button on the tool bar, then enter text in the **Search** combo box to scroll intelligently to items in that column. The tool matches text as you type.

2. Click the BN button on the tool bar to match business names only. Click both the TERM button and the BN button to match terms and business names (search terms are ORed).
3. Click the 🔍 button on the tool bar. The results of the search are displayed in the Search Result pane.

To use the advanced search facility:

1 In the **Glossary** menu, choose **Advanced Search**. The Advanced Search dialog opens (Figure 7-2).

Figure 7-2 *Advanced Search Dialog*



Advanced Search

Look In

Source Types: (All)

□ File Name

Look For

Repository Entities

☑ Assembler File
☑ BMS File
☑ Cobol File
☑ COMS Program

Hypercode Entities

☑ BMS Declaration
☑ BMS Field
☑ BMS Screen
☑ COBOL Declaration

With Tags

Where

Term

○ Plain String
● Wildcards
○ Regular Expressions

acct

□ Match Case

● AND
○ OR

Business Name

○ Plain String
● Wildcards
○ Regular Expressions
○ Same As Term

□ Match Case

Filter Name: Default

2A To create a new search filter, enter the name of the filter in the **Filter Name** combo box at the bottom of the dialog.

3A To edit an existing search filter, choose the filter in the **Filter Name** combo box at the bottom of the dialog.

4 To filter on source file type, select the type in the drop-down at the top of the Look In pane.

5. To filter on source file name, select the **File Name** check box, then select the matching pattern for the name in the accompanying combo box. Enter a new matching pattern by typing in the combo box.

   You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA), as described in *Getting Started* in the workbench documentation set. Click the ▼ button next to the drop-down to display a pop-up menu of elements you can insert in the pattern. Click an element to insert it. The matching pattern is saved when you save or execute the filter.

6. To filter on repository object type, select each object type you want to filter on in the Repository Entities pane.

7. To filter on HyperCode construct type, select each construct type you want to filter on in the Hypercode Entities pane.

8. To filter on tags, select each tag you want to filter on in the With Tags pane. The search will return matching repository objects that have been assigned the tag and matching HyperCode constructs contained in repository objects that have been assigned the tag.

9. To filter on the text for a term and/or business name, enter a matching pattern for the text in the combo boxes for terms and/or business names in the Where pane. You can enter plain character strings, regular expressions, or wildcard patterns. Click the radio button for the type of pattern matching you want to use.

   Select the **Match Case** check box if you want the search to match case. Select **AND** if you want the search terms to be ANDed. Select **OR** if you want the search terms to be ORed

   For regular expressions and wildcard patterns, click the ▼ button next to the drop-down to display a list of elements you can insert in the pattern. Click an element to insert it.

   You can use Posix regular expressions and wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA), as described in *Getting Started* in the workbench documentation set. The matching pattern is saved when you save or execute the filter.

10. Click **Save** to save the filter. Click **OK** to execute the filter. The Glossary tool displays the search results in the Search Result pane.

Note: If you execute another advanced search with the same criteria for the term or business name, the contents of the Search Result pane act as a filter. If you don't want to filter on existing results, choose **Clear Search Result** in the **Glossary** menu to clear the Search Result pane.

Every search filter you execute is saved, whether or not you save it explicitly. To delete a search filter, select it in the **Filter Name** drop-down and click **Delete**.

Assigning Business Attributes

You can assign business names manually or in batch mode. You can export terms and business attributes from the glossary for one workspace for import into the glossary of another.

Assigning Business Attributes Manually

You can assign business names and descriptions in the Glossary pane or in any other HyperView pane in which an eligible item is visible.

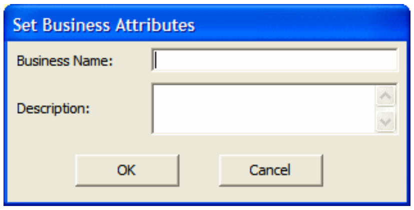
To assign business attributes:

1A In the Terms or Search Result pane, select the term you want to create a business name and description for and choose **Set Business Attributes** in the right-click menu. The Set Business Attributes dialog opens (Figure 7-3).

1B In a HyperView pane, select the object you want to create a business name and description for and choose **Set Business Attributes** in the right-click menu. The Set Business Attributes dialog opens (Figure 7-3).

2 In the Set Business Attributes dialog, enter the business name for the term in the **Business Name** field and the business description in the **Description** field. Click **OK** to dismiss the dialog and return to the Glossary pane.

Figure 7-3 *Set Business Attributes Dialog*



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 replacement text for each string you want to change in a term. For the string NUM, for example, you would enter the replacement text Number. You can import terms and business names into the batch interface from a file.

To assign business names in batch mode:

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

2 In the **String to find** combo box, enter the string you want to replace and click **Add**. The Glossary tool displays the string in the String column. In the Replacement column, click in the cell, type the replacement text for the string in the text field, and click outside the cell.

Tip: To import terms and business names from a file, click **Import**. To export terms and attributes to a file, click **Export**.

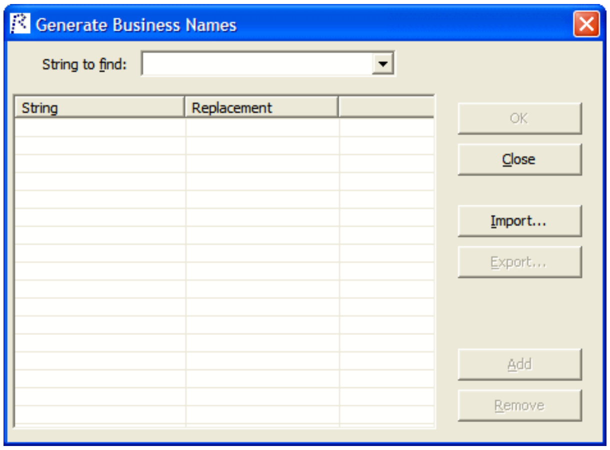
3 Repeat step 2 for each string you want to replace.

Tip: To remove an entry, select it and click **Remove**.

4 When you are satisfied with your entries, click **OK**. The Glossary tool automatically generates business names for each term or business name that contains the string, replacing the string with the text

you specified. Each replace operation acts on the output of the previous replace operation.

Figure 7-4 *Generate Business Names Window*



Importing and Exporting Business Attributes

You can export terms and business attributes from the glossary for one workspace for import into the glossary of another. You can also import terms and business attributes propagated programmatically. Term (source), business name (replacement), and business description (description) must appear in XML file format as follows:

```
<glossary>
   <item source="CUST_INFO" replacement="CUSTOMER
   INFORMATION" description="customer information"/>
   ...
</glossary>
```

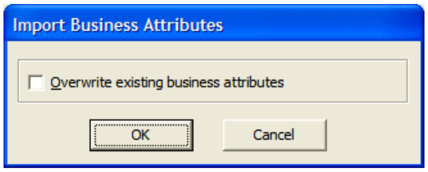
To export business names from a glossary:

1 In the **Glossary** menu, choose **Export Business Attributes**. A save as dialog opens where you can specify the name and location of the file for the exported business attributes.

To import business names into a glossary:

1 In the **Glossary** menu, choose **Import Business Attributes**. The Import Business Attributes dialog opens (Figure 7-5).

Figure 7-5 *Import Business Attributes Dialog*



□ Overwrite existing business attributes

2 In the Import Business Attributes dialog, select **Overwrite existing business attributes** if you want to replace existing business attributes as well as add new ones. Click **OK**. An open dialog appears where you can select the file for the business attributes you want to import.

Deleting Business Attributes

To delete business attributes for a given term, select the term in the Terms or Search Result pane and choose **Set Business Attributes** in the right-click menu. The Set Business Attributes dialog opens (Figure 7-3), where you can delete the attributes.

To delete all business names, right-click in the Terms or Search Result pane and choose **Delete All Business Names** in the pop-up menu. To delete only manually assigned business names, right-click in the Terms or Search Result pane and choose **Delete Manually Entered** in the pop-up menu. In either case, you are prompted to confirm the deletion. Click **Yes**. Select **Delete Descriptions** in the confirmation dialog if you also want to delete descriptions.

To delete all business descriptions, right-click in the Terms or Search Result pane and choose **Delete All Descriptions** in the pop-up menu. You are prompted to confirm the deletion. Click **Yes**.

Synchronizing Business Names

When you *synchronize* business names, you propagate the business name for one identifier to a related identifier with an identical meaning. Business names can be propagated in either direction of the relationship and transitively:

- For the statements MOVE A to B and MOVE B to C, if A has a business name, it is propagated to B and C.
- For the statements MOVE A to B and MOVE B to C, if C has a business name, it is propagated to B and A.

No business name is assigned to a variable if it receives values from variables with different business names:

```
Working-Storage Section.
10 Cust-No PIC 9(7).
10 Ord-No PIC 9(7).
10 Display-No PIC 9(7).

Procedure Division.
…
Move Cust-No to Display-No.
Display Display-No.
Move Ord-No to Display-No.
Display Display-No.
…
```

If Cust-No has the business name Customer Number, and Ord-No has the business name Order Number, no business name is assigned to Display-No, since it receives values from variables with different business names.

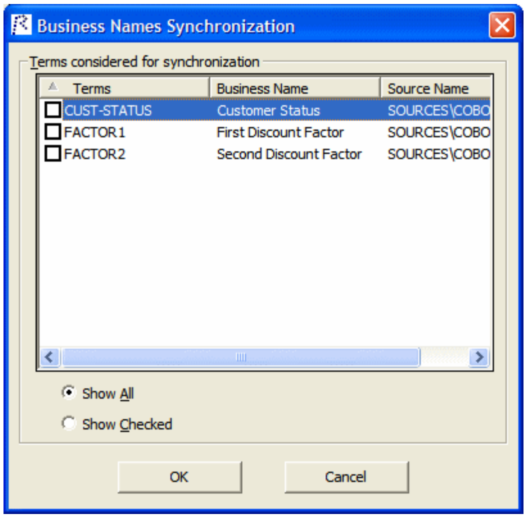
To synchronize business names:

1 Specify the impact relationships you want to synchronize business names for, as described in "Setting Options on the Synchronization Tab" on page 7-12.

2 In the **Glossary** menu, choose **Synchronize Business Names**. The Business Names Synchronization window opens (Figure 7-6).

Note: If you assigned business attributes in another tool during the current Glossary session, you are prompted to refresh the glossary. In the **Glossary** menu, choose **Refresh Glossary**.

Figure 7-6 *Business Names Synchronization Window*


Business Names Synchronization
Terms considered for synchronization
OK
Cancel

Terms	Business Name	Source Name
□ CUST-STATUS	Customer Status	SOURCES\COBO
□ FACTOR1	First Discount Factor	SOURCES\COBO
□ FACTOR2	Second Discount Factor	SOURCES\COBO

● Show All

○ Show Checked

3 In the Business Names Synchronization window, place a check mark next to each term whose business name you want to propagate.

Tip: Select **Show All** to display all terms with business names. Select **Show Checked** to display only checked business names.

4 When you are satisfied with your choices in the Business Names Synchronization window, click **OK**. The Glossary tool propagates the selected business names.

Setting Glossary Options

Workspace options for the Glossary tool determine:

- The types of impact relationships for which business names are synchronized.
- The types of entities you want to include identifiers for.

Setting Options on the Synchronization Tab

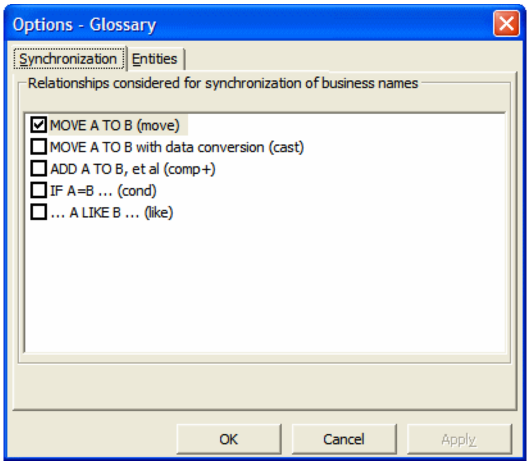
Use the Synchronization tab in the Workspace options for the Glossary tool to specify the impact relationships you want to synchronize business names for. For more information, see "Synchronizing Business Names" on page 7-10.

To specify options on the Synchronization tab:

1 Click the button on the tool bar. The Glossary options window opens. Click the Synchronization tab (Figure 7-7).

2 Place a check mark next to each impact relationship you want to synchronize business names for. For a description of the impact relationships, see "Impact Trace Relationships" on page 5-11.

Figure 7-7 *Synchronization Tab*



Relationships considered for synchronization of business names

☑ MOVE A TO B (move)
□ MOVE A TO B with data conversion (cast)
□ ADD A TO B, et al (comp+)
□ IF A=B ... (cond)
□ ... A LIKE B ... (like)

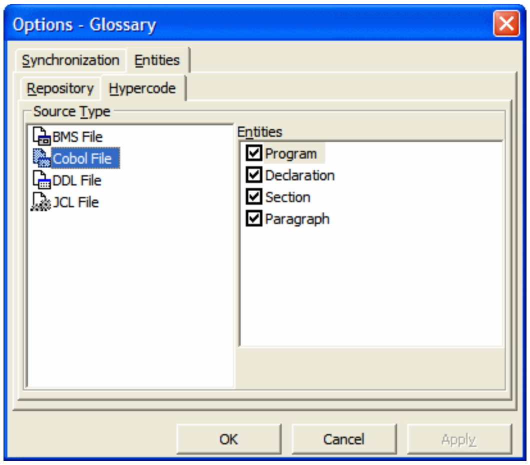
Setting Options on the Entities Tab

Use the Entities tab in the Workspace options for the Glossary tool to specify the types of entities you want to include identifiers for in the glossary.

To specify options on the Entities tab:

1. Click the button on the tool bar. The Glossary options window opens. Click the Entities tab, then the Hypercode tab (Figure 7-8).
2. In the Hypercode tab, select the programming language for the application in the Source Type pane. In the Hypercode Entities pane, select each type of Hypercode construct you want to include identifiers for in the glossary. Repeat this step for each programming language in use in the workspace.
3. Click the Repository tab. In the Repository tab, select each type of repository object you want to include identifiers for in the glossary.

Figure 7-8 *Entities Tab*



Source Type
- BMS File
- Cobol File
- DDL File
- JCL File

Entities
- ☑ Program
- ☑ Declaration
- ☑ Section
- ☑ Paragraph

Refreshing a Glossary

You need to refresh a glossary if, during the current Glossary session, you assign a business name to an object in another tool and want the change to be reflected in the glossary. Follow the instructions below to refresh a glossary.

To refresh a glossary:

1 In the **Glossary** menu, choose **Refresh Glossary**. A dialog box prompts you to confirm that you want to refresh the glossary. Click **OK**.

Generating Glossary Reports

The Glossary tool offers two reports:

- The Terms report prints the contents of the Terms pane. Choose **Terms Report** in the **Glossary** menu to generate the Terms report.
- The Search report prints the contents of the Search Result pane. Choose **Search Report** in the **Glossary** menu to generate the Search 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.

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.

8 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 data transfer and other tasks.

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.

Business Rule Manager (BRM) 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 for program code that performs a discrete task in a business process. It identifies and documents the *code segment* that performs this task. 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 segment only. The same segment can be assigned to different rules. Rules are identified in the Rules pane with a 🗎 symbol.

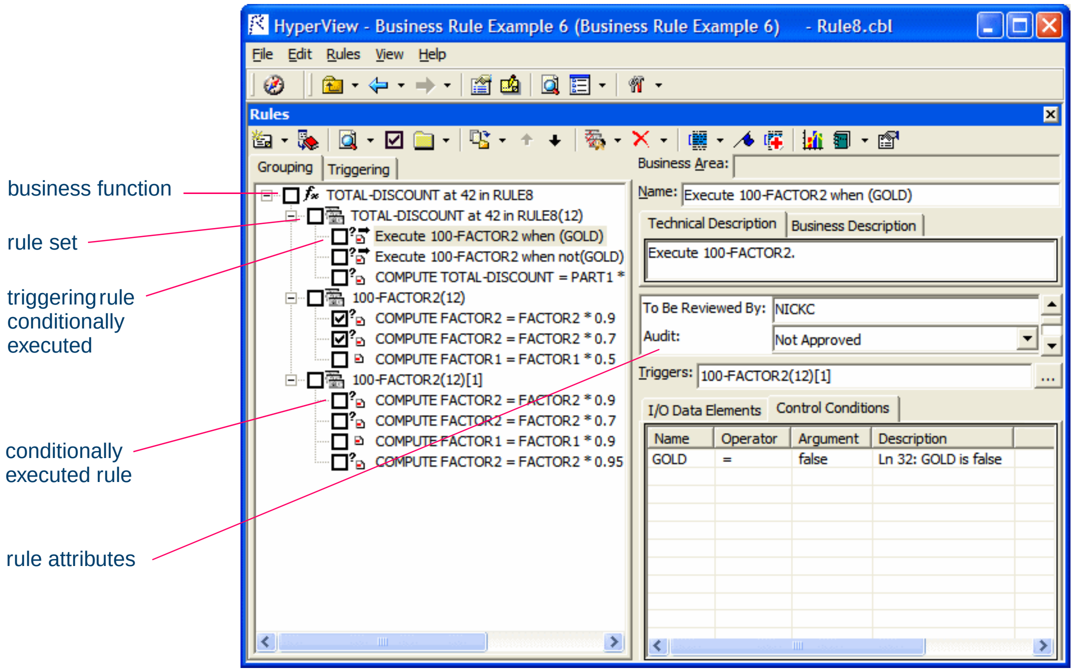
How Business Rules Are Organized

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). Think of a rule set as a business process, like Validate Input or Update File. Rule sets are identified in the Rules pane with a ▤ symbol.

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. Business functions are identified in the Rules pane with a *fx* symbol.

Business function names must be unique in the workspace. The same rule set can exist in multiple business functions; the same rule can exist in multiple rule sets. Figure 8-1 shows the HyperView Rules pane with a typical hierarchy of business functions, rule sets, and business rules.

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


HyperView - Business Rule Example 6 (Business Rule Example 6) - Rule8.cbl
File Edit Rules View Help
Rules
Grouping
Triggering

business function

rule set

triggering rule conditionally executed

conditionally executed rule

rule attributes

```
□ TOTAL-DISCOUNT at 42 in RULE8
  □ TOTAL-DISCOUNT at 42 in RULE8(12)
    □ Execute 100-FACTOR2 when (GOLD)
    □ Execute 100-FACTOR2 when not(GOLD)
    □ COMPUTE TOTAL-DISCOUNT = PART1 *
  □ 100-FACTOR2(12)
    ☑ COMPUTE FACTOR2 = FACTOR2 * 0.9
    ☑ COMPUTE FACTOR2 = FACTOR2 * 0.7
    □ COMPUTE FACTOR1 = FACTOR1 * 0.5
  □ 100-FACTOR2(12)[1]
    □ COMPUTE FACTOR2 = FACTOR2 * 0.9
    □ COMPUTE FACTOR2 = FACTOR2 * 0.7
    □ COMPUTE FACTOR1 = FACTOR1 * 0.9
    □ COMPUTE FACTOR2 = FACTOR2 * 0.95
```

Business Area:

Name: Execute 100-FACTOR2 when (GOLD)

Technical Description | Business Description

Execute 100-FACTOR2.

To Be Reviewed By: NICKC

Audit: Not Approved

Triggers: 100-FACTOR2(12)[1]

I/O Data Elements | Control Conditions

Name	Operator	Argument	Description
GOLD	=	false	Ln 32: GOLD is false

How to Assign Segments to Business Rules

You 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 predefined attributes.

Understanding Business Rule Attributes

Business rule *attributes* define the characteristics of the rule: its name, classification, status, business description, and the like. You can view rule attributes in the righthand pane of the Rules window (Figure 8-1).

Tip: Use the Technical Description, Business Description, Audit, Classification, Status, and Transition attributes for a business function or rule set to indicate that a value applies for all the rules included in the business function or rule set.

You can propagate rule attribute values to one or more rules (see "Propagating Rule Attributes" on page 8-28), and you can use a variety of methods to batch edit rule attributes:

- Clipper lets you generate business rules in a specified rule set with predefined attributes. For more information, see "Generating Rules in a Specified Rule Set" on page 4-9.
- The search facility in the Rules pane lets you locate existing rules with common characteristics, which you can then modify in bulk with the Change Rules feature. For more information, see "Searching for Rules" on page 8-35 and "Rule Search Window" on page 8-36.

You can modify attribute characteristics and valid values, and you can create custom attributes suited to your particular needs. For more information, see "Setting Business Rule Manager Project Options" on page 8-40.

Understanding Triggers, I/O Data Elements, and Control Conditions

In addition to the attributes of a rule, the righthand pane of the Rules window (Figure 8-1) displays the input/output data elements of the rule, its control conditions, and whether it triggers a rule set:

- *I/O Data Elements* identify the input, output, and input/output fields for a rule.
- *Control Conditions* identify the conditions that govern the execution of a rule. A rule may be executed, for example, only when the value of the State Field is California. Rules that execute conditionally are identified in the Rules pane with ? symbol.
- A *trigger* is a rule that causes another business process to be started. After the triggered process completes, the triggering process may either continue or terminate. The Valid Phone Number rule in the rule set Validate Input might trigger the rule set Update File, for example. Triggering rules are identified in the Rules pane with a ➡ symbol.

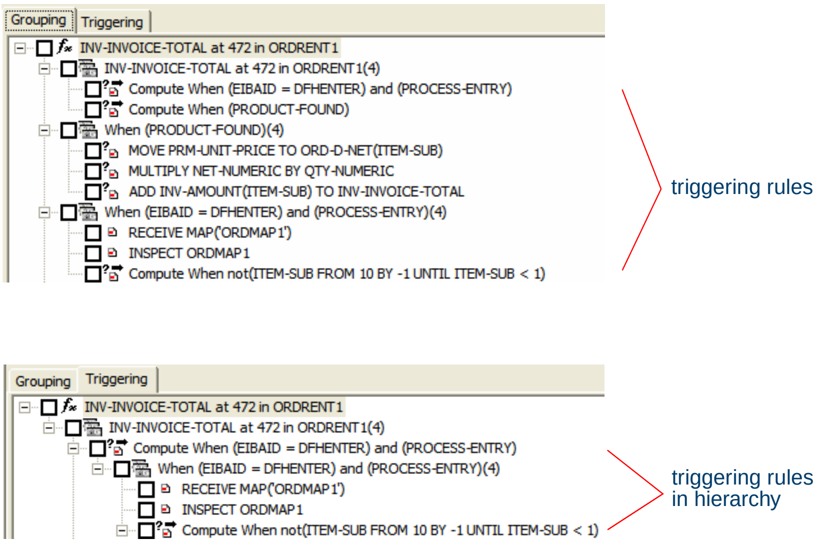
Understanding the Rules Pane Display

Choose **Rules** in the HyperView **View** menu to open the Rules pane (Figure 8-1). The lefthand pane displays the business functions in your workspace in tree form:

- The Grouping tab lists rules in order of execution, if they were auto-detected, or in the order you created them manually, organized by business function and rule set (Figure 8-2).
- The Triggering tab lists rules in a hierarchy of triggers, in which each triggering rule can be a parent to triggering rules that depend on it in program logic (Figure 8-2).

The righthand pane displays the attributes and other characteristics of the business function, rule set, or rule selected in the lefthand pane.

Figure 8-2 *Grouping Tab vs. Triggering Tab*



```
INV-INVOICE-TOTAL at 472 in ORDRENT1
  INV-INVOICE-TOTAL at 472 in ORDRENT1(4)
    Compute When (EIBAID = DFHENTER) and (PROCESS-ENTRY)
    Compute When (PRODUCT-FOUND)
  When (PRODUCT-FOUND)(4)
    MOVE PRM-UNIT-PRICE TO ORD-D-NET(ITEM-SUB)
    MULTIPLY NET-NUMERIC BY QTY-NUMERIC
    ADD INV-AMOUNT(ITEM-SUB) TO INV-INVOICE-TOTAL
  When (EIBAID = DFHENTER) and (PROCESS-ENTRY)(4)
    RECEIVE MAP('ORDMAP1')
    INSPECT ORDMAP1
    Compute When not(ITEM-SUB FROM 10 BY -1 UNTIL ITEM-SUB < 1)
```

```
INV-INVOICE-TOTAL at 472 in ORDRENT1
  INV-INVOICE-TOTAL at 472 in ORDRENT1(4)
    Compute When (EIBAID = DFHENTER) and (PROCESS-ENTRY)
      When (EIBAID = DFHENTER) and (PROCESS-ENTRY)(4)
        RECEIVE MAP('ORDMAP1')
        INSPECT ORDMAP1
        Compute When not(ITEM-SUB FROM 10 BY -1 UNTIL ITEM-SUB < 1)
```

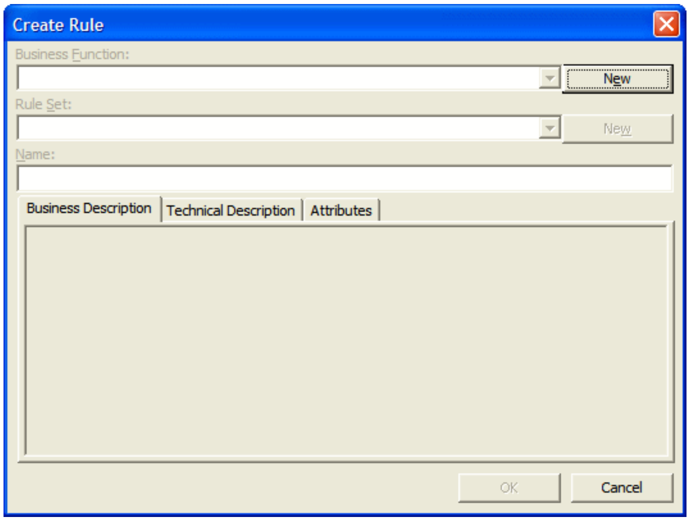
Extracting Business Rules Manually

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 8-22.

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 business function or rule set (if any) for the new rule, then choose **Create Rule** in the **Rules** menu. The Create Business Rule dialog opens (Figure 8-3).

Figure 8-3 *Create Business Rule Dialog*



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.

2 In the **Business Function** drop-down, select the business function for the rule. The current business function is displayed. To create a new business function, click **New**, then follow the instructions in "Creating Business Functions" on page 8-19.

3 In the **Rule Set** drop-down, select the rule set for the rule. The current rule set is displayed. To create a new rule set, click **New**, then follow the instructions in "Creating Rule Sets" on page 8-19.

4 Specify the rule's attributes:

- In the **Name** field, enter the name of the rule.
- On the Business Description tab, enter the business description for the rule.
- On the Technical Description tab, enter the technical description for the rule.
- On the Attributes tab, specify the other attributes of the rule.

For more information on rule attributes, see "Editing Rule Attributes" on page 8-23. Click **OK**. As shown in Figure 8-1, 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 8-18.
- 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 8-23.

Autodetecting Business Rules

BRM offers two autodetect methods:

- The backward autodetect method identifies statements containing fields that impact a *seed item:* a variable, dataport, or I/O statement with nested variables.
- The screen validation autodetect method identifies statements that validate screen data.

Tip: For either method, BRM creates an HTML diagram of the results in the \Workspace\\Output folder.

Invoking the Backward Autodetect Method

The backward autodetect method identifies statements containing fields that impact a *seed item:* a variable, dataport, or I/O statement with nested variables. The method:

- Traces the flow of data from the seed item to its source, assigning each statement in the flow to a new rule.
- Detects when a rule’s execution completes one business process and triggers another, assigning the rules for each business process to a discrete rule set, and, optionally, assigning the rule sets for a called program to a discrete business function.
- Identifies the control conditions that govern the execution of a rule, and any business names you have assigned to controlling variables in your project glossary.

Tip: Control conditions are also listed in the Technical Description of the business function.

- 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 backward autodetect method in BRM as described below, or for every variable in a Clipper list, as described in “Autodetecting Business Rules in Clipper” on page 4-8. In either case, you can “prune out” of the trace conditional logic not of interest to you.

Options determine the impact relationships traced by the backward autodetect, how it handles called programs, whether it groups or filters out rules based on the paragraphs they are mined from, and how it handles data items. For naming conventions, see “Understanding Autodetect Results” on page 8-13.

Important: Set **Enable Impact Report** and **Enable Data Element Flow** in the project verification options when you verify the project to enable the impact trace, as described in *Preparing Projects* in the workbench documentation set.

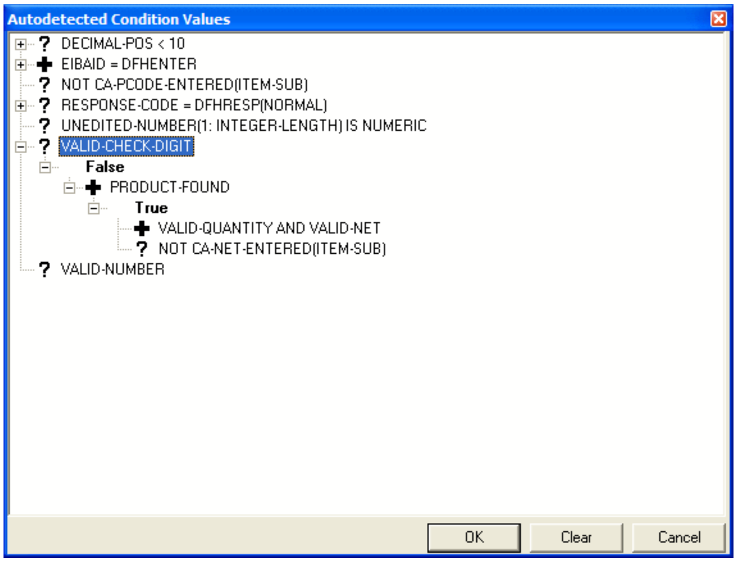
To invoke the backward autodetect method in BRM:

1. Set autodetect options, as described in “Setting Options on the AutoDetect Tab” on page 8-43.
2. In the HyperView Source or Context pane, select the seed item.
3. In the Rules pane, click the ▾ button next to the button on the tool bar and choose **Backward AutoDetect** in the drop-down menu. Cancel autodetection at any time during the process by clicking the **Cancel** button in the upper righthand corner of the HyperView window.
4. If the autodetect trace detects conditional logic, the Autodetected Condition Values dialog (Figure 8-4) prompts you to “prune out” logic branches not of interest to you, as described in “Pruning Autodetect Results” on page 8-12. If you set a condition to True, for example, the False branch (assuming one exists) is not explored by the trace. When you are satisfied with your choices in the Autodetected Condition Values dialog, click **OK**.
5. The autodetect method creates a rule for each code segment that impacts the selected item. HyperView displays an informational message that tells you how many rules it created. Click **OK**.

Pruning Autodetect Results

If the autodetect trace detects conditional logic, the Autodetected Condition Values dialog (Figure 8-4) gives you the option of “pruning out” logic branches not of interest to you. If you set a condition to True, for example, the False branch (assuming one exists) is not explored by the trace.

Figure 8-4 *Autodetected Condition Values Dialog*



```
⊞ ? DECIMAL-POS < 10
⊞ ✚ EIBAID = DFHENTER
  ? NOT CA-PCODE-ENTERED(ITEM-SUB)
⊞ ? RESPONSE-CODE = DFHRESP(NORMAL)
  ? UNEDITED-NUMBER(1: INTEGER-LENGTH) IS NUMERIC
⊟ ? VALID-CHECK-DIGIT
    ⊟ False
        ⊟ ✚ PRODUCT-FOUND
            ⊟ True
                ✚ VALID-QUANTITY AND VALID-NET
                ? NOT CA-NET-ENTERED(ITEM-SUB)
  ? VALID-NUMBER
```

In the dialog, click the plus sign (+) next to an item to expand its tree one level. Click the minus sign (-) to collapse the tree one level. Symbols are as follows:

- ✚ means that the value of the condition is True.
- ▬ means that the value of the condition is False.
- ? means that the value of the condition is not determined. Select the item and choose **True** in the right-click menu to set the value to True, **False** to set it to False. Choose **Any Value** to restore the item to its original state. Click **Clear** to restore all the items to their original state.

When you are satisfied with your choices in the Autodetected Condition Values dialog, click **OK** to dismiss the dialog and continue the trace.

Note: Deselect **Set values of conditions manually** on the AutoDetect tab in the Project options for Business Rule Manager to disable condition pruning.

Understanding Autodetect Results

The business function and first rule set in the autodetect results have names of the form *Seed Name* at *Line Number* in *Program*. After the first rule set:

1. Rules mined from the seed program appear in rule sets named for the paragraphs they are mined from, except as described in item 3.
2. Rules grouped by the paragraph they are mined from (see “Grouping Autodetected Rules” on page 8-15) appear in rule sets with names of the form Paragraph *Paragraph Name.*
3. If a rule set contains rules mined from three or more statements with at least one common condition, and the rule set is triggered by a single rule executing under the conditions, the rule set has a name of the form When *Conditions*.
4. Rules mined from a different program appear in rule sets named for the program.

After the first autodetect run in which a rule set is mined, a number in parentheses ((*n*)) is appended to its name to indicate each subsequent run in which it was mined (whether or not you have deleted an instance in the meantime). For business functions and rule sets with the same name, a number in square brackets ([*n*]) is appended to the name to guarantee uniqueness.

Rules are named for the statement they are mined from, unless:

1. They are “containers” for filtered rules, in which case they have names of the form Execute *Paragraph Name*.
2. They trigger rule sets that group rules by the paragraph they are mined from, in which case they have names of the form Execute *Paragraph Name*.

3 They trigger rule sets that contain rules mined from three or more statements with at least one common condition, in which case they have names of the form:

- Compute *Triggered Rule Set* When *Conditions,* if the rule set is triggered by a single rule.
- Execute *Triggered Rule Set* When *Conditions,* if the rule set is triggered by multiple rules.

Autodetecting Called Programs

When you set the Impact tool to trace Calls relationships, backward autodetect generates a rule for each CALL statement, with a trigger to the first rule set of the called program. By default, both the triggering rule and the triggered rule set are displayed in the same business function. Use the Business Rule Manager autodetect options to:

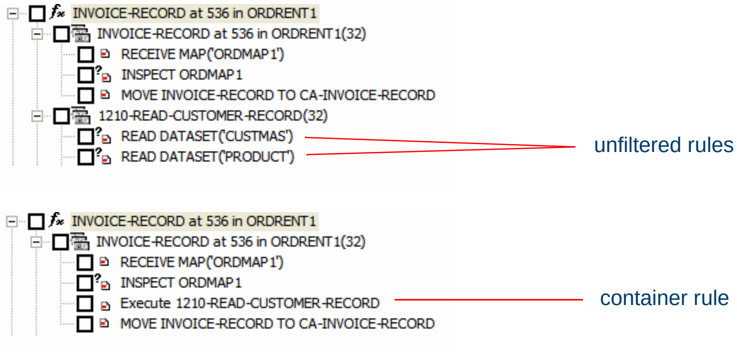
- Skip specified called programs.
- Group rule sets for specified called programs in discrete business functions.

For more information, see “Setting Options on the AutoDetect Tab” on page 8-43.

Filtering Autodetected Rules

Use the Business Rule Manager autodetect options to filter out autodetected rules based on the paragraph they are mined from. The filtered rules are replaced by a single “container” rule with a name of the form Execute *Paragraph Name,* in the executing rule set (Figure 8-5). For more information, see “Setting Options on the AutoDetect Tab” on page 8-43.

Figure 8-5 *Unfiltered vs. Filtered Autodetect*



```
□ INVOICE-RECORD at 536 in ORDRENT1
  □ INVOICE-RECORD at 536 in ORDRENT1(32)
    □ RECEIVE MAP('ORDMAP1')
    □ INSPECT ORDMAP1
    □ MOVE INVOICE-RECORD TO CA-INVOICE-RECORD
  □ 1210-READ-CUSTOMER-RECORD(32)
    □ READ DATASET('CUSTMAS')
    □ READ DATASET('PRODUCT')

□ INVOICE-RECORD at 536 in ORDRENT1
  □ INVOICE-RECORD at 536 in ORDRENT1(32)
    □ RECEIVE MAP('ORDMAP1')
    □ INSPECT ORDMAP1
    □ Execute 1210-READ-CUSTOMER-RECORD
    □ MOVE INVOICE-RECORD TO CA-INVOICE-RECORD
```

Grouping Autodetected Rules

Use the Business Rule Manager autodetect options to group autodetected rules based on the paragraphs they are mined from. Rules mined from the "paragraph of interest" you specify are assigned to a rule set with a name of the form Paragraph *Paragraph Name*. A triggering rule with a name of the form Execute *Paragraph Name* is created in the executing rule set (Figure 8-6). For more information, see "Setting Options on the AutoDetect Tab" on page 8-43.

Tip: Use this feature when you want to isolate the rules in a key paragraph, or when you want to use the name of the triggered rule set to search for rules.

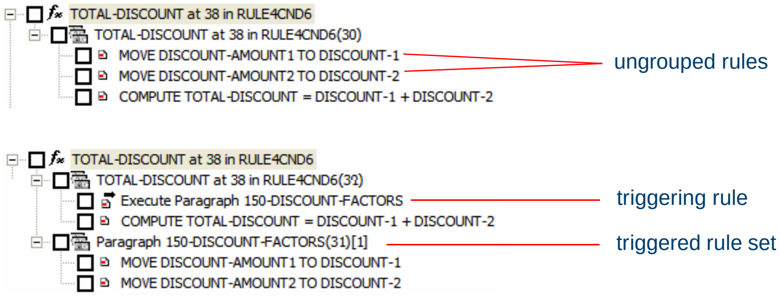
If the grouping rule set contains rules mined from three or more statements with at least one common condition, a distinct rule set is created for the rules with a triggering rule in the grouping rule set. For naming conventions, see "Understanding Autodetect Results" on page 8-13.

For statements executed multiple times:

- If the entire paragraph of interest is executed multiple times (if the paragraph is performed at different points in the program, for example), a single rule set is created with a triggering rule for each execution instance; each rule is appended with a number to guarantee uniqueness.

- If discrete statements in the paragraph of interest are executed at different times, a rule set is created for each statement, and a triggering rule is created for the rule set.

Figure 8-6 *Ungrouped vs. Grouped Autodetect*



```
□ TOTAL-DISCOUNT at 38 in RULE4CND6
  □ TOTAL-DISCOUNT at 38 in RULE4CND6(30)
    □ MOVE DISCOUNT-AMOUNT1 TO DISCOUNT-1
    □ MOVE DISCOUNT-AMOUNT2 TO DISCOUNT-2
    □ COMPUTE TOTAL-DISCOUNT = DISCOUNT-1 + DISCOUNT-2

□ TOTAL-DISCOUNT at 38 in RULE4CND6
  □ TOTAL-DISCOUNT at 38 in RULE4CND6(32)
    □ Execute Paragraph 150-DISCOUNT-FACTORS
    □ COMPUTE TOTAL-DISCOUNT = DISCOUNT-1 + DISCOUNT-2
  □ Paragraph 150-DISCOUNT-FACTORS(31)[1]
    □ MOVE DISCOUNT-AMOUNT1 TO DISCOUNT-1
    □ MOVE DISCOUNT-AMOUNT2 TO DISCOUNT-2
```

Autodetect Data Item Handling

Many data item evaluations do not involve business processes. When you set backward autodetect to trace constant moves (MOVE 1 to B), for example, there may be constant initializations you want the method to ignore.

Similarly, you may want backward autodetect to skip specified I/O elements, such as system variables, or variables initalized by zero or a blank line. Use the Business Rule Manager autodetect options to skip these items. For more information, see "Setting Options on the AutoDetect Tab" on page 8-43.

Invoking the Screen Validation Autodetect Method

The screen validation autodetect method identifies statements that validate screen data. The method:

- Traces from a receive screen statement each statement that validates screen input, assigning each statement to a new rule. The execution path from a screen field to its validation statement appears in the Technical Description attribute of the autodetected rule. A screen field may be validated directly or in a downstream variable to which its value is passed, or both.

- Detects the input/output data elements for each rule, and any business names you have assigned to them in your project glossary.

The business function and rule set for the autodetect results have names of the form Validation of Screen *Receive Screen Statement Name* at *Line Number* in *Program*. Rules have names of the form Validation of Screen *Statement Name* at *Line Number* in *Program.*

After the first autodetect run in which a rule set is mined, a number in parentheses ((*n*)) is appended to its name to indicate each subsequent run in which it was mined (whether or not you have deleted an instance in the meantime). For business functions and rule sets with the same name, a number in square brackets ([*n*]) is appended to the name to guarantee uniqueness.

To invoke the screen validation autodetect method:

1 In the HyperView Source or Context pane, select the receive screen statement from which you want to trace screen validations.

2 In the Rules pane, click the ▾ button next to the button on the tool bar and choose **Screen Validation** in the drop-down menu. Cancel autodetection at any time during the process by clicking the **Cancel** button in the upper righthand corner of the HyperView window.

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 “Autodetecting Business Rules in Clipper” 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 “Generating Rules in a Specified Rule Set” on page 4-9.

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 attributes in the righthand pane of the window.
- A rule set to view its attributes 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 8-23. For instructions on how to create rules manually, see "Extracting Business Rules Manually" on page 8-7.

Creating Business Functions To create a business function, choose **Create Business Function** in the **Rules** menu. The Business Function dialog opens. In the Business Function dialog, enter the attributes of the new business function and click **OK**. For a description of the attributes, see "Editing Rule Attributes" on page 8-23.

Creating Rule Sets To create a rule set, select the business function for the rule set in the lefthand pane and choose **Create Rule Set** in the **Rules** menu. The Create Rule Set dialog opens, with the current business function displayed in the **Business Function** drop-down. Select another business function if necessary, then enter the attributes of the new rule set and click **OK**. For a description of the attributes, see "Editing Rule Attributes" on page 8-23.

Tip: To create a new business function, click the **New** button next to the **Business Function** drop-down, then follow the instructions in "Creating Business Functions."

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 the rules in a rule set, click the check box for the rule set. To check all the rule sets and rules in a business function, click the check box for the business function.

To check all rules, choose **Select All** in the **Rules** menu. To uncheck all rules, choose **Unselect All** in the **Rules** menu. To invert the current selection, so that unselected rules are selected and selected rules are unselected, choose **Invert Selection** in the **Rules** menu.

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 or Copying Rules or Rule Sets To move or copy rules, place a check mark next to the rules and choose **Copy/Move Selected Rules** in the **Rules** menu. The Copy/Move Selected Rules dialog opens, where you can specify the rule set for the rules and whether you want to move or copy them.

To move or copy rule sets, place a check mark next to the rule sets and choose **Copy/Move Selected Rule Sets** in the **Rules** menu. The Copy/Move Selected Rule Sets dialog opens, where you can specify the business function for the rule sets and whether you want to move, copy, or reuse them. For copied rule sets, a number in curly brackets ({*n*}) is appended to the name to guarantee uniqueness. For reused rule sets, no change is made to the name.

Note: If you modify the attributes of a reused rule set, the changes appear in every instance. If you attempt to delete a reused rule set, you are prompted to confirm whether you want to delete every instance or only the selected instance.

Copying a Rule or Rule Set with a Different Name To copy a rule with a different name, select the rule and choose **Copy/Move** in the **Rules** menu. The Copy/Move dialog opens, where you can specify the rule set for the rule and the new name of the rule.

Note: If you select **Move** rather than **Copy** in this dialog, the rule is moved unchanged to the new location.

To copy a rule set with a different name, select the rule set and choose **Copy/Move** in the **Rules** menu. The Copy/Move dialog opens, where you can specify the business function for the rule set and the new name of the rule set. You can also specify a technical description or business description. The rules in the rule set are not copied to the new location.

Note: If you select **Move** rather than **Copy** in this dialog, the rule set and all its rules are moved unchanged to the new location.

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. To undo the deletion, choose **Undo Delete** in the **Rules** menu.

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, set options for the export, as described in "To specify options on the Rule Export tab:" on page 8-48. Select the items you want to export, then choose **Rule Export** in the **Rules** menu. You are prompted to continue. 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.

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.

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

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.

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.

Editing Rule Attributes

Before a 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, the last time it was validated, and so forth. You can view rule attributes in the righthand pane of the HyperView Rules window (Figure 8-1).

Tip: Use the Technical Description, Business Description, Audit, Classification, Status, and Transition attributes for a business function or rule set to indicate that a value applies for all the rules included in the business function or rule set. Use the Business Area attribute of a business function to classify business functions as members of a still broader category of activity, such as Customer Management.

You can propagate rule attribute values to one or more rules (see "Propagating Rule Attributes" on page 8-28), and you can use a variety of methods to batch edit rule attributes:

- Clipper lets you generate business rules in a specified rule set with predefined attributes. For more information, see "Generating Rules in a Specified Rule Set" on page 4-9.
- The search facility in the Rules pane lets you locate existing rules with common characteristics, which you can then modify in bulk with the Change Rules feature. For more information, see "Searching for Rules" on page 8-35 and "Rule Search Window" on page 8-36.

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 8-49.

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 business function, rule set, or rule when you create it. To edit the Name attribute, enter a new name in the **Name** field.

Editing the Business Area Attribute

The Business Area attribute of a business function identifies the business activity of which the business function is a part, such as Customer Management. To edit the Business Area attribute, enter appropriate text in the **Business Area** 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.

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 is supplied when a segment is validated or invalidated. For more information, see “Validating Rule Segments after Refreshing or Editing Code” on page 8-39.

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 8-49. 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 the 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.
- **Complete** if the rule has been redeveloped.
- **Obsolete** if the rule is obsolete.
- **Requires modification** if the rule requires modification for redevelopment.
- **Duplicate** if the item is a duplicate of a rule being used for redevelopment.

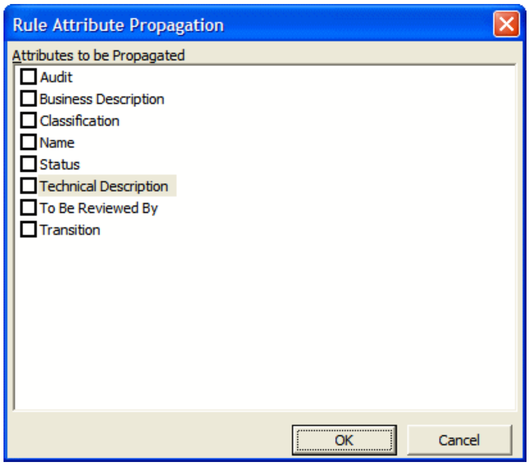
Propagating Rule Attributes

Follow the steps below to propagate one or more attribute values for a rule to another rule or a group of rules.

To propagate rule attributes:

1. Select the rules to which you want to propagate attribute values.
2. Click the rule whose attribute values you want to propagate and choose **Propagate from Highlighted to Selected** in the **Rules** menu. You are prompted to continue. Click **Yes**. The Attribute Propagation window opens (Figure 8-7).

Figure 8-7 *Attribute Propagation Window*



Rule Attribute Propagation

Attributes to be Propagated

- □ Audit
- □ Business Description
- □ Classification
- □ Name
- □ Status
- □ Technical Description
- □ To Be Reviewed By
- □ Transition

3 In the Attribute Propagation window, place a check mark next to each attribute whose value you want to propagate, then click **OK**.

Identifying Triggered Rule Sets

A *trigger* is a rule that causes another business process to be started. After the triggered process completes, the triggering process may either continue or terminate. The Valid Phone Number rule in the rule set Validate Input might trigger the rule set Update File, for example.

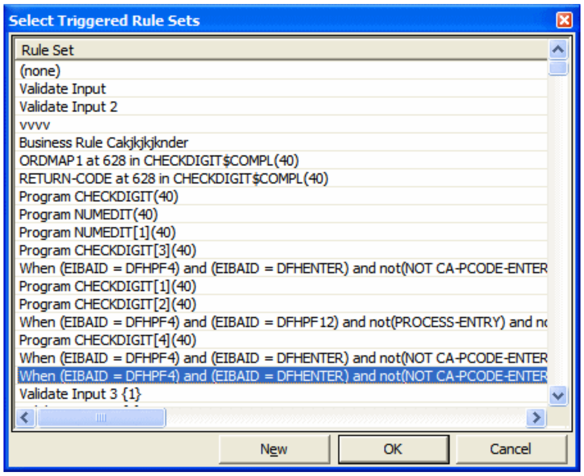
The **Triggers** field for a rule identifies the rule set the rule triggers. Triggering rules are identified in the Rules pane with a symbol. A rule set can be triggered by multiple rules. A rule can trigger the rule set it belongs to.

If a rule set is triggered by a rule, the trigger is listed in the lower portion of the righthand pane for the rule set. Double-click the name of a trigger in the list to navigate to the corresponding rule in the lefthand pane.

To identify a triggered rule set:

1. Select the triggering rule and click ⬚ button next to the **Triggers** field in the righthand pane. The Select Triggered Rule Sets window opens (Figure 8-8).
2. 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 "Creating Rule Sets" on page 8-19.

Figure 8-8 *Select Triggered Rule Sets Dialog*



Rule Set
(none)
Validate Input
Validate Input 2
vvvv
Business Rule Cakjkjkjknder
ORDMAP1 at 628 in CHECKDIGIT$COMPL(40)
RETURN-CODE at 628 in CHECKDIGIT$COMPL(40)
Program CHECKDIGIT(40)
Program NUMEDIT(40)
Program NUMEDIT[1](40)
Program CHECKDIGIT[3](40)
When (EIBAID = DFHPF4) and (EIBAID = DFHENTER) and not(NOT CA-PCODE-ENTER
Program CHECKDIGIT[1](40)
Program CHECKDIGIT[2](40)
When (EIBAID = DFHPF4) and (EIBAID = DFHPF12) and not(PROCESS-ENTRY) and nc
Program CHECKDIGIT[4](40)
When (EIBAID = DFHPF4) and (EIBAID = DFHENTER) and not(NOT CA-PCODE-ENTER
When (EIBAID = DFHPF4) and (EIBAID = DFHENTER) and not(NOT CA-PCODE-ENTER
Validate Input 3 {1}

Identifying I/O Data Elements

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

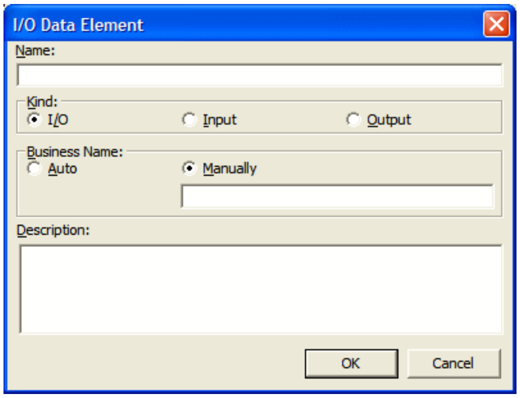
Use the autoretrieve feature to match I/O elements with the business names you've assigned to them in your project glossary, as described in Chapter 7, "Setting Up a Glossary." A business name identifies the business function of a variable in natural language, and is invaluable in documenting and retrieving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

Tip: Use the I/O Data Elements tab for a business function or rule set to specify the data elements for all the rules included in the business function or rule set.

To specify I/O data elements manually:

1 Select a business function, rule set, or rule, then right-click in the I/O Data Elements tab and choose **Create** in the pop-up menu. The I/O Data Element dialog opens (Figure 8-9).

Figure 8-9 *I/O Data Element Dialog*



I/O Data Element

Name:

Kind: ● I/O ○ Input ○ Output

Business Name: ○ Auto ● Manually

Description:

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 business function, rule set, or rule, and a data element with the same name but a different type is autodetected for the item, 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 step 1 through step 6 for each field you want to add to the rule.

Tip: To edit a field in the table, select the field and choose **Edit** in the right-click menu. The I/O Data Element dialog opens (Figure 8-9), where you can make the necessary changes. To delete a field, select it and choose **Delete** in the right-click menu.

To autodetect I/O data elements:

1 Select a rule, then right-click in the I/O Data Elements tab and choose **Detect I/O Data Elements** in the pop-up menu. You are prompted to continue. Click **Yes**.

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 from a glossary:

1. Select a business function, rule set, or rule, then right-click in the I/O Data Elements tab and choose **Populate I/O Business Names** in the pop-up menu. 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.

Identifying Control Conditions

Control Conditions identify the conditions that govern the execution of a rule. A rule may be executed, for example, only when the value of the State Field is California. Rules that execute conditionally are identified in the Rules pane with ? symbol.

Tip: Use the Control Conditions tab for a business function or rule set to specify the control conditions for all the rules included in the business function or rule set.

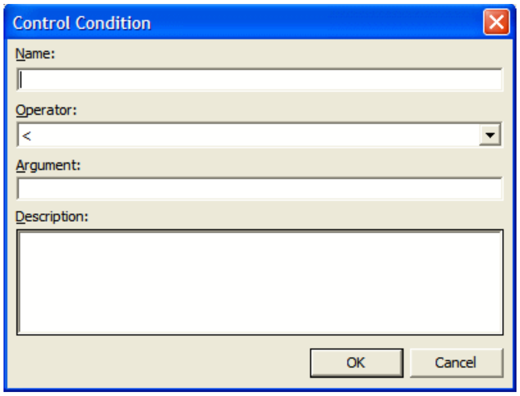
Use the autoreplace feature to replace variable names in control conditions with the business names you've assigned to them in your project glossary, as described in Chapter 7, "Setting Up a Glossary." A business name identifies the business function of a variable in natural language, and is invaluable in documenting and retrieving business rules. The field PLC_NUM, for example, might have the business name Policy Number.

Tip: Move a control condition up or down in the Control Conditions tab by selecting it and choosing **Move Up** or **Move Down** in the right.-click menu.

To specify control conditions:

1 Select a business function, rule set, or rule, then right-click in the Control Conditions tab and choose **Create Control Condition** in the pop-up menu. The Control Condition dialog opens (Figure 8-10).

Figure 8-10 *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, select the field and choose **Edit** in the right-click menu. The Control Condition dialog opens (Figure 8-10), where you can make the necessary changes. To delete a field, select it and choose **Delete** in the right-click menu.

7. Repeat step 1 through step 6 for each control condition you want to add to the rule.

To autoreplace variable names with business names from a glossary:

1. Select a business function, rule set, or rule, then right-click in the Control Conditions tab and choose **Substitute Business Names** in the pop-up menu. You are prompted to continue. Click **Yes**.
2. When the autoreplace process is complete, a dialog box displays the number of autoreplaced variable names. Click **OK**.

Searching for Rules

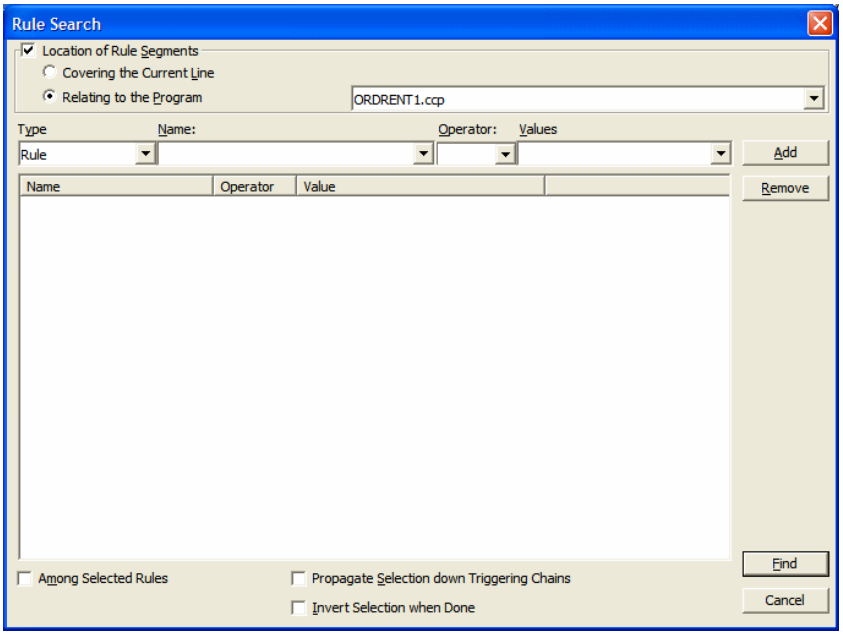
Use the search facility in the Rules pane to find rules based on segment location and/or attribute values. Use the Change Rules feature to batch-edit the rules returned by the search. For more information, see "Batch Processing Rules" on page 8-37.

To search for rules:

1. Select a business function, rule set, or rule and choose **Start Search** in the **Rules** menu. The Rule Search window opens (Figure 8-11).
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 type of object that contains the attribute in the **Type** combo box.
   - 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.

Figure 8-11 *Rule Search Window*


Rule Search
Location of Rule Segments
Covering the Current Line
Relating to the Program
Type
Name:
Operator:
Values
Add
Name
Operator
Value
Remove
Among Selected Rules
Propagate Selection down Triggering Chains
Invert Selection when Done
Find
Cancel

☑ Location of Rule Segments

○ Covering the Current Line

● Relating to the Program: ORDRENT1.ccp

Type: Rule

□ Among Selected Rules

□ Propagate Selection down Triggering Chains

□ Invert Selection when Done

4 To search only the selected rules, select **Among Selected Rules**.

5 To find rules in the triggering hierarchy for the selection (see "Understanding the Rules Pane Display" on page 8-6), select **Propagate Selection down Triggering Chains**.

6 To invert the search, so that rules not identified by the search criterion are found, choose **Invert Selection when Done**.

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

Batch Processing Rules

Use the Change Rules feature to batch process rule attributes. You can select batch-processing candidates manually, or use the search facility to select candidates automatically.

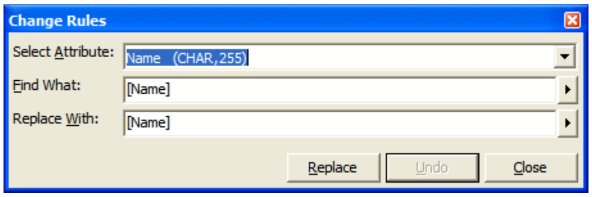
To batch process rules:

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 **Change Rule Attributes**. You are prompted to continue. Click **Yes**. The Change Rules window opens (Figure 8-12).

Figure 8-12 *Change Rules Window (After Step 3)*



Select Attribute: Name (CHAR,255)
Find What: [Name]
Replace With: [Name]

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.

For attributes with enumerated values, click the ▼ button next to the **Find What** and **Replace With** fields to display the values in a drop-down. For the remaining attributes, click the ▶ button to display the available attributes in a drop-down. Select an item in a drop-down to populate the adjacent field.

Note: Attributes of I/O data elements and control conditions can only be used to replace a Business Description or Technical Description. If there are multiple instance of I/O data elements or control conditions, they are displayed in the Business Description or Technical Description on separate lines.

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**. BRM modifies the selected rules. Click **Close** to dismiss the window and return to the Rules pane.

Synchronizing Sources

Suppose a rule set in the current project is triggered by a rule in another project. When you synchronize sources, BRM includes the program that contains the segment for the triggering rule in the current project.

To synchronize sources:

1 To synchronize sources, choose **Sync Sources** in the **Rules** menu.

Validating Rule Segments after Refreshing or Editing Code

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

The Segment Validity attribute of a rule indicates whether a rule segment no longer is valid. Use the Automatic Rule Validation feature to have the segment:

- Deleted from the rule, with the attribute set to Valid.
- Retained with the rule, with the attribute set to Invalid.
- Resynchronized with the rule (subject to the limitations described in “Some Limitations on Automatic Rule Validation”).

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**. The Automatic Rule Validation dialog opens. Follow the instructions for specifying how you want to handle invalid segments in “To specify options on the Automatic Rule Validation tab:” on page 8-46, then click **OK**. The validation process begins.

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.

Setting Business Rule Manager Project Options

Project options for Business Rule Manager determine:

- The rule name templates available when you use Clipper to create rules in a specified rule set.
- Autodetect behavior, including how backward autodetect handles called programs, whether it groups or filters out rules based on the paragraphs they are mined from, and how it handles data items.
- How BRM handles invalidated rule segments.
- Whether BRM automatically includes triggered rule sets when it performs a rule export.

Setting Options on the Rule Defaults Tab

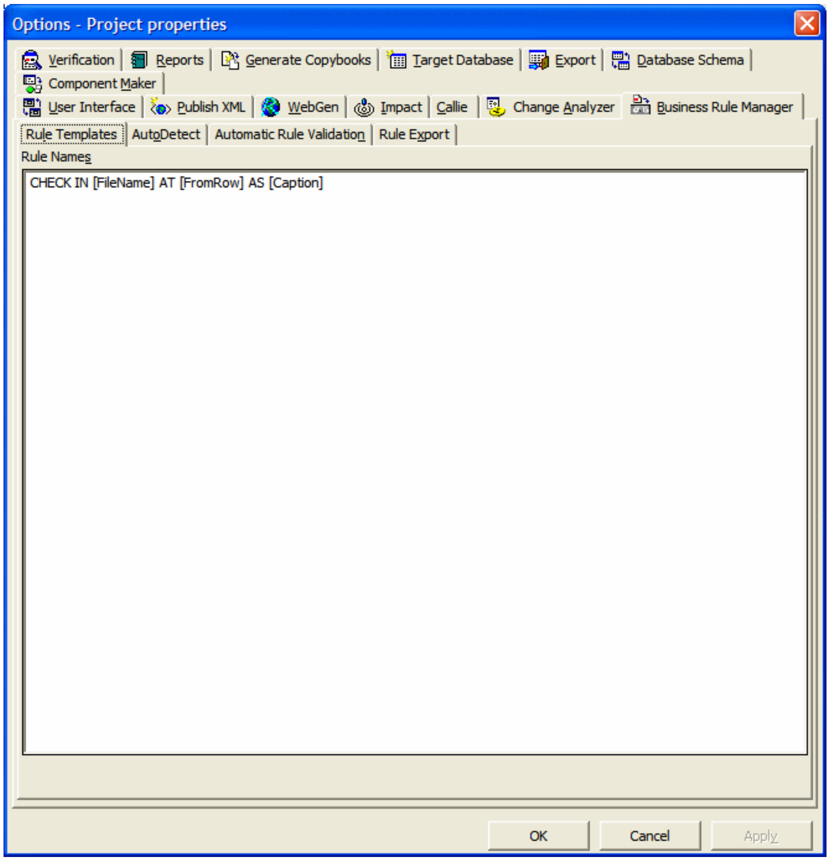
Use the Rule Defaults tab in the Project options for Business Rule Manager to determine the rule name templates available when you use Clipper to create rules in a specified rule set. For more information, see "Generating Rules in a Specified Rule Set" on page 4-9.

To specify options on the Rule Defaults tab:

1 Click the ▾ arrow next to the button on the tool bar and choose **Project Options** in the pull-down menu. The Project Options win-

dow opens. Click the Business Rule Manager tab, then the Rule Defaults tab (Figure 8-13).

Figure 8-13 *Rule Defaults Tab*



CHECK IN [FileName] AT [FromRow] AS [Caption]

2 To add a template to the list, right-click in the Used Rule Name Templates pane and choose **Add** in the pop-up menu. The system displays an empty text field next to a selected check box. Enter the new template in the field and click outside the field.

You can use any combination of text and bracketed property names in the template. So if you are creating rules for 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 Chapter 7, "Setting Up a Glossary."

3 To edit a template, select it and choose **Edit** in the right-click menu. To delete a template, select it and choose **Delete** in the right-click menu.

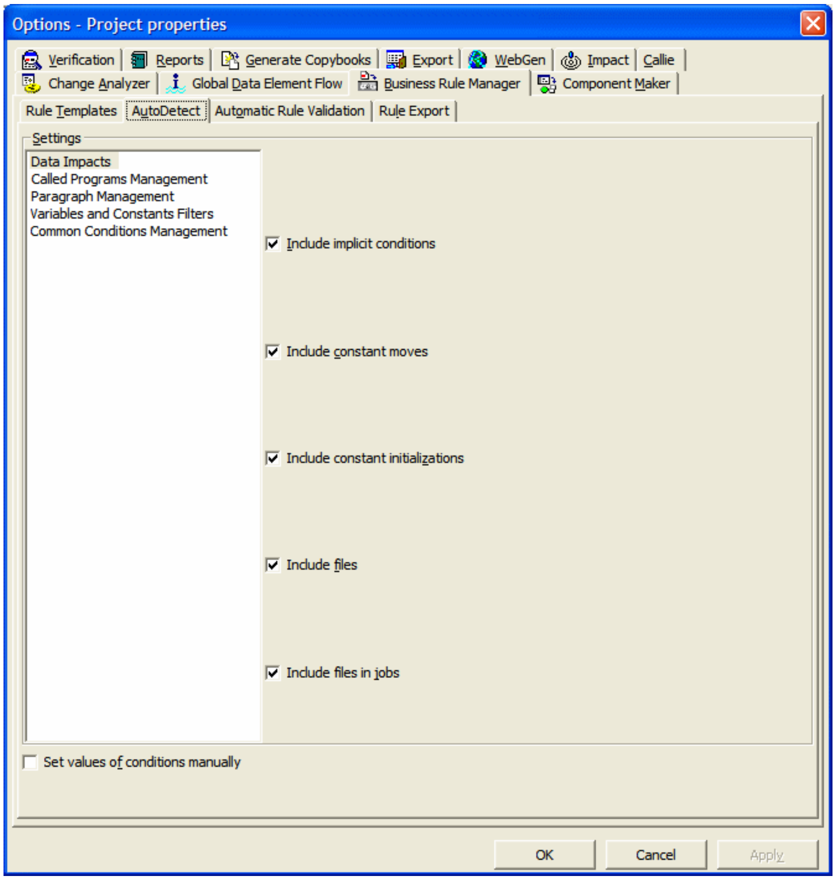
Setting Options on the AutoDetect Tab

Use the AutoDetect tab in the Project options for Business Rule Manager to specify autodetect behavior, including the impact relationships traced by the method, how it handles called programs, whether it groups or filters out rules based on the paragraphs they are mined from, and how it handles data items.

To specify options on the AutoDetect tab:

1 Click the ▾ arrow next to the button on the tool bar and choose **Project Options** in the pull-down menu. The Project Options window opens. Click the Business Rule Manager tab, then the AutoDetect tab (Figure 8-14).

Figure 8-14 *AutoDetect Tab*



☑ Include implicit conditions

☑ Include constant moves

☑ Include constant initializations

☑ Include files

☑ Include files in jobs

□ Set values of conditions manually

2 In the Settings pane, select Data Impacts. Select the impact relationships you want the autodetect to trace:

- **Include implicit conditions** if you want to trace relationships that depend on a statement that executes under the control of a condition, typically, GO TO, EXIT, or PERFORM...THRU.

- **Include constant moves** if you want to trace constants moved into data items (MOVE 1 TO B).
- **Include constant initializations** if you want to trace data items initialized by constants (03 A ... VALUE 1).
- **Include files** if you want to trace data communication via files. Traced only when corresponding JCL, ECL, FCT, or CSD files are verified.
- **Include files in jobs** if you want to trace data flow in JCL datasets when **Include files** is selected.

Note: The remaining panes in the AutoDetect tab prompts you to place a check mark next to an item listed in the pane.

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

To edit an item, select it and choose **Edit** in the right-click menu. To delete an item, select it and choose **Delete** in the right-click menu.

3. In the Settings pane, select Called Paragraph Management. In the:
   - Exclusion of called programs pane, place a check mark next to each called program you want the autodetect method to skip.
   - Business Functions from called programs pane, place a check mark next to each called program whose rule sets you want the autodetect method to group in a discrete business function.
4. In the Settings pane, select Paragraph Management. In the:
   - Filtered Paragraphs pane, place a check mark next to each paragraph whose rules you want the autodetect method to filter out of the results.

   - Paragraphs of interest pane, place a check mark next to each paragraph of interest you want the autodetect method to use in identifying results.

5. In the Settings pane, select Variables and Constants Filters. Select **Ignore variable initialization by zero or blank line** if you want the autodetect method to ignore variables initialized by zero or a blank line. In the:

   - Constants to ignore as initial values pane, place a check mark next to each initial value of a constant you want the autodetect method to ignore.
   - Templates of I/O data elements names to be skipped pane, place a check mark next to the template for each I/O data element you want the autodetect method to skip.

6. Select **Set values of conditions manually** if you want to enable condition pruning during the autodetect.

7. In the Settings pane, select Common Conditions Management. Specify how you want common control conditions for rules in a business function to be displayed:

   - **Business Function** if you want to display common control conditions in the business function only.
   - **Rules** if you want to display common control conditions in rules only.
   - **Business Function and Rules** if you want to display common control conditions in the business function and rules.

Setting Options on the Automatic Rule Validation Tab

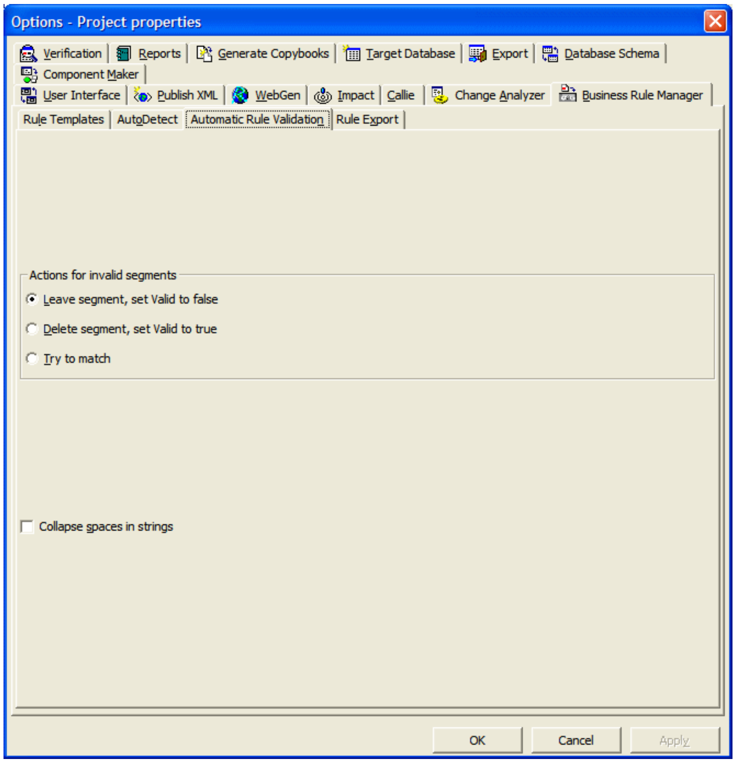
Use the Automatic Rule Validation tab in the Project options for Business Rule Manager to specify how you want BRM to handle invalidated rule segments. For more information, see “Validating Rule Segments after Refreshing or Editing Code” on page 8-39.

To specify options on the Automatic Rule Validation tab:

1. Click the ▾ arrow next to the 🛠 button on the tool bar and choose **Project Options** in the pull-down menu. The Project Options win-

dow opens. Click the Business Rule Manager tab, then the Automatic Rule Validation tab (Figure 8-15).

Figure 8-15 *Automatic Rule Validation Tab*



Actions for invalid segments

● Leave segment, set Valid to false

○ Delete segment, set Valid to true

○ Try to match

□ Collapse spaces in strings

2 In the Actions for invalid segments pane, select:

- **Leave segment, set Valid to false** if you want BRM to keep the invalid segment with the rule, but set the Segment Validity attribute of the rule to Invalid.
- **Delete segment, set Valid to true** if you want BRM to delete the invalid segment from the rule and set the Segment Validity attribute of the rule to Valid.
- **Try to match** if you want BRM to resynchronize the segment with the rule (subject to the limitations described in “Some Limitations on Automatic Rule Validation” on page 8-40).

3. Place a check mark next to **Collapse spaces in strings** if you have added spaces to strings in segments and want BRM to ignore the spaces.

Setting Options on the Rule Export Tab

Use the Rule Export tab in the Project options for Business Rule Manager to determine whether BRM automatically includes triggered rule sets when it performs a rule export. For more information, see “Exporting Rules, Rule Sets, and Business Functions” on page 8-22.

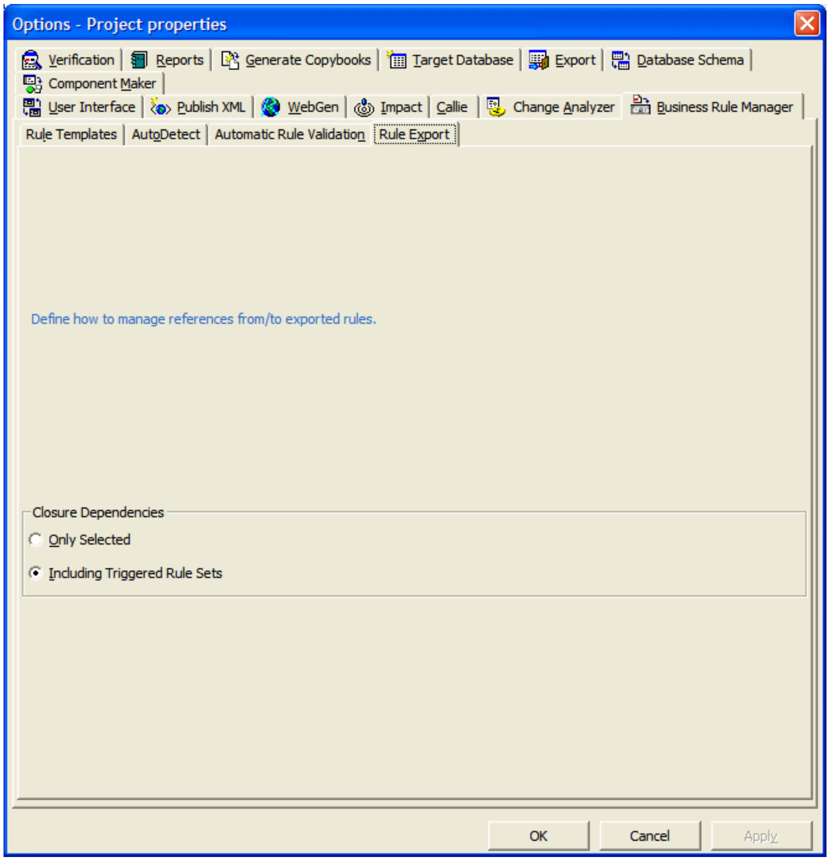
To specify options on the Rule Export tab:

1. Click the ▾ arrow next to the 🛠 button on the tool bar and choose **Project Options** in the pull-down menu. The Project Options window opens. Click the Business Rule Manager tab, then the Rule Export tab (Figure 8-16).

2. In the Close Dependencies pane, select:

   - **Only Selected** if you want BRM to export only the selected rules.
   - **Including Triggered Rule Set** if you want BRM to include triggered rule sets when it performs the rule export.

Figure 8-16 *Rule Export Tab*



Define how to manage references from/to exported rules.

Closure Dependencies

○ Only Selected

● Including Triggered Rule Sets

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. Modernization

Workbench 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 functions, rule sets, rules, the default value of the attribute is restored.

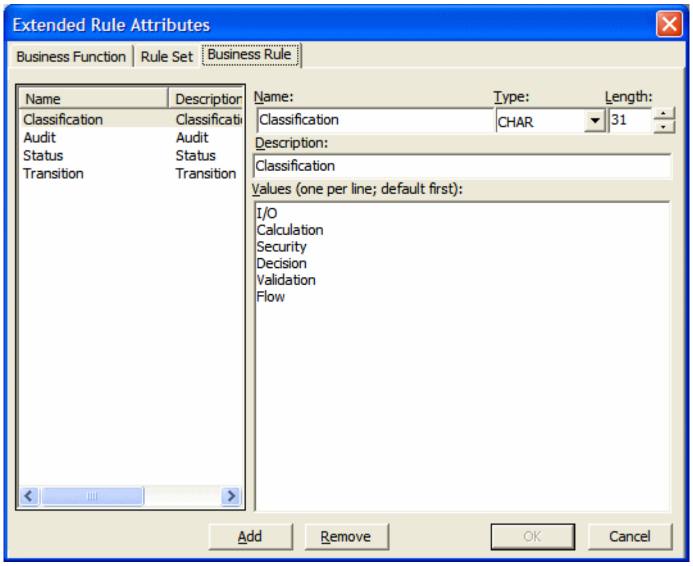
To customize attributes:

1. In the Windows **Start** menu, go to the **Programs** folder. Choose **Micro Focus**, then **Modernization Workbench Administration**. The Modernization Workbench Administration tool window opens.
2. In the **Administer** menu, choose **Edit Rule Attributes**.
   - If you have never customized attributes, the window opens automatically with the default attributes.
   - If you have already customized 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 Extended Rule Attributes window opens (Figure 8-17).
   - If you have already customized attributes, and the **Use Latest File** choice is selected in the **Administer** menu, the Extended Rule Attributes window opens with the customized attributes for the *.Repstry.xml file you last saved in the 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. Click the Business Function, Rule Set, or Business Rule tab as appropriate.

Figure 8-17 *Extended Rule Attributes Window*



Name	Description
Classification	Classificati
Audit	Audit
Status	Status
Transition	Transition

Name: Classification
Type: CHAR
Length: 31
Description: Classification
Values (one per line; default first):

```
I/O
Calculation
Security
Decision
Validation
Flow
```

4 To modify an existing attribute, select the attribute in the lefthand pane, then edit the characteristics or values you want to modify in the righthand pane. To remove an attribute, select it and click **Remove**.

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.

5 To create a new attribute, click **Add**, then edit the characteristics of the new attribute as necessary. The new attribute is added to the list in the lefthand pane.

6 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. Modernization Workbench applies the definitions to your business functions, rule sets, or 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.

Generating Reports

BRM reports make it easy to develop the kind of detailed specification you'll need for a code extraction or redevelopment project. The following reports are available:

- The *Business Rules Report* lists the business function, rule set, segments, attributes, data elements, and control condition definitions of selected rules.
- The *Process Outline Report* shows the call context in which rules are executed.
- The *Coverage Report* shows the percentage of program logic (including blank lines and comments) that contains rule segments.

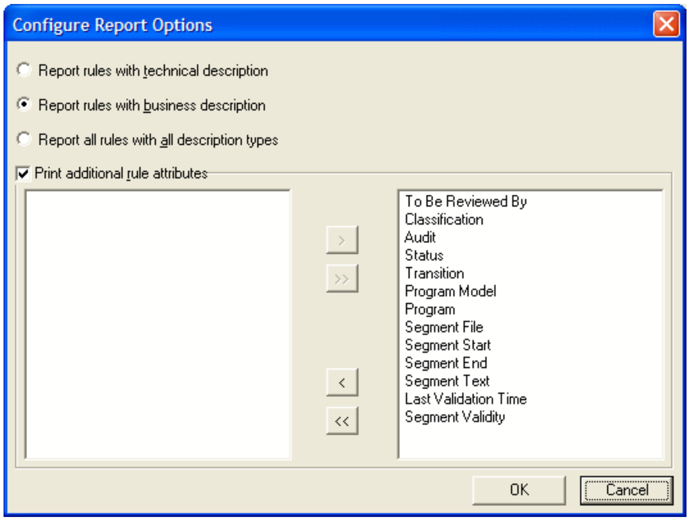
Generating Business Rule Reports

Business Rule Reports list the business function, rule set, segments, attributes, data elements, and control condition definitions of selected rules. Options let you filter out rules from the report and control the attributes displayed in the report.

To generate Business Rule Reports:

1. Select the business functions, rule sets, or rules you want to report on, click the button on the tool bar, and choose **Business Rule Report** in the pull-down menu. The Configure Report Options window opens (Figure 8-18).

Figure 8-18 *Configure Report Options Window*



Configure Report Options

○ Report rules with technical description

● Report rules with business description

○ Report all rules with all description types

☑ Print additional rule attributes

To Be Reviewed By
Classification
Audit
Status
Transition
Program Model
Program
Segment File
Segment Start
Segment End
Segment Text
Last Validation Time
Segment Validity

2 Select:

- **Report rules with technical description** if you want only rules with a technical description to be included in the report.
- **Report rules with business description** if you want only rules with a business description to be included in the report.
- **Report all rules with all description types** if you want all rules to be included in the report.

3 Place a check mark next to **Print additional rule attributes** if you want to display other rule attributes in the report.

In the righthand pane, select each attribute you want to display in the report and click the < button to move the attribute into the lefthand pane. Attributes appear in the report in the order they are listed in the lefthand pane. Click the << button to move all the attributes into the lefthand pane.

In the lefthand pane, select an attribute and click the > button to move the attribute back into the righthand pane. Click the >> button to move all the attributes into the righthand pane.

4 Click **OK**. The Business Rule Report window opens.

Generating Process Outline Reports

Process Outline Reports show the context in which rules are executed. The report lists every paragraph in the selected program, the call flow, or *process outline,* of the paragraph, every rule with a segment that starts in the paragraph, and the segment itself.

To generate Process Outline Reports:

1 Select the program you want to report on, click the ▤ button on the tool bar, and choose **Process Outline Report** in the pull-down menu. The Process Outline Report window opens.

Generating Coverage Reports

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.

To generate Coverage Reports:

1 Click the ▤ button on the tool bar and choose **Coverage Report** in the pull-down menu. The Coverage Report window opens.

Printing Reports

Click **Page Setup** in a report window to set up the page for a printed report. Click **Print t**o 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.

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.

Using the Batch Duplicate Finder

A

Use the Batch Duplicate Finder (BDF) to perform more sophisticated comparisons of similarly structured paragraphs than you can with the online duplicate search tool (see “Searching for Similarly Structured Code” on page 3-6). Unlike the online tool, BDF can be used to compare PL/I or RPG procedures as well as Cobol paragraphs. You can also use it to analyze paragraphs or procedures across workspaces, and to compare entire programs for their similarity.

Finding Duplicates

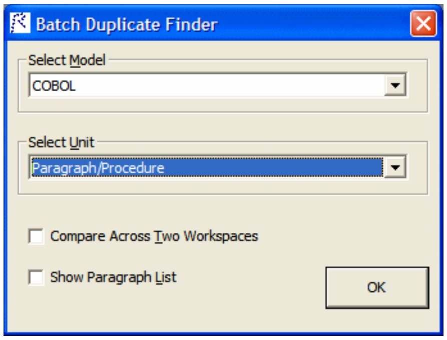
Make sure to verify the workspace files you plan to compare before running BDF.

To find duplicates:

1A If BDF is already open, choose **Startup Options** in the **Options** menu. The BDF scope window opens (Figure A-1).

1B If BDF is not already open, double-click the file *workbench home*\Bin\BDF.exe. The BDF scope window opens (Figure A-1)

Figure A-1 *BDF Scope Window*



Batch Duplicate Finder

Select Model: COBOL

Select Unit: Paragraph/Procedure

□ Compare Across Two Workspaces

□ Show Paragraph List

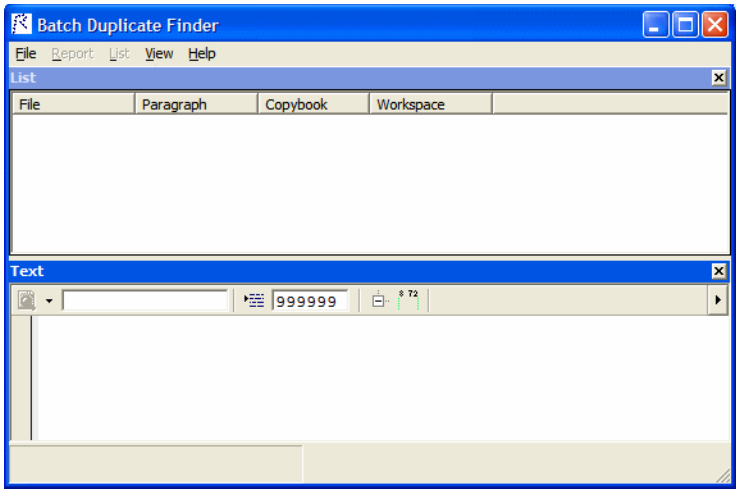
2 Specify the scope of the comparison:

- In the **Select Model** drop-down, select the object model for the workspace(s) to be analyzed: Cobol, PL/I, or RPG.
- In the **Select Unit** drop-down, select the unit of comparison: Paragraph/Procedure or Program.
- Select **Compare Across Two Workspaces** if you want to perform the comparison across workspaces.
- Select **Show Paragraph List** to view a list of paragraphs in the workspace(s) to be analyzed. You must choose this option if you want to select and mark the paragraphs to compare, as described in “Marking Items for Comparison” on page A-6. Otherwise, all paragraphs are marked for comparison.

Tip: For extremely large workspaces, deselect **Show Paragraph List** to avoid the performance cost of loading paragraphs in the BDF main window.

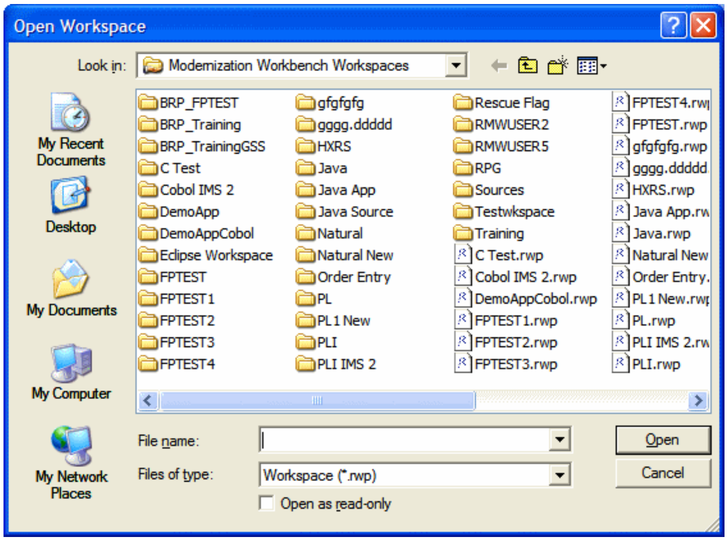
Click **OK**. The BDF main window opens (Figure A-2).

Figure A-2 *BDF Main Window*



3 In the **File** menu, choose **Open Workspace**. The Open Workspace dialog opens (Figure A-3)

Figure A-3 *Open Workspace Dialog*



Look in: Modernization Workbench Workspaces

File name:

Files of type: Workspace (*.rwp)

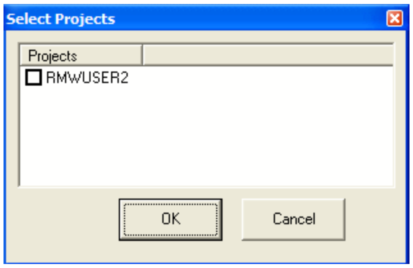
□ Open as read-only

4 In the Open Workspace dialog, select the workspace (.rwp) for the comparison, then click **Open**.

Note: If you are performing the comparison across workspaces, the Open Workspace dialog appears again. Select the second workspace and click **Open**.

The Select Projects window opens (Figure A-4).

Figure A-4 *Select Projects Window*



5 In the Select Projects window, select the projects for the comparison, then click **OK**.

Note: If you are performing the comparison across workspaces, the Select Projects window appears again. Select the projects for comparison from the second workspace and click **Open**.

- If you are performing the comparison in a single workspace, BDF loads the units of comparison in the List pane.
- If you are performing the comparison across workspaces, BDF loads the units of comparison for the first workspace at the top of the List pane with a value of A in the Workspace column, and for the second workspace at the bottom of the List pane with a value of B in the Workspace column.

Click an item to view its source in the Text pane. Click a column heading to sort items by that column. Grab-and-drag the border of a column heading to increase or decrease the width of the column. Select the appropriate choice in the **View** menu to show or hide a pane.

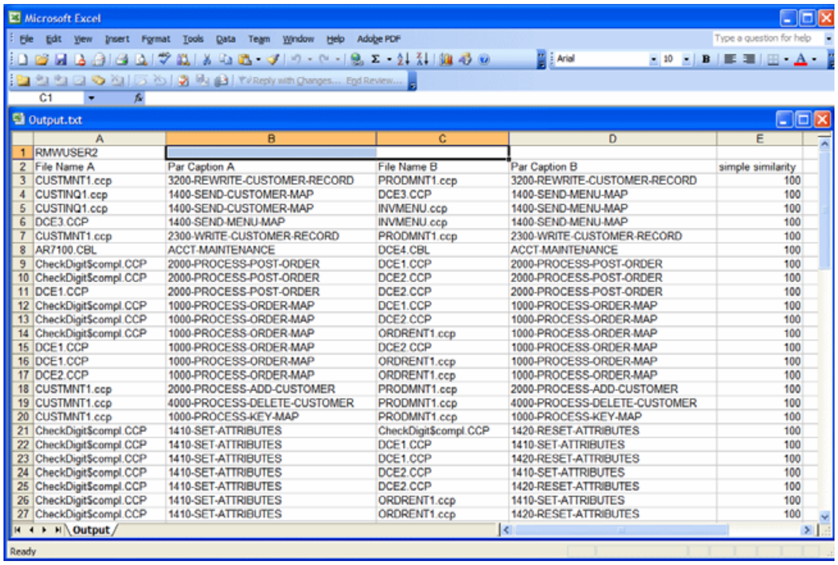
Note: If an item is contained in a copybook, the name of the copybook is displayed in the Copybook column.

6 Select and mark the items you want to compare, as described in "Marking Items for Comparison" on page A-6.

7 Set BDF search options as described in "Setting BDF Search Options" on page A-7.

8 In the **File** menu, choose **Find Duplicates**. BDF writes the results to the output file specified in the BDF search options. The output file renders the results in tab-delimited text. Use a spreadsheet editor to view and manipulate the results (Figure A-5).

Figure A-5 *BDF Output in Excel*


Microsoft Excel
File Edit View Insert Format Tools Data Team Window Help Adobe PDF
Type a question for help
Arial
10
Reply with Changes... End Review...
C1
Output.txt
Output
Ready

	A	B	C	D	E
1	RMWUSER2				
2	File Name A	Par Caption A	File Name B	Par Caption B	simple similarity
3	CUSTMNT1.ccp	3200-REWRITE-CUSTOMER-RECORD	PRODMNT1.ccp	3200-REWRITE-CUSTOMER-RECORD	100
4	CUSTINQ1.ccp	1400-SEND-CUSTOMER-MAP	DCE3.CCP	1400-SEND-MENU-MAP	100
5	CUSTINQ1.ccp	1400-SEND-CUSTOMER-MAP	INVMENU.ccp	1400-SEND-MENU-MAP	100
6	DCE3.CCP	1400-SEND-MENU-MAP	INVMENU.ccp	1400-SEND-MENU-MAP	100
7	CUSTMNT1.ccp	2300-WRITE-CUSTOMER-RECORD	PRODMNT1.ccp	2300-WRITE-CUSTOMER-RECORD	100
8	AR7100.CBL	ACCT-MAINTENANCE	DCE4.CBL	ACCT-MAINTENANCE	100
9	CheckDigit$compl.CCP	2000-PROCESS-POST-ORDER	DCE1.CCP	2000-PROCESS-POST-ORDER	100
10	CheckDigit$compl.CCP	2000-PROCESS-POST-ORDER	DCE2.CCP	2000-PROCESS-POST-ORDER	100
11	DCE1.CCP	2000-PROCESS-POST-ORDER	DCE2.CCP	2000-PROCESS-POST-ORDER	100
12	CheckDigit$compl.CCP	1000-PROCESS-ORDER-MAP	DCE1.CCP	1000-PROCESS-ORDER-MAP	100
13	CheckDigit$compl.CCP	1000-PROCESS-ORDER-MAP	DCE2.CCP	1000-PROCESS-ORDER-MAP	100
14	CheckDigit$compl.CCP	1000-PROCESS-ORDER-MAP	ORDRENT1.ccp	1000-PROCESS-ORDER-MAP	100
15	DCE1.CCP	1000-PROCESS-ORDER-MAP	DCE2.CCP	1000-PROCESS-ORDER-MAP	100
16	DCE1.CCP	1000-PROCESS-ORDER-MAP	ORDRENT1.ccp	1000-PROCESS-ORDER-MAP	100
17	DCE2.CCP	1000-PROCESS-ORDER-MAP	ORDRENT1.ccp	1000-PROCESS-ORDER-MAP	100
18	CUSTMNT1.ccp	2000-PROCESS-ADD-CUSTOMER	PRODMNT1.ccp	2000-PROCESS-ADD-CUSTOMER	100
19	CUSTMNT1.ccp	4000-PROCESS-DELETE-CUSTOMER	PRODMNT1.ccp	4000-PROCESS-DELETE-CUSTOMER	100
20	CUSTMNT1.ccp	1000-PROCESS-KEY-MAP	PRODMNT1.ccp	1000-PROCESS-KEY-MAP	100
21	CheckDigit$compl.CCP	1410-SET-ATTRIBUTES	CheckDigit$compl.CCP	1420-RESET-ATTRIBUTES	100
22	CheckDigit$compl.CCP	1410-SET-ATTRIBUTES	DCE1.CCP	1410-SET-ATTRIBUTES	100
23	CheckDigit$compl.CCP	1410-SET-ATTRIBUTES	DCE1.CCP	1420-RESET-ATTRIBUTES	100
24	CheckDigit$compl.CCP	1410-SET-ATTRIBUTES	DCE2.CCP	1410-SET-ATTRIBUTES	100
25	CheckDigit$compl.CCP	1410-SET-ATTRIBUTES	DCE2.CCP	1420-RESET-ATTRIBUTES	100
26	CheckDigit$compl.CCP	1410-SET-ATTRIBUTES	ORDRENT1.ccp	1410-SET-ATTRIBUTES	100
27	CheckDigit$compl.CCP	1410-SET-ATTRIBUTES	ORDRENT1.ccp	1420-RESET-ATTRIBUTES	100

Marking Items for Comparison

For small projects, you can mark the items you want to compare manually. For larger projects, you can mark the items in batch mode, by defining a mask with the Select Paragraphs or Select Programs feature.

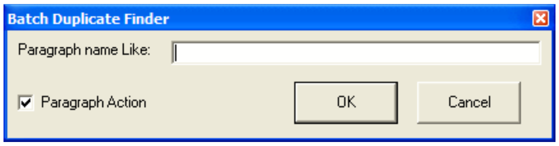
Marking Items Manually Place a check mark next to an item to mark it. To mark all the items in a section, choose **Check All** in the **List** menu. To unmark all the items in the section, choose **Uncheck All**.

To mark selected items in a section, select the items and choose **Check Selected** in the **List** menu. To unmark selected items in the section, choose **Uncheck Selected**.

Note: To select a range of items, hold down the Shift key, click the first item in the range, then click the last item in the range. To select items that are not in a range, hold down the Control key, then click each item you want to select.

Marking Items in Batch Mode To mark items in batch mode, choose **Select Paragraphs** or **Select Programs i**n the **List** menu. The BDF Define Mask window opens (Figure A-6).

Figure A-6 *Define Mask Window (for Paragraphs)*



☑ Paragraph Action

Enter the mask for the items you want to select in the **Paragraph name Like** or **Program name Like** field, then click **OK**. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA). Deselect **Paragraph Action** or **Program Action** to select items that do not match the mask. To mark the items selected with the mask, choose **Check Selected** in the **List** menu.

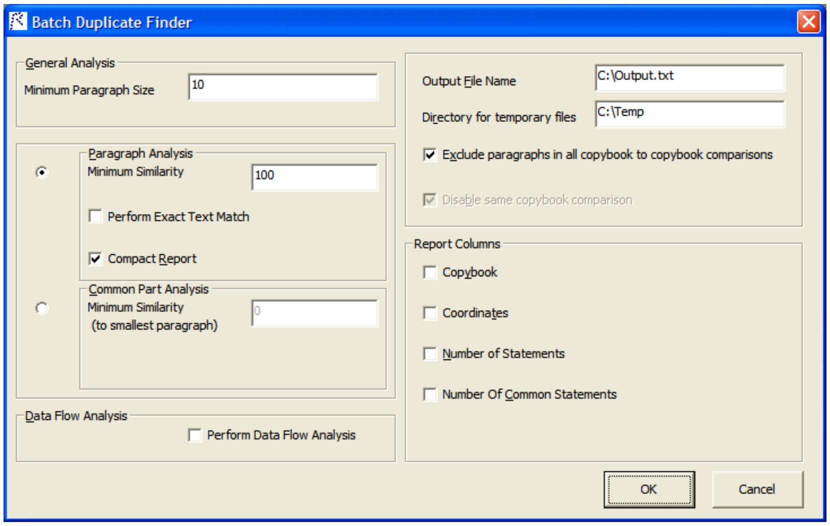
Setting BDF Search Options

BDF search options control the types of comparison BDF performs, the columns it displays in the output, the name and location of the output file, and how it handles copybooks.

To set BDF search options:

1 In the **Options** menu, choose **Search Options**. The BDF Options window opens (Figure A-7).

Figure A-7 *BDF Options Window (for Paragraphs)*



Batch Duplicate Finder

General Analysis
Minimum Paragraph Size: 10

● Paragraph Analysis
Minimum Similarity: 100
□ Perform Exact Text Match
☑ Compact Report

○ Common Part Analysis
Minimum Similarity (to smallest paragraph): 0

Data Flow Analysis
□ Perform Data Flow Analysis

Output File Name: C:\Output.txt
Directory for temporary files: C:\Temp
☑ Exclude paragraphs in all copybook to copybook comparisons
☑ Disable same copybook comparison

Report Columns
□ Copybook
□ Coordinates
□ Number of Statements
□ Number Of Common Statements

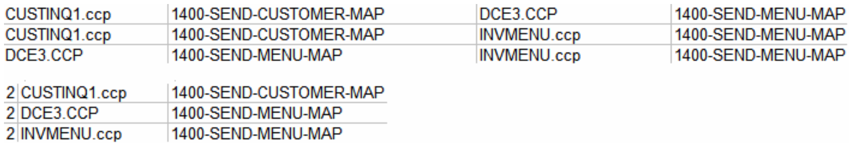
2 Choose **General Analysis** if you want to compare items on the basis of the minimum number of executable statements they contain. Enter the number in the **Minimum Paragraph Size** or **Minimum Program Size** field. An entry of 0 means that all items satisfy the test.

3 Choose the type of comparison you want to perform. If a pair of paragraphs or programs fails to satisfy the specified value for the comparison, the pair is excluded from the report. Select:

- **Paragraph Analysis** or **Program Analysis** if you want to compare the statements in the pair for their minimum similarity on the *Levenshtein edit distance* metric, as described in "Understanding Paragraph or Program Analysis" on page A-9. Select **Perform Exact Text Match** if you want to compare the entire text of the pair (including paragraph names, parameter names, comments, and blank lines) for exact matches.

Tip: Select **Compact Report** if you want the output to show items with 100% similarity in a numbered group, rather than with a line for each pair (Figure A-8)

Figure A-8 *Comparing Non-compact and Compact Report Output in Excel*



CUSTINQ1.ccp	1400-SEND-CUSTOMER-MAP	DCE3.CCP	1400-SEND-MENU-MAP
CUSTINQ1.ccp	1400-SEND-CUSTOMER-MAP	INVMENU.ccp	1400-SEND-MENU-MAP
DCE3.CCP	1400-SEND-MENU-MAP	INVMENU.ccp	1400-SEND-MENU-MAP

2	CUSTINQ1.ccp	1400-SEND-CUSTOMER-MAP
2	DCE3.CCP	1400-SEND-MENU-MAP
2	INVMENU.ccp	1400-SEND-MENU-MAP

- **Common Part Analysis** if you want to compare the statements in the pair for their minimum similarity on the *longest common sequence (LCS)* metric, as described in "Understanding Common Part Analysis" on page A-9.

For either type of comparison, select **Perform Data Flow Analysis** if you want to perform an additional test to determine whether similar items perform data flow operations with the same pattern, as described in "Understanding Data Flow Analysis" on page A-10.

4 In the **Output File Name** field, enter the pathname of the BDF output file. In the **Directory for temporary files** field, enter the pathname of the directory for temporary files.

5 Specify copybook handling. Select:

- **Exclude paragraphs in all copybook to copybook comparisons** if you want to prevent comparison of a paragraph in a copybook against any paragraph from any copybook, including the same copybook.
- **Disable same copybook comparison** if you want to prevent comparison of a paragraph in a copybook against any paragraph in the same copybook.

6 In the Report Columns group box, select:

- **Copybook** to include the Copybook column in the output.

- **Coordinates** to include source code coordinates in the output.
- **Number of statements** to include the number of statements in an item in the output.
- **Number of common statements** to include the number of common statements in an item in the output.

Understanding Paragraph or Program Analysis

Select **Paragraph Analysis** or **Program Analysis** in the BDF search options if you want to compare items for their minimum similarity on the *Levenshtein edit distance* metric, namely, the number of operations needed to transform one item into the other, where an operation is an insertion, deletion, or substitution of a single character. The formula for this comparison type is:

```
output = 100 -(d/Max(a,b)) * 100
```

where d is the Levenshtein edit distance, a is the length of item A, and b is the length of item B.

Note: Only the statements themselves, not their parameters, are considered in performing the comparison.

Enter the minimum similarity in the **Minimum Similarity** field. A value of 100 means that only items containing exactly the same statements in exactly the same order satisfy the test. Enter 0 to disable the test.

Tip: You cannot use this comparison type to find paragraphs that CONTAIN or are INCLUDED in other paragraphs.

Understanding Common Part Analysis

Select **Common Part Analysis** in the BDF search options if you want to compare items for their minimum similarity on the *longest common sequence (LCS)* metric, namely, the number of characters in the longest sequence of characters the items have in common. The formula for this comparison type is:

```
output = (L/Min(a,b)) * 100
```

where L is the LCS, a is the length of item A, and b is the length of item B.

Note: Only the statements themselves, not their parameters, are considered in performing the comparison.

Enter the minimum similarity in the **Minimum Similarity** field. A value of 100 means that only items containing exactly the same statements in exactly the same order satisfy the test. Enter 0 to disable the test.

Tip: Use this comparison type to narrow the items of interest to those with common character sequences.

Understanding Data Flow Analysis

For either Paragraph/Program Analysis or Common Part Analysis, select **Perform Data Flow Analysis** in the BDF search options to perform an additional test to determine whether similar items perform data flow operations with the same pattern. In data pattern analysis, if item A contains:

```
MOVE A TO B
MOVE B TO C
```

and item B contains:

```
MOVE X TO Y
MOVE Y TO Z
```

their data pattern is regarded as identical. Whereas if item A contains:

```
MOVE A TO B
MOVE B TO C
```

and item B contains:

```
MOVE X TO Y
MOVE X TO Z
```

their data pattern is regarded as different.

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 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.

Index