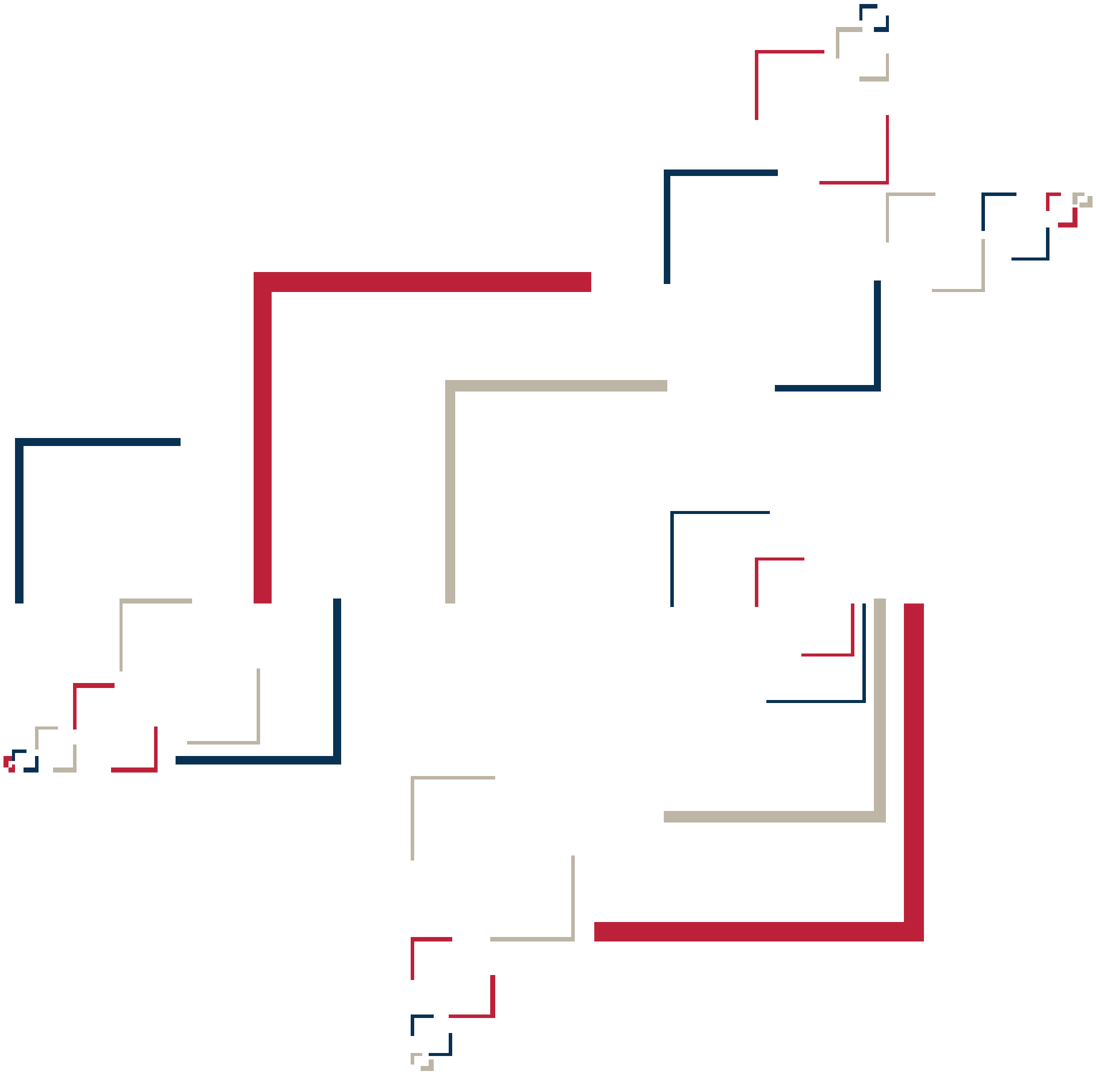


Micro Focus®

Modernization Workbench™

Analyzing Programs

MICRO FOCUS®

Micro Focus (IP) Ltd.
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
http://www.microfocus.com

Contents

Introducing HyperView

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, extract business rules, and much more.

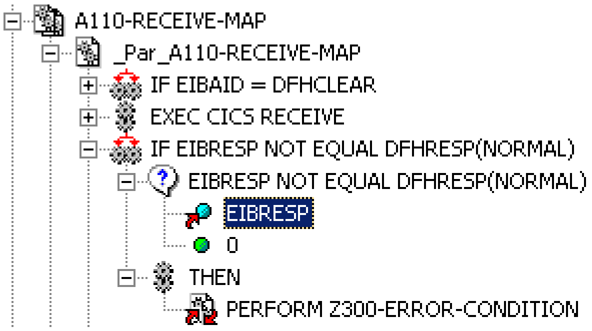
HyperView is designed primarily for programs, but you can also use it to analyze map files, JCL or ECL files, IDMS schemas, and the like.

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.

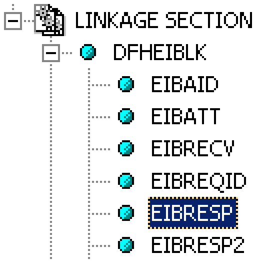
A variable, for example, can be related in the construct model to its declaration, a dataport (if it is used in an I/O statement), or a condition (if the condition uses an expression of which the variable forms a part). You view the construct model for a source file in the HyperView Context pane.

The figure below 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).



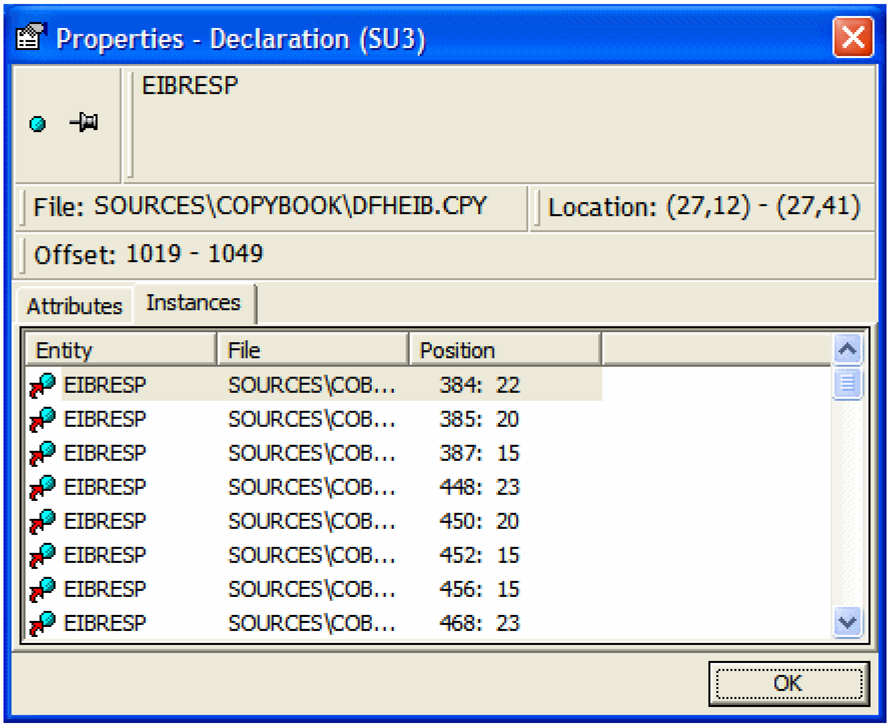
```
⊟ A110-RECEIVE-MAP
    ⊟ _Par_A110-RECEIVE-MAP
        ⊞ IF EIBAID = DFHCLEAR
        ⊞ EXEC CICS RECEIVE
        ⊟ IF EIBRESP NOT EQUAL DFHRESP(NORMAL)
            ⊟ EIBRESP NOT EQUAL DFHRESP(NORMAL)
                  EIBRESP
                  0
            ⊟ THEN
                  PERFORM Z300-ERROR-CONDITION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

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. Select EIBRESP in the Context pane and choose **Edit > Declaration**.



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. Select the declaration in the Context pane and choose **Edit > Instances**.



Properties - Declaration (SU3)

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.

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.

You can invoke HyperView directly by selecting a source file in the Repository Browser and choosing **Analyze > Interactive Analysis**. 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.

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. 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 pane. Choose the type of file to display in the drop-down under the title bar of the Objects pane. Select the file you want to open in the list below the drop-down.

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. **Edit** menu choices and pane-specific menu choices have equivalent choices in the right-click menu.

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.

The HyperView window's state (size, location, configuration, and option settings) is saved across sessions. HyperView provides access to the workbench Activity Log and relevant options windows. To view these items, select the appropriate choice in the **View** menu.

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

For Component pane usage, see *Creating Components* in the workbench documentation set.

Using Basic Navigation Controls

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

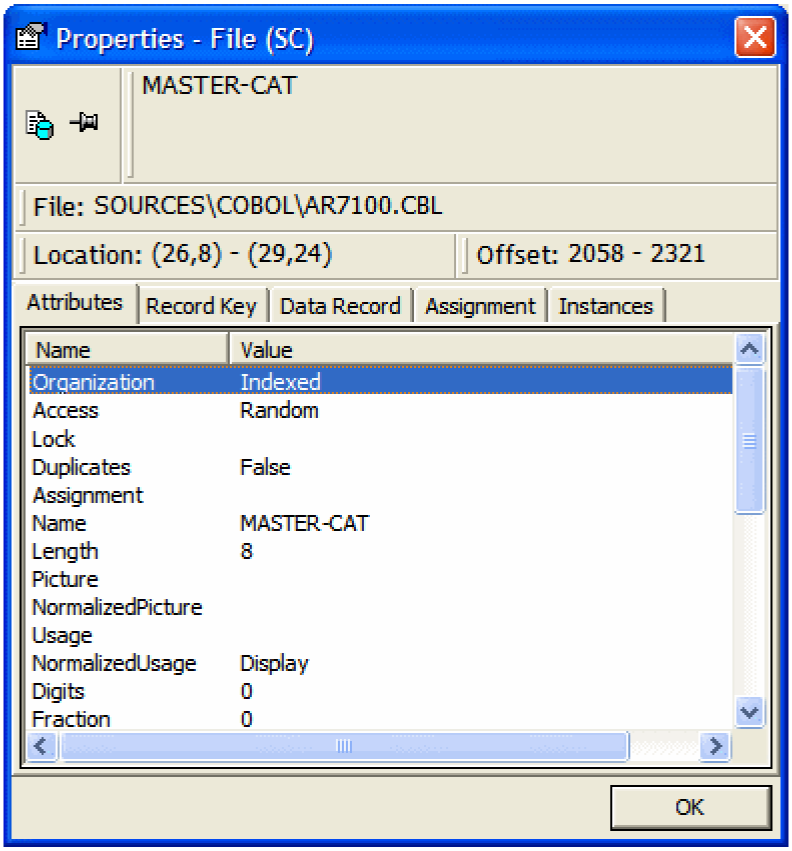
- Click the ⇦ button on the tool bar to navigate backward in the history of your selections in the HyperView window (regardless of the pane in which they were made). Click the adjacent ▾ button to display the selection history in a drop-down menu. Choose a selection by clicking it in the menu.
- Click 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

The HyperView Properties window for a construct identifies its attributes and related constructs. Attributes are the construct's type, name, location, and characteristics. Related constructs are the constructs with which the construct interacts. The Properties window is not modal, so you can leave it up throughout your HyperView session.

Opening the Properties Window

You can open the Properties window from 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.



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

To view related constructs in the Properties window, 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.

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

Opening a Properties Window for a Related Construct

To open a Properties window for a related construct, select the related construct in the Properties window and choose **Properties** in the right-click menu.

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 Manually

To assign a business name and business description to an object, select the object and choose **Set Business Attributes** in the right-click menu. A dialog box opens, where you can enter the business name and business description. To unassign a business name or business description, simply delete the value in the dialog box.

Understanding Program Context

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:

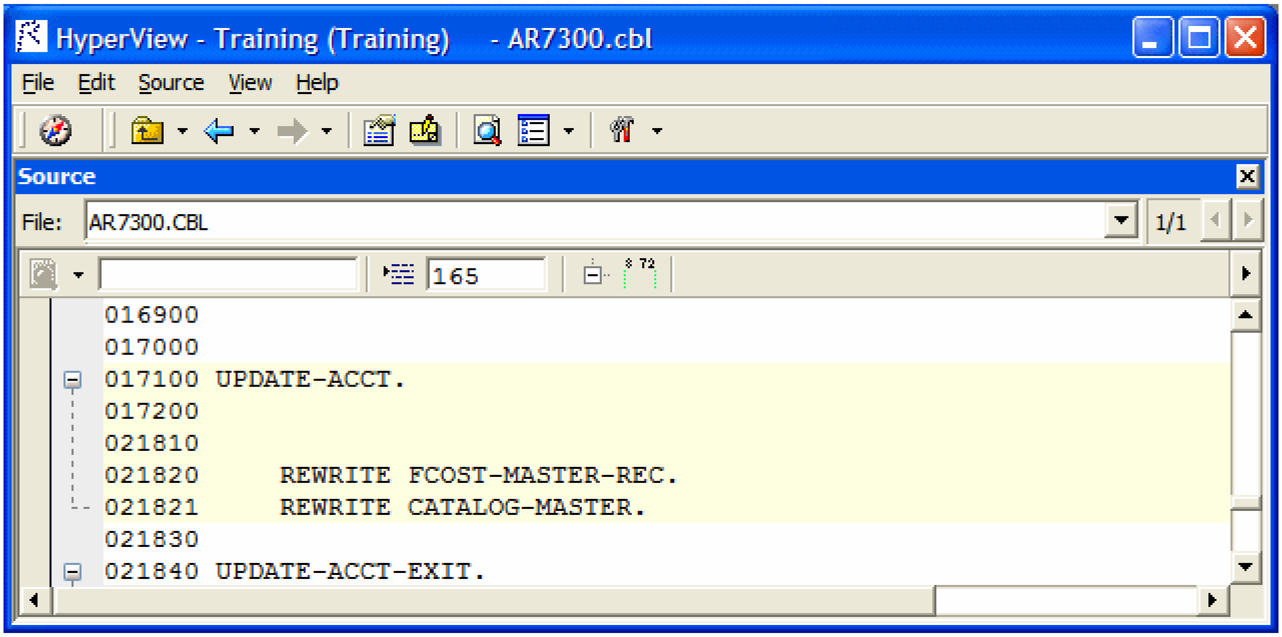
- The Source pane displays view-only source for the selected file.
- The Context pane displays the same code in hierarchical form, in a parse tree that defines the relationships between the code constructs that comprise the source.
- The Objects pane lists the files in the project. Select a file to open 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 displays view-only source for the selected file. 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.

Note: Use the Editor in the Modernization Workbench main window to modify program source code.



File: AR7300.CBL

```
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.

Using the Simple Search Facility

HyperView provides simple and advanced search facilities for source code. 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 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.

Navigating to Multiple Occurrences of an Included 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 right-hand corner of the Source pane 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 right-hand 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.

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 **Source > Copy Selected Text**.

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.

Setting Source Pane User Preferences

Use the HyperView > Source tab of the User Preferences window to specify the color-coding used in the Source pane and other display features.

1. Choose **Tools > User Preferences**. The User Preferences window opens. Click the HyperView > Source tab.
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.
4. In the **List Highlighting Color** field, click the adjacent ▾ button to edit the color used to highlight lists.
5. In the Color Scheme pane, click the construct types whose text you want to colorize, or click **Select All** to select all the construct types. The current color of the selected types (if any) is displayed in the **Fore Color** drop-down. Click the adjacent ▾ button to edit the color of the construct types.
6. Select **Display Business Names** to display business names rather than original identifiers in the Context pane.

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 **Edit > Collapse All**. 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 **Edit > Expand All**.

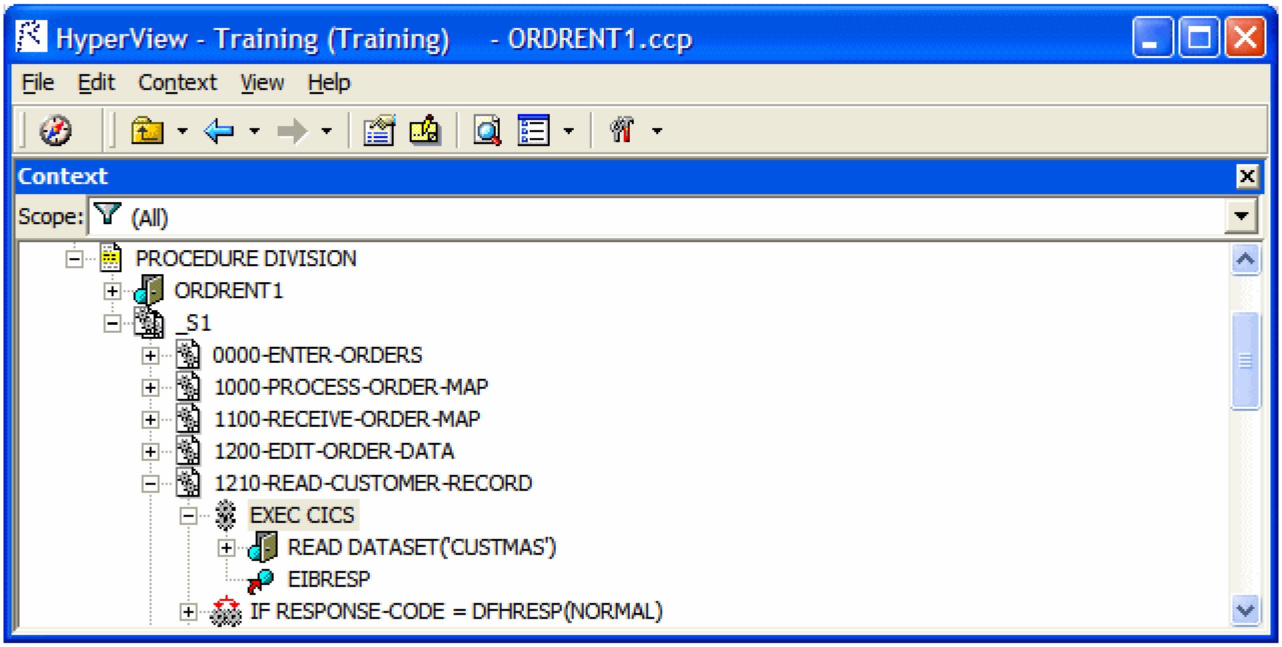
Showing Source Code Boundaries

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

Using the Context Pane

The Context pane 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.

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.



```
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)
```

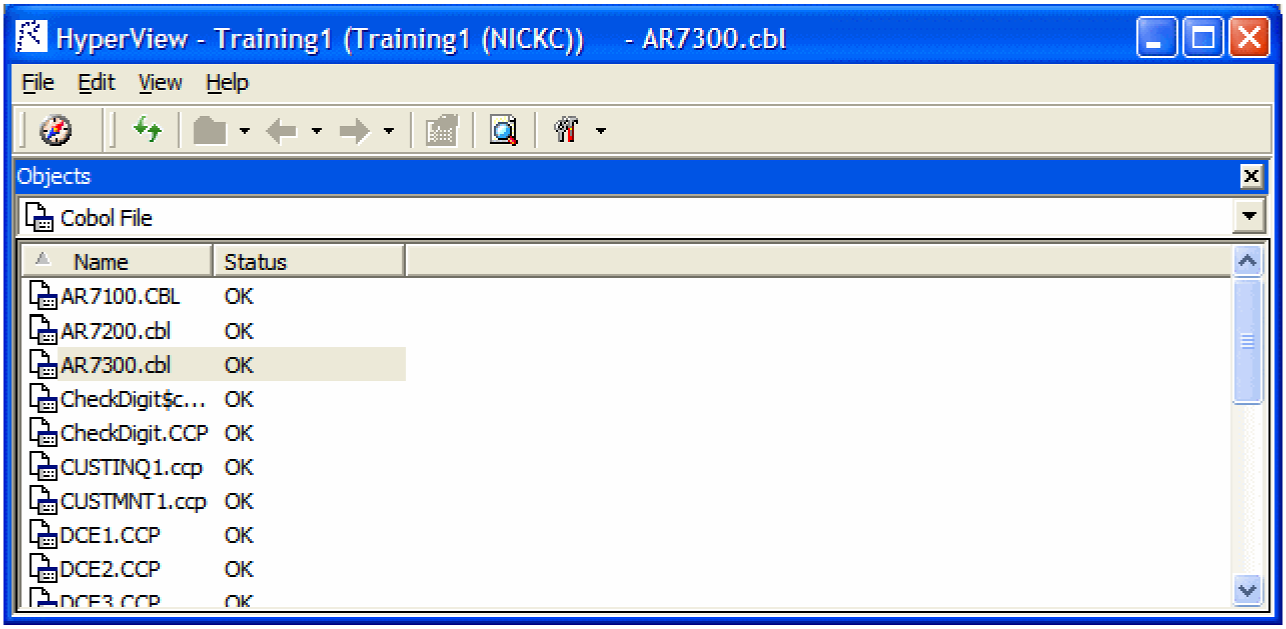
The scope of the parse tree determines the constructs and relationships it displays: all constructs, only control statements, only declarations, and so forth. Scopes include the lists HyperView generates when you perform advanced searches, navigate to a related construct, or identify candidates in Clipper. 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 **Display Business Names** in the User Preferences for the Source pane to display business names rather than original identifiers in the Context pane.

Using the Objects Pane

The Objects pane 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.



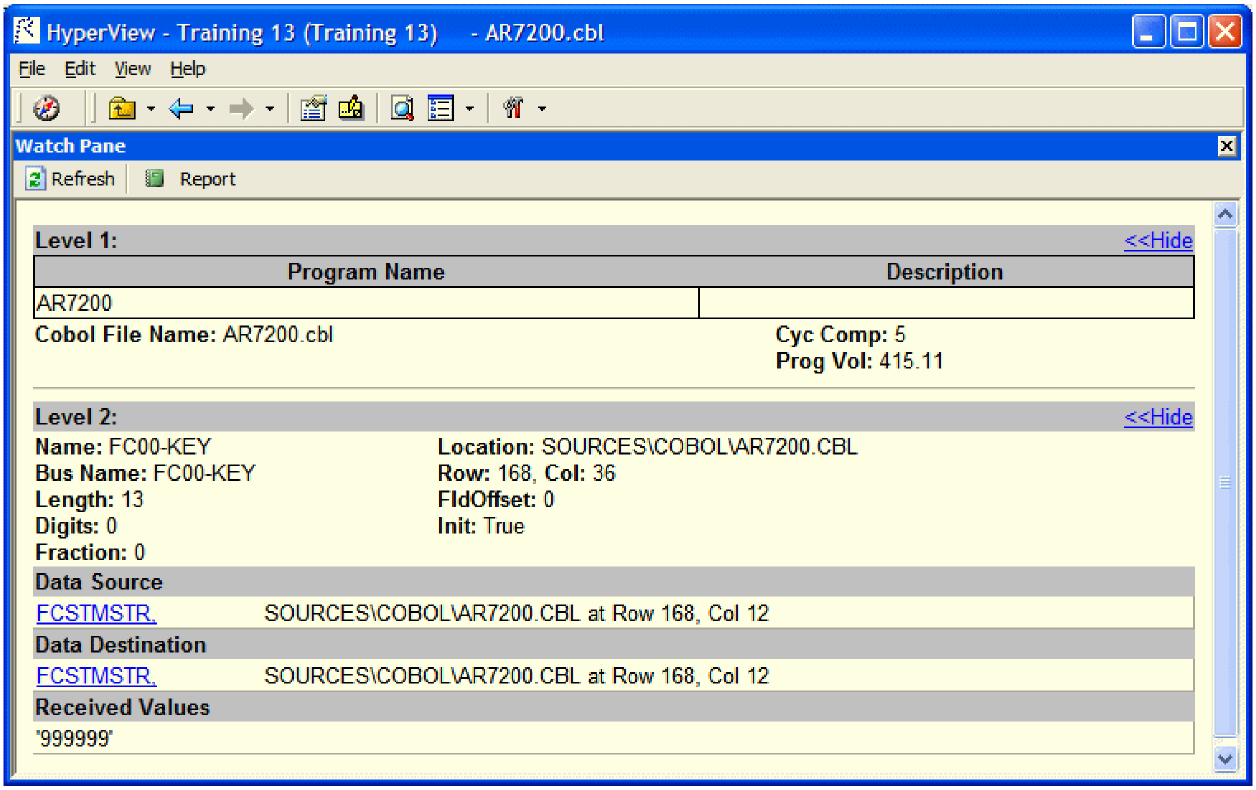
Cobol File

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.

Follow the instructions below to view current context summaries in the 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

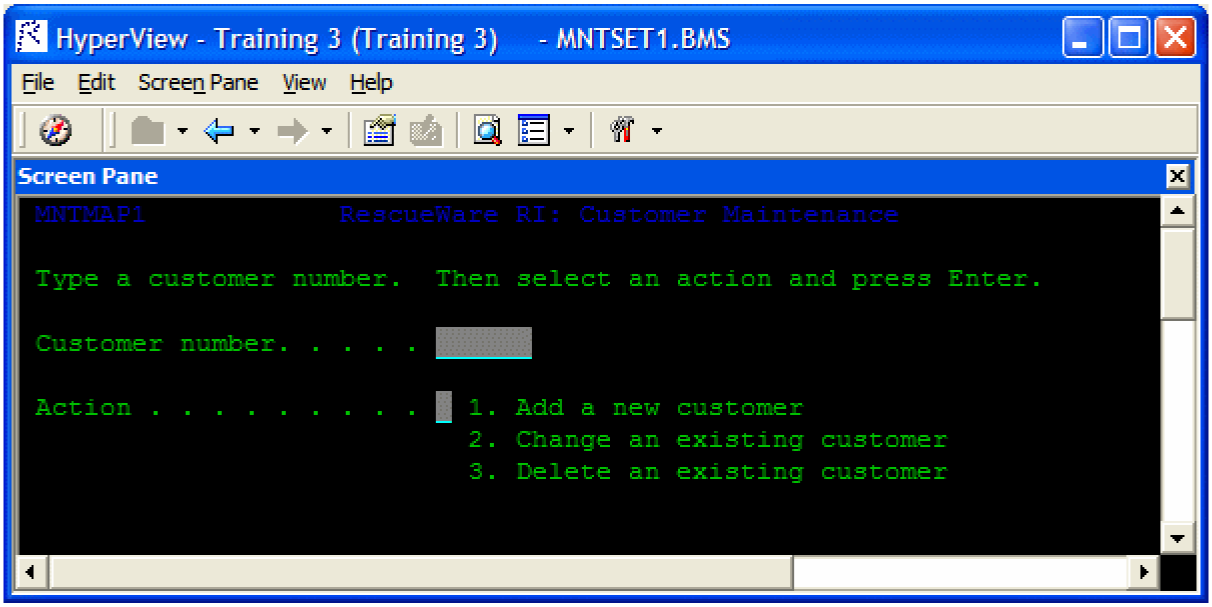
'999999'

1. Select the source file you want to view in the Objects pane.
2. Select the construct you want to view in the Source or Context pane.
3. Click the button in the Watch pane 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.
   - 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.
     - Data Destination displays the flow of data through each port in the trace to the destination port.
     - Received Values displays the value of the watched variable at each location in the impact trace.
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.

Using the Screen Pane

The Screen pane displays the screen object defined in the selected map file. 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.

To view a screen object in the Screen pane, select the map file in the Objects pane, then select the screen in the Source or Context panes. Choose **Screen > Best Fit** to fit the screen to the Screen pane. Choose **Screen > Actual Size** to restore the screen to its actual size.



```
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
```

Gray areas on the Screen pane represent screen fields. Click on a field or screen label to select it. Select a field and choose **Data Items** in the right-click menu to view the declarations for the field in the Properties window.

Manually Extracting Business Names for Program Variables

One of the most powerful features of Modernization Workbench lets you *auto-extract screen labels* as business names for program variables. For each screen field with a label immediately to its left in the Screen pane, the Glossary tool generates a business name for the program variable whose value is displayed in the field. If the screen field has the label **Customer Number**, for example, Glossary assigns the business name "Customer Number" to the variable whose value is displayed in the field.

Occasionally, however, you may need a finer-grained approach than the auto-extract feature offers. The field of interest may not be immediately to the right of the label, or the field may have no label at all. In these situations, you can extract business names for program variables manually in the Screen pane.

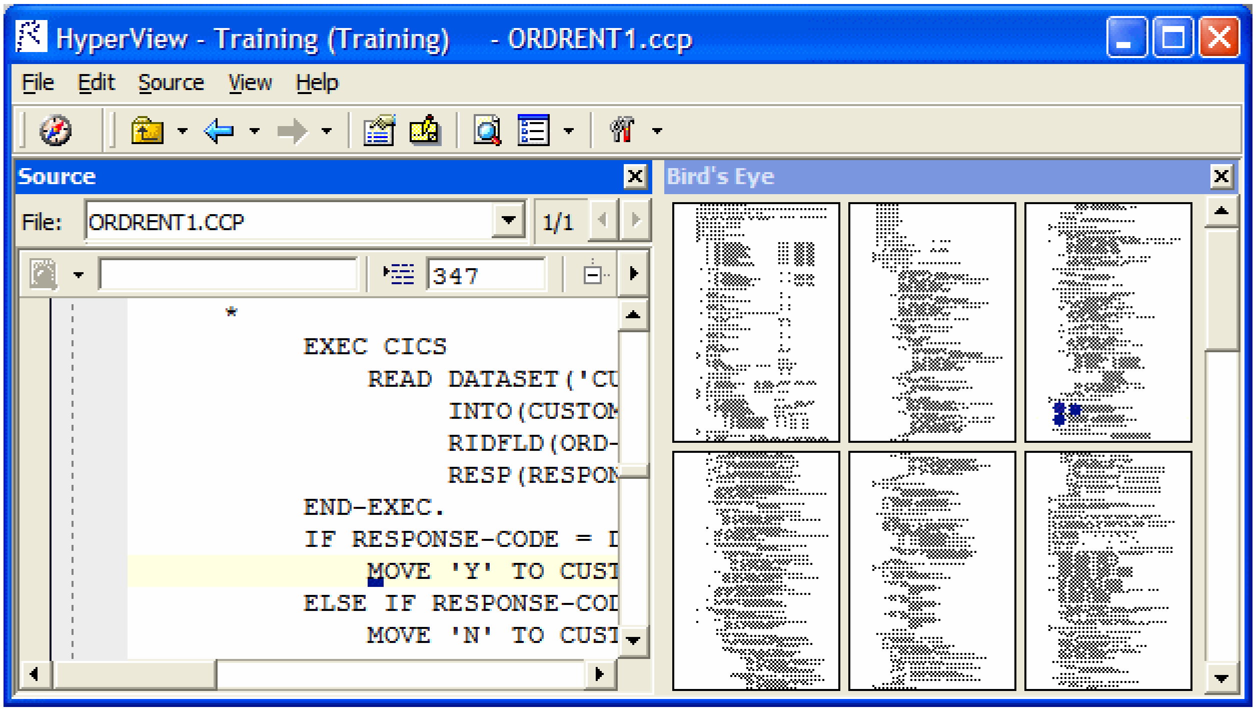
1. Double-click a screen label and select the portion of the label you want to use as the business name, then click Ctrl+C.

   Note: You don't have to use the screen label. You can put any text on the clipboard and assign it as the business name.

2. Select the field of interest and click Ctrl+V. HyperView paints a yellow box in the lower lefthand corner of the field to indicate that the program variable for the field has been assigned a business name. Double-click the field to navigate to the variable in the Glossary.

Using the Bird's Eye Pane

The Bird's Eye pane lets you quickly identify the location of a code construct relative to the entire program and any included files. Click a construct in the Source or Context panes to view its location in the Bird's Eye pane.



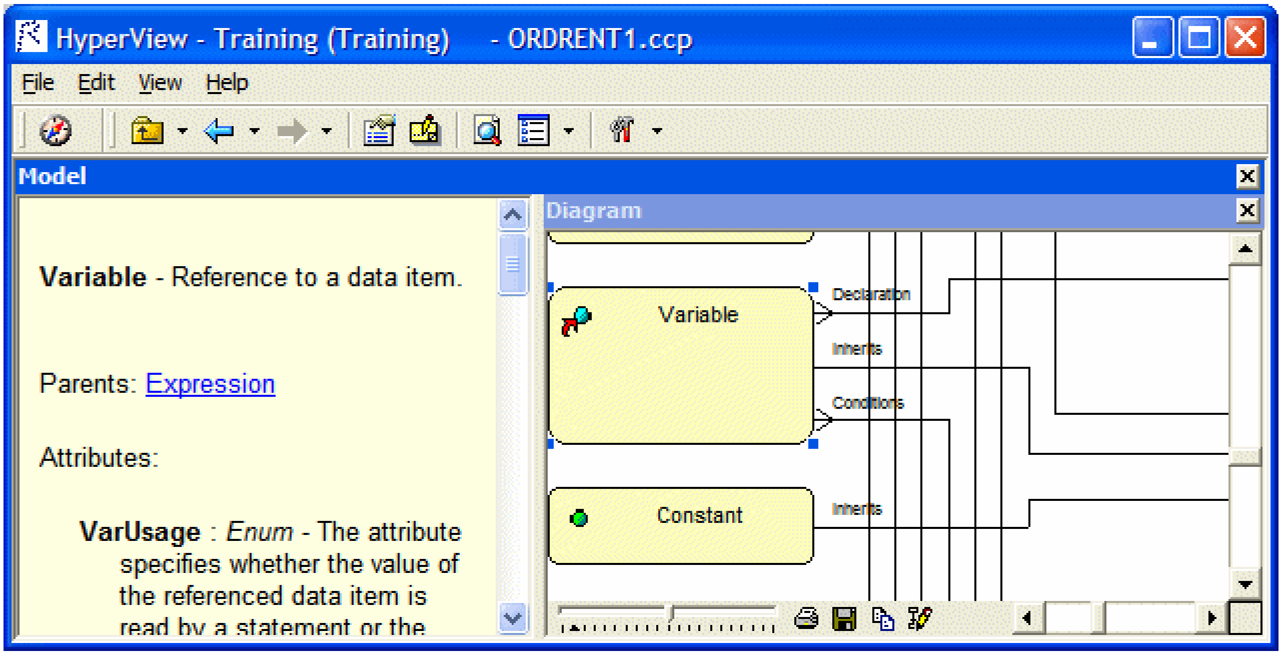
File: ORDRENT1.CCP 1/1

347

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

Using the Model Reference Pane

The Model Reference pane 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 searches.



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

Performing Advanced Searches

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

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

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: 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.

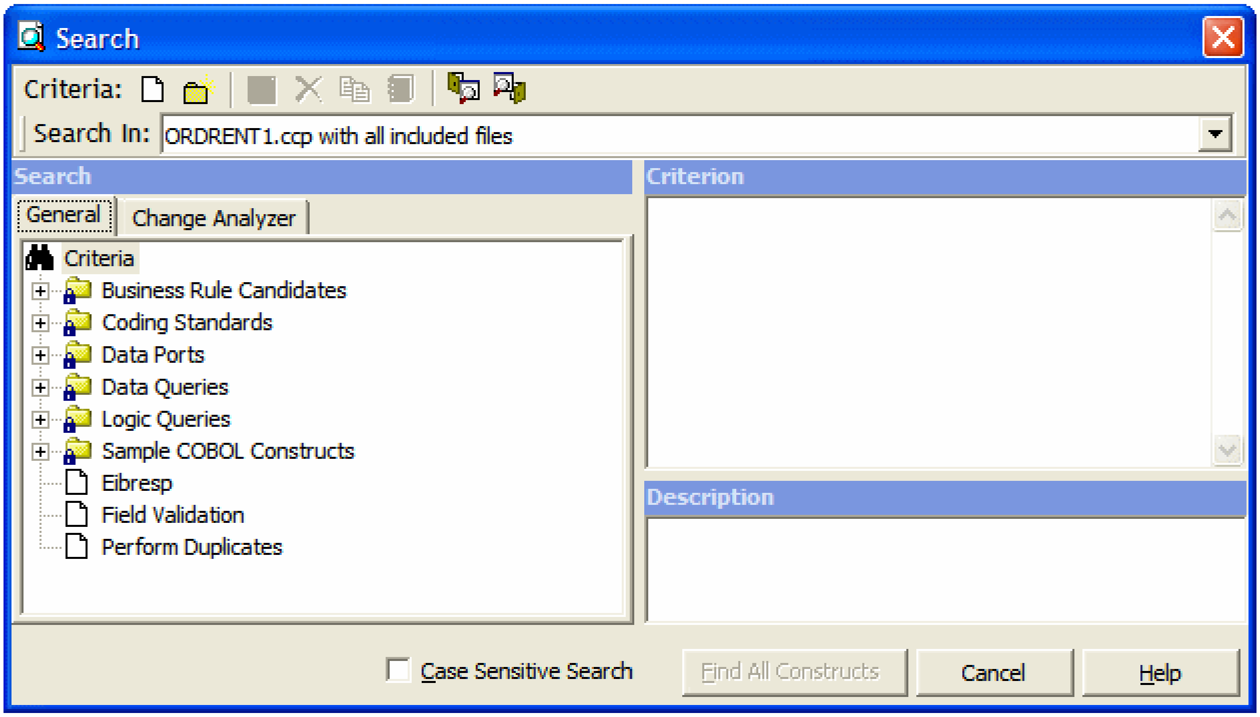
Defining a Search Filter

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 for the conditions a program uses to validate the variable EIBRESP.

1. Select the file you want to search in the Objects pane, then choose **Edit > Find**. The Search window opens.

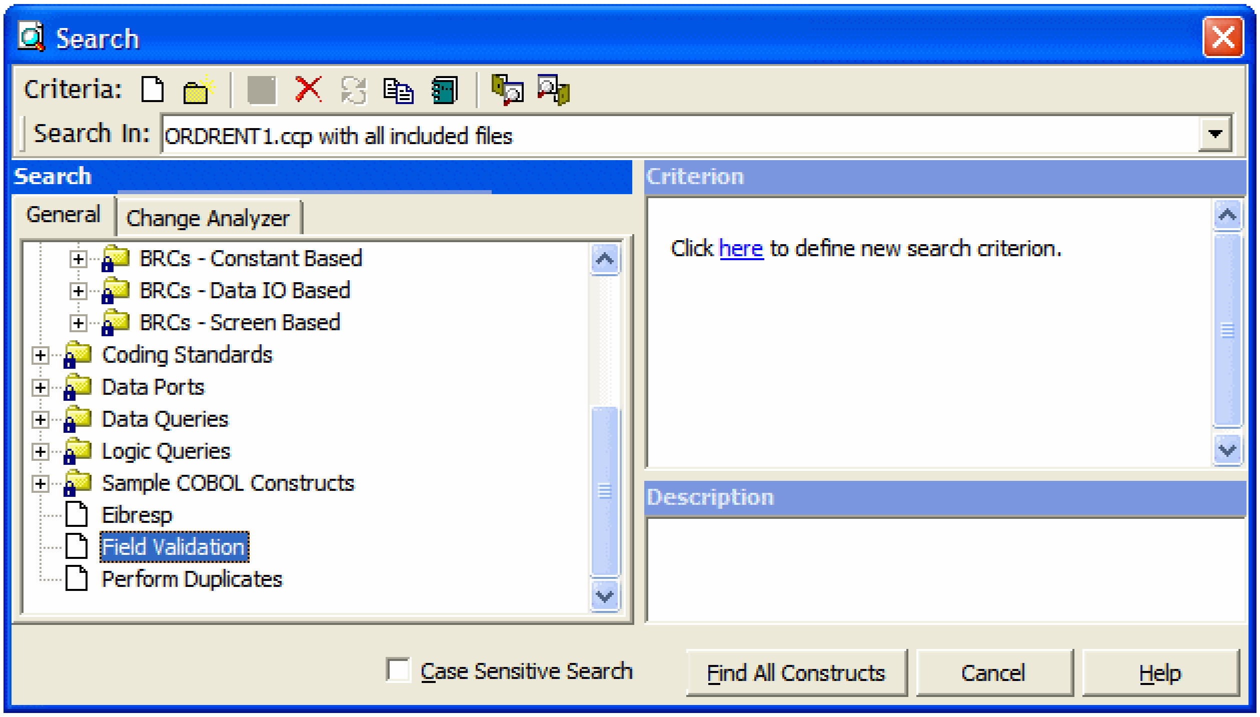
   Note: Skip this step if you are using Clipper to perform a batch search.

2. Click the General tab.



Search In: ORDRENT1.ccp with all included files

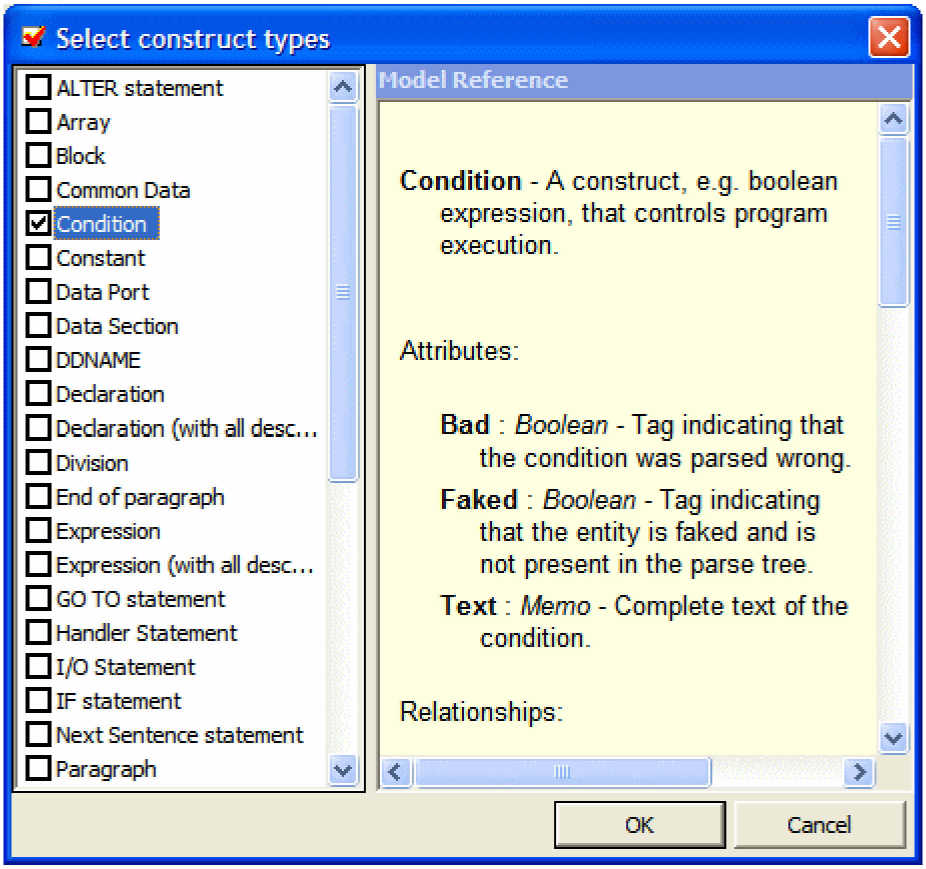
3. In the General tab, select a folder for the new criterion, then click the 🗋 button on the tool bar. The New Criterion dialog opens. Enter Field Validation in the text field and click **OK.** HyperView creates the Field Validation criterion in the selected folder.



Search In: ORDRENT1.ccp with all included files

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.



Select construct types

- □ 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

Model Reference

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.

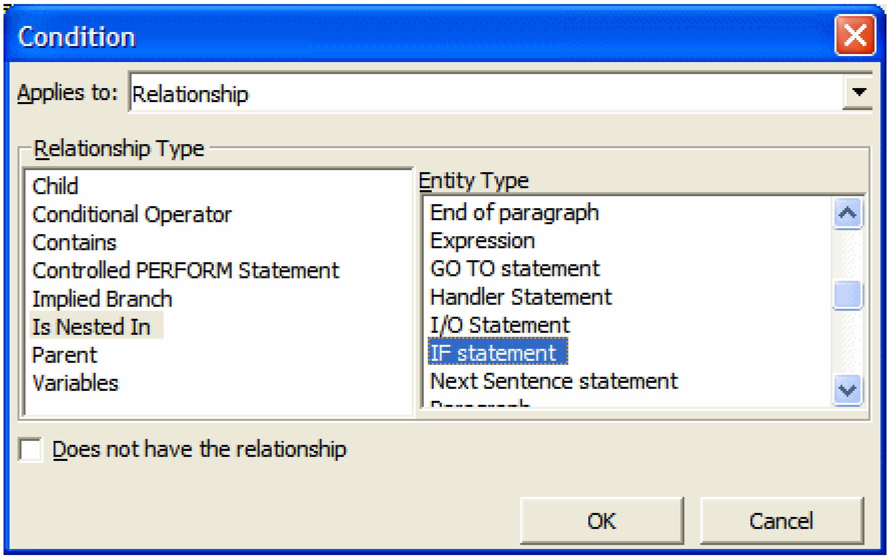
Text : *Memo* - Complete text of the condition.

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.



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: Except for the # symbol, 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*"
```

Note: 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.

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.

1. 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.
2. Choose the criterion for the search you want to execute from the list of criteria in the lefthand pane.
3. Select **Case Sensitive Search** if you want the search to match case.
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.

Working with Advanced Search Criteria

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.

Staging Program Analysis with Clipper

Use the Clipper pane to stage the results of a batch search 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.

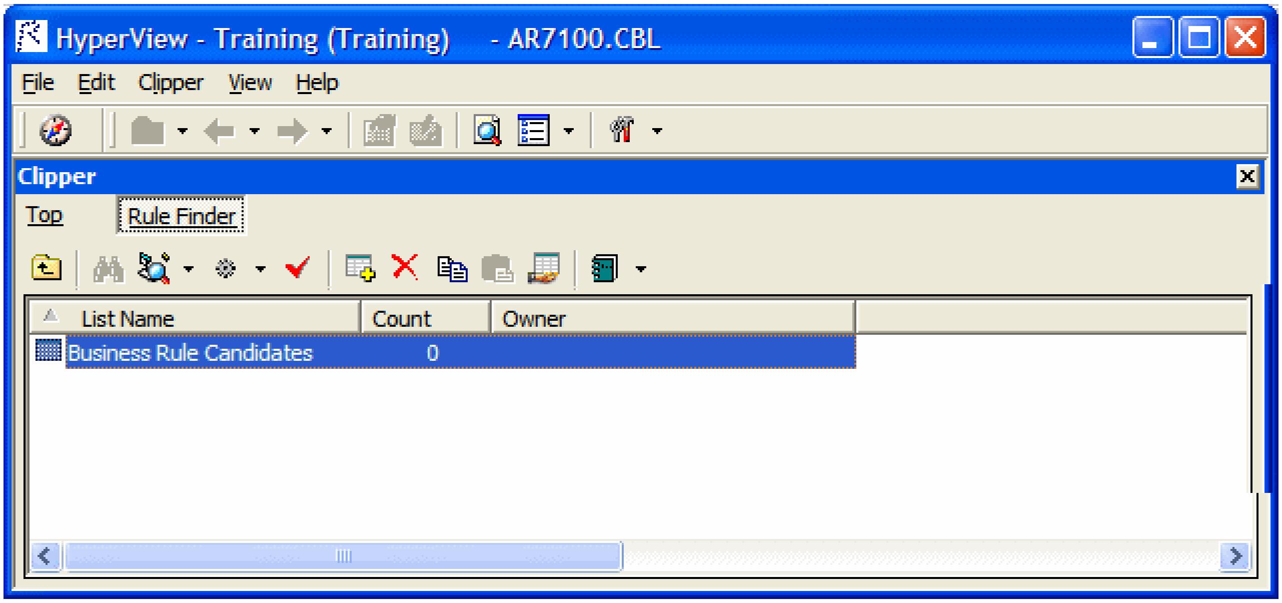
Clipper searches are executed against the current project. You can also execute searches against the results of a previous search.

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

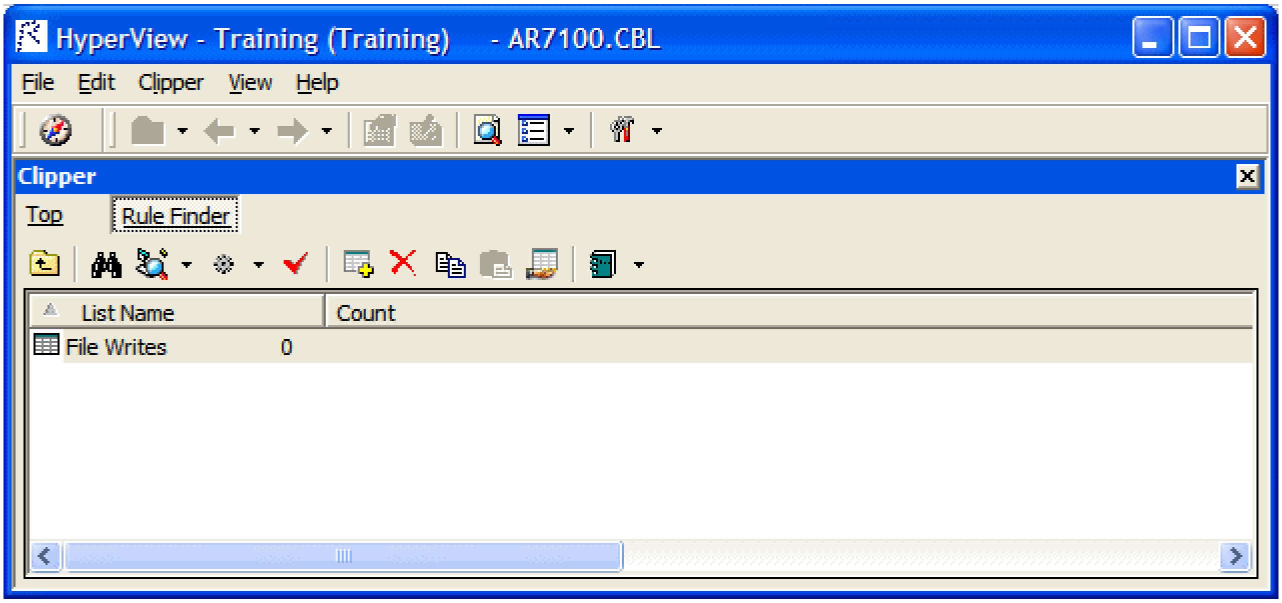
This section describes a sample use of Clipper that should help you get oriented in the tool. Assume that you want to create business rules for segments of code that write to a file.

1. In the Clipper pane, click the **Top** view link. The Top view opens.
2. Double-click the Rule Finder category. The Rule Finder view opens.



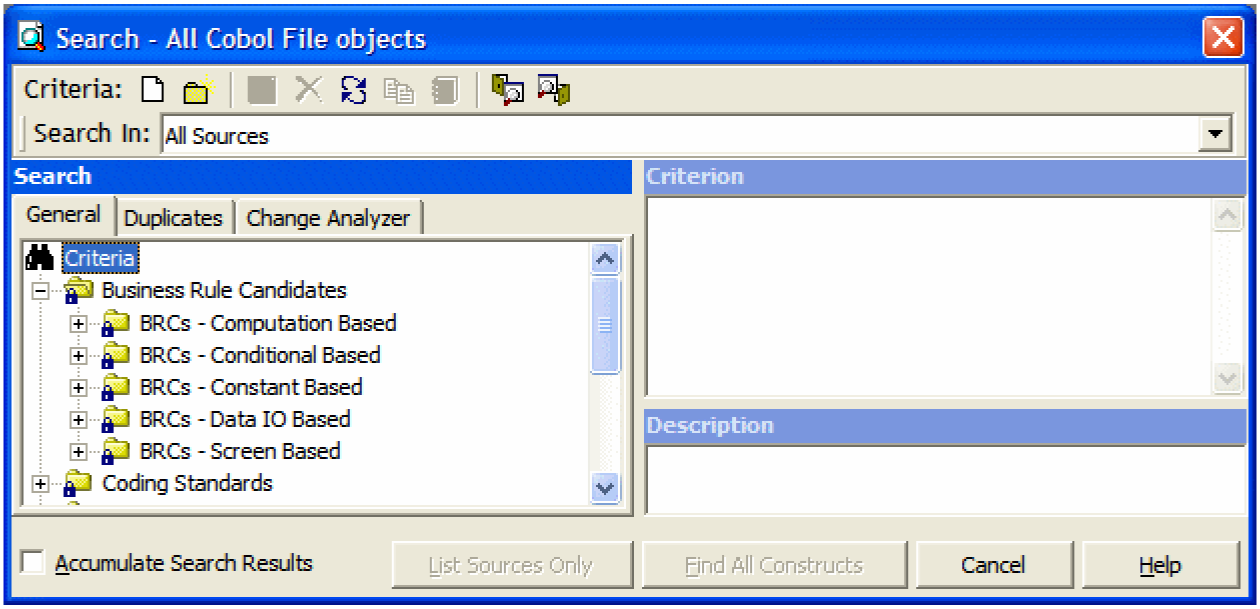
List Name	Count	Owner
Business Rule Candidates	0	

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.


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.

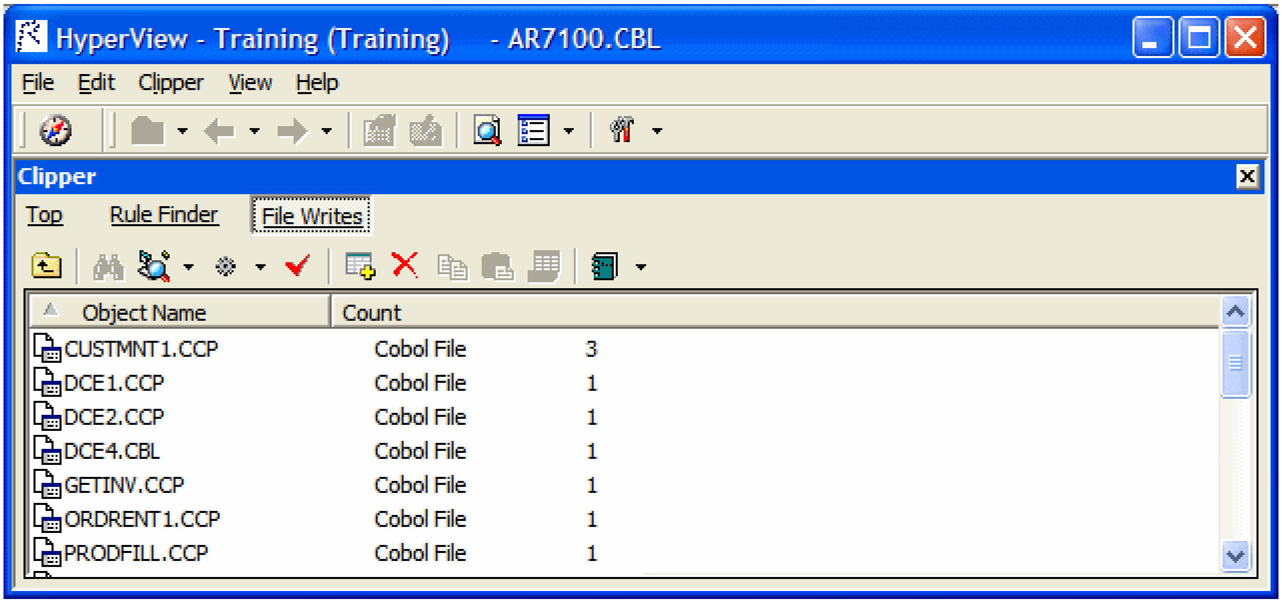


Search In: All Sources

□ Accumulate Search Results

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.

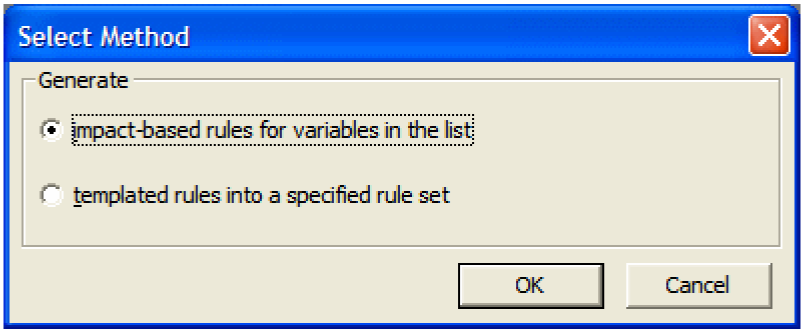
Note: Modify the predefined search criterion or create your own search criterion as you do in the Advanced Search facility.



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

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.



Generate

● 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 for generating business rules.

Understanding 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 display HyperView lists and perform other list-related tasks:

- The Top view displays the folders, or *categories*, for lists.
- The Category view displays the lists in the selected category.

- The List view displays source files with non-zero counts in the selected 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 other HyperView 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.

Working with Categories

The Top view of the Clipper pane displays the categories for HyperView lists. Categories are displayed only for lists you can create for the source file type selected in the Objects pane.

You can use predefined categories or create your own. All predefined categories are shared. Use the General and Internal categories for unenumerated tasks.

Note: HyperView puts the results of standalone advanced searches in a predefined list called Last Search Results in the Internal categories.

To create a category, select the parent category and choose **Clipper > Add**. The New category dialog opens, where you can specify the new category name. A list with the category name is automatically created in the category. To edit the category name, click the name and type over it in the text box.

By default, user-defined categories are not shared. To share a user-defined category, select it and click the button on the tool bar. A symbol indicates that the category is shared. Click the button again to turn sharing off.

Note: Sharing a user-defined category automatically makes its parent categories shared. Unsharing a user-defined category automatically makes its children unshared.

Working with Lists

The Category view of the Clipper pane displays the lists in the selected category. You can use predefined lists or create your own. Use a Clipper search to populate lists.

Creating a List

To create a list, double-click the category for the list in the Top view and choose **Clipper > Add**. The New List dialog opens, where you can specify the name of the new list.

Deleting a List

To delete a list, select it and click the ✕ button on the tool bar. You are prompted to confirm the deletion. Click **OK**.

Copying a List

To copy a list, select it and click the button on the tool bar. 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 a List

By default, lists are not shared. To share a list, select it and click the

button on the tool bar. A symbol indicates that the list is shared. Click the button again to turn sharing off.

Adding a Construct to a List Manually

To add a construct to a list, select a source file in the List view, 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 a Construct from a List

To delete a construct from a list, select the construct in the File view and click the ✕ button on the tool bar. You are prompted to confirm the deletion. Click **OK**.

Executing Clipper Searches

Use a Clipper search to generate Clipper lists. You can modify the predefined search criteria or create your own criteria as needed. You can use a construct list itself to narrow the search.

1. 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.
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 folder.
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 needed.
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.
5. Select:
   - **Find All Constructs** if you want HyperView to create for each searched file a list of all the constructs found by the search.
   - **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.

   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, storing the results in a list called Matched Paragraphs in the General folder. 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.

1. 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.
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.
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. Choose:

   - Relationship in the **Applies to** drop-down.
   - Called Paragraph in the **Relationship Type** list box.
6. 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. Choose:

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

   You can filter out list constructs by selecting **Does not belong to the list** in the Condition window.
8. 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
       General.Matched Paragraphs
   ```
9. Execute the search.
   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.

Searching for Similar Constructs in Clipper

To search for similar constructs, select a construct in the Source or Context panes, then select a list. 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.

Extracting Business Rules in Clipper

Clipper offers two methods for extracting 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 **Rule > Create Rule**.

Autodetecting Business Rules in Clipper

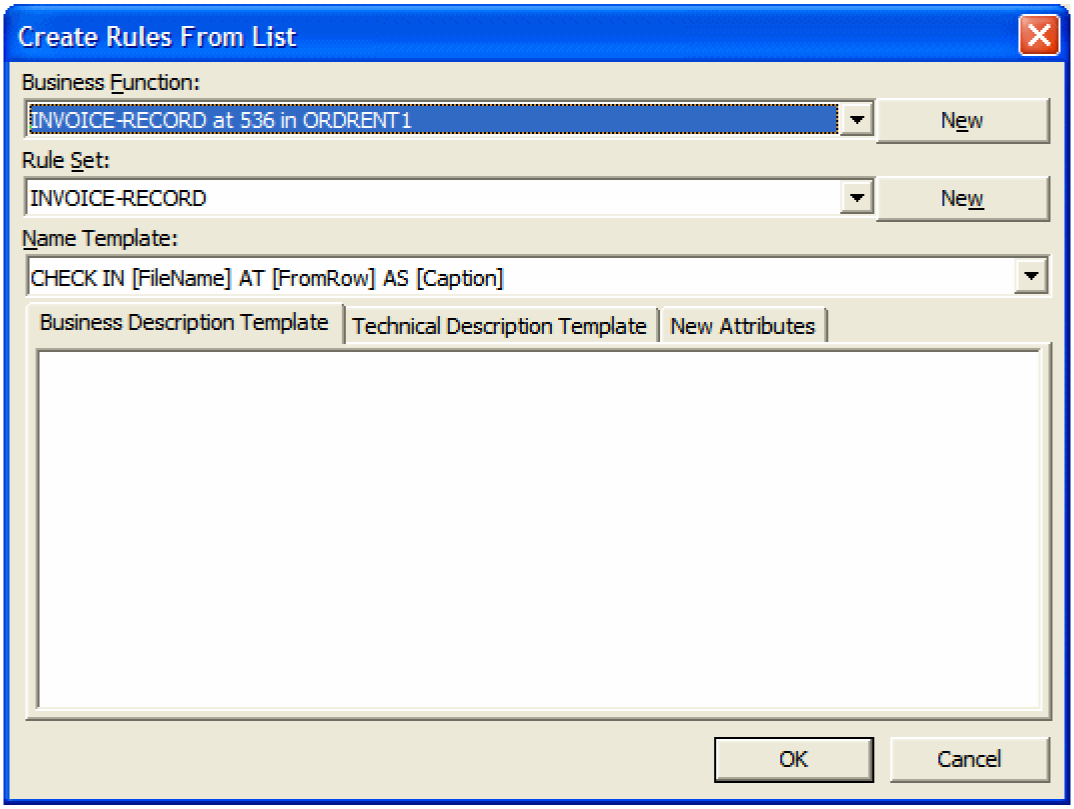
The *backward autodetect* method traces the flow of data from a variable to its source, assigning each statement in the flow to a new rule. Follow the instructions below to backward-autodetect code segments that impact constructs in a Clipper list. Clipper creates a rule for each autodetected segment. The constructs must be variables.

1. Select a list in the Rule Finder category, then 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.
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 prompts you to "prune 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. 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 Business 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.

1. Select a list in the Rule Finder category, then 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.
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**.
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.



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.
6. On the Business Description Template and Technical Description Template tabs, enter templates for the description and business description attributes, respectively.
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 in Clipper

To generate an impact subtree report for all the constructs in a list, select the list in the Impact Report category, then 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.

Marking and Colorizing Constructs in Clipper

The Components pane offers a variety of advanced algorithms for *slicing* logic from legacy source: all the code you need for a computation, for example, or the code you need 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.

To mark and colorize list constructs in the Source pane, select the list 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.

Creating Projects in Clipper

You can create a project based on the results of a Clipper analysis.

1. Select a list, then 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**.
2. The Change Magnitude window opens. Select **Automatically calculate change magnitude** 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. Modify the ranges as needed. The Clipper settings will override any existing change magnitudes for the files.

   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**.
3. When you are satisfied with your choices, click **OK**.

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 Folder 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.

Analyzing Impact Traces

An *impact trace* describes the flow of data within and between programs in an application: how data items exchange values, use each other in computations, and so forth. 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.

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: Projects must have been verified with the **Enable Impact Report** and **Enable Data Element Flow** options set in the Project Verification options.

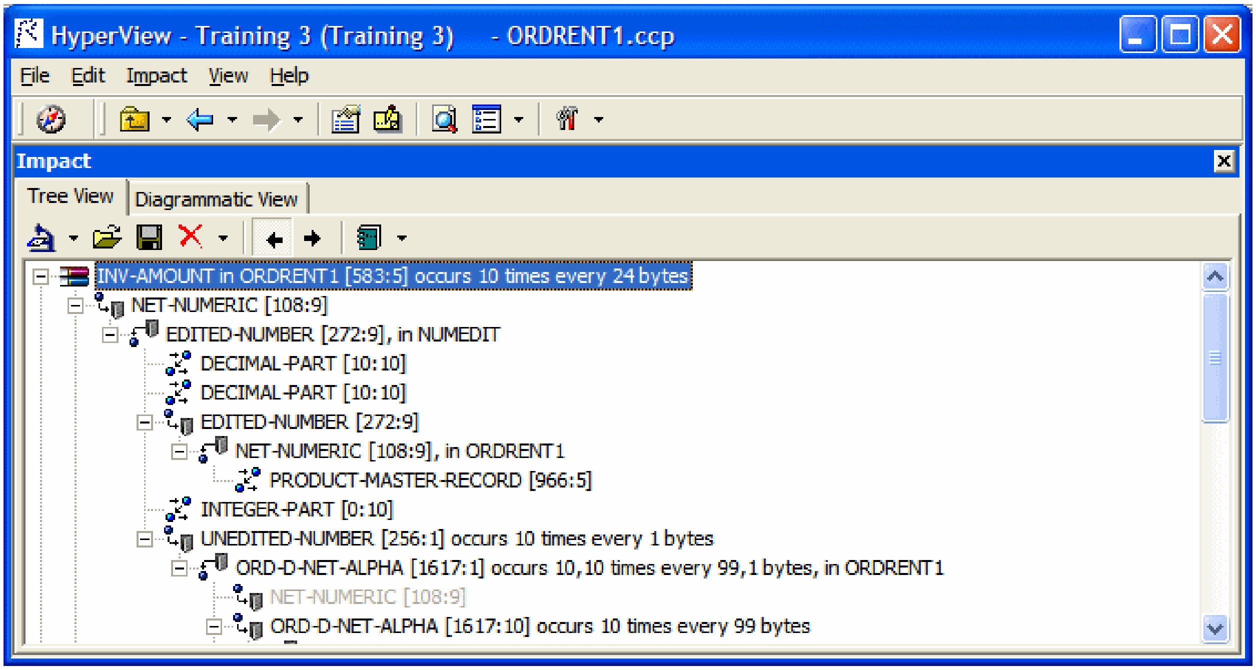
Generating an Impact Trace

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*.

1. In the HyperView Source or Context pane, select the construct that contains the startup item or items and choose **View > Impact**. The Impact pane opens.
2. Choose:
    - **Impact > Backward Analysis** to view the flow of data into the startup items.
    - **Impact > Forward Analysis** to view to view the flow of data out of the startup items.

    Note: Once a trace is generated, click the ← or → buttons on the tool bar to reverse the direction of the trace.
3. Choose:
    - **Impact > Detailed Analysis** to generate the entire tree for a trace.
    - **Impact > 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.

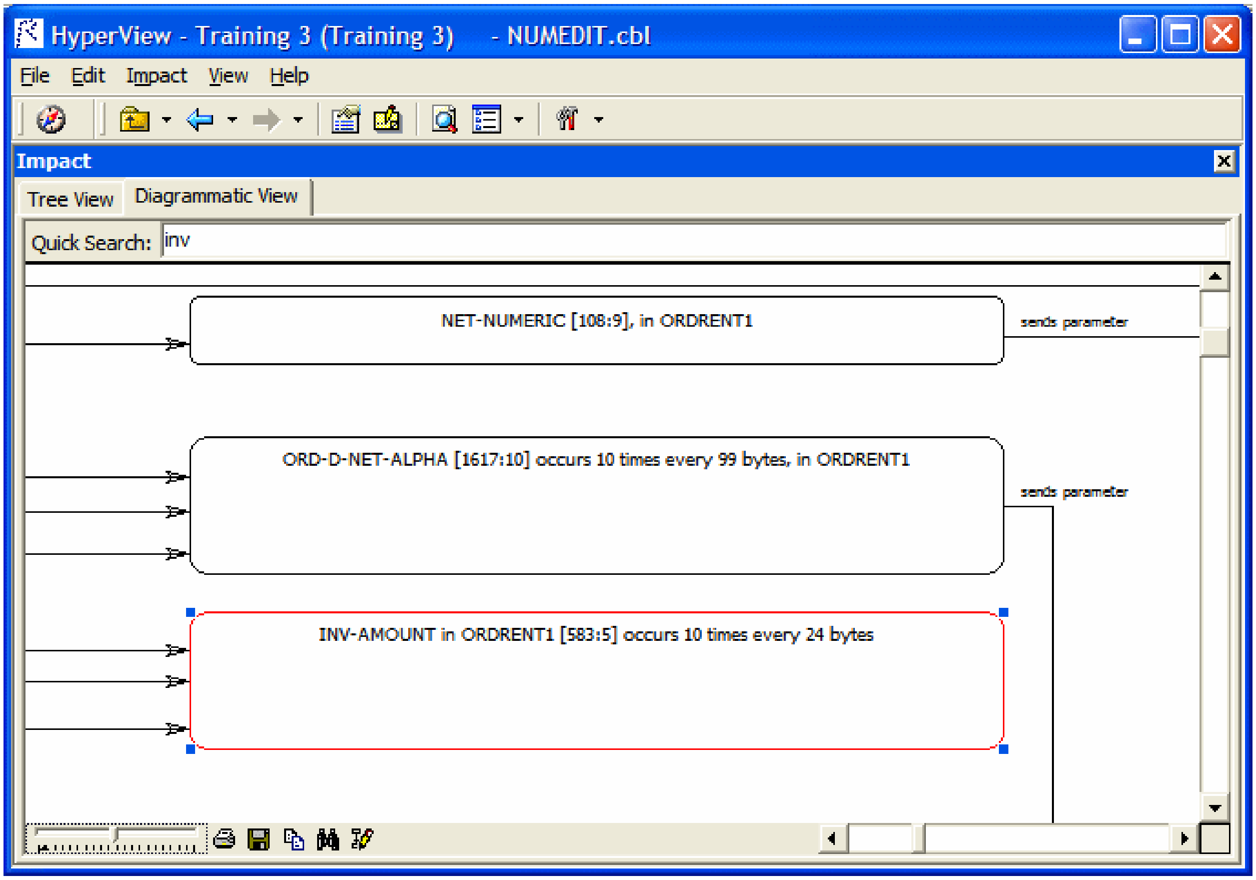
    HyperView displays the trace results. Click the plus sign (+) next to an item to expand its hierarchy. Click the minus sign (-) to collapse the hierarchy.



HyperView - Training 3 (Training 3) - ORDRENT1.ccp
File Edit Impact View Help
Impact
Tree View
Diagrammatic 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

4. Click the Diagrammatic View tab to display the impact trace in a diagram.



HyperView - Training 3 (Training 3) - NUMEDIT.cbl
File Edit Impact View Help
Impact
Tree View
Diagrammatic View
sends parameter
sends parameter

Quick Search: inv

NET-NUMERIC [108:9], in ORDRENT1

ORD-D-NET-ALPHA [1617:10] occurs 10 times every 99 bytes, in ORDRENT1

INV-AMOUNT in ORDRENT1 [583:5] occurs 10 times every 24 bytes

5. Select an item in the trace and repeat the procedure to start a trace for that item. The new trace is displayed below the previous trace.
6. To delete a trace, select its startup item and choose **Impact > Remove Root**. To clear the pane, choose **Impact > Remove All**.

Understanding the Hierarchical View

In the hierarchical view, 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. Options give you complete control over the information displayed in the label. The table shows the meaning of the typefaces used in the labels.

Typeface	Meaning
bold	An item that has not been visited by the trace.
gray	A repeated item.
blue	A repeated item at a different location.
bold blue	A repeated item at a different location that has not been visited by the trace.
red	An item filtered out of the trace.

Understanding the Diagram View

In the diagram view, each box in the trace contains the data item name and the name of the program it belongs to (unless the parent item belongs to the same program). Relationship lines contain the name of the relationship. Options let you display the memory offset and size of a data item. Place your cursor over a variable for a moment to display the memory offset and size in a tool tip. The table shows the meaning of the colors used in the diagram.

Color	Meaning
red outline	Startup item.
blue background	An item that has not been visited by the trace.
black outline	An item filtered out of the trace.

Data Flow Relationships

The table below describes the intraprogram and interprogram relationships detected by the data flow analysis tools.

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.
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. **Perform Unisys Common-Storage Area Analysis** must be set in the project verification options.
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.
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.

Relationship	Definition	Type	Icon	Description
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.
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.
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.
used	MOVE ... TO A ... MOVE A TO ...	intraprogram		A value assigned in a statement used as an argument in another statement.

Setting Impact Pane User Preferences

Use the HyperView > Impact tab of the User Preferences window 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 these options to specify the level of detail to include in an Impact Subtree Report.

1. Choose **Tools > User Preferences**. The User Preferences window opens. Click the HyperView > Impact tab.
2. Click 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 attributes reported in the Subtree Report. 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 attributes reported in the Subtree Report. You can modify unescaped text as necessary.
5. Click **More** in the Report Attributes pane. In the Report Attributes dialog, select the attributes you want to include in the Subtree Report.
6. Choose **Display Business Names** if you want to display business names in the impact trace rather than original identifiers.

Setting Impact Pane Project Options: Processing Tab

Use the Impact > Processing tab of the Project Options window to specify 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.

1. Choose **Tools > Project Options**. The Project Options window opens. Click the Impact > Processing tab.
2. 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, 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:

          ```
          IF A > 0 THEN
          MOVE A TO B  (1)
          ELSE
          GOBACK.
          MOVE B TO C.   (2)
          ```
        - **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. Add patterns as necessary. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).

Setting Impact Pane Project Options: Relationships Tab

Use the Impact > Relationships tab of the Project Options window to specify the intraprogram and interprogram data flow relationships to include in the trace. The table below describes the intraprogram and interprogram relationships detected by the data flow analysis tools.

1. Choose **Tools > Project Options**. The Project Options window opens. Click the Impact > Relationships tab.
2. In the Intraprogram pane, place a check mark next to each intraprogram data flow relationship you want the trace to include.

3. In the Interprogram pane, place a check mark next to each interprogram data flow relationship you want the trace to include.
4. 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.

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.
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. **Perform Unisys Common-Storage Area Analysis** must be set in the project verification options.
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.

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.
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.
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.
used	MOVE ... TO A ... MOVE A TO ...	intraprogram		A value assigned in a statement used as an argument in another statement.

Setting Impact Pane Project Options: Impact Reports Tab

Use the Impact > Impact Reports tab of the Project Options window to exclude repeated items from the Impact Subtree Report and to limit the depth of the trace in all reports.

1. Choose **Tools > Project Options**. The Project Options window opens. Click the Impact > Impact Reports tab.
2. Select **Output duplicates in the SubTree report** if you want to include repeated items in the Impact Subtree Report.
3. Select **Limit Impact Depth** if you want to limit the depth of the trace reported, then specify the maximum depth of the trace in the **Max Depth** combo box.

Exporting and Importing an Impact Trace

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

To import a saved impact trace, click the 📂 button on the tool bar. An Open dialog appears, where you can select the trace to import.

Generating a Subtree Report

A Subtree Report shows the trace for a subtree in XML format or in a database. To generate a Subtree Report, click the ▾ button next to the 📗 button on the tool bar and choose **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.

The following table shows the columns displayed in the database report format. You can control the level of detail displayed in the report in the Impact Pane User Preferences.

Note: The attributes listed in the table are a superset of the attributes listed in the Impact Pane User Preferences. Database IDs refer to IDs generated for the database report format.

Attribute	Description
CallStack	Not used.
Cnt	Not used.
Col	Number of the source column that contains the beginning of the affected data item.
Context	Internal ID of the context for the affected data item.
DeclHCID	Internal ID of the declaration for the affected data item.
Depth	The depth of the trace.
Direction	The direction of the trace, forward or backward.
Duplicate	Indicates that the affected data item is repeated in the trace.
EntityType	Construct type of the affected data item.
Excluded	Indicates that the affected data item is filtered from the trace.
File	Name of the file that contains the affected data item.
FromSel	Character offset from the start of the file to the beginning of the affected data item, counting from 0.
HCID	Internal ID of the affected data item.
ID	Database ID of the affected data item.

Attribute	Description
JobName	Name of the job that contains the affected data item.
JobStepID	Database ID of the job step that contains the affected data item.
Keyword	The operation on the affected data item.
Ln	Number of the source line that contains the affected data item.
Name	Name of the affected data item.
ObjName	Name of the object that contains the affected data item.
ObjType	Type of the object that contains the affected data item.
Offset	Memory offset of the affected data item.
OS	Memory offset and size in bytes of the affected data item, separated by a colon (:).
PortFlag	Port type of the affected data item: I for input, O for output, N for not a port.
PortName	Name of the port referenced in the operation identified by the Keyword.
PrevID	Database ID of the affected data item's parent node.
RelType	Data flow relationship type for the affected data item.
RootID	Database ID of the root node of the tree.
Size	Size in bytes of the affected data item.
SourceID	Database ID of the file that contains the affected data item.
SourceText	Text of the source line that contains the affected data item.
SourceType	Source type of the file that contains the affected data item.
StepName	Name of the job step that contains the affected data item.
StepNum (also Step)	Number of the job step that contains the affected data item.
ToSel	Character offset from the start of the file to the end of the affected data item, counting from 0.

Generating a Pane Report

A Pane Report shows the entire tree for a trace in XML format. To generate a Pane Report, click the ▾ button next to the 📓 button on the tool bar and choose **Pane Report** in the drop-down menu. A Save As dialog opens, where you can specify the name, location, and file type of the report.

Generating an Affected Code Report (Cobol and APS Only)

An Affected Code Report shows code that would be impacted by changing a data item's definition or usage (Cobol and APS only). To generate an Affected Code Report, click the ▾ button next to the 📒 button on the tool bar and choose **Affected Code Report** in the drop-down menu. A Save As dialog opens, where you can specify the name, location, and file type of the report.

Analyzing Program Control Flows

Control flows describe the processing paths in a program. Call flows, decision flows, and statement flows each offer a different way of understanding the procedures in a program. HyperView offers the following tools 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.

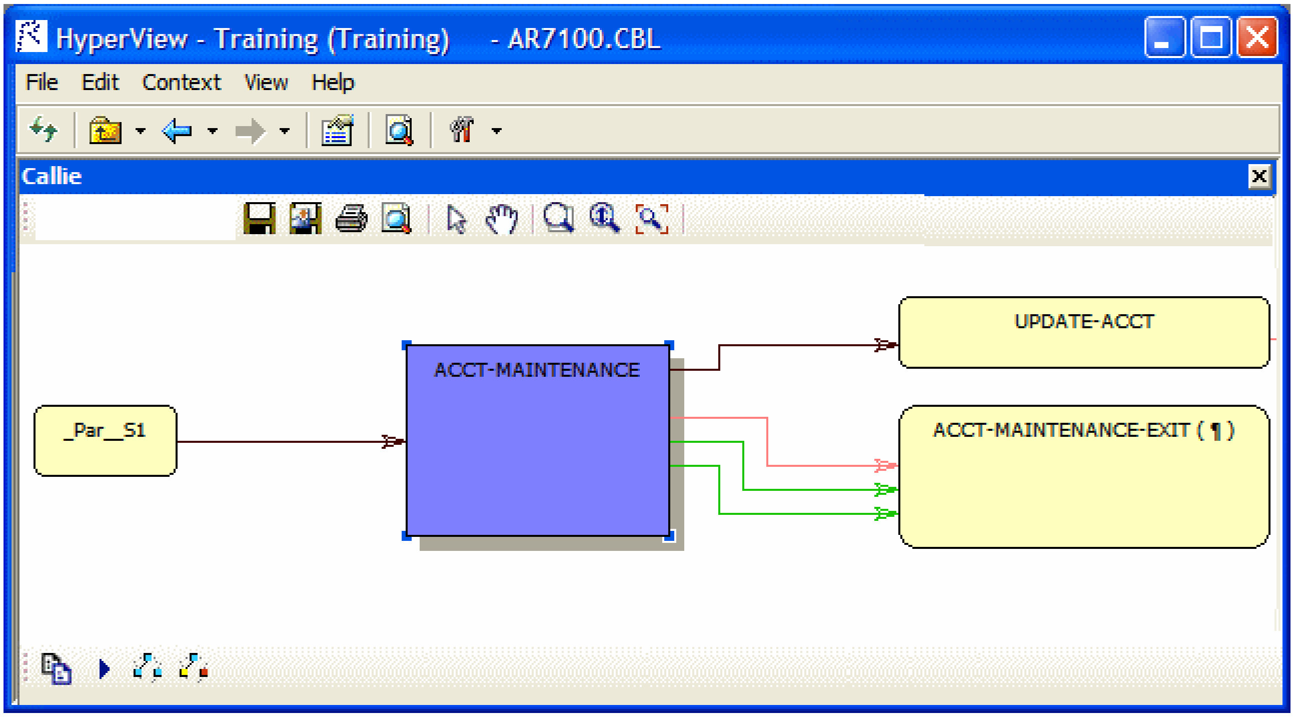
Using the Callie Pane

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).

Call relationship lines are color-coded as shown in the table below.

Color	Relationship
Green	GO TO
Red	Fall Thru
Brown	PERFORM
Blue	External call

The toolbar at the top of the pane displays icons used in the main Diagrammer window. For Diagrammer usage, see *Analyzing Projects* in the workbench documentation set.


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

Drawing the Diagram

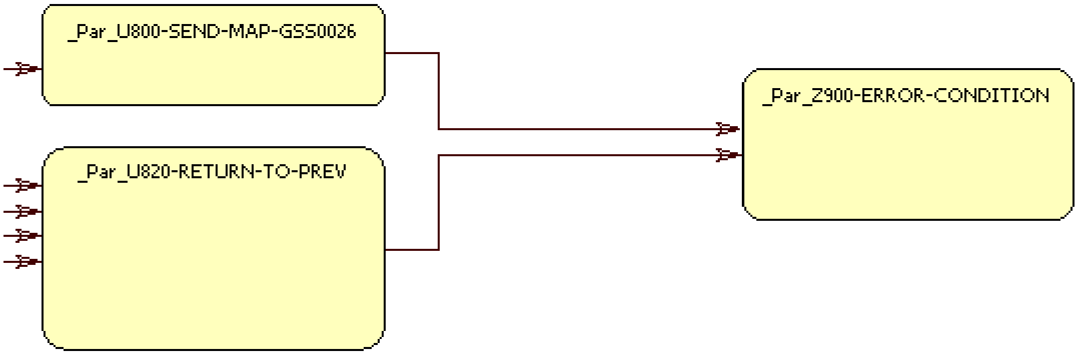
To display a call flow diagram in the Callie pane, select the seed item in the Source or Context panes. The diagram is automatically redrawn when you select a new seed item in the Source or Context pane, unless the new seed already appears in the current diagram. To redraw the diagram in that case, select the new seed item in the diagram and click the ▶ button on the toolbar at the bottom of the pane.

Choosing the Diagram View

The Callie pane offers two views of program control flows, a subgraph view and a subtree view. The figures below show the same paragraphs in both views.

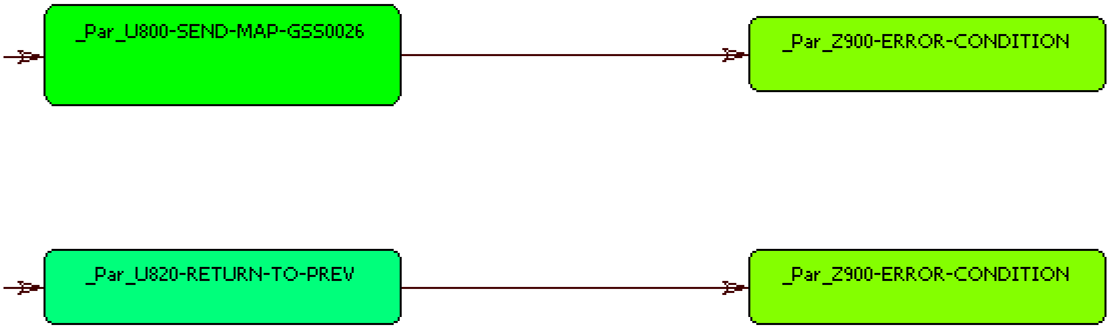
Subgraph View

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. Click the button on the toolbar at the bottom of the pane to select the 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. Click the button on the toolbar at the bottom of the pane to select the subgraph view.


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

Selecting Items

Select an internal item in a call flow diagram 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.

Setting Callie User Preferences

Use the HyperView > Callie tab of the User Preferences window to specify the default box color for items in subgraph views and the color of the selected item in subgraph views. You can also specify the number of visited nodes that Callie displays in different colors in subtree views. For both views, you can specify that business names replace original identifiers in the diagram.

1. Choose **Tools > User Preferences**. The User Preferences window opens. Click the HyperView > Callie tab.
2. The current default background color of the box representing an item in the subgraph view is displayed in the **Default Box Color** drop-down. The current background color of the box representing the selected item in the subgraph view is displayed in the **Selected Box Color** drop-down. Click the adjacent ▾ button to edit the color of the boxes.
3. In the **Track Length** combo box, enter the number of visited nodes in the subtree view that Callie displays in different colors.
4. Select **Display Business Names** to display business names rather than original identifiers in the Callie pane.

Setting Callie Project Options

Use the Callie tab of the Project Options window to 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 limit the relationship types the diagram displays and filter out items by name.

1. Choose **Tools > Project Options**. The Project Options window opens. Click the Callie tab.
2. In the Connections pane, select the relationship types you want to view in the diagram. Select:
   - **Show GO TO** to show GO TO relationships.
   - **Show Performs** to show PERFORMS relationships.
   - **Show External Calls** to show external call relationships.
   - **Show Follows** to show Fall Thru relationships.
   - **Merge Duplicate Links** to show a single relationship line rather than multiple lines when constructs are connected by identical call relationships.
3. 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 these options to values lower than 5.
4. 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. Add patterns as needed.

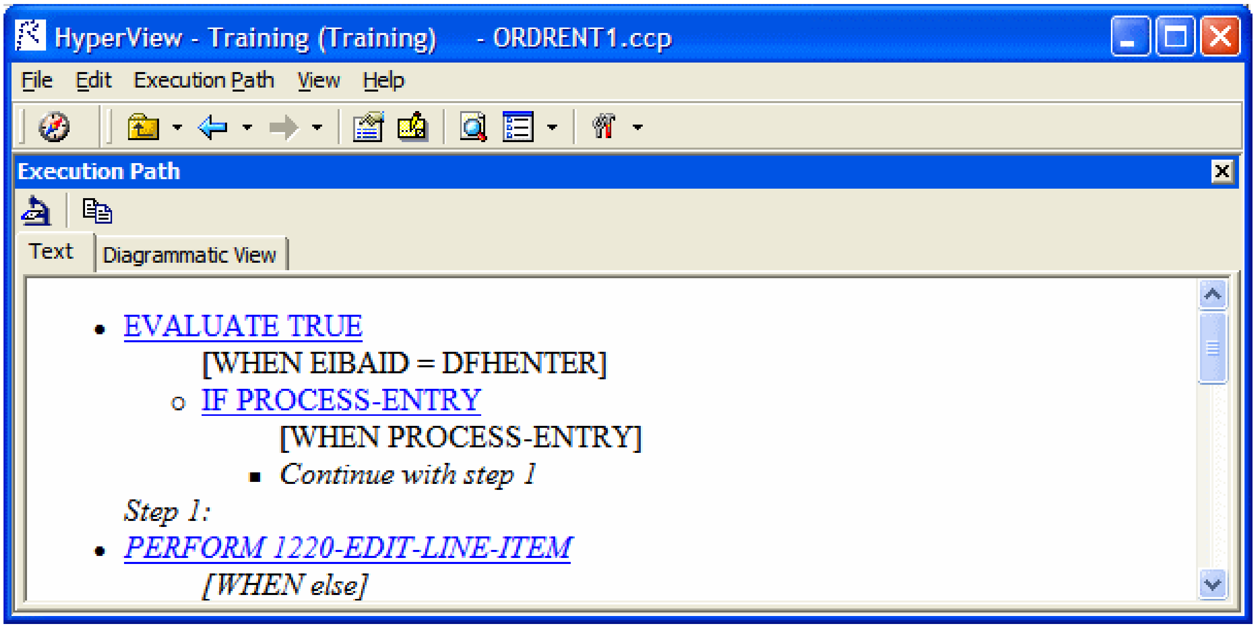
Using the Execution Path Pane

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 GO TO statements.

1. Select the seed construct in the Source or Context pane.
2. Choose **Execution Path > Analyze Selected Construct**. HyperView displays the execution path for the seed construct

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

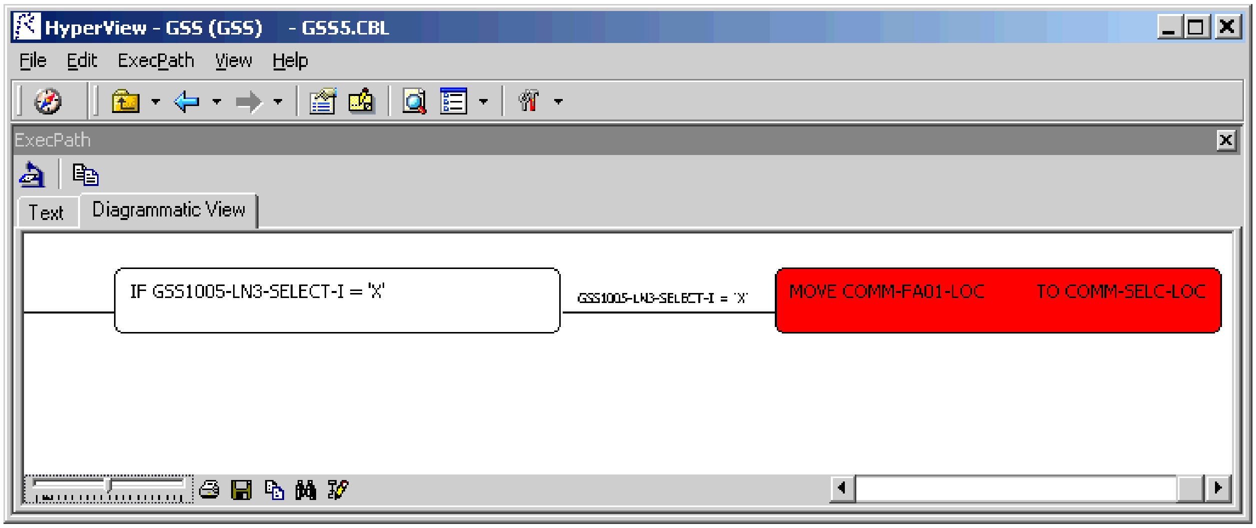


- EVALUATE TRUE
  [WHEN EIBAID = DFHENTER]
  - IF PROCESS-ENTRY
    [WHEN PROCESS-ENTRY]
    - *Continue with step 1*

Step 1:

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

3. Click the Diagrammatic View tab to display the execution path in a diagram. The seed construct is displayed in red. The tool bar displays icons used in the common diagrammer. For more information, see *Analyzing Projects* in the workbench documentation set.



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

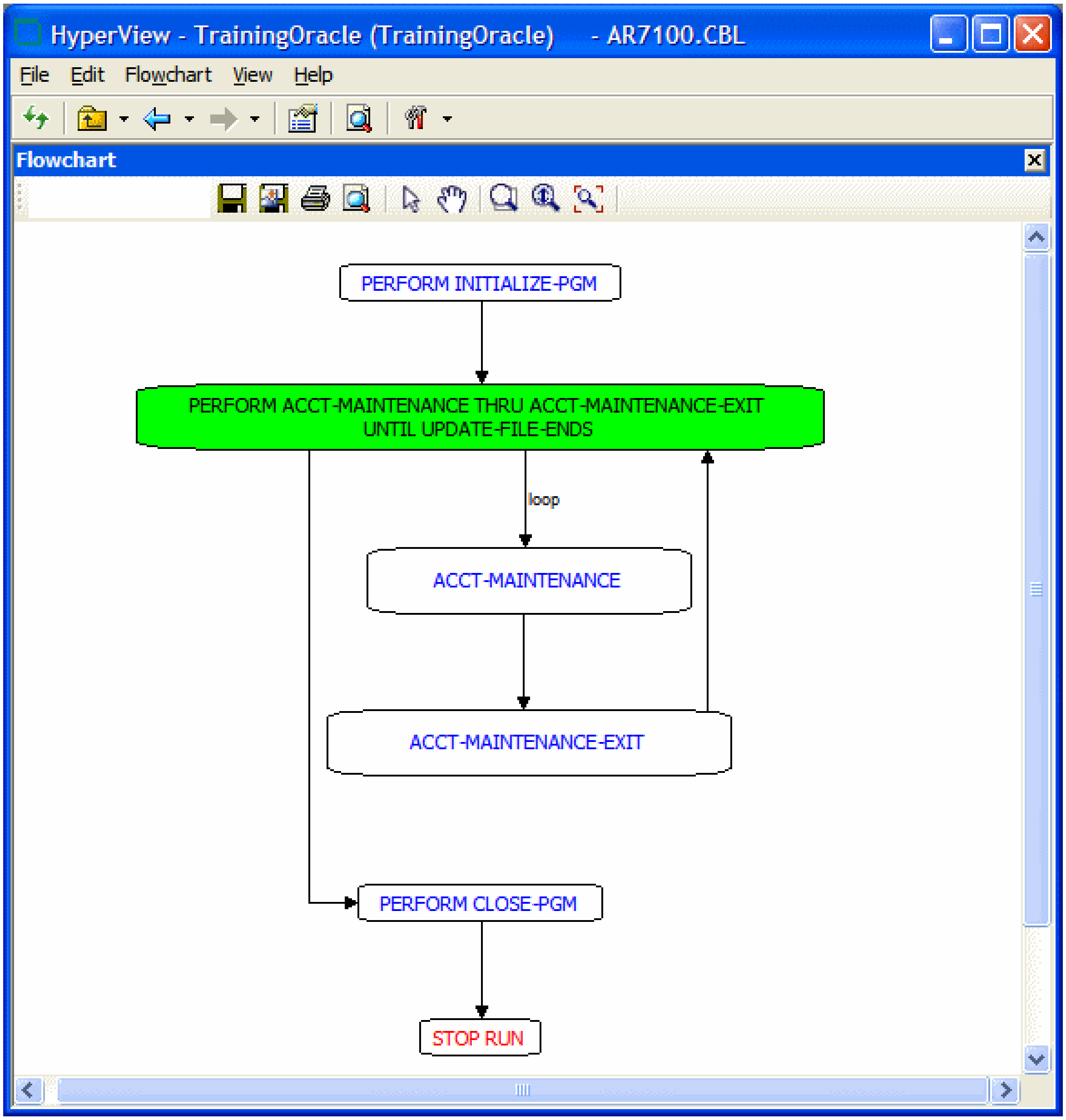
Using the Flowchart Pane

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 statements are displayed with a blue label. GO TO, RETURN, and STOP RUN statements are displayed with a red label. 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. Select a statement or condition to navigate to it in the Source and Context panes.

Job step relationships are collapsed in a group. Double-click the maximize button for the group to expand the relationships. Double-click the minimize button for the group to collapse the relationships.

Options control grouping of same-level statements and whether business names are displayed. The toolbar at the top of the pane displays icons used in the main Diagrammer window. For Diagrammer usage, see *Analyzing Projects* in the workbench documentation set.


HyperView - TrainingOracle (TrainingOracle) - AR7100.CBL
File
Edit
Flowchart
View
Help
Flowchart
PERFORM INITIALIZE-PGM
PERFORM ACCT-MAINTENANCE THRU ACCT-MAINTENANCE-EXIT UNTIL UPDATE-FILE-ENDS
loop
ACCT-MAINTENANCE
ACCT-MAINTENANCE-EXIT
PERFORM CLOSE-PGM
STOP RUN

Drawing the Diagram

To display a flowchart diagram in the Flowchart pane, select the paragraph, procedure, job, or JCL procedure you want to analyze in the Source or Context panes. The diagram is automatically redrawn when you select a new startup item in the Source or Context pane.

Setting Flowchart User Preferences

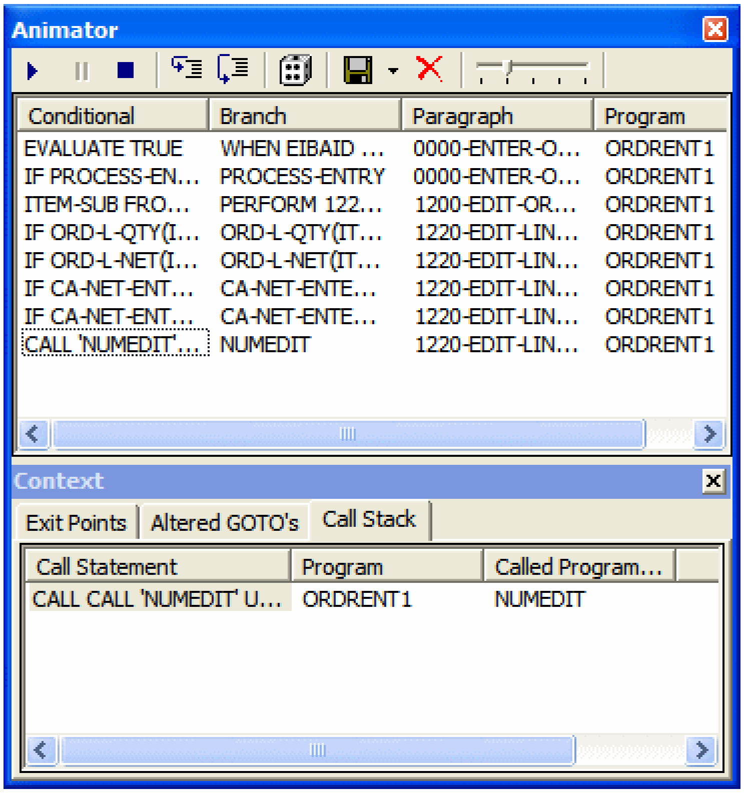
Use the HyperView > Flowchart tab of the User Preferences window to control grouping of same-level statements and whether business names are displayed.

1. Choose **Tools > User Preferences**. The User Preferences window opens. Click the HyperView > Flowchart tab.
2. Select **Group Same Level Names** to collapse same-level statements in a group. Double-click the maximize button for the group to expand the statements. Double-click the minimize button for the group to collapse the statements.
3. Select **Display Business Names** to display business names rather than original identifiers in the Flowchart pane.

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



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

1. Select the construct you want to start the animation from in the Source or Context pane.
2. Click the ⚄ button on the tool bar if you want Animator to choose logical branches randomly.

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

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

4. In the Context pane at the bottom of the window, click the appropriate tab to view exit points, altered GO TOs, or the call stack for the paragraph currently being executed. You can hide the Context pane by choosing the **Animator > Show Context**. Choose it again to show the Context pane.

   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.

5. 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.

6. 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.
   - 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.

7. Click the ✕ to delete the results from the Animator window.

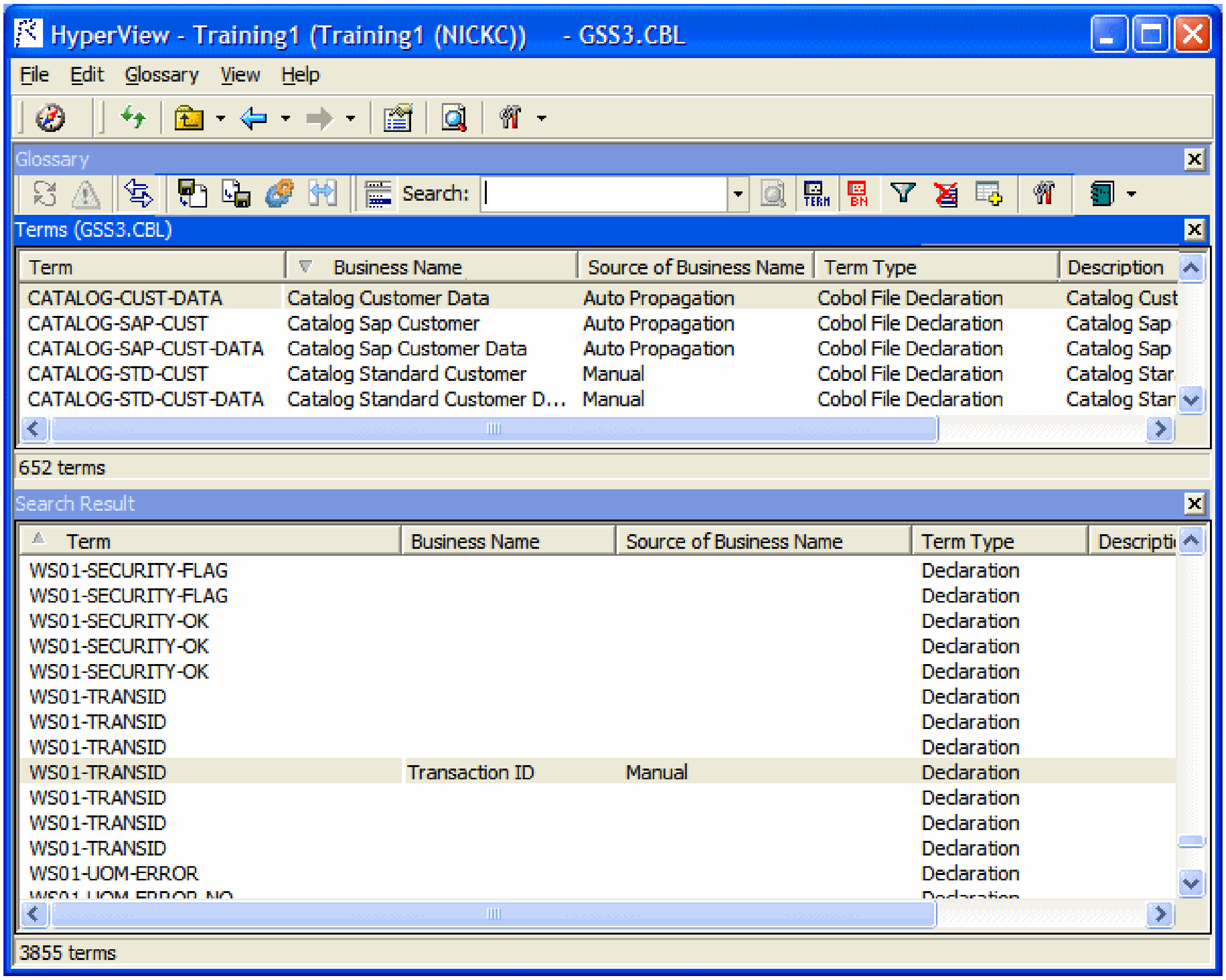
Setting Up a Glossary

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 auto-extract business names from screens, import business names into a glossary from a file, and much more.

Understanding the Glossary Pane

A *term* is the name of an item of interest in your workspace. The Glossary pane automatically generates a list of terms for the file selected in the Objects pane and any included files. Glossary search facilities let you home in on terms of interest in the workspace.



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

The Glossary tool consists of two panes:

- The Terms pane displays the terms in the selected file, filtered by your choices in the workspace options for the Glossary pane.
- The Search Result pane displays the terms returned by the most recent Glossary search. The scope of the search is workspace-wide, unless you use the advanced search feature to filter explicitly by project.

Each pane displays the term type, the business name and description assigned to the term, and how the business name and description were assigned (Manual, Automatic, Extracted from Screen, or From Propagation). To hide a pane, choose **Glossary > View Terms** or **Glossary > View Search Result**, as appropriate. Deselect the menu item to show the pane again.

To synch the Terms pane with other HyperView panes, choose **Glossary > Synchronize with other views**. Select a term in the Terms pane to navigate to the term in other HyperView panes. Deselect **Glossary > Synchronize with other views** to turn off synchronization.

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

Searching for Terms

Glossary search facilities let you home in on terms of interest in your workspace:

- Use a *simple search* to perform coarse-grained searches for terms.
- Use an *advanced search* to perform narrowly targeted searches for terms.

The scope of all searches is workspace-wide, unless you use the advanced search feature to filter explicitly by project. Results are returned in the Search Result pane. To clear the Search Result pane, choose **Glossary > Clear Search Result**.

Using the Simple Search Facility

Use a *simple search* to perform coarse-grained searches for terms. Follow the instructions below 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).

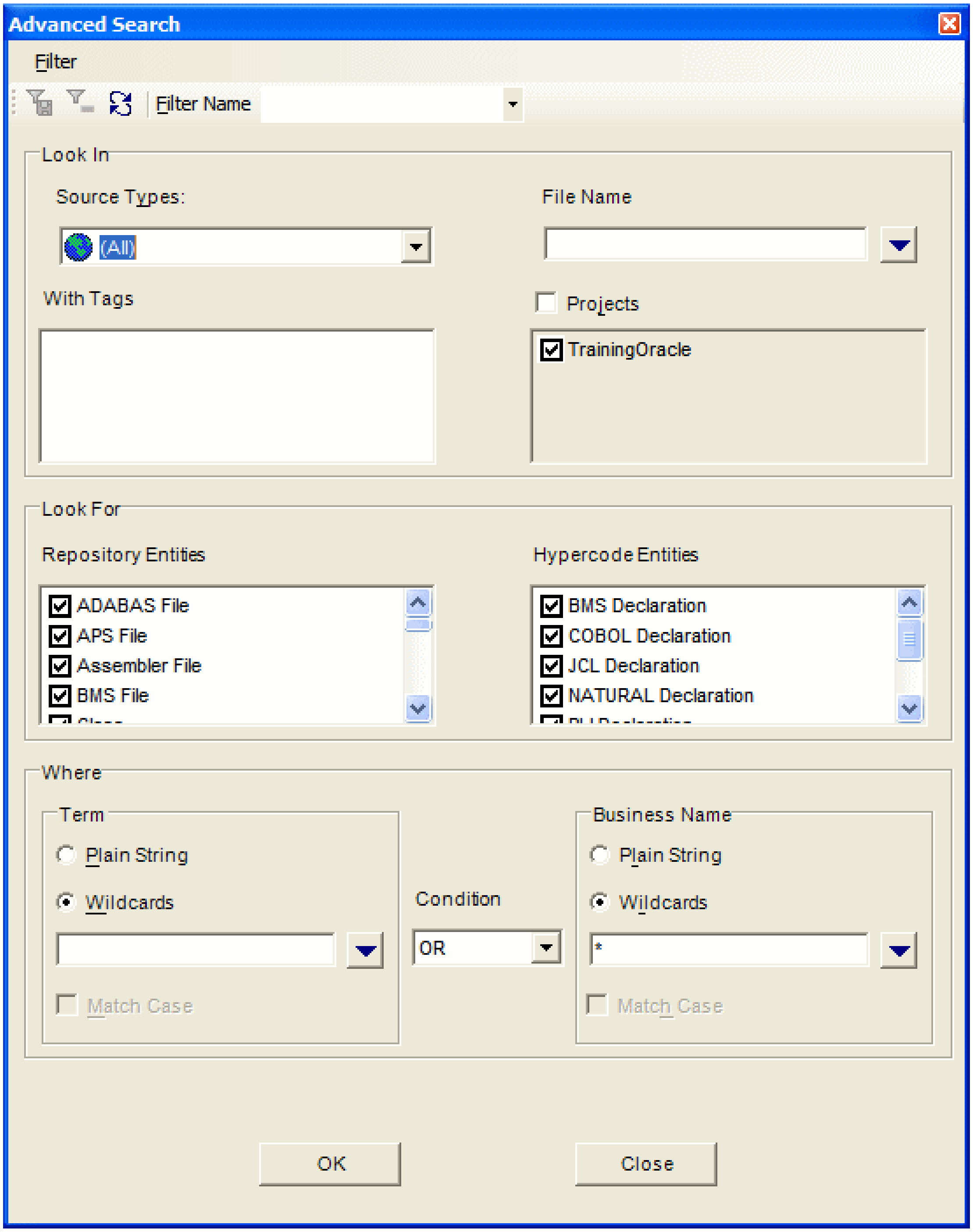
   Note: Select a column heading and click the button on the tool bar, then enter text in the Search combo box to scroll intelligently to items in that column. The tool matches text as you type.

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.

Using the Advanced Search Facility

Use an *advanced search* to perform narrowly targeted searches for terms. Follow the instructions below to use the advanced search facility.

1. Choose **Glossary > Advanced Search**. The Advanced Search dialog opens.



Source Types: (All)

File Name

With Tags

□ Projects

☑ TrainingOracle

Repository Entities

☑ ADABAS File
☑ APS File
☑ Assembler File
☑ BMS File

Hypercode Entities

☑ BMS Declaration
☑ COBOL Declaration
☑ JCL Declaration
☑ NATURAL Declaration

Term

○ Plain String
● Wildcards

□ Match Case

Condition: OR

Business Name

○ Plain String
● Wildcards

*

□ Match Case

2. You can create a new search filter or edit an existing search filter:
   - To create a new search filter, enter the name of the filter in the **Filter Name** combo box.
   - To edit an existing search filter, select the filter in the **Filter Name** combo box.

   Note: To save a search filter, choose **Filter > Save Current Filter**. To delete a search filter, choose **Filter > Delete Filter**. To restore a filter to its initial values, choose **Filter > Reset Values**.
3. To filter on source file type, select the type in the **Source Type** drop-down.
4. To filter on source file name, select the **File Name** check box, then select the matching pattern for the name in the combo box below. 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). 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.

5. 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.
6. To filter on projects, select the **Project** check box, then select the projects in the list pane below.
7. To filter on repository object type, select each object type you want to filter on in the Repository Entities pane.
8. To filter on HyperCode construct type, select each construct type you want to filter on in the Hypercode Entities pane.

   Note: Both repository and Hypercode entities are filtered by your choices in the workspace options for the Glossary pane.
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 or wildcard patterns.
   a) Click the radio button for the type of pattern matching you want to use.
   b) For 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 wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA). The matching pattern is saved when you save or execute the filter.
   c) In the **Condition** drop-down, select **AND** if you want the search terms to be ANDed, **OR** if you want the search terms to be ORed.
   d) Select the **Match Case** check box if you want the search to match case.
10. 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 **Glossary > Clear Search Result** to clear the Search Result pane.

Creating a List of Search Results

To create a Clipper list containing search results, choose **Glossary > Add to Clipper List**. Specify the list name in the **New list** dialog box and click **OK**. The list is displayed in the Glossary category in Clipper.

Assigning Business Attributes

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

Assigning Business Names Manually

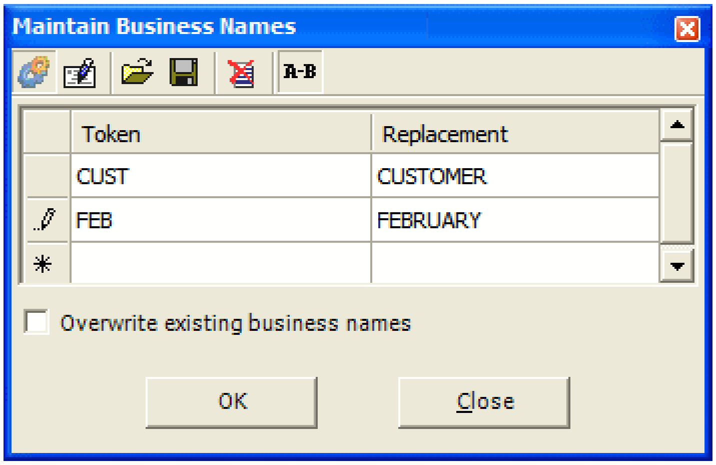
To assign a business name and business description to an object, select the object and choose **Set Business Attributes** in the right-click menu. A dialog box opens, where you can enter the business name and business description. To unassign a business name or business description, simply delete the value in the dialog box.

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 token or string you want to change in a term. For the string NUM, for example, you would enter the replacement text Number. You can also modify business names themselves in batch mode.

A token is an element in a program identifier delimited by a hyphen (-) or underscore (_). In the identifier WS01-CUST-FIELD, for example, there are three tokens: WS01, CUST, and FIELD.

1. Choose **Glossary > Generate/Modify Business Names.** The Maintain Business Names dialog opens.



Token	Replacement
CUST	CUSTOMER
FEB	FEBRUARY

□ Overwrite existing business names

2. Click the button on the toolbar to enable business name generation.
3. Click the A-B button on the toolbar to enable token or string replacement. The button is a toggle.
4. In the **Token/String** field, enter the token or string you want to replace. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA). In the **Replacement** field, enter the replacement string. Repeat the procedure for each token or string you want to replace.

   Note: Click the button on the toolbar to clear the dialog. Click the button on the toolbar to save the list of entries to a file. Click the button on the toolbar to load a list of entries into the dialog.

5. Select **Overwrite existing business names** if you want existing business names to be overwritten by the matching text.
6. Click **OK**. The Glossary tool automatically generates business names for each term that contains the token or string, replacing the string with the text you specified.
7. To modify business names, click the button on the toolbar, then repeat the procedure described above for string replacement.

Extracting Business Names from Screens

One of the most powerful features of Modernization Workbench lets you *auto-extract screen labels* as business names for program variables. For each screen field with a label immediately to its left in the

Screen pane, the Glossary tool generates a business name for the program variable whose value is displayed in the field. If the screen field has the label **Customer Number**, for example, Glossary assigns the business name "Customer Number" to the variable whose value is displayed in the field.

To extract business names from all the screens in the workspace, choose **Glossary > Extract From All Screens**. To extract business names only from the screen selected in the Source or Context panes, choose **Glossary > Extract From Current Screen**.

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="CUSTOMERINFORMATION"
   description="customer information"/>
  ...
</glossary>
```

To export business attributes, choose **Glossary > Export Business Attributes**. A save as dialog opens, where you can specify the name and location of the file to export.

To import business attributes, choose **Glossary > Import Business Attributes**. An open dialog displays, where you can specify the file to import. You are prompted to confirm whether you want to overwrite existing business attributes. Select **Overwrite existing business attributes** if you want to replace existing business attributes as well as add new ones. Click **OK**.

Propagating Business Names

You can *propagate* the business name for one identifier to a related identifier with an identical meaning in the data flow. 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 propagate business names from all existing business names, choose **Glossary > Propagate All**. To propagate business names from selected business names, choose a single business name in the Terms pane or multiple business names in the Search Result pane and choose **Glossary > Propagate Selected**. To propagate only from business names assigned in a specific way, choose **Glossary > Propagate <Assignment Method>**.

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, where you can delete the values in the **Business Name** and **Description** fields.

To delete all business names, right-click in the Terms or Search Result pane and choose **Delete Business Names > Delete All Business Names** in the pop-up menu. To delete only business names assigned in a specific way, right-click in the Terms or Search Result pane and choose **Delete Business Names > Delete <Assignment Method>** in the pop-up menu.

To delete all business descriptions, right-click in the Terms or Search Result pane and choose **Delete All Descriptions** in the pop-up menu.

Note: Mass delete operations are workspace-wide, and should normally be performed by the master user only.

Setting Glossary Workspace Options

Use the Glossary tab of the Workspace Options window to filter the objects and constructs available in the Terms pane and the Advanced Search dialog, and to specify the types of data flow relationships for which business names are propagated.

1. Choose **Tools > Workspace Options**. The Workspace Options window opens. Click the Glossary tab.
2. Click the Entities tab, then the Hypercode tab. In the Source Type pane, select the source file type whose constructs you want to include, then select the constructs in the HyperCode Entities pane.
3. Click the Repository tab. Select each type of repository object you want to include.
4. Click the Propagation tab. Select each data flow relationship you want to propagate business names for.

   Note: Click **Select All** to select all the items on a tab. Click **Select All** again to deselect all the items.

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. To refresh a glossary, choose **Glossary > Refresh Glossary**.

Generating Glossary Reports

The Glossary tool offers two reports:

- The Terms report prints the contents of the Terms pane. To generate a Terms report, choose **Glossary > Terms Report**.
- The Search report prints the contents of the Search Result pane. To generate a Terms report, choose **Glossary > Search Report**.

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. The rule 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.
```

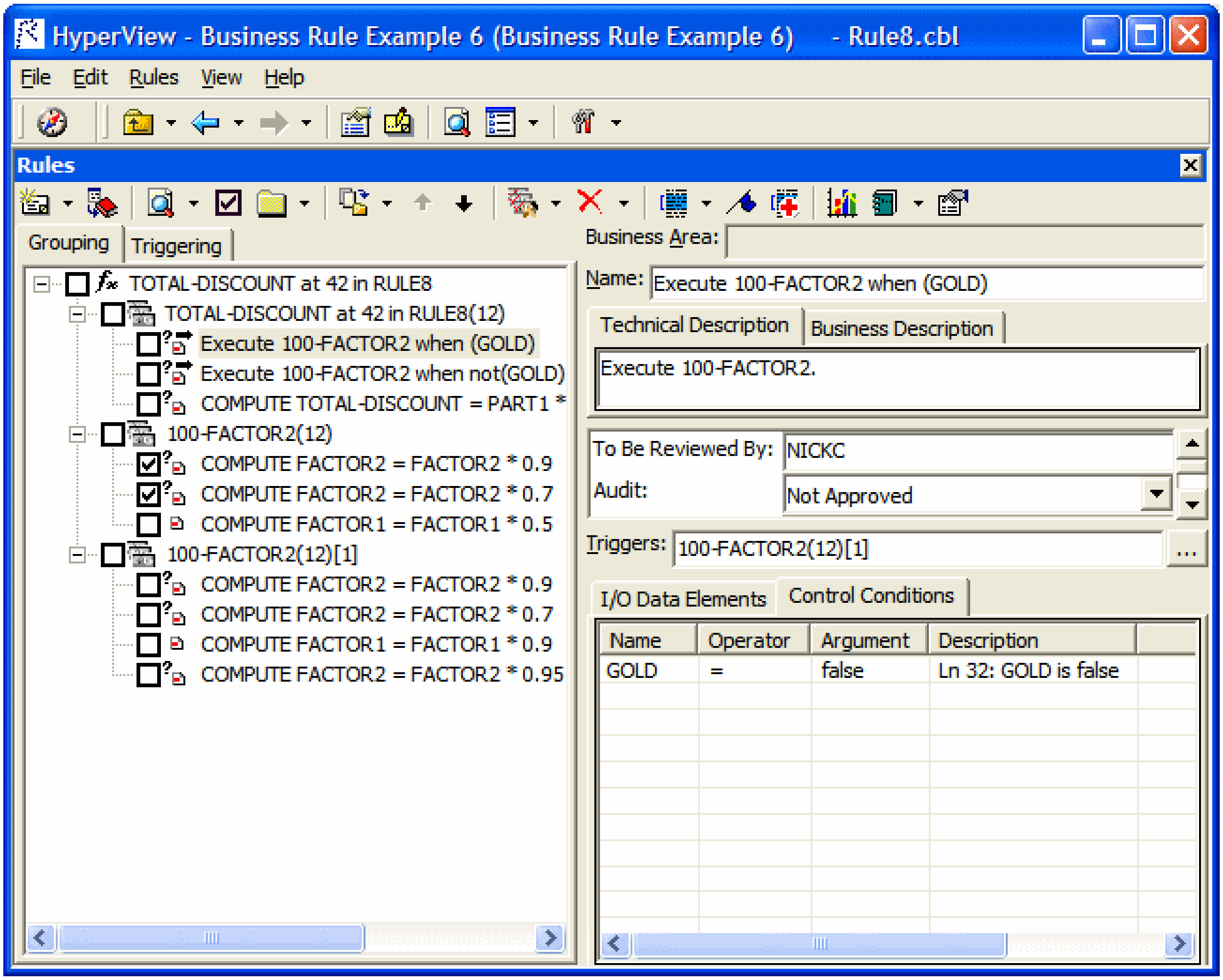
A business rule can contain one segment only. The same segment can be assigned to different rules. Rules are identified in the Rules pane with a ▣ symbol.

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. The figure below shows the HyperView Rules pane with a typical hierarchy of business functions, rule sets, and business rules.



- □ 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)

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

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.

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, 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.
- 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.

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

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

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

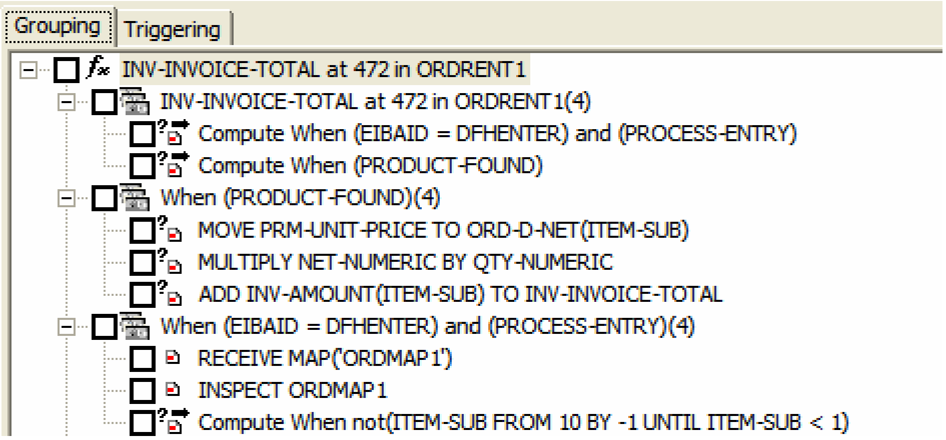
- *I/O data elements* identify the input, output, and input/output fields for a rule.
- *Control conditions* identify the conditions that govern the execution of a rule. A rule may be executed, for example, only when the value of the State Field is California. Rules that execute conditionally are identified 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

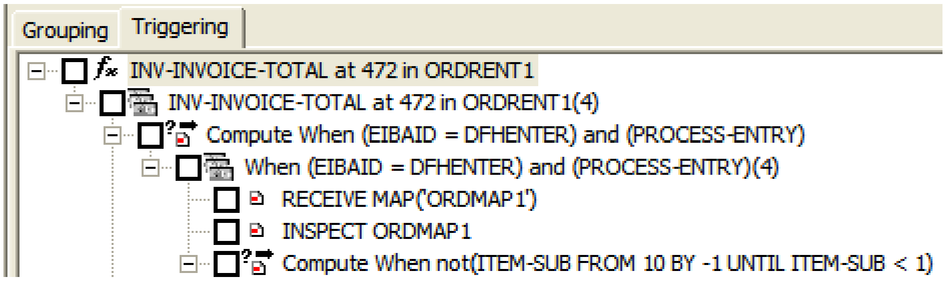
The lefthand portion of the Rules pane displays the business functions in your workspace in tree form:

- The Grouping tab lists rules in order of execution, if they were autodetected, or in the order you created them manually, organized by business function and rule set.
- 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.

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



```
⊟ □ fx 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)
```



Grouping | Triggering

- □ 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 instructions describe 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.

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.

   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 for creating business functions.
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 for creating rule sets.
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 remaining attributes of the rule.
5. Click **OK**.

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 Output folder for the workspace.

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.
- 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. In either case, you can "prune out" of the trace conditional logic not of interest to you.

BRM project options determine the impact relationships traced by the backward autodetect, how the 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.

Note: When you verify the project, set **Enable Impact Report** and **Enable Data Element Flow** in the project verification options to enable the impact trace.

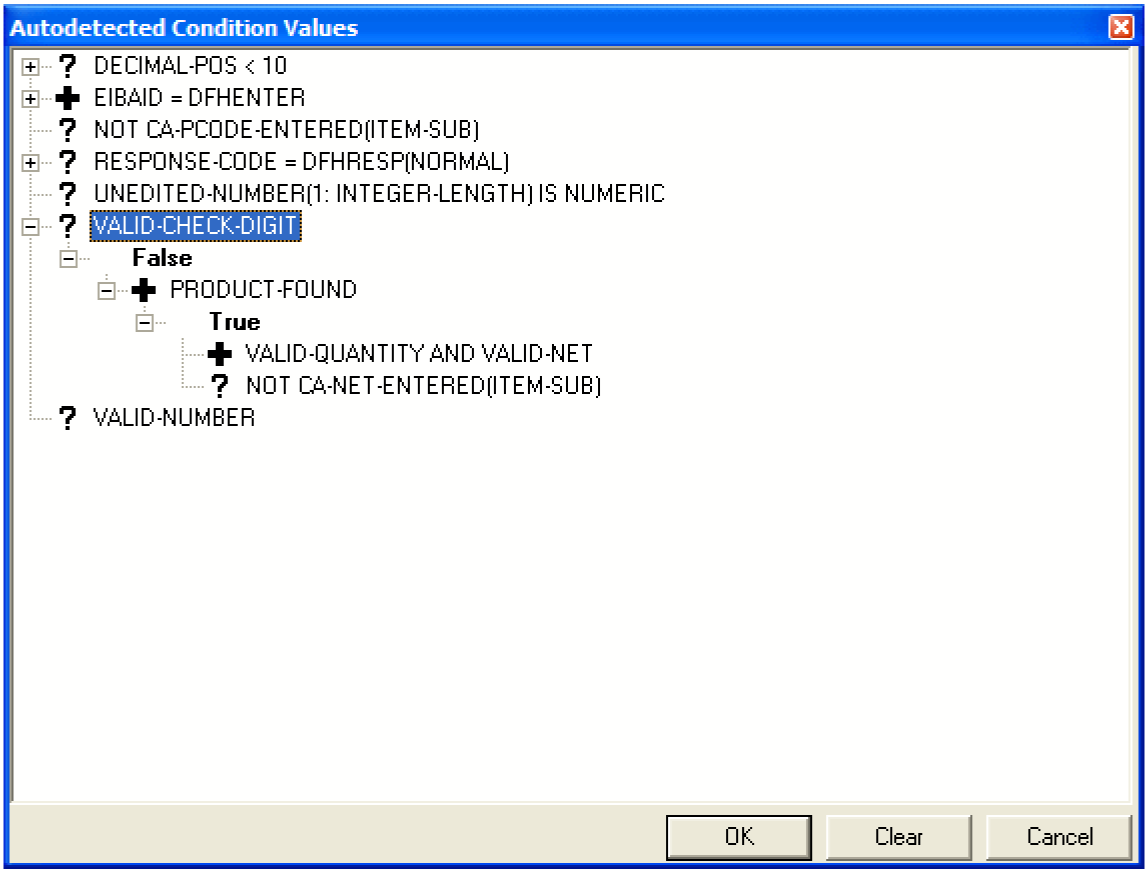
1. Select the seed item in the HyperView Source or Context pane.
2. In the Rules pane, click the ▾ button next to the button on the tool bar and choose **Backward AutoDetect** in the drop-down menu.

   Tip: Cancel autodetection at any time during the process by clicking the **Cancel** button in the upper righthand corner of the HyperView window.
3. If the autodetect trace detects conditional logic, the Autodetected Condition Values dialog prompts you to "prune 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. When you are satisfied with your choices in the Autodetected Condition Values dialog, click **OK**.

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 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.



Autodetected Condition Values

```
⊞ ? 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 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.

If a business name exists for a variable, it replaces the variable name in a business function, rule set, or rule. If a business name exists for a program, it replaces the program name in a rule set.

Note: 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. Recreate the rule manually.

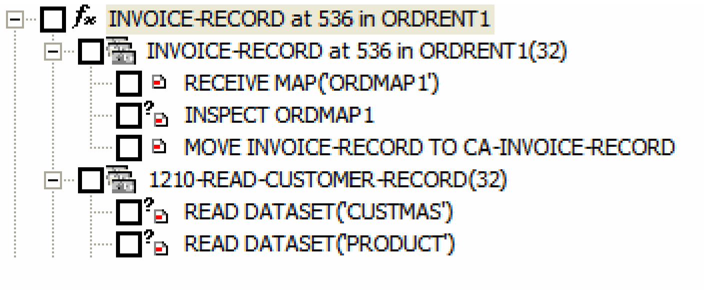
Autodetecting Called Programs

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:

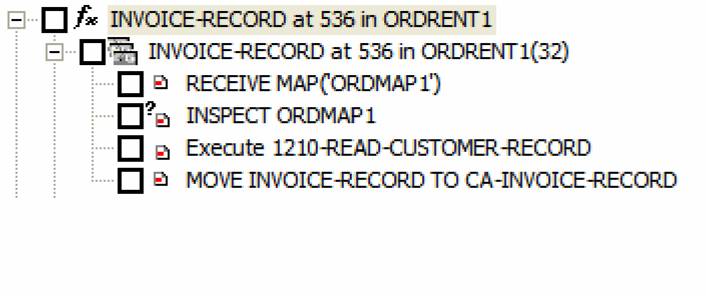
- Skip specified called programs.
- Group rule sets for specified called programs in discrete business functions.

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.



```
□ fx 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

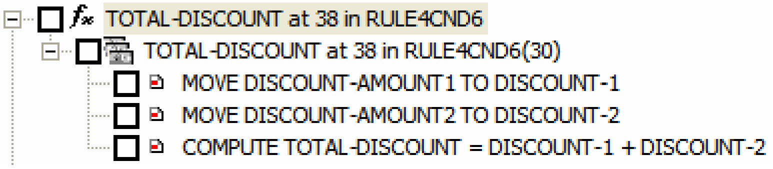
Use the Business Rule Manager autodetect options to group autodetected rules based on the paragraphs they are mined from. Rules mined from a specified "paragraph of interest" are assigned to a rule set with a name of the form Paragraph*Paragraph Name*. A triggering rule with a name of the form Execute*Paragraph Name* is created in the executing rule set, as shown in the comparison below.

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.

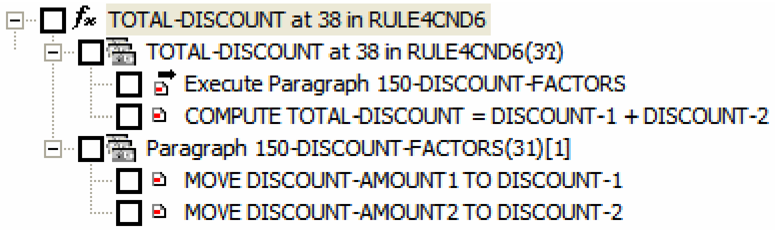
If the grouping rule set contains rules mined from three or more statements with at least one common condition, a distinct rule set is created for the rules with a triggering rule in the grouping rule set.

For 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.



□ 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 initialized by zero or a blank line. Use the Business Rule Manager autodetect options to skip these items.

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. 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.

If a business name exists for a variable, it replaces the variable name in a business function, rule set, or rule. If a business name exists for a program, it replaces the program name in a rule set.

1. Select the receive screen statement from which you want to trace screen validations in the HyperView Source or Context pane.
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.

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.

Creating Business Functions

To create a business function, choose **Rules > Create Business Function**. The Business Function dialog opens. In the Business Function dialog, enter the attributes of the new business function and click **OK**.

Creating Rule Sets

To create a rule set, select the business function for the rule set in the lefthand pane and choose **Rules > Create Rule Set**. 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**.

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

Note: Choose **Rules > Selected Only** 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 **Rules > Copy/Move Selected Rules**. 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 **Rules > Copy/Move Selected Rule Sets**. 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 **Rules > Copy/Move**u. 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 **Rules > Copy/Move**. 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 **Rules > Delete**. 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 **Rules > Undo Delete**.

Changing the Order of Rules and Rule Sets

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

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

Exporting Rules, Rule Sets, and Business Functions

To export rules, rule sets, or business functions to an ERD file, for import to another workspace or a third-party tool, select the items you want to export and choose **Rules > Rule Export**. 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 **Rules > Rule Import**. 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 **Rules > Assign segment**.

Deleting a Segment from a Rule

To delete a segment from a rule, select the rule and choose **Rules > Delete segment**.

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 for a rule, select the rule and choose **Rules > Activity Log**.

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.

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, 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.
- 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.

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

Editing System Attributes

The names and characteristics of 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. For autodetected rules, the field is automatically populated with the code segment for the rule, in the form *Segment*at*Line Number*in*Program*. 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 To Be Reviewed By 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.

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.

Editing User-Defined Attributes

User-defined attributes can be renamed, defined with different values, or deleted altogether. 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

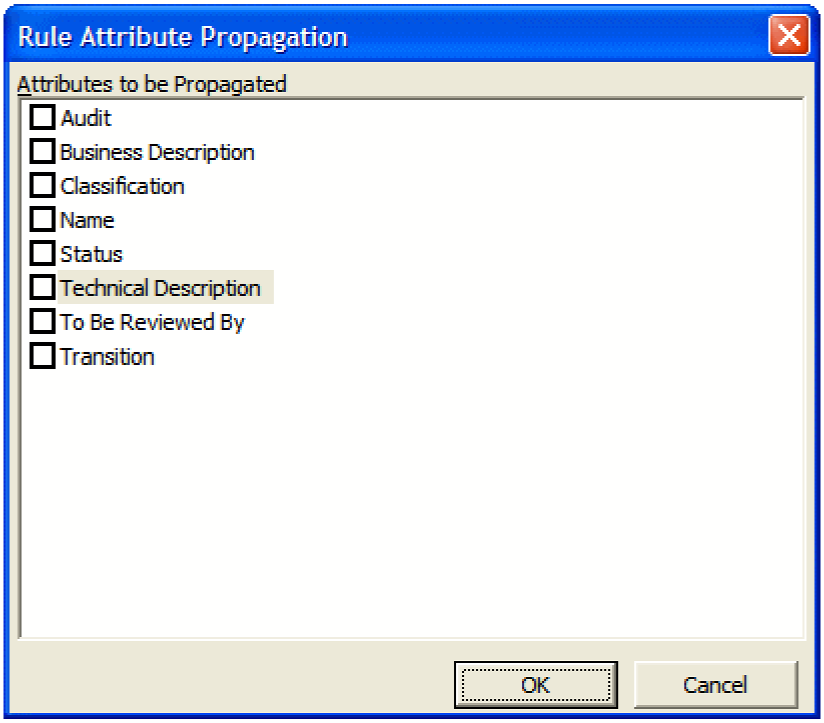
The Transition attribute identifies the status of a rule in an application redevelopment project. To edit the Transition attribute, choose one of the following from the **Transition** drop-down:

- 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 to a group of rules

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 **Rules > Propagate from Highlighted to Selected**. You are prompted to continue. Click **Yes.** The Attribute Propagation window opens.



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**.

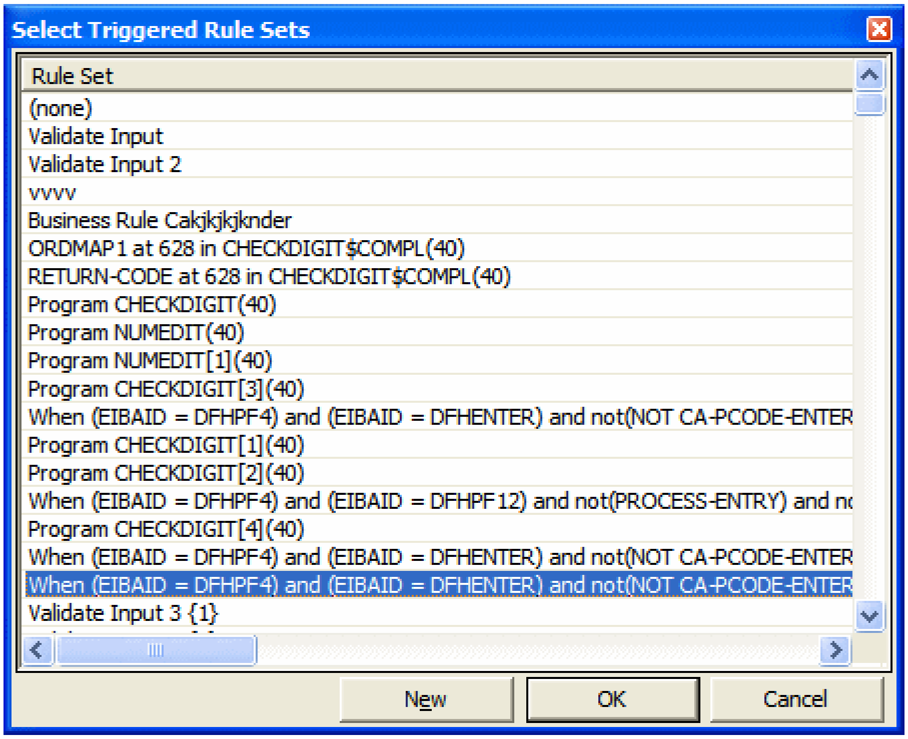
Identifying Triggered Rule Sets

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, for example, in the rule set Validate Input might trigger the rule set Update File.

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.

1. Select the triggering rule and click the button next to the **Triggers** field in the righthand pane. The Select Triggered Rule Sets window opens.



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}

2. In the Triggered Rule Sets window, 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.

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.

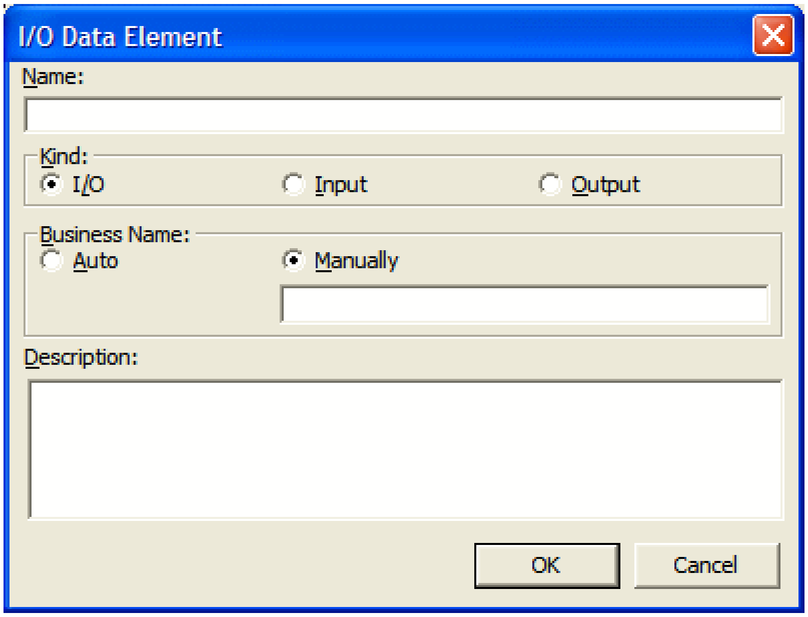
Use the autoretrieve feature to match I/O data elements with the business names you've assigned to them in your project glossary. A business name identifies the business function of a variable in natural language, and is invaluable in documenting and retrieving business rules. The field PLC_NUM, for 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.

Identifying I/O Data Elements Manually

Follow the instructions below to identify 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.



Kind: ● I/O ○ Input ○ Output

Business Name: ○ Auto ● Manually

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 this procedure 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, where you can make the necessary changes. To delete a field, select it and choose **Delete** in the right-click menu.

Autodetecting I/O Data Elements

To autodetect I/O data elements, 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**.

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: When you verify the project, set **Enable Impact Report** and **Enable Data Element Flow** in the project verification options 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.

Retrieving Business Names for I/O Data Elements

To match I/O data elements with the business names you've assigned to them in your project glossary, 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**.

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.

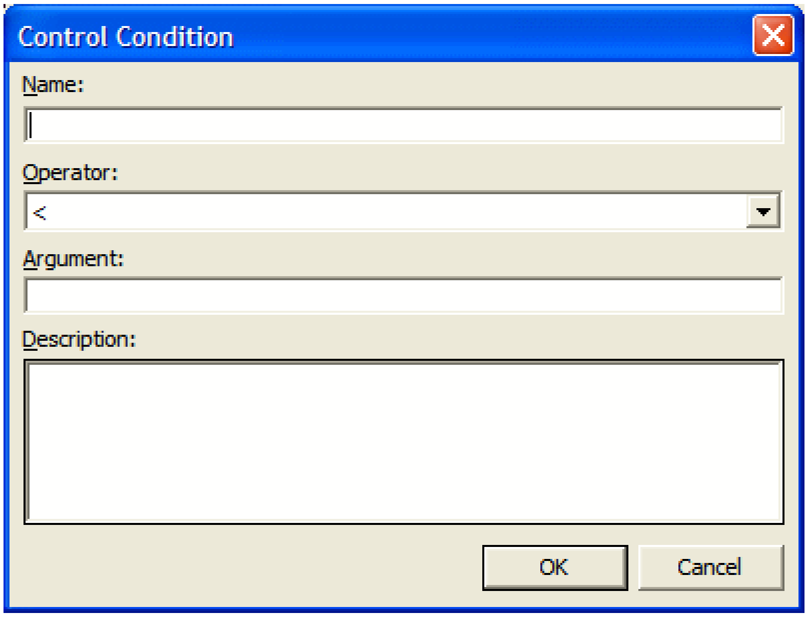
Use the autoreplace feature to replace variable names in control conditions with the business names you've assigned to them in your project glossary. A business name identifies the business function of a variable in natural language, and is invaluable in documenting and retrieving business rules. The 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.

Specifying Control Conditions

Follow the instructions below to specify control conditions for a rule.

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.



Operator: <

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 , where you can make the necessary changes. To delete a field, select it and choose **Delete** in the right-click menu.

7. Repeat this procedure for each control condition you want to add to the rule.

Autoreplacing Variable Names in Control Conditions with Business Names

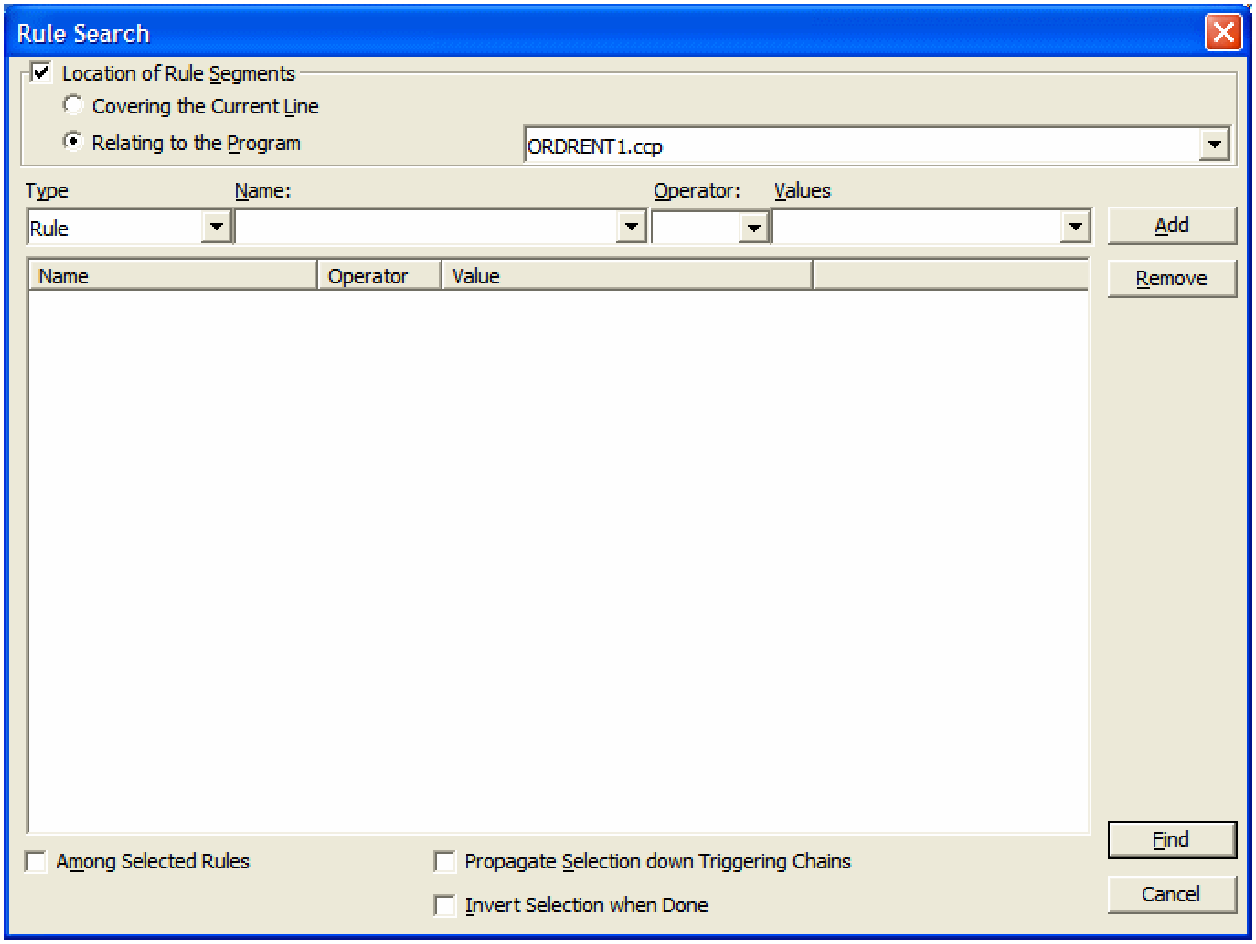
To autoreplace variable names in control conditions with their business names in a project glossary, select a business function, rule set, or rule, then right-click in the Control Conditions tab and choose **Substitute Business Names** in the pop-up menu. You are prompted to continue. Click **Yes**.

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.

1. Select a business function, rule set, or rule and choose **Rules > Start Search**. The Rule Search window opens.



Rule Search

☑ Location of Rule Segments

○ Covering the Current Line

● Relating to the Program ORDRENT1.ccp

Type: Rule

Name:

Operator:

Values

Name	Operator	Value	

□ Among Selected Rules

□ Propagate Selection down Triggering Chains

□ Invert Selection when Done

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.
4. To search only the selected rules, select **Among Selected Rules**.
5. To find rules in the triggering hierarchy for the selection, 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 Rule Attributes

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.

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.

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 **Rules > Previous Selected**. To return to the rules selected in the next most recent search, select **Rules > Next Selected**.
2. Choose **Rules > Change Rule Attributes.** You are prompted to continue. Click **Yes.** The Change Rules window opens.
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.
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.

Usage Example

Suppose you wanted to use the Change Rules feature 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
```

1. Select the check boxes for Rule 1 of program ABC and Rule 2 of program XYX.
2. Choose **Rules > Change Rule Attributes.** You are prompted to continue. Click **Yes.** The Change Rules window opens.
3. In the **Select Attribute** field, choose the name attribute.
4. In the **Find What** field, enter:

   ```
   of program [Program]
   ```
5. In the **Replace With** field, enter:

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

Substituting a Business Name for an I/O Element in an Attribute Value

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 in the Change Rules window.

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]
```

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, choose **Rules > Sync Sources**.

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 reverification.

BRM automatically handles invalidated segments based on your instructions in the Business Rule Manager > Automatic Rule Validation tab of the Project options window. Use the procedure below when you need to override the specified handling for a rule.

1. Select the rules you want to validate and choose **Rules > Validate**. You are prompted to continue. Click **Yes.** The Automatic Rule Validation dialog opens.
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.
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.
4. Click **OK**.

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.

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 Business Rule Manager > Rule Defaults tab of the Project options window to define the rule name templates available when you use Clipper to create rules in a specified rule set.

1. Choose **Tools > Project Options**. The Project Options window opens. Click the Business Rule Manager > Rule Defaults tab.
2. The Rule Defaults tab displays the rule name templates available when you use Clipper to create rules in a specified rule set. Add templates as necessary.

Defining Rule Name Templates

Use the Business Rule Manager > Rule Defaults tab of the Project options window to define the rule name templates available when you use Clipper to create rules in a specified rule set.

You can use any combination of text and bracketed property names in a rule name 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
```

Setting Options on the AutoDetect Tab

Use the Business Rule Manager > AutoDetect tab of the Project options window to specify BRM 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.

1. Choose **Tools > Project Options**. The Project Options window opens. Click the Business Rule Manager > AutoDetect tab.
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.
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. Add called programs as necessary.
   - 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. Add called programs as necessary.
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. Add paragraphs as necessary.
   - Paragraphs of interest pane, place a check mark next to each paragraph of interest you want the autodetect method to use in identifying results. Add paragraphs as necessary.
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. Add initial constant values as necessary.
   - 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. Add I/O data element template names as necessary.

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

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 reverification. BRM automatically handles invalidated segments based on your instructions in the Business Rule Manager > Automatic Rule Validation tab of the Project options window.

1. Choose **Tools > Project Options**. The Project Options window opens. Click the Business Rule Manager > Automatic Rule Validation tab.
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.
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 Business Rule Manager > Rule Export tab of the Project options window to specify whether BRM automatically includes triggered rule sets when it performs a rule export.

1. Choose **Tools > Project Options**. The Project Options window opens. Click the Business Rule Manager > Rule Export tab.
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.

Customizing Rule Attributes

Use the Modernization 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.

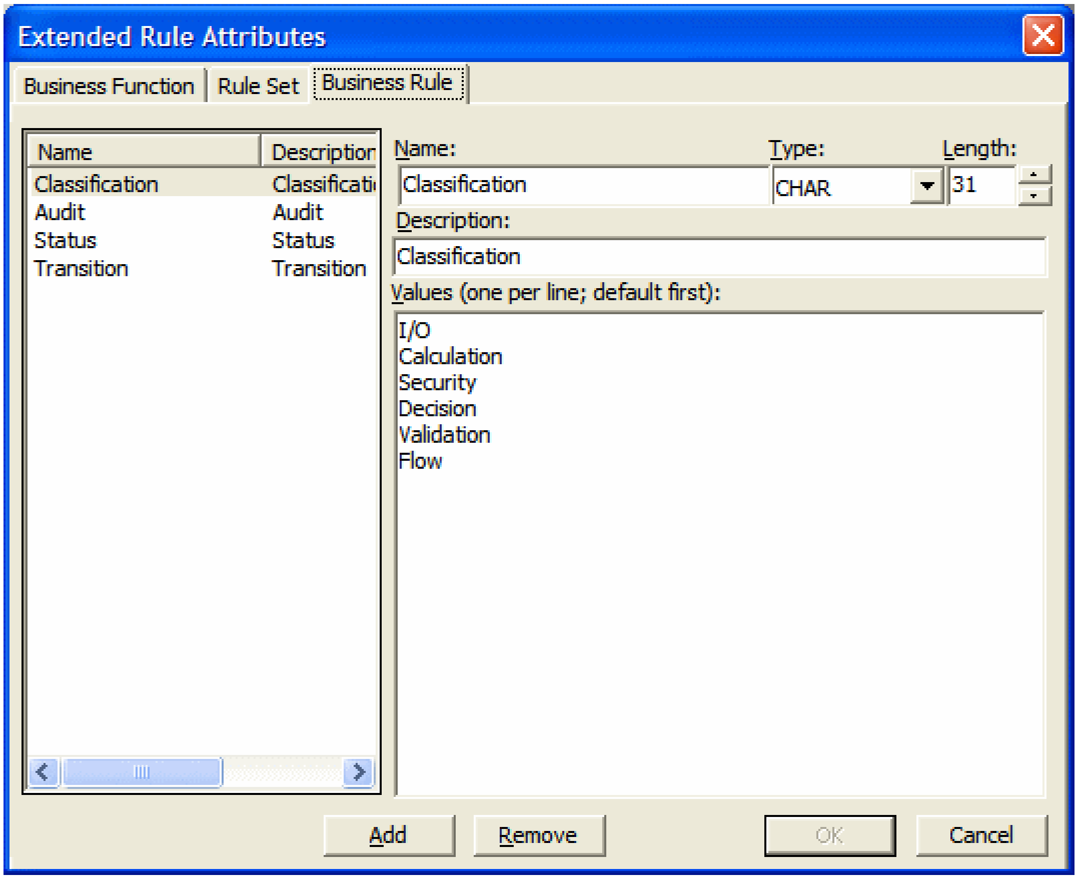
Choose **Rules > Extended Rule Attributes** to open a limited version of this tool in which you can modify or delete the values for character-type attributes. If you delete values in use in existing business functions, rule sets, or rules, the default value of the attribute is restored.

Note: You must upgrade workspaces after customizing rule attributes. For upgrade instructions, see the installation manual for your product.

1. In the Modernization Workbench Administration window, choose **Administer > Edit Rule Attributes**.

    - If you have never customized attributes, the Extended Rule Attributes window opens automatically with the default attributes.
    - If you have already customized attributes, and **Administer > Use Latest File** is not selected, 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.
    - If you have already customized attributes, and **Administer > Use Latest File** is selected, 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.



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

2. Click the Business Function, Rule Set, or Business Rule tab as appropriate.

3. 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**.
4. 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.

   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. 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 **Administer > Use Latest File**, 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.

You must upgrade workspaces after customizing rule attributes. For upgrade instructions, see the installation manual for your product.

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.

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.
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 a Process Outline Report, 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 a Coverage Report, click the 📒 button on the tool bar and choose **Coverage Report** in the pull-down menu. The Coverage Report window opens.

Using the Batch Duplicate Finder

Use the Batch Duplicate Finder (BDF) to perform sophisticated comparisons of similarly structured paragraphs. You can also use it to analyze paragraphs or procedures across workspaces, and to compare entire programs for their similarity.

Finding Duplicates

Follow the instructions below to find duplicate paragraphs with BDF. Make sure to verify the workspace files you plan to compare before running the tool.

1. If BDF is not already open, double-click the file \<*Workbench Home*>\Bin\BDF.exe. If BDF is already open, choose **Options > Startup Options**. In either case, the BDF scope window opens.
2. In the BDF scope window, specify the scope of the comparison:

   - In the **Select Model** drop-down, select the object model for the workspace(s) to be analyzed.
   - 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. 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.
3. Click **OK**. The BDF main window opens.
4. Choose **File > Open Workspace.** The Open Workspace dialog opens. 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.**
5. The Select Projects window opens. 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**.
6. 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.

   Tip: Click an item to view its source in the Text pane. Select the appropriate choice in the **View** menu to show or hide a pane.
7. Set BDF search options.
8. Select and mark the items you want to compare, then choose **File > 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.

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 **List > Check All**. To unmark all the items in the section, choose **List > Uncheck All.**

To mark selected items in a section, select the items and choose **List > Check Selected**. To unmark selected items in the section, choose **List > 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 **List > Select Paragraphs** or **List > Select Programs**. The BDF Define Mask window opens.

Enter the mask for the items you want to select in the **Paragraph name Like** field, then click **OK.** You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).

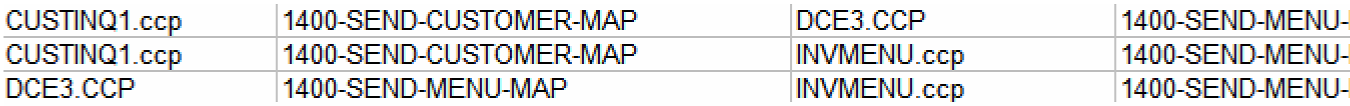
Deselect **Paragraph Action** or **Program Action** to select items that do not match the mask. To mark the items selected with the mask, choose **Check Selected** in the **List** menu.

Setting BDF Search Options

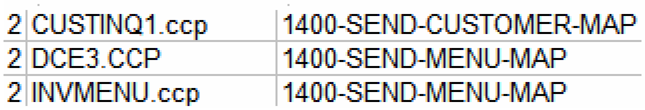
BDF search options control the types of comparison BDF performs, the columns it displays in the output, the name and location of the output file, and how it handles copybooks.

1. Choose **Options > Search Options**. The BDF Options window opens.
2. In the BDF Options window, 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. 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.



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



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.

4. 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.
5. 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.
6. 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.
7. 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.

Index