IBM Asset Transformation Workbench
v1.1

Analyzing Projects

SC31-6880-00

IBM Asset Transformation Workbench
v1.1

Analyzing Projects

SC31-6880-00

Note:

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

First Edition (February 2005)

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

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

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

Product Reference: IBM Asset Transformation Workbench v1.1

Document Reference: REL7.3.07.DOC03.A

Contents

Preface

Audience xi
Organization xii
Conventions xiii
Related Manuals xiii
Online Help xiv

1 Overview

Analyzing Relationship Flows 1-1
Analyzing Data Operations 1-3
Estimating Complexity and Effort 1-3
Defining an Application Namespace 1-4
Identifying Classes of Data Items 1-4
What's Next? 1-5

2 Analyzing Relationship Flows

Understanding Relationship Flow Diagrams 2-2
Diagram Scopes 2-2
Diagram Views 2-3

Using the Diagrammer 2-4
Generating Project Diagrams. 2-4
Working with Project Diagrams 2-5
Editing the Color Scheme for Project Diagrams. 2-7
Defining Scopes for Project Diagrams 2-9
Searching for Objects in Project Diagrams 2-13
Exporting Project Diagrams and Diagram-Based Reports. 2-13
Using the Diagram Editor 2-14
Opening a Diagram in the Diagram Editor 2-14
Searching for Objects in the Diagram Editor 2-15
Selecting Objects in the Diagram Editor 2-17
Editing the Layout of a Diagram in the Diagram Editor 2-17
Editing Objects in the Diagram Editor 2-19
Editing Relationships in the Diagram Editor. 2-20
Editing General Properties of a Diagram in the Diagram Editor. . 2-21
Saving Diagrams in the Diagram Editor. 2-22
Copying Diagrams with a Different Name (Saving As) in the Diagram Editor 2-22
Printing Diagrams in the Diagram Editor 2-23
Exporting Diagrams in the Diagram Editor. 2-23
Using the Quick Diagrammer 2-23
Generating Object Diagrams 2-23
Changing the Scope of an Object Diagram 2-24
Defining Scopes for Object Diagrams 2-25
Mirroring the Scope of an Object Diagram. 2-25
Exporting Multiple Object Diagrams. 2-26
Moving or Copying Source Files for Diagrammed Objects into Projects 2-26
What's Next?. 2-27

3 Analyzing Global Data Flows

Understanding Global Data Flows 3-2
Searching for the Variable 3-2
Viewing Memory Usage for the Variable 3-3
Viewing the Data Flow for the Variable 3-4

Using the Global Data Flow Tool 3-5
Generating Global Data Flow Information 3-5
Working with Global Data Flow Information 3-7
Setting Global Data Flow Options 3-9
Copying Global Data Flow Diagrams to the Clipboard 3-13
Printing Global Data Flow Diagrams 3-13
Exporting Global Data Flow Diagrams 3-13
What's Next? 3-13

4 **Analyzing Batch Applications**

Using the Batch Application Viewer 4-1
Generating Batch Application Information 4-2
Working with Batch Application Table Views 4-5
Searching Batch Application Table Views 4-7
Working with Batch Application Diagrams 4-7
Working with HyperView in Batch Application Viewer 4-12
What's Next? 4-12

5 **Analyzing PL/I Applications**

Understanding PL/I Call Flow Diagrams 5-1
Using the PL/I Call Diagrammer 5-2
Generating PL/I Call Information 5-2
Setting PL/I Call Diagram Options 5-3
Copying PL/I Call Diagrams to the Clipboard 5-5
Printing PL/I Call Diagrams 5-5
Exporting PL/I Call Diagrams 5-6
What's Next? 5-6

6 **Analyzing Data Operations**

Viewing CRUD Reports 6-1
Enabling IMS Port Analysis 6-3
What's Next? 6-5

7 Estimating Complexity and Effort

Using the Complexity Metrics Tool 7-2
Using the Effort Estimation Tool 7-3
Specifying Change Magnitudes 7-6
Exporting Complexity and Effort Reports 7-7
Supported Complexity Metrics 7-8
What's Next?. 7-14

8 Defining an Application Namespace

Understanding Application Namespaces 8-1
Getting Started in the Application Namespace Tool 8-3
Using the Application Namespace Tool 8-4
- *Working with a Project Dictionary* 8-4
- *Setting Application Namespace Options* .8-8
- *Refreshing a Project Dictionary* 8-9
- *Generating and Exporting Application Namespace Reports* 8-9

What's Next?. 8-10

9 Identifying Classes of Data Items

Understanding Data Item Classification 9-1
Getting Started in Change Analyzer 9-3
Using Change Analyzer 9-7
- *Working with Change Impact Information* 9-7
- *Setting Change Analyzer Options* 9-10
- *Searching for Seed Fields in Change Analyzer* 9-13
- *Creating Projects in Change Analyzer* 9-15
- *Generating and Exporting Change Analyzer Reports* 9-16

What's Next?. 9-17

Glossary

Bibliography

Notices

Index

Preface

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

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

Audience

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

Organization

This guide contains the following chapters:

- Chapter 1, “Overview,” provides an overview of the ATW project analysis tools.
- Chapter 2, “Analyzing Relationship Flows,” describes how to use the Diagrammer to perform high-level analysis of applications.
- Chapter 3, “Analyzing Global Data Flows,” describes how to use the Global Data Flow tool to perform low-level analysis of program data flows.
- Chapter 4, “Analyzing Batch Applications,”describes how to use the Batch Application Viewer to perform low-level analysis of batch processes.
- Chapter 5, “Analyzing PL/I Applications,” describes how to use the PL/I Call Diagrammer to perform low-level analysis of PL/I programs.
- Chapter 6, “Analyzing Data Operations,” describes the workbench CRUD report and IMS port analysis feature.
- Chapter 7, “Estimating Complexity and Effort,” describes how to use the legacy estimation tools to estimate project complexity and effort.
- Chapter 8, “Defining an Application Namespace,” describes how to use the Application Namespace tool to change identifiers in system source code without undermining the system’s consistency.
- Chapter 9, “Identifying Classes of Data Items,” describes how to use the Change Analyzer to identify the class of data items used to perform a business function in a legacy application.
- The Glossary defines the names, acronyms, and special terminology used in this guide.

Conventions

This guide uses the following typographic conventions:

- **Bold type** — Indicates a specific area within the graphical user interface, such as a button on a screen, a window name, or a command or function.
- *Italic type* — Indicates a new term. Also indicates a document title. Occasionally, italic type is used for emphasis.
- `Monospace type` — Indicates computer programming code.
- **`Bold monospace type`** — Indicates input you type on the computer keyboard.
- **1A**/**1B**, **2A**/**2B** — In task descriptions, indicates mutually exclusive steps: perform step A or step B, but not both.

Related Manuals

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

- *Getting Started* introduces ATW. This guide provides an overview of the workbench tools and discusses basic concepts. It describes how to install the product and how to manage licenses. It also describes how to use common product features.
- *Preparing Projects* describes how to set up ATW projects. This guide describes how to load applications in the repository and how to use reports and other tools to ensure that the entire application is available for analysis.
- *Analyzing Programs* describes how to analyze applications at the program level. This guide describes how to use HyperView tools to view programs interactively and perform program analysis in stages. It also describes how to analyze procedure and data flows, search the repository, and extract business rules with HyperView.
- *Profiling Projects* describes how to create and browse Web-generated views of the repositories in your organization.

- *Creating Components* describes how to extract program components from a legacy application.
- *Transforming Applications* describes how to generate legacy application components in modern languages.
- *Parser Reference Manual* describes legacy constructions supported by Application Analyzer in reference format.
- *Architecture Reference Manual* describes legacy constructions supported by Application Architect in reference format.
- *Transformation Reference Manual* describes legacy constructions supported by Transformation Assistant in reference format.

Online Help

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

You can display:

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

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

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

Overview

1

When you verify a legacy application, the parser generates a model of the application that defines the objects it uses and how they interact. The tools described in this guide let you analyze the model at the project level. You can view high- and low-level process and data flows, estimate project effort and complexity, define an application namespace, and identify classes of data items that may be affected by a change in a field.

Analyzing Relationship Flows

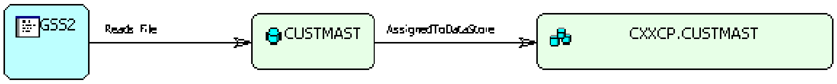
ATW offers four related tools for analyzing relationship flows in legacy applications. The core Diagrammer tool lets you analyze high-level relationship flows. The other three tools let you drill down deeper into data flows, batch processes, and PL/I programs. All the tools let you browse source code interactively.

Analyzing Relationships with the Diagrammer

The *Diagrammer* lets you view the relationships between application objects interactively — programs, files, DDL, JCL, screen maps, and more. These relationships describe the ways in which application objects interact.

In Figure 1-1, for example, the GSS2 program *reads* the data file CUSTMAST. The CUSTMAST file, in turn, is *assigned to* the CXXCP.CUSTMAST data store.

Figure 1-1 *Relationships in Diagrammatic Form*



The Diagrammer's high-level analysis of relationships lets you quickly trace the flow of information in an application. Use the Diagrammer to view:

- Program to program calls
- Program, transaction, and screen flows
- Program to table or data store flows
- Job, program, and data store flows

Analyzing Data Flows with the Global Data Flow Tool

The *Global Data Flow* tool lets you perform low-level analysis of program data flows. Use it to determine how changes to a variable may affect other variables and to trace assignments to and from a variable across programs.

Analyzing Batch Applications with the Batch Application Viewer

The *Batch Application Viewer* lets you perform low-level analysis of batch processes. Use it to determine whether jobs are dependent on one another, the programs that use a data store, and the flow of data into or out of a data store.

Analyzing PL/I Applications with the PL/I Call Diagrammer

The *PL/I Call Diagrammer* lets you perform low-level analysis of PL/I programs. You can use it to examine call flows for external procedures as you would in the Diagrammer. But you can also use it to examine call flows for internal procedures that the Diagrammer is unable to model.

Analyzing Data Operations

The workbench parser generates relationships that show the data operations a program performs — Insert, Read, Update, or Delete — and the data objects on which the program operates — files, segments, tables, and so forth.

The workbench *CRUD report* lets you view these relationships in a convenient format. The *IMS port analysis* feature determines the database segments or screens an IMS program operates on by tracing PSB usage through an entire application call sequence.

Estimating Complexity and Effort

Suppose you are planning to implement a change request for a program and want to know how long it will take to complete the change. ATW *legacy estimation* tools let you compare programs based on weighted values for selected complexity metrics. Based on the comparison, you can develop a credible estimate of the time required to make the change.

The complexity metrics used in the calculation are a combination of industry standard and ATW-generated statistics (Table 7-1 on page 7-8). If your own analysis shows that a given program is more or less complex than the weighted calculation would suggest, you can use a *change magnitude* to override the calculated value.

The program might have thousands of source lines, for example, increasing its calculated complexity, while actually being very easy to modify. When you use a change magnitude, your “subjective” estimate of the effort involved — Small, Medium, Large, Extra-large — becomes an input to the effort calculation, along with the weighted values.

Defining an Application Namespace

The *Application Namespace* tool creates a conveniently organized dictionary that helps you navigate through your system's terminology and modify it as necessary. You can also use the tool to provide pseudo-code versions of application names that are easier for business analysts to understand.

Namespace *mapping* lets you change identifiers in system source code without undermining the system's consistency. If your legacy application uses the identifiers CUSTMR and CSTMR to refer to a customer, you can use the tool to rename both identifiers to CUSTOMER, and propagate the change to all the source files that need it.

Identifying Classes of Data Items

Suppose your organization is considering adding support for a new currency and that you are going to have to expand the existing exchange rate data field from 9(5)V9(3) to 9(5)V9(6) to accommodate the currency. You will need to know the data fields that are affected in the database, intermediate fields that may contain or use the exchange rate field in calculations, and so forth.

The *Change Analyzer* identifies the *class* of data items that may need to be modified to support the new currency — not only the exchange rate fields themselves, but any fields related to them by assignment. In this way it lets you answer the kinds of "What if?" questions posed in the recent past by the industry-wide changes for Y2K, Zip+4, and the Euro dollar: "What if I change the type of this variable, or the length of this field — what *other* fields will I also have to change?"

Use change analysis results to prepare project plans and technical specifications. You can generate reports showing source entities that may require modification, lines of code affected, and the like.

What’s Next?

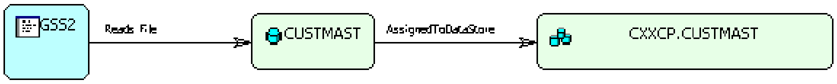
That’s all you need to know before you begin analyzing ATW projects. Now let’s look at how you use the Diagrammer to perform high-level analysis of legacy applications.

Analyzing Relationship Flows

2

The *Diagrammer* lets you view the relationships between application objects interactively — programs, files, DDL, JCL, screen maps, and more. These relationships describe the ways in which program objects interact. In Figure 2-1, for example, the GSS2 program *reads* the file CUSTMAST. The CUSTMAST file, in turn, is *assigned to* the CXXCP.CUSTMAST data store.

Figure 2-1 *Relationships in Diagrammatic Form*



The Diagrammer's high-level analysis of relationships lets you quickly trace the flow of information in an application. Use the Diagrammer to view:

- Program to program calls
- Program, transaction, and screen flows

- Program to table or data store flows
- Job, program, and data store flows

Understanding Relationship Flow Diagrams

Relationship flow diagrams use boxes and lines to show the relationships between objects in an application. You can display the relationships for all the objects in a workspace, or only the objects in the selected project.

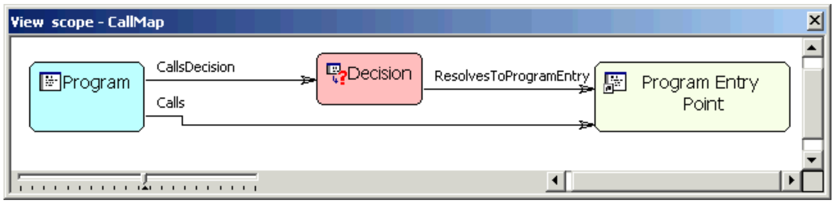
Objects are color-coded, based on a color scheme of your choosing. Hyperlinks let you navigate easily through a diagram to objects of interest, and browse source code interactively.

You can use the built-in editor to edit diagrams, cut-and-paste from a diagram, and so forth. You can export diagrams to a variety of standard formats and generate printed reports based on your diagrams.

Diagram Scopes

The *scope* of a diagram determines the objects and relationships it displays. Figure 2-2 shows the scope of the Call Map diagram.

Figure 2-2 *Scope of Call Map Diagram*



Tip: To view the definition of a scope, choose the scope in the **Scope** drop-down, then choose **View Scope** in the **Scope** menu.

The Diagrammer provides default scopes that you can use for most analysis tasks, but you can create your own scopes if you like. You might want to exclude data stores from a data flow, or include source files in a call map. For more information, see "Defining Scopes for Project Diagrams" on page 2-9.

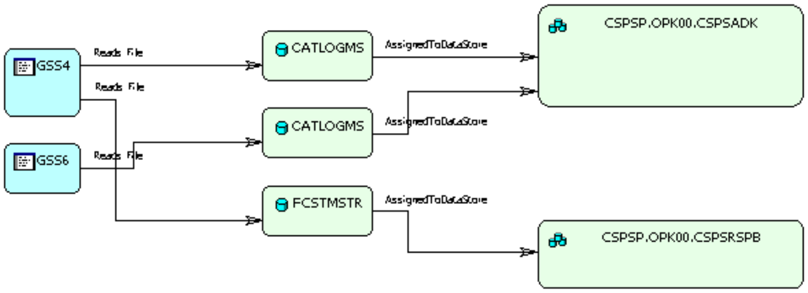
Diagram Views

The Diagrammer offers two views of relationship flows, a *compact view* and an *expanded view.* Figure 2-3 and Figure 2-4 show the same objects in both views. (Note that the diagram in Figure 2-4 has been modified to fit the page.)

Compact View

The compact view offers a cyclic representation of the information in a diagram. Objects are drawn once. Relationship lines cross. Compact views are often easier to understand than expanded views.

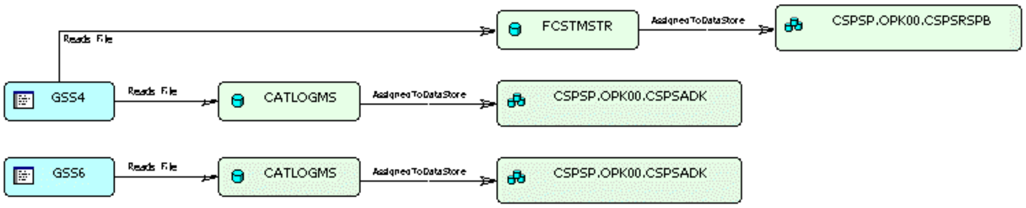
Figure 2-3 *Compact View*



Expanded View

The expanded view offers a linear representation of the information in a diagram. Objects are drawn as many times as necessary. Relationship lines do not cross. Use this view if too many intersecting lines make a compact view hard to read.

Figure 2-4 *Expanded View*



Using the Diagrammer

You generate relationship flow diagrams for the current project. You can include all the objects in the workspace by deselecting **Intraproject Relationships** in the **View** menu.

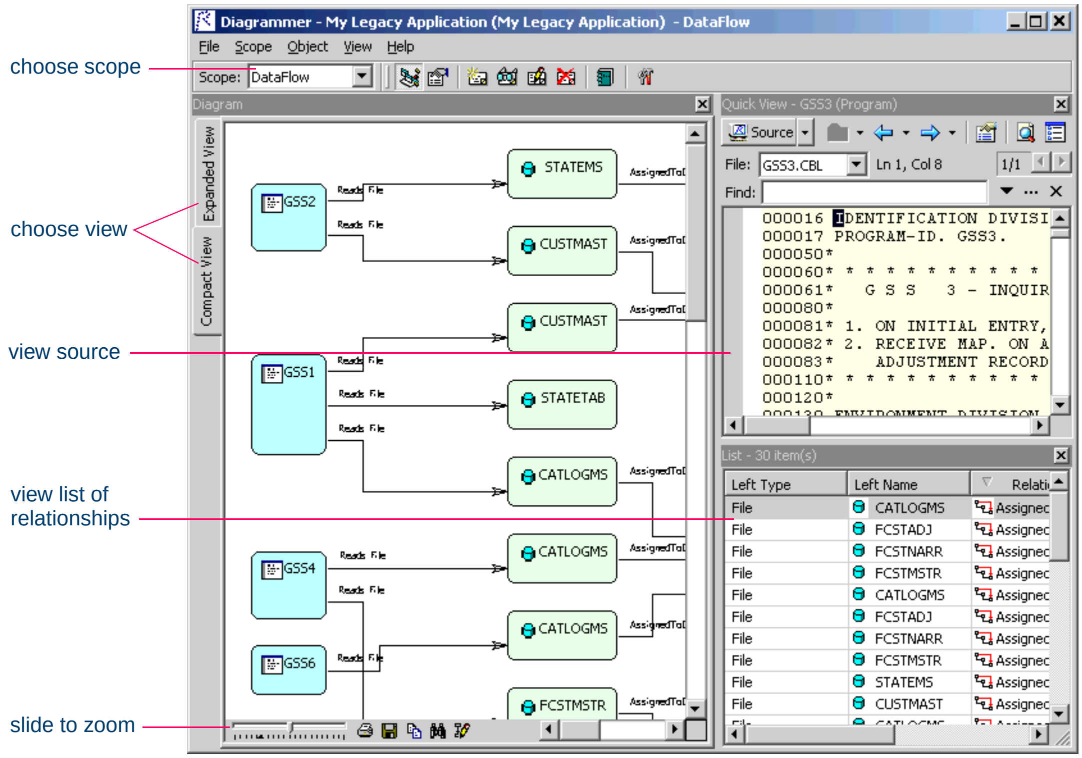
Generating Project Diagrams

Follow the steps below to generate a relationship flow diagram.

To generate a relationship flow diagram:

1 In the Repository Browser, select the project for the diagram and choose **Diagram** in the **Analyze** menu. A blank Diagrammer window opens. Choose the scope of the diagram in the **Scope** drop-down. Diagrammer draws the selected diagram (Figure 2-5).

Figure 2-5 *Diagrammer Window*



Scope: DataFlow

File: GSS3.CBL Ln 1, Col 8 1/1

```
000016 IDENTIFICATION DIVISI
000017 PROGRAM-ID. GSS3.
000050*
000060* * * * * * * * * *
000061*   G S S   3 - INQUIR
000080*
000081* 1. ON INITIAL ENTRY,
000082* 2. RECEIVE MAP. ON A
000083*    ADJUSTMENT RECORD
000110* * * * * * * * * *
000120*
```

Left Type	Left Name	Relati
File	CATLOGMS	Assignec
File	FCSTADJ	Assignec
File	FCSTNARR	Assignec
File	FCSTMSTR	Assignec
File	CATLOGMS	Assignec
File	FCSTADJ	Assignec
File	FCSTNARR	Assignec
File	FCSTMSTR	Assignec
File	STATEMS	Assignec
File	CUSTMAST	Assignec

2 In the **View** menu, deselect **Intraproject Relationships** if you want to include all related objects in the workspace in the diagram.

What If Your Project Is Too Large?

The Diagrammer issues a warning and fails to draw the diagram for the selected scope if the diagram requires more than 2500 lines or 32,000 boxes. (Note that the Diagrammer will still populate the List pane with the relationships in the selected scope.) Either reduce the size of the project or edit the scope to filter out unnecessary objects.

Working with Project Diagrams

The Diagrammer window consists of a Diagram pane, List pane, Quick View pane, and Activity Log. You can hide a pane by clicking the close box in the upper righthand corner. Select the appropriate choice in the **View** menu to show the pane again.

Diagram Pane

The Diagram pane displays the diagram for the selected scope. Use the slider on the tool bar in the lower lefthand corner of the pane to zoom in or out on the diagram. Place your cursor over an object for a moment to display a tool tip that describes the object in more detail.

Choosing the View Click the **Compact View** tab to choose the compact view of relationship flows. Click the **Expanded View** tab to choose the expanded view of relationship flows. For background on views, see "Diagram Views" on page 2-3.

Tip: Right-click an object in the expanded view and choose **Root** in the pop-up menu to navigate to the first instance of the object in the diagram. Select the scope again to redraw the diagram.

Viewing Source Select an object in the diagram to view its source in the Quick View pane. The object is displayed with a background shadow.

Navigating to an Object Select the relationship for an object in the List pane to navigate to the object in the Diagram pane.

Copying Diagrams In the **File** menu, choose **Copy *Type* View** to copy the diagram to the clipboard in EMF format. You can paste the diagram from the clipboard to a document in a third-party tool such as Word.

Printing Diagrams In the **File** menu, choose **Print *Type* View** to print the diagram. The Print Preview window opens, where you can change the zoom factor for the diagram to control the number of pages in the print job. Click **OK**.

List Pane

The List pane displays the relationships in the selected scope, as they appear from left to right in the diagram. Select the relationship for an object in the List pane to navigate to the object in the Diagram pane.

Sorting Entries Click a column heading in the List pane to sort the relationship entries by that column.

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

Tip: Even if the Diagrammer is unable to draw a diagram because the scope contains too many objects or relationships, it lists the relationships in the List pane. For more information, see "What If Your Project Is Too Large?" on page 2-5.

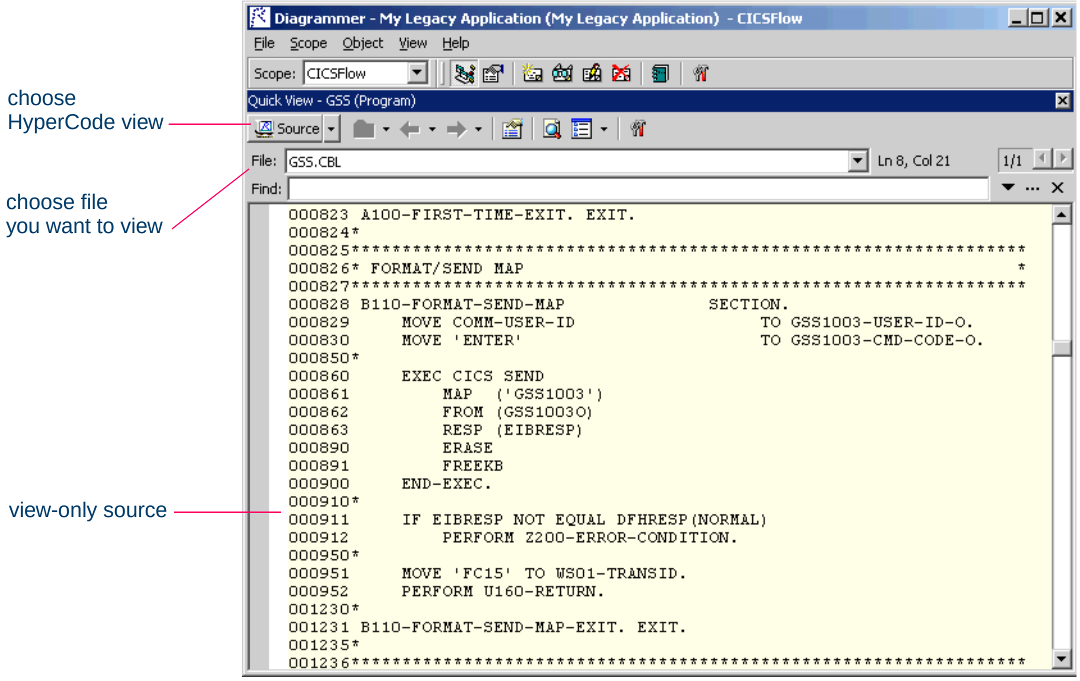
Quick View Pane

The Quick View pane lets you browse information about the object selected in the Diagram pane. The information available depends on the type of object selected — you see only source code for a copybook, for example, but full HyperCode for a program.

Note: "HyperCode" is shorthand for the information displayed in the ATW HyperView tool. For HyperView usage information, see *Analyzing Programs* in the ATW document set.

Select an object in the Diagram pane to view it in the Quick View pane. Choose the information you want to view for the object from the **Source** drop-down. Use the search facilities to navigate to the source you want to view.

Figure 2-6 *Quick View Pane*


Diagrammer - My Legacy Application (My Legacy Application) - CICSFlow
File Scope Object View Help
Scope: CICSFlow
Quick View - GSS (Program)
Source
File: GSS.CBL
Ln 8, Col 21
1/1
Find:

choose HyperCode view

choose file you want to view

view-only source

```
000823 A100-FIRST-TIME-EXIT. EXIT.
000824*
000825******************************************************************
000826* FORMAT/SEND MAP                                                *
000827******************************************************************
000828 B110-FORMAT-SEND-MAP              SECTION.
000829     MOVE COMM-USER-ID                  TO GSS1003-USER-ID-O.
000830     MOVE 'ENTER'                       TO GSS1003-CMD-CODE-O.
000850*
000860     EXEC CICS SEND
000861         MAP  ('GSS1003')
000862         FROM (GSS1003O)
000863         RESP (EIBRESP)
000890         ERASE
000891         FREEKB
000900     END-EXEC.
000910*
000911     IF EIBRESP NOT EQUAL DFHRESP(NORMAL)
000912         PERFORM Z200-ERROR-CONDITION.
000950*
000951     MOVE 'FC15' TO WS01-TRANSID.
000952     PERFORM U160-RETURN.
001230*
001231 B110-FORMAT-SEND-MAP-EXIT. EXIT.
001235*
001236******************************************************************
```

Note: The Quick View pane shows a view-only copy of application source.

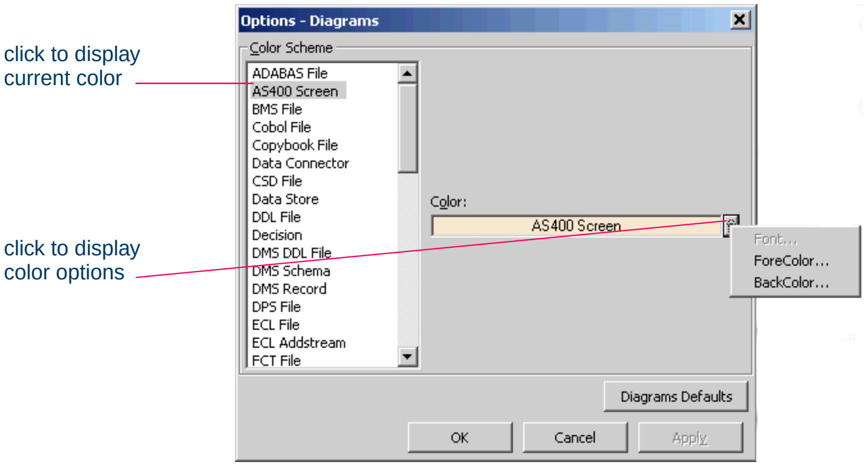
Editing the Color Scheme for Project Diagrams

Object types are color-coded in relationship flow diagrams so that you can distinguish one type from another — programs from transactions, transactions from screens, and so forth. You can use the default color scheme or create your own.

To edit the color scheme for diagrams:

1. In the **View** menu, choose **Options**. The Options window opens (Figure 2-7 on page 2-8).
2. In the Color Scheme pane, click the object type whose color you want to edit. The current background color of the type (if any) is displayed in the **Color** drop-down.

Figure 2-7 *Options Window*



3 Click the arrow beside the **Color** drop-down and choose **ForeColor** from the pop-up menu if you want to edit the color of the caption for the object type, or **BackColor** if you want to edit the color of the background for the object type.

4 A standard Windows color control is displayed. Use the **Palette** tab to select the color of the caption or background from the Windows palette. Use the **System** tab to match the color of the caption or background with the color of standard Windows elements.

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

5A Click **Apply** if you want to save your settings without dismissing the Options window.

5B Click **OK** if you want to save your settings and dismiss the Options window.

Defining Scopes for Project Diagrams

The *scope* of a diagram determines the objects and relationships it displays. Figure 2-2 on page 2-2 shows the scope of the Call Map diagram. To view the definition of a scope, choose the scope in the **Scope** drop-down, then choose **View Scope** in the **Scope** menu.

The Diagrammer provides default scopes that you can use for most analysis tasks, but you can create your own scopes if you like. You might want to exclude data stores from a Data Flow diagram, for example, or include Cobol source files in a Call Map diagram.

You can modify an existing scope, create a new scope based on an existing scope, or create a new scope from scratch. In each case, you specify the relationships you want the diagram to display

Note: Scopes are created or modified in the current workspace only.

Editing Scopes in the Quick Diagrammer

If your project is very large, you will probably want to edit scopes in the Quick Diagrammer, which does not automatically redraw a diagram each time you select a new scope. For more information, see "Using the Quick Diagrammer" on page 2-23.

Basic and Composite Relationships

Relationships are of two types, or *classes:*

- A *basic* relationship defines the direct interaction of two objects — a program and a program entry point, for example.
- A *composite* relationship defines the indirect interaction of two objects — if a job *runs* a program entry point that *belongs* to a program, for example, the relationship between the job and program is said to be composite. A composite relationship is indicated by a symbol.

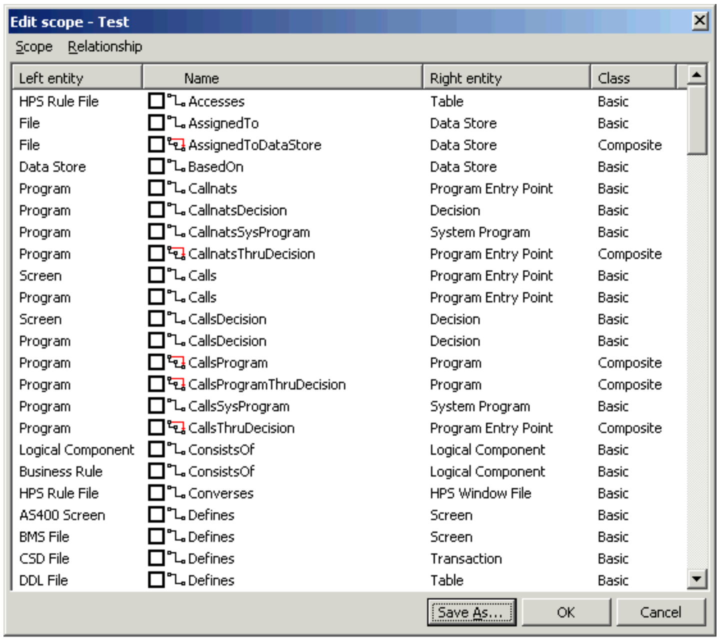
Tip: For convenience, you can create custom composite relationships that define an entire chain of relationships for an object.

To define the scope of a diagram:

1A To create a new scope from scratch, choose **New Scope** in the **Scope** menu. The New Scope dialog opens. Enter the name of the new scope in the text field and click **OK**. The Edit Scope window opens (Figure 2-8).

1B To modify an existing scope, or create a new scope based on an existing scope, choose the existing scope in the **Scope** drop-down, then choose **Edit Scope** in the **Scope** menu. The Edit Scope window opens (Figure 2-8).

Figure 2-8 *Edit Scope Window*

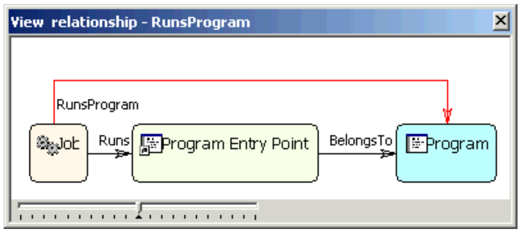


Left entity	Name	Right entity	Class
HPS Rule File	□ Accesses	Table	Basic
File	□ AssignedTo	Data Store	Basic
File	□ AssignedToDataStore	Data Store	Composite
Data Store	□ BasedOn	Data Store	Basic
Program	□ Callnats	Program Entry Point	Basic
Program	□ CallnatsDecision	Decision	Basic
Program	□ CallnatsSysProgram	System Program	Basic
Program	□ CallnatsThruDecision	Program Entry Point	Composite
Screen	□ Calls	Program Entry Point	Basic
Program	□ Calls	Program Entry Point	Basic
Screen	□ CallsDecision	Decision	Basic
Program	□ CallsDecision	Decision	Basic
Program	□ CallsProgram	Program	Composite
Program	□ CallsProgramThruDecision	Program	Composite
Program	□ CallsSysProgram	System Program	Basic
Program	□ CallsThruDecision	Program Entry Point	Composite
Logical Component	□ ConsistsOf	Logical Component	Basic
Business Rule	□ ConsistsOf	Logical Component	Basic
HPS Rule File	□ Converses	HPS Window File	Basic
AS400 Screen	□ Defines	Screen	Basic
BMS File	□ Defines	Screen	Basic
CSD File	□ Defines	Transaction	Basic
DDL File	□ Defines	Table	Basic

2 In the Edit Scope window, select each relationship you want to include in the scope. The Class column displays the class of the relationship, basic or composite.

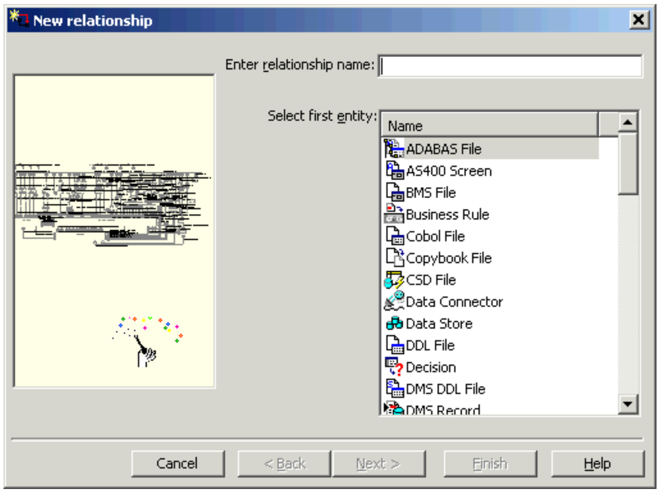
3 To view a composite relationship, select it and choose **View relationship** in the **Relationship** menu. The View Relationship window opens (Figure 2-9 on page 2-11).

Figure 2-9 *View Relationship Window*



4 To create a composite relationship, choose **New relationship** in the **Relationship** menu. The New Relationship Wizard opens (Figure 2-10).

Figure 2-10 *New Relationship Wizard (Screen One)*

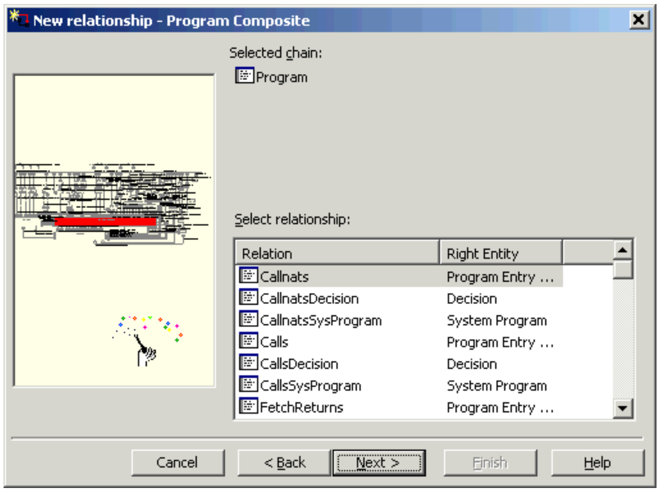


5 In the **Enter relationship name** field, enter the name of the composite relationship. In the Select first entity pane, select the object you want to appear on the left side of the relationship chain. Click **Next**. The screen shown in Figure 2-11 on page 2-12 appears.

6 In the Select relationship pane, select the relationship you want to appear next in the relationship chain. Click **Next**. A screen similar to the one shown in Figure 2-11 on page 2-12 appears.

7 Repeat step 6 on page 2-11 for each relationship you want to include in the chain. When you are done, click **Finish**. The New Relationship Wizard closes and the new composite relationship is added to the Edit Scope window.

Figure 2-11 *New Relationship Wizard (Screen Two)*



Selected chain:
Program

Select relationship:

Relation	Right Entity
CallNats	Program Entry ...
CallnatsDecision	Decision
CallnatsSysProgram	System Program
Calls	Program Entry ...
CallsDecision	Decision
CallsSysProgram	System Program
FetchReturns	Program Entry ...

8 To delete a composite relationship, select it and choose **Delete relationship** in the **Relationship** menu. You are prompted to confirm the deletion. Click **Yes**.

Note: You cannot create or delete a basic relationship.

8A Click **Save As** in the Edit Scope window to create a new scope based on an existing scope. The Save Scope window opens. Enter the name of the new scope in the text field and click **OK**. The new scope is displayed in the **Scope** drop-down in the Diagrammer window.

8B Click **OK** in the Edit Scope window to save a modified scope or a new scope created from scratch. The new scope is displayed in the **Scope** drop-down in the Diagrammer window.

Deleting a Scope

To delete a scope, select it in the **Scope** drop-down and choose **Delete Scope** in the **Scope** menu. You are prompted to confirm the deletion. Click **Yes**.

Searching for Objects in Project Diagrams

Use the Diagrammer’s Quick Search facility to navigate quickly to objects in a relationship flow diagram. To invoke Quick Search, click the 🔍 button on the tool bar in the lower lefthand corner of the Diagram pane. The Quick Search facility opens at the top of the Diagram pane. Enter the text for the search in the **Quick Search** field.

Tip: The 🔍 button is a toggle. Click the button again to hide the Quick Search facility.

Exporting Project Diagrams and Diagram-Based Reports

You can export diagrams to a variety of standard formats and generate printed reports based on your diagrams. Choose **Save *Type* View As** in the **File** menu to export a diagram to BED, bitmap, JPEG, Visio, Visio XML, DOT, or EMF.

Note: Visio 2002 must be installed to save a diagram to Visio. Visio 2002 is not required to save to Visio XML.

Choose **Save *Diagram* Report** in the **File** menu to export a printed report for a diagram to HTML, Excel, RTF, Word, or formatted text.

Note: For more information on exporting diagrams and reports, see *Getting Started* in the ATW document set.

You can also export a diagram to the ATW WebGen tool by clicking **Create WebGen Page** in the **File** menu. All diagram types are exported to the page. The page is stored in the file *Workspace*\Projects*Project*\Webgen\Diagrammer\Main.htm. Click the hyperlink for a diagram to view it in the browser.

Using the Diagram Editor

Use the built-in Diagram Editor to edit diagrams, change the appearance of diagrams, cut-and-paste from a diagram, and other tasks.

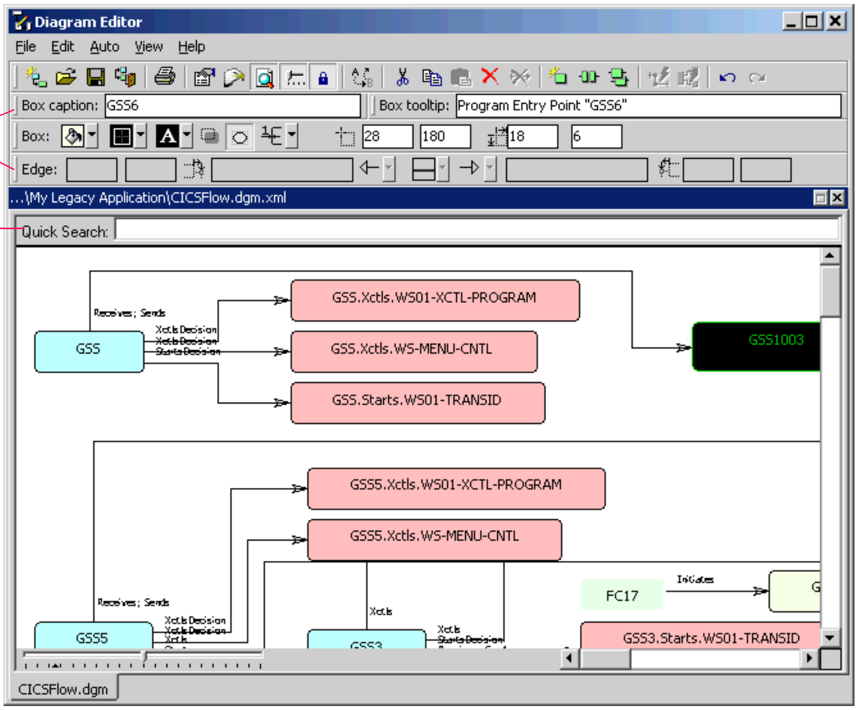
Opening a Diagram in the Diagram Editor

You can invoke the Diagram Editor from any tool that displays a diagram, or by double-clicking `BEDit.exe` in the ATW Bin directory. If you open the Diagram Editor from within another tool, the Editor creates a copy of the diagram on view in the tool.

To open a diagram in the Diagram Editor:

1A If the Diagram Editor is not open, click the button on the tool bar in the lower lefthand corner of the Diagram pane. The diagram on view in the Diagram pane opens in the Diagram Editor (Figure 2-5).

Figure 2-12 *Diagram Editor Window*



Box caption: GSS6
Box tooltip: Program Entry Point "GSS6"
28 180 18 6
...\My Legacy Application\CICSFlow.dgm.xml
Quick Search:
CICSFlow.dgm

edit diagram

quick search

1B If the Diagram Editor is already open, choose **Open Diagram** in the **File** menu. The Open diagram dialog appears. Select the diagram you want to open and click **Open**.

Tip: Choose **New Diagram** in the **File** menu to create a new blank diagram. You can paste portions of an existing diagram into the blank diagram, as described in "Copying Objects" on page 2-18.

Searching for Objects in the Diagram Editor

The Diagram Editor offers facilities for navigating to general locations in a diagram, or to specific objects. You can use the search and replace feature to modify captions or tool tips globally.

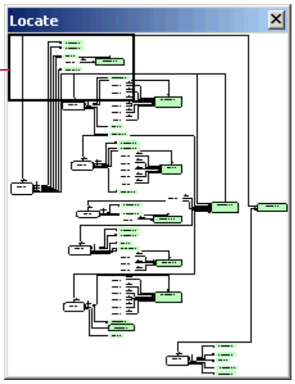
Using the Locate Facility

Use the Locate facility to navigate to general locations in a diagram. To invoke the Locate facility, click **Locate** in the **View** menu. The Locate window opens (Figure 2-13).

The Locate window displays the entire diagram on view in the Diagram Editor main window. A frame surrounds the portion of the diagram visible in the main window.

Drag the frame to the area of the diagram you want to view. The Diagram Editor displays the selected area in the main window.

Figure 2-13 *Locate Window*



Using the Quick Search Facility

Use the Quick Search facility to navigate to objects in a diagram. To invoke Quick Search, choose **Search pane** in the **View** menu. The Quick Search facility opens at the top of the window.

Enter the text for the search in the **Quick Search** field. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).

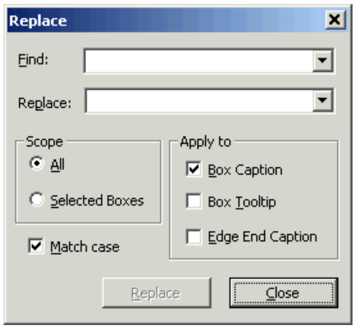
Using the Search and Replace Facility

Use the Search and Replace facility to modify captions or tool tips globally. The Diagram Editor modifies the diagram only, not the repository.

To search for and replace text in a diagram:

1 In the **Edit** menu, choose **Replace**. The Replace window opens (Figure 2-14).

Figure 2-14 *Replace Window*



Replace

Find:

Replace:

Scope
● All
○ Selected Boxes

☑ Match case

Apply to
☑ Box Caption
☐ Box Tooltip
☐ Edge End Caption

2 Enter the text you want to replace in the **Find** field. Enter the text you want to substitute in the **Replace** field.

3 Select **Match case** if you want to match the case of the entered text.

4 You can apply the change to any combination of object captions, relationship captions, or tool tips by selecting the appropriate choices in the Apply to pane. If you apply the change to object captions, select the appropriate choice in the Scope pane to apply the change to

all the objects that contain the text or only the selected objects that contain the text.

5 Click **Replace** to run the operation and dismiss the Replace window.

Selecting Objects in the Diagram Editor

Select an object or relationship in the Diagram Editor by clicking it. Deselect an object by clicking on white space or selecting another object. To select all the objects in the diagram, choose **Select All** in the **Edit** menu.

To select a range of objects, click in the diagram, hold down the right mouse button, drag the selection box over the objects you want to select, and release the mouse button. To select objects that are not in a range, hold down the Control key, then click each object you want to select.

Editing the Layout of a Diagram in the Diagram Editor

You can change the layout of a diagram, and copy, add or delete objects from the diagram. The Diagram Editor modifies the diagram only, not the repository.

Unlocking the Diagram Layout

When you open a diagram in the Diagram Editor, the diagram layout is *locked* — you cannot move or resize the objects in the diagram. To unlock the diagram layout, deselect **Lock Layout** in the **Edit** menu.

Autoplacing Objects

To rearrange the objects in a diagram automatically, choose **Auto Place** in the **Auto** menu. You are prompted to confirm that you want to autoplace the objects. Click **Yes**.

Splitting Objects

To “split” an object that has multiple relationships into an equal number of objects with a single relationship, select the object and choose **Split** in the **Edit** menu. Rearrange the objects manually or use the autoplace feature to rearrange the objects automatically.

Moving and Resizing Objects

To move an object in a diagram, drag-and-drop the object to the new location, or enter the grid coordinates of the location in the fields beside the symbol on the Box tool bar. The Diagram Editor redraws the relationships for the object.

To resize an object in a diagram, grab the border of the object with the mouse and drag it to a new location, or enter the grid coordinates for the height and width of the resized box in the fields beside the symbol on the Box tool bar.

Copying Objects

To create a copy of an object and all its relationships, select the object and choose **Clone** in the **Edit** menu.

To copy an object but not its relationships to the clipboard, select the object and choose **Copy** in the **Edit** menu. To paste the object from the clipboard, choose **Paste** in the **Edit** menu.

Tip: Choose **Clipboard** in the **View** menu to view the contents of the clipboard.

Adding Objects

To add an empty object to a diagram, click **New Box** in the **Edit** menu. To create a relationship between the new object and another object, hold down the Alt key, select either object, and drag-and-drop the relationship to the other object.

Deleting Objects

To delete an object from a diagram, select the object and click **Delete** in the **Edit** menu. To delete an object from the diagram and copy it to the clipboard, select the object and choose **Cut** from the **Edit** menu. The Diagram Editor also deletes the relationships for the object.

Autoattaching Relationships

To redraw a relationship line in a diagram automatically, select the relationship line and choose **Auto Attach** in the **Auto** menu.

Redrawing Relationships

To redraw a relationship in a diagram, grab the relationship line with the mouse and drag it to a new location, or enter the grid coordinates for the length and offset of each end of the redrawn line in the fields beside the and symbols on the Edge tool bar.

Deleting Relationships

To delete a relationship from a diagram, select the relationship and click **Delete Edge** in the **Edit** menu.

Editing Objects in the Diagram Editor

You can change the caption, color scheme, and border characteristics of an object. The Diagram Editor modifies the diagram only, not the repository.

Editing the Captions of an Object and Tool Tip

To edit the caption of an object, select the object and enter the new caption in the text field on the Box caption tool bar. To edit the caption of the tool tip for an object, select the object and enter the new caption in the text field on the Box tooltip tool bar.

Editing the Color Scheme for an Object

You can change the color of the background, border, and caption of an object:

- To change the background color of an object, select the object and click the arrow beside the symbol on the Box tool bar.
- To change the border color of an object, select the object and click the arrow beside the symbol on the Box tool bar.
- To change the color of the caption for an object, select the object and click the arrow beside the A symbol on the Box tool bar.

A standard Windows color control is displayed. Use the **Palette** tab to select the color of the background, border, or caption from the Windows palette. Use the **System** tab to match the color of the background, border, or caption with the color of standard Windows elements.

Editing the Border of an Object

You can change the thickness of the border for an object, specify that the border have rounded or square corners, and add a background shadow to the border:

- To change the thickness of the border for an object, select the object and click the arrow beside the drop-down on the Box tool bar. In the drop-down, choose the thickness you want, where 1 is the thinnest border and 4 is the thickest.
- To specify that the border of an object has rounded corners, select the object and click the button on the Box tool bar. The button is a toggle. Click the button again to specify that the border has square corners.
- To add a background shadow to the border for an object, select the object and click the button on the Box tool bar. The button is a toggle. Click the button again to specify that the border has no shadow.

Editing Relationships in the Diagram Editor

You can add a caption to, or delete a caption from, either end of a relationship, edit the captions, and edit the cardinality of the relationship. The Diagram Editor modifies the diagram only, not the repository.

Adding, Deleting, or Editing a Caption for a Relationship

There are two text fields for relationship captions on the Edge tool bar. The field on the left contains the caption for the left end of the relationship, the field on the right contains the caption for the right end of the relationship.

To add a caption to a relationship, select the relationship and enter the new caption in the appropriate text field. To delete a caption from a relationship, select the relationship and clear the caption in the appropriate text field. To edit the caption for a relationship, select the relationship and enter the new caption in the appropriate text field.

Editing the Cardinality of a Relationship

There are two drop-downs for relationship cardinalities on the Edge tool bar. The drop-down on the left contains the cardinality for the left end of the relationship, the drop-down on the right contains the cardinality for the right end of the relationship.

To edit the cardinality of a relationship, select the relationship and click the appropriate drop-down on the Edge tool bar. In the drop-down, choose the cardinality for the relationship.

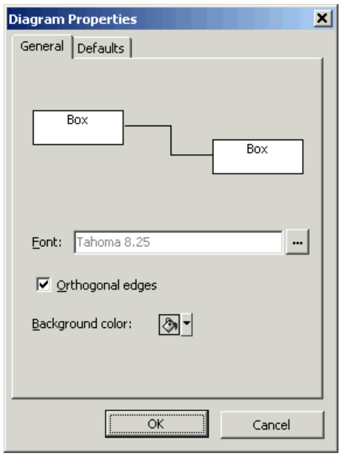
Editing General Properties of a Diagram in the Diagram Editor

Use the Diagram Properties window to specify general properties of a diagram in the Diagram Editor — the font for captions, whether relationships are displayed as orthogonal (right-angled) lines, and the background color of the Diagram Editor window.

To edit the general properties of a diagram:

1 In the **File** menu, choose **Properties**. The Diagram Properties window opens (Figure 2-15).

Figure 2-15 *Diagram Properties Window*



Font: Tahoma 8.25

☑ Orthogonal edges

Background color:

2 Click the button next to the **Font** drop-down. A standard Windows font control is displayed, where you can choose the font type, style, and size for diagram captions.

3 Select **Orthogonal Edges** if you want the Diagram Editor to display relationships as orthogonal (right-angled) lines.

4 Click the arrow beside the **Background color** drop-down to specify the background color for the Diagram Editor window. A standard Windows color control is displayed. Use the **Palette** tab to select the color of the text or background from the Windows palette. Use the **System** tab to match the color of the text or background with the color of standard Windows elements.

5 Click **OK** to apply your changes and dismiss the Diagram Properties window.

> ***Zooming in the Diagram Editor***
>
> Use the slider on the tool bar in the lower lefthand corner of the Diagram Editor window to zoom in or out on the diagram. To display the slider, choose **Zoom** in the **View** menu.

Saving Diagrams in the Diagram Editor

Choose **Save** in the **File** menu to save a diagram in the Diagram Editor. The Save Diagram dialog opens. In the **Save in** drop-down, choose the folder to save the new diagram in. In the **File name** field, enter the name of the new diagram. Choose a name that describes the diagram as closely as possible. Click **Save**.

Copying Diagrams with a Different Name (Saving As) in the Diagram Editor

Choose **Save as** in the **File** menu to copy a diagram with a different name in the Diagram Editor. The Save Diagram dialog opens. In the **Save in** drop-down, choose the folder to save the new diagram in. In the **File name** field, enter the name of the new diagram. Choose a name that describes the diagram as closely as possible. Click **Save**.

Printing Diagrams in the Diagram Editor

Choose **Print** in the **File** menu to print a diagram in the Diagram Editor. The Print Preview window opens, where you can change the zoom factor for the diagram to control the number of pages in the print job. Click **OK**.

Exporting Diagrams in the Diagram Editor

You can export diagrams to a variety of standard formats in the Diagram Editor. Choose **Export** in the **File** menu to export a diagram to BED, bitmap, JPEG, Visio, Visio XML, DOT, or EMF. For more information on exporting diagrams, see *Getting Started* in the ATW document set.

Note: Visio 2002 must be installed to save a diagram to Visio. Visio 2002 is not required to save to Visio XML.

Using the Quick Diagrammer

The Quick Diagrammer tool lets you view relationships for *selected objects only,* rather than an entire project. For that reason, it’s an ideal tool for initial program analysis — fast and focused.

For the most part, tool usage is similar to that for the main Diagrammer and Diagram Editor, but there are some handy additional features that you may want to explore.

Generating Object Diagrams

The Quick Diagrammer opens in the righthand section of the main ATW window. You can open as many Quick Diagrammer windows as you need.

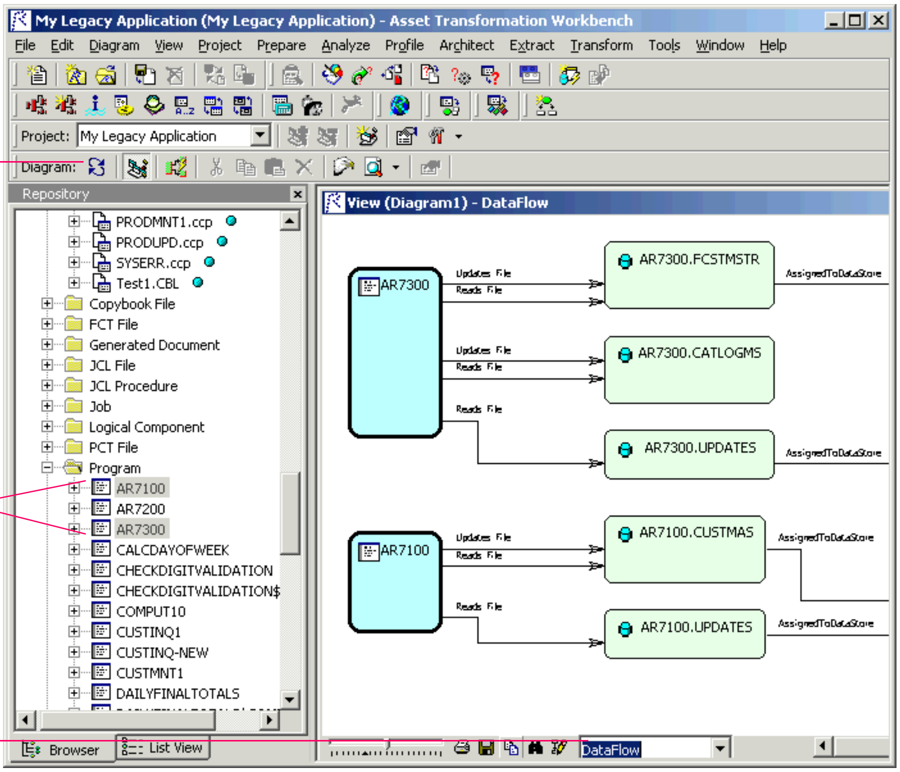
To open a diagram in the Quick Diagrammer:

1. In the ATW **Analyze** menu, choose **Quick Diagram**. The Quick Diagrammer opens in the righthand section of the main ATW window.
2. Choose the scope of the diagram in the **Scope** drop-down.

3 Drag-and-drop objects in the diagram's scope from the Repository Browser to the Quick Diagrammer window. Quick Diagrammer draws the selected diagrams (Figure 2-16).

Tip: Make sure to select objects in the diagram's scope — most scopes include extracted objects rather than source files.

Figure 2-16 *ATW Window with Quick Diagrammer*



rebuild diagram

drag-and-drop objects

choose scope

Changing the Scope of an Object Diagram

Unlike the main Diagrammer, Quick Diagrammer *does not automatically redraw a diagram after you select a new scope.* You must choose **Rebuild** in the **Diagram** menu to redraw the diagram. This is a considerable advantage, of course, if your diagram is very large — you don't have to wait for it to redraw each time you select a scope to edit.

Defining Scopes for Object Diagrams

You can create, edit, view, or delete a scope in the Quick Diagrammer, just as you can in the main Diagrammer. Any modifications in either tool are reflected in the other.

To define the scope of a diagram:

1A To create a new scope from scratch, choose **Scopes:New** in the **Diagram** menu. The New Scope dialog opens. Enter the name of the new scope in the text field and click **OK**. The Edit Scope window opens (Figure 2-8 on page 2-10).

1B To modify an existing scope, choose the scope you want to modify in the **Scope** drop-down, then choose **Scopes:Edit** in the **Diagram** menu. The Edit Scope window opens (Figure 2-8 on page 2-10).

Tip: You do not have to draw a diagram in the Quick Diagrammer to modify a scope.

2 Follow the instructions in steps 2-8B on pages 2-10-2-12 to define the scope.

To view the definition of a scope:

1 To view the definition of a scope, choose the scope in the **Scope** drop-down, then choose **Scopes:View** in the **Diagram** menu.

To delete a scope:

1 To delete a scope, select it in the **Scope** drop-down and choose **Scopes:Delete** in the **Diagram** menu. You are prompted to confirm the deletion. Click **Yes**.

Mirroring the Scope of an Object Diagram

By default, Quick Diagrammer shows the flow of relationships *from* a diagrammed object rather than *to* the object. Choose **Mirror Scope** in the **Diagram** menu to show the flow of relationships to the object. Choose **Mirror Scope** again to return to the original view.

Exporting Multiple Object Diagrams

Use Quick Diagrammer to export one or more diagrams to the supported formats in a single step.

To export object diagrams:

1 In the **Scope** drop-down, choose the scope of the diagrams you want to export.

Tip: Although the Quick Diagrammer window must be open to export diagrams, you do not actually have to draw a diagram to export it. It's enough to choose the diagram scope. Of course, you *can* draw the diagrams if you like.

2 In the Repository Browser, select the objects whose diagrams you want to export, then click in the Quick Diagrammer window and choose **Save Multiple** in the **Diagram** menu. A standard Save As window opens, where you can export the diagrams to BED, bitmap, JPEG, Visio, Visio XML, DOT, or EMF.

Note: You cannot specify the names of the exported diagrams. Diagrams are named after the diagrammed object.

Visio 2002 must be installed to save a diagram to Visio. Visio 2002 is not required to save to Visio XML.

Moving or Copying Source Files for Diagrammed Objects into Projects

Use Quick Diagrammer to move or copy the source files for the objects in a diagram to different projects.

To move or copy source files for diagrammed objects into projects:

1 In the **Diagram** menu, choose **Include into Project**. The Select Project window opens (Figure 2-17 on page 2-27).

2 In the Select Project window, select the project you want to move or copy source files into. Click **New** to create a new project.

3 Select the **Include All Related Objects** check box if you want to move or copy the objects in the workspace related to the diagram

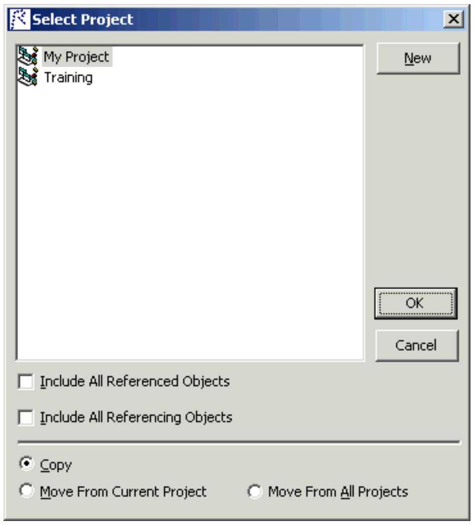
objects — the Cobol copybooks included in a Cobol program file, for example.

4 Choose one of the following:

- **Copy** to copy the source files to the specified project.
- **Move From Current Project** to move the source files to the specified project.
- **Move From All Projects** to move the source files from all projects to the selected project.

5 Click **OK** to move or copy the selection.

Figure 2-17 *Select Project Window*



My Project
Training

□ Include All Referenced Objects

□ Include All Referencing Objects

● Copy
○ Move From Current Project ○ Move From All Projects

What’s Next?

Now that you have performed high-level analysis of relationship flows in the Diagrammer and Quick Diagrammer, you are ready to drill down deeper into your application. The next chapter shows you how to analyze program data flows.

Analyzing Global Data Flows

3

The *Global Data Flow* tool lets you analyze incoming and outgoing data flows in a program. You can view the memory allocation and offset for a variable to determine how changes to the variable may affect other variables, and trace assignments to and from the variable across programs.

Note: Projects must have been verified with the **Enable Data Element Flow** option set in the **Verification** tab of the Project Options window.

How the Global Data Flow Tool Is Implemented

The Global Data Flow tool is implemented as both a standalone and HyperView-based tool. This chapter describes the HyperView version. Usage is identical for the standalone tool. For Context and Clipper pane usage, and for HyperView usage in general, see *Analyzing Programs* in the ATW document set.

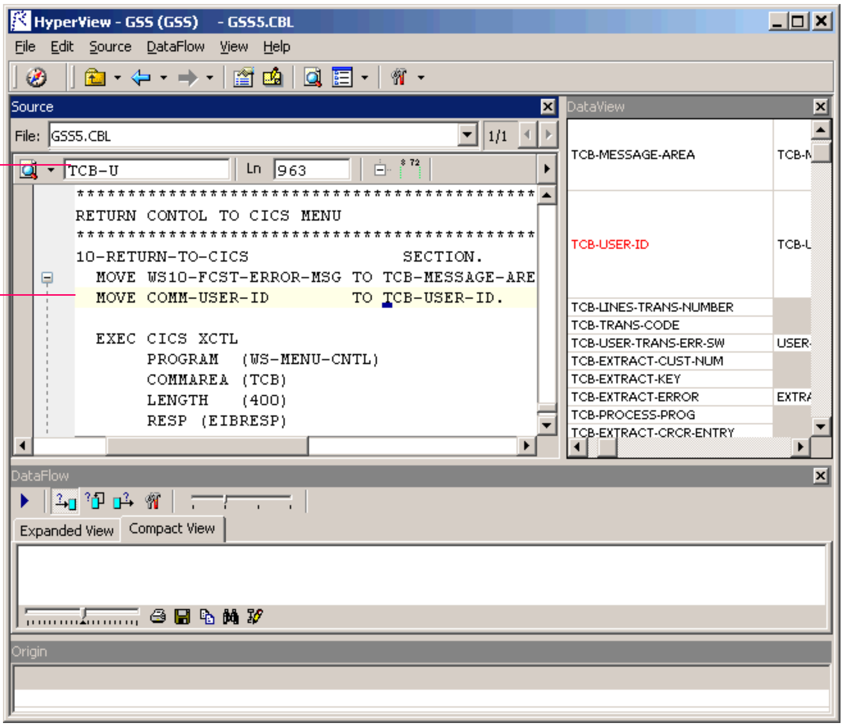
Understanding Global Data Flows

Suppose your task is to learn how data flows to and from the variable TCB-USER-ID in the GSS5 program in the GSS project. The Global Data Flow tool offers four views of a project that together let you perform low-level analysis of application data flows. Let's look at how you use these views to analyze the TCB-USER-ID variable.

Searching for the Variable

When you open the Global Data Flow tool in HyperView, the Source pane displays the source code for the selected program. In the text field next to the 🔍 button on the tool bar, enter TCB-USER-ID to navigate to the variable in the source code (Figure 3-1). We can see that a MOVE statement assigns the value of the COMM-USER-ID variable to TCB-USER-ID, and that the Data View pane highlights the variable in red.

Figure 3-1 *Global Data Flow Tool — Searching for a Variable*



```
*********************************************
RETURN CONTOL TO CICS MENU
*********************************************
10-RETURN-TO-CICS                SECTION.
  MOVE WS10-FCST-ERROR-MSG TO TCB-MESSAGE-ARE
  MOVE COMM-USER-ID         TO TCB-USER-ID.

  EXEC CICS XCTL
       PROGRAM  (WS-MENU-CNTL)
       COMMAREA (TCB)
       LENGTH   (400)
       RESP (EIBRESP)
```

TCB-MESSAGE-AREA	TCB-M
TCB-USER-ID	TCB-U
TCB-LINES-TRANS-NUMBER	
TCB-TRANS-CODE	
TCB-USER-TRANS-ERR-SW	USER-
TCB-EXTRACT-CUST-NUM	
TCB-EXTRACT-KEY	
TCB-EXTRACT-ERROR	EXTRA
TCB-PROCESS-PROG	
TCB-EXTRACT-CRCR-ENTRY	

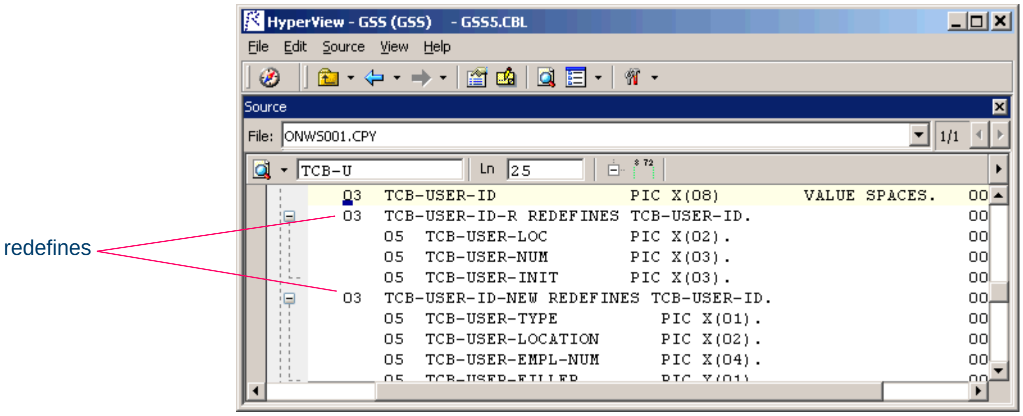
enter search text for variable

MOVE statement

Viewing Memory Usage for the Variable

Although we have located a reference to TCB-USER-ID in the GSS5 source code, we still don't know where the variable is declared. Click TCB-USER-ID in the Data View pane to display the copybook source for the declaration (Figure 3-2).

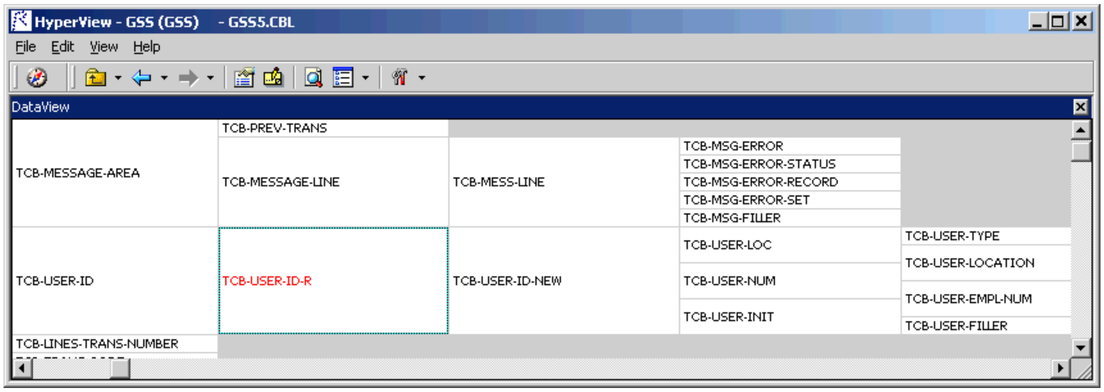
Figure 3-2 *Global Data Flow Tool — Viewing Copybook Source*



```
03  TCB-USER-ID                PIC X(08)         VALUE SPACES.   00
03  TCB-USER-ID-R REDEFINES TCB-USER-ID.                        00
    05  TCB-USER-LOC           PIC X(02).                       00
    05  TCB-USER-NUM           PIC X(03).                       00
    05  TCB-USER-INIT          PIC X(03).                       00
03  TCB-USER-ID-NEW REDEFINES TCB-USER-ID.                      00
    05  TCB-USER-TYPE              PIC X(01).                   00
    05  TCB-USER-LOCATION          PIC X(02).                   00
    05  TCB-USER-EMPL-NUM          PIC X(04).                   00
    05  TCB-USER-FILLER            PIC X(01)                    00
```

We can see from the source that TCB-USER-ID is a structure that is redefined by the structures TCB-USER-ID-R and TCB-USER-ID-NEW. Let's take a closer look at these structures in the Data View pane (Figure 3-3).

Figure 3-3 *Global Data Flow Tool — Viewing Memory Usage*



TCB-PREV-TRANS

TCB-MESSAGE-AREA | TCB-MESSAGE-LINE | TCB-MESS-LINE | TCB-MSG-ERROR, TCB-MSG-ERROR-STATUS, TCB-MSG-ERROR-RECORD, TCB-MSG-ERROR-SET, TCB-MSG-FILLER

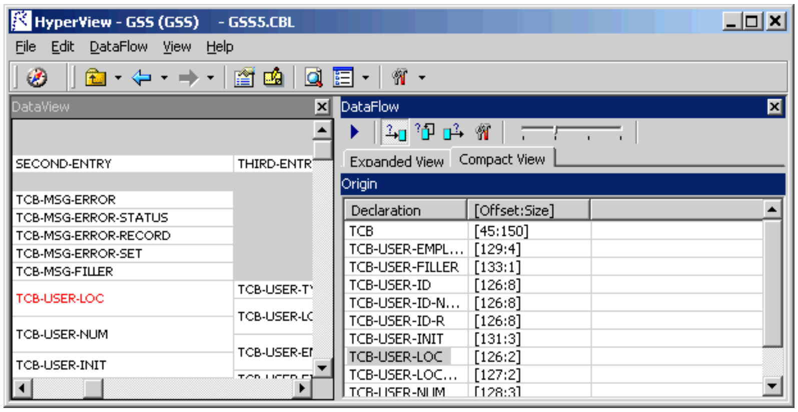
TCB-USER-ID | TCB-USER-ID-R | TCB-USER-ID-NEW | TCB-USER-LOC, TCB-USER-NUM, TCB-USER-INIT | TCB-USER-TYPE, TCB-USER-LOCATION, TCB-USER-EMPL-NUM, TCB-USER-FILLER

TCB-LINES-TRANS-NUMBER

The Data View pane provides a visual representation of memory usage by the redefining structures. We can see which fields in TCB-USER-ID-NEW (in the fifth column) share memory locations with fields in TCB-USER-ID-R (in the fourth column).

Knowing this, we can be alert to the possibility that changing one of these variables might affect another variable at the same location. If we double-click TCB-USER-ID in the Data View pane, we can pinpoint the memory locations in the Origin pane, which shows the offset and size of each variable in both structures (Figure 3-4).

Figure 3-4 *Global Data Flow Tool — Viewing Memory Locations*



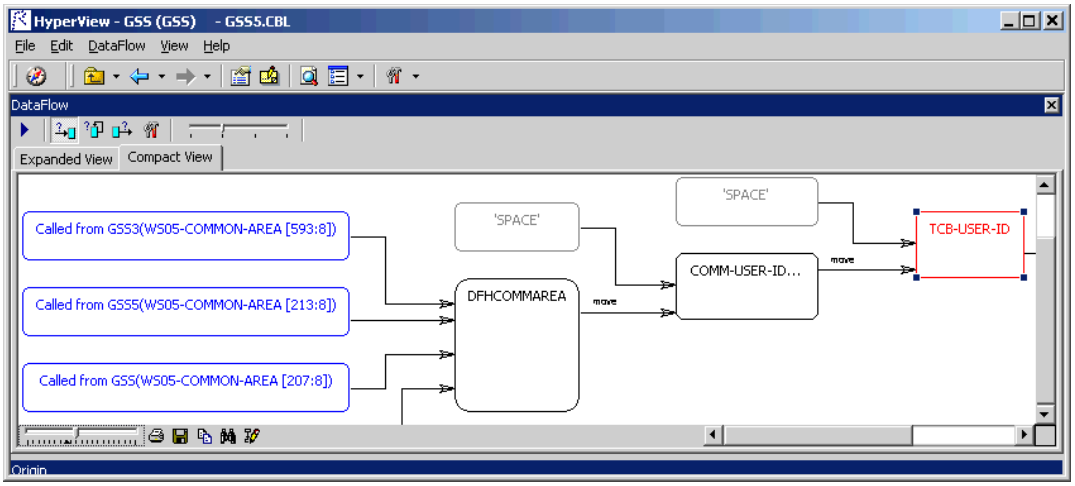
Declaration	[Offset:Size]
TCB	[45:150]
TCB-USER-EMPL…	[129:4]
TCB-USER-FILLER	[133:1]
TCB-USER-ID	[126:8]
TCB-USER-ID-N…	[126:8]
TCB-USER-ID-R	[126:8]
TCB-USER-INIT	[131:3]
TCB-USER-LOC	[126:2]
TCB-USER-LOC…	[127:2]
TCB-USER-NUM	[128:3]

Viewing the Data Flow for the Variable

Double-click TCB-USER-ID in the Data View pane to display a diagram that shows how data flows to and from the variable in the GSS5 program (Figure 3-5 on page 3-5). The diagram traces the incoming and outgoing data flows in the program up to a *dataport* — an I/O statement or a call to or from another program.

At the dataport for the call from GSS3, for example, we can trace the COMM-USER-ID field to an 8-byte variable at memory offset 593 in the WS-05-COMMON-AREA structure. When we select the dataport and choose **Reconstruct** in the **Diagram** menu, the Global Data Flow tool displays the data flow diagram for COMM-USER-ID in the GSS3 program.

Figure 3-5 *Global Data Flow Tool — Viewing Program Data Flow*


HyperView - GSS (GSS) - GSS5.CBL
File Edit DataFlow View Help
DataFlow
Expanded View
Compact View
Called from GSS3(WS05-COMMON-AREA [593:8])
Called from GSS5(WS05-COMMON-AREA [213:8])
Called from GSS(WS05-COMMON-AREA [207:8])
'SPACE'
DFHCOMMAREA
move
'SPACE'
COMM-USER-ID...
move
TCB-USER-ID
Origin

Using the Global Data Flow Tool

You can start the Global Data Flow tool as a standalone or HyperView-based tool. Start the standalone tool by selecting a project in the Repository Browser and choosing **Data Flow** in the **Analyze** menu. Start the HyperView-based tool as described below.

Generating Global Data Flow Information

Follow the steps below to start the Global Data Flow tool in HyperView.

To generate global data flow information in HyperView:

1 In the Repository Browser, select the program you want to analyze and choose **Interactive Analysis** in the **Analyze** menu. The HyperView window opens. In the **View** menu, choose **Data Flow**. The Global Data Flow tool opens (Figure 3-6 on page 3-6).

2 In the Source pane, use the Quick Search field next to the 🔍 button on the tool bar to navigate to the variable you want to analyze. The Global Data Flow tool highlights the variable name in the Source pane and displays the name in red in the Data View pane. Click the

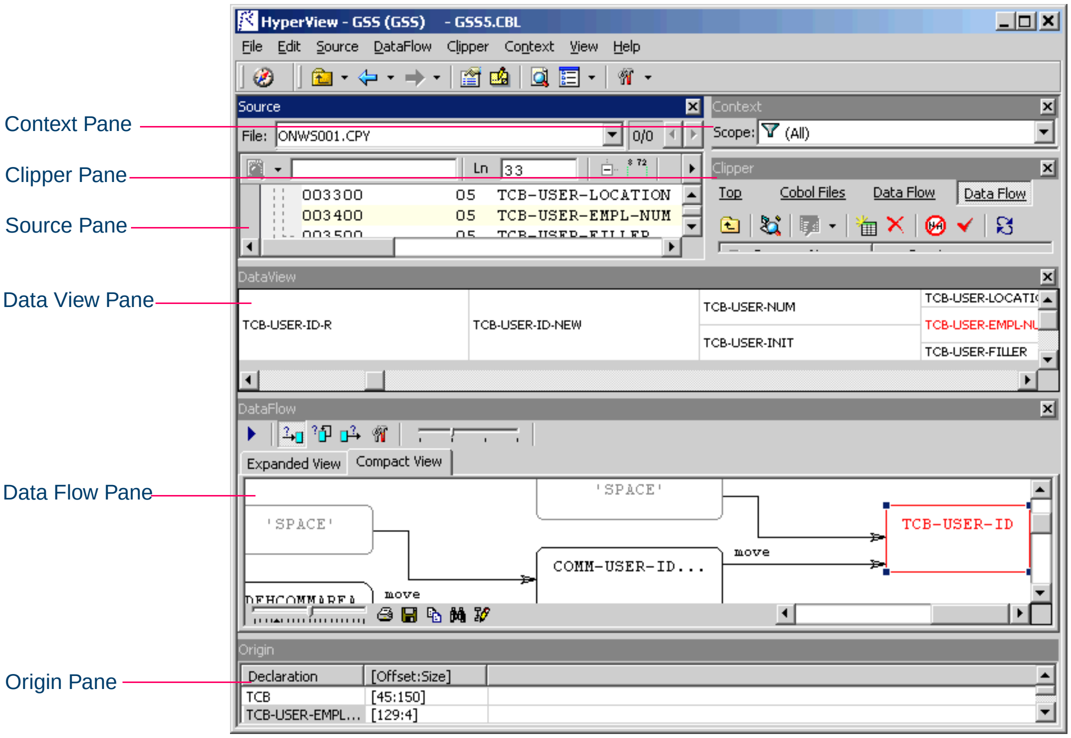
variable name in the Data View pane to display the source code for the variable declaration in the Source pane.

3 The Data View pane shows variable structures, substructures, and fields in hierarchical order from left to right. Double-click a variable name to generate a data flow diagram for the variable in the Data Flow pane, and a list of offsets and memory allocations for the variable and any related variables in the Origin pane.

4 Select a dataport in the Data Flow pane and choose **Reconstruct** in the **Data Flow** menu to display the data flow diagram for the variable identified at the dataport. For more information on dataports, see “Viewing the Data Flow for the Variable” on page 3-4.

Note: For Context and Clipper pane usage, see *Analyzing Programs* in the ATW document set.

Figure 3-6 *Global Data Flow Tool Window*



File: ONWS001.CPY 0/0

Scope: (All)

Ln 33

```
003300      05  TCB-USER-LOCATION
003400      05  TCB-USER-EMPL-NUM
003500      05  TCB-USER-FILLER
```

Declaration	[Offset:Size]
TCB	[45:150]
TCB-USER-EMPL...	[129:4]

Working with Global Data Flow Information

The window for the Global Data Flow tool consists of a Source pane, Data View pane, Data Flow pane (which includes the Origin pane), Context pane, Clipper pane, and Activity Log. You can hide a pane by clicking the close box in the upper righthand corner. Select the appropriate choice in the **View** menu to show the pane again.

Note: For Context and Clipper pane usage, see *Analyzing Programs* in the ATW document set.

Source Pane

The Source pane lets you browse the source code for the program selected in the Global Diagram pane or the variable selected in the Data View or Data Flow panes. For usage information, see *Analyzing Programs* in the ATW document set.

Data View Pane

For the program selected in the Source pane, the Data View pane shows variable structures, substructures, and fields in hierarchical order from left to right. The selected variable is displayed in red.

Double-click a variable name to generate a data flow diagram for the variable in the Data Flow pane, and a list of offsets and memory allocations for the variable and any related variables in the Origin pane.

Origin Pane

For the variable selected in the Data View or Data Flow panes, the Origin pane displays a list of offsets and memory allocations for the variable and any related variables. For the relationship selected in the Data Flow pane, it lists all the statements that determine the data flow between the related variables. For the dataport selected in the Data Flow pane, it displays related variables.

Data Flow Pane

The Data Flow pane displays a data flow diagram for the variable selected in the Data View pane. Usage is similar to that for the Diagrammer tool described in Chapter 2.

The selected variable is displayed in red, constants in gray, and dataports in blue. Place your cursor over a variable for a moment to display a tool tip that identifies the memory offset and allocation for the variable.

Select a dataport in the Data Flow pane and choose **Reconstruct** in the **Data Flow** menu to display the data flow diagram for the variable identified at the dataport. For more information on dataports, see “Viewing the Data Flow for the Variable” on page 3-4.

Tip: If the Data View pane is closed, you can generate the data flow diagram by selecting the variable name in the Source pane and choosing **Reconstruct** in the **Data Flow** menu.

Select a variable in the Data Flow pane to view its memory usage in the Data View and Origin panes and its source code in the Source pane. Select a relationship line in the Diagram pane to display all the statements that determine the data flow between variables in the Origin pane.

Note: For descriptions of the relationships between data items, see Table 6-1 in *Analyzing Programs* in the ATW document set.

In the **Diagram** menu, choose any combination of

- **Show Causes** to show data flows into the selected variable.
- **Show Consequences** to show data flows from the selected variable.
- **Show Self-Dependencies** to show recursive data flows for the selected variable.

Setting Global Data Flow Options

Use the Global Data Flow User Preferences window to edit the display in the Data View pane. Use the Global Data Flow Project Options window to edit the display in the Data Flow pane. Global Data Flow User Preferences apply across workspaces. Global Data Flow Project Options apply only in the selected project.

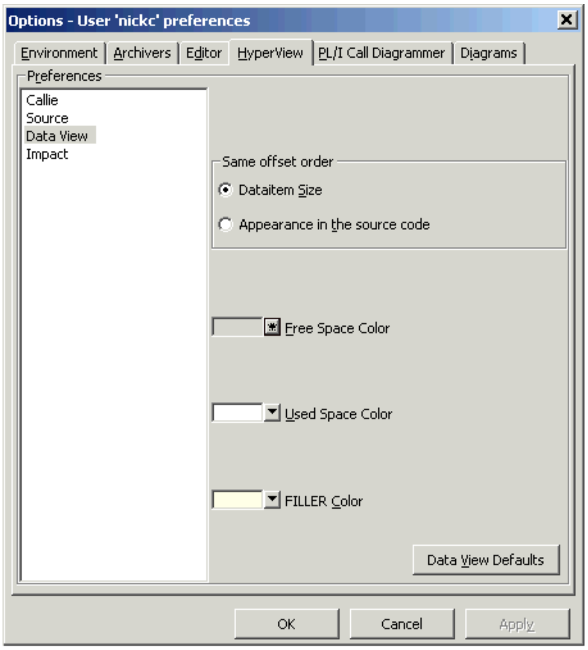
Setting Global Data Flow User Preferences

Global Data Flow User Preferences determine the order of display of variables with the same offset and the colors used in the Data View pane.

To set Global Data Flow User Preferences:

1 In the **View** menu, choose **User Preferences**. The User Preferences window opens. Click the **HyperView** tab. In the Preferences pane, click **Data View**. The Global Data Flow User Preferences window opens (Figure 3-8).

Figure 3-7 *Global Data Flow User Preferences Window*



Options - User 'nickc' preferences

Preferences: Callie, Source, Data View, Impact

Same offset order
● Dataitem Size
○ Appearance in the source code

Free Space Color
Used Space Color
FILLER Color

2 In the Same offset order group box, specify the order you want variables with the same offset to be displayed in the Data View pane. Choose:

- **Data item size** if you want variables with the same offset to be displayed in size order, largest to smallest.
- **Appearance in the source code** if you want variables with the same offset to be displayed in the order they appear in the source code.

3 The current default background color of the box representing free space in memory is displayed in the **Free Space Color** drop-down. The current background color of the box representing used space in memory is displayed in the **Used Space Color** drop-down. The current background color of the box representing filler is displayed in the **FILLER Color** drop-down.

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

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

4A Click **Apply** if you want to save your settings without dismissing the User Preferences window.

4B Click **OK** if you want to save your settings and dismiss the User Preferences window.

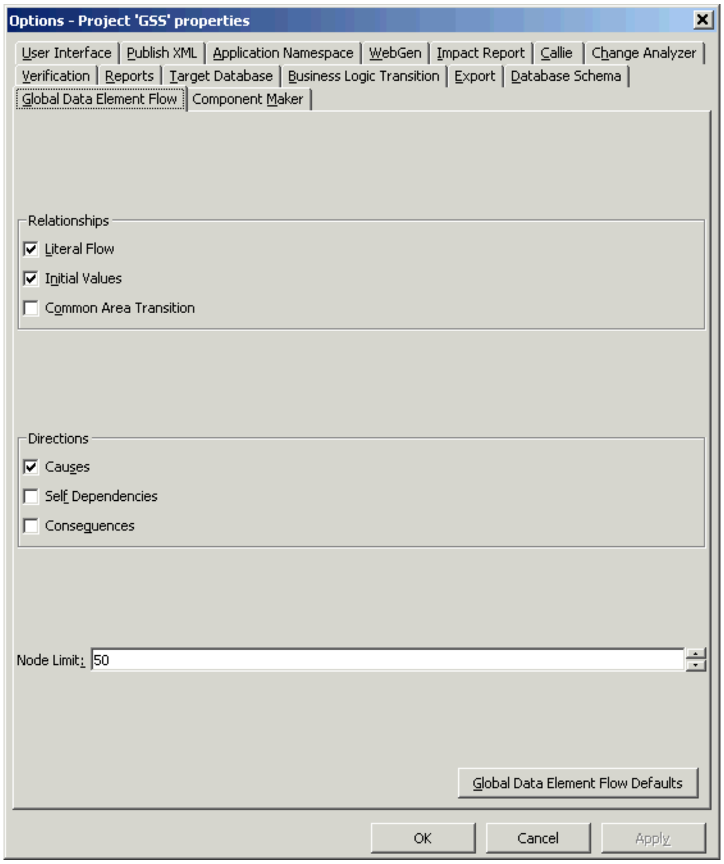
Setting Global Data Flow Project Options

Global Data Flow Project Options determine whether data flow diagrams include literals, variable values set at initialization, and Unisys Cobol common storage variables; whether they show incoming, outgoing, and recursive data flows; and the number of nodes they display.

To set Global Data Flow Project Options:

1 In the **View** menu, choose **Project Options**. The Project Options window opens. Click the **Global Data Flow** tab. The Global Data Flow Options window opens (Figure 3-8).

Figure 3-8 *Global Data Flow Options Window*



Relationships
☑ Literal Flow
☑ Initial Values
☐ Common Area Transition

Directions
☑ Causes
☐ Self Dependencies
☐ Consequences

Node Limit: 50

2 In the Relationships pane, choose any combination of:

- **Literal Flow** to include literals in the data flow diagram for the selected variable.
- **Initial Values** to include variable values set at initialization in the data flow diagram for the selected variable.

- **Common Area Transition** to include Unisys Cobol common storage variables in the data flow diagram for the selected variable. Common storage variables are not explicitly declared in CALL statements.

Note: To include Unisys Cobol common storage variables, you must have verified the project with the **Perform Unisys Common-Storage Analysis** option set in the **Verification** tab of the Project Options window. For more information, see *Preparing Projects* in the ATW document set.

3 In the Directions pane, choose any combination of:

- **Causes** to generate data flows into the selected variable.
- **Consequences** to generate data flows from the selected variable.
- **Self-Dependencies** to show recursive data flows for the selected variable.

4 In the **Node Limit** combo box, enter the maximum number of diagram nodes you want to display. You might restrict the number of nodes to improve performance or make the diagram easier to read. You can also use the slider on the Node Limit tool bar to adjust the number of displayed nodes.

Note: All children of a parent node are displayed even if some of them exceed the specified maximum.

5A Click **Apply** if you want to save your settings without dismissing the Options window.

5B Click **OK** if you want to save your settings and dismiss the Options window.

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

Copying Global Data Flow Diagrams to the Clipboard

Choose **Copy:*Diagram*** in the **Data Flow** menu to copy a global data flow diagram to the clipboard in EMF format. You can paste the diagram from the clipboard to a document in a third-party tool such as Word.

Printing Global Data Flow Diagrams

Choose **Print:*Diagram*** in the **Data Flow** menu to print a global data flow diagram. The Print Preview window opens, where you can change the zoom factor for the diagram to control the number of pages in the print job. Click **OK**.

Exporting Global Data Flow Diagrams

You can export global data flow diagrams to a variety of standard formats. Choose **Save *Diagram*** in the **Data Flow** menu to export a diagram to BED, bitmap, JPEG, Visio, Visio XML, DOT, or EMF. For more information on exporting diagrams, see *Getting Started* in the ATW document set.

Note: Visio 2002 must be installed to save a diagram to Visio. Visio 2002 is not required to save to Visio XML.

What's Next?

That's all you need to know to analyze program data flows in the Global Data Flow tool. Now let's look at how you perform low-level analysis of batch programs in the Batch Application Viewer. That's the subject of the next chapter.

Analyzing Batch Applications

4

The *Batch Application Viewer* lets you perform low-level analysis of batch processes. Use the Batch Application Viewer's rich set of tables and diagrams to determine whether batch jobs are dependent on one another, the programs that use a data store, and the flow of data into or out of a data store.

Note: Batch Application Viewer analyzes JCL and ECL jobs. Usage is identical except for language-specific differences in terminology.

Using the Batch Application Viewer

You generate batch application information for the current project. The information is presented in linked table and diagram views. The tables cross-reference jobs, steps, procedures and programs, and data stores. The diagrams show the relationships between jobs, procedures and programs, and data stores.

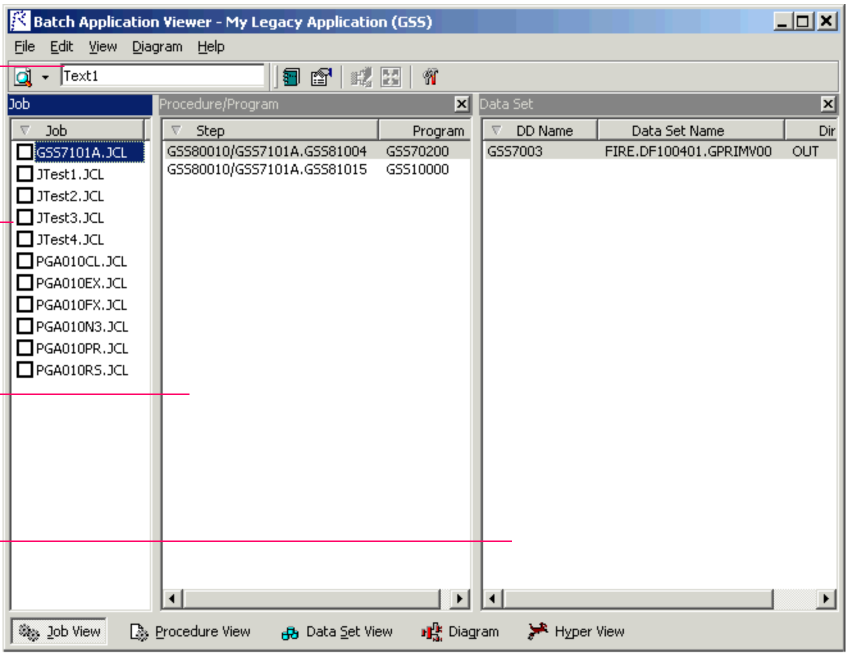
Generating Batch Application Information

Follow the steps below to generate batch application information.

To generate batch application information:

1 In the Repository Browser, select the project you want to analyze and choose **Batch Application** in the **Analyze** menu. The Batch Application Viewer window opens (Figure 4-1).

Figure 4-1 *Batch Application Viewer Window — Job View*



Job
□ GSS7101A.JCL
□ JTest1.JCL
□ JTest2.JCL
□ JTest3.JCL
□ JTest4.JCL
□ PGA010CL.JCL
□ PGA010EX.JCL
□ PGA010FX.JCL
□ PGA010N3.JCL
□ PGA010PR.JCL
□ PGA010RS.JCL

Step	Program
GSS80010/GSS7101A.GSS81004	GSS70200
GSS80010/GSS7101A.GSS81015	GSS10000

DD Name	Data Set Name	Dir
GSS7003	FIRE.DF100401.GPRIMV00	OUT

Search for items in lefthand column

Job Pane

Procedure/ Program Pane

Data Set Pane

2 In the Job pane, select a job to view the steps the job uses, the procedure or program each step executes, and the data stores each procedure or program reads from or writes to.

Tip: Use the search facility described in "Searching Batch Application Table Views" on page 4-7 to locate jobs.

3 Click **Procedure View** at the bottom of the window to open the procedure view. In the Procedure/Program pane, select a procedure or

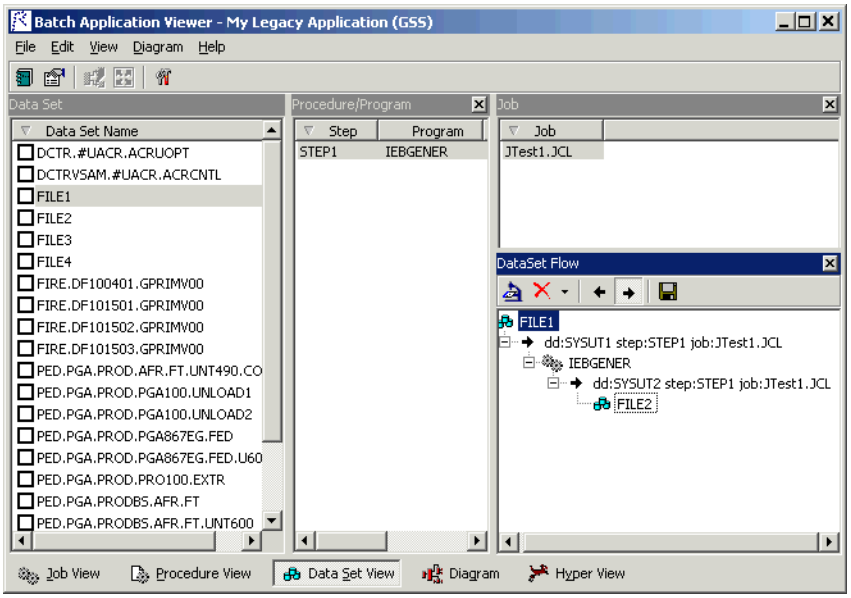
program to view the jobs and steps that execute it, and the data stores the procedure or program reads from or writes to. For procedures, Batch Application Viewer displays data stores referenced by DD statements added after the substitution of the procedure.

4 Click **Dataset View** at the bottom of the window to open the dataset view. In the Dataset pane, select a dataset to view the procedure or program that reads or writes to it, the step that executes the procedure or program, and the job that contains the step.

5 When you select the dataset view, Batch Application Viewer displays the Data Set Flow pane in the bottom right part of the window. Select a dataset in the Data Set pane and a step in the Procedure/Program pane, then click the button in the Data Set Flow pane to view the flow of data into or out of the selected dataset (Figure 4-2).

Click the ← button in the Data Set Flow pane to view the flow of data into the dataset, click the → button to view the flow of data out of the dataset.

Figure 4-2 *Batch Application Viewer Window — Data Set Flow*



Data Set Name
□ DCTR.#UACR.ACRUOPT
□ DCTRVSAM.#UACR.ACRCNTL
□ FILE1
□ FILE2
□ FILE3
□ FILE4
□ FIRE.DF100401.GPRIMV00
□ FIRE.DF101501.GPRIMV00
□ FIRE.DF101502.GPRIMV00
□ FIRE.DF101503.GPRIMV00
□ PED.PGA.PROD.AFR.FT.UNT490.CO
□ PED.PGA.PROD.PGA100.UNLOAD1
□ PED.PGA.PROD.PGA100.UNLOAD2
□ PED.PGA.PROD.PGA867EG.FED
□ PED.PGA.PROD.PGA867EG.FED.U60
□ PED.PGA.PROD.PRO100.EXTR
□ PED.PGA.PRODBS.AFR.FT
□ PED.PGA.PRODBS.AFR.FT.UNT600

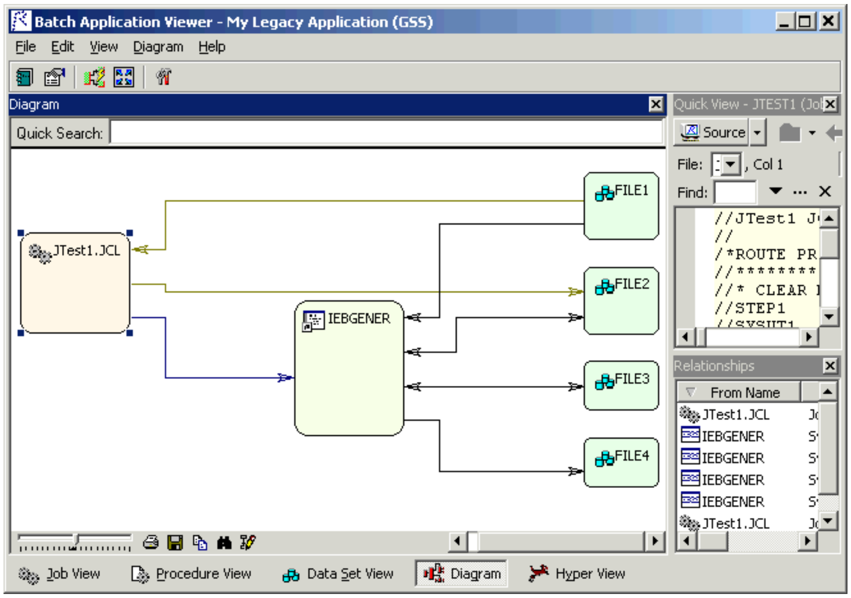
Step	Program
STEP1	IEBGENER

Job
JTest1.JCL

```
FILE1
⊟ → dd:SYSUT1 step:STEP1 job:JTest1.JCL
    ⊟ IEBGENER
        ⊟ → dd:SYSUT2 step:STEP1 job:JTest1.JCL
              FILE2
```

6 In any view, place a check mark next to one or more objects and click **Diagram** at the bottom of the window to display a diagram of batch application objects related to the objects. The diagram shows the relationships between jobs, programs or procedures, and data stores (Figure 4-3). Usage is similar to that for the Diagrammer tool described in Chapter 2.

Figure 4-3 *Batch Application Viewer Window — Diagram View*



```
//JTest1 J
//
/*ROUTE PR
//*********
//* CLEAR I
//STEP1
//SYSUT1
```

From Name	
JTest1.JCL	J
IEBGENER	S
IEBGENER	S
IEBGENER	S
IEBGENER	S
JTest1.JCL	J

7 In any table view or diagram, select an object and click **HyperView** at the bottom of the window to view the object in the ATW HyperView tool. For procedures or data stores referenced by procedures, HyperView navigates to the object in the source code for the procedure. Otherwise, it navigates to the object in the JCL file.

Note: The HyperView pane shows a view-only copy of application source. For HyperView usage information, see *Analyzing Programs* in the ATW document set.

Working with Batch Application Table Views

Batch Application Viewer offers three table views of a batch application:

- The *Job View* cross-references the selected job with the steps the job contains, the procedure or program each step executes, and the DD and dataset names of the files each procedure or program reads from or writes to. Click **Job View** at the bottom of the Batch Application Viewer window to open the procedure view.
- The *Procedure View* cross-references the selected procedure or program with the jobs and steps that execute it, and the DD and dataset names of the files it reads from or writes to. Click **Procedure View** at the bottom of the Batch Application Viewer window to open the procedure view.

Note: For procedures, Batch Application Viewer displays the DD and dataset names of files referenced by DD statements added after the substitution of the procedure.

- The *Dataset View* cross-references the selected dataset with the procedure or program that reads or writes to it, the step that executes the procedure or program, and the job that contains the step. Click **Dataset View** at the bottom of the Batch Application Viewer window to open the dataset view.

Batch Application table views consist of a Job pane, Program/Procedure pane, and Data Set pane.

Note: For the data set view, Batch Application Viewer also displays a Data Set Flow pane, where you can investigate the flow of data into or out of a data store.

You can hide a pane by clicking the close box in the upper righthand corner. Select the appropriate choice in the **View** menu to show the pane again.

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

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

Selecting Objects Click an object in a table view pane to select it. To select a range of objects, hold down the Shift key, click the first object in the range, then click the last object in the range. To select objects that are not in a range, hold down the Control key, then click each object you want to select.

Selecting Entries to Diagram Place a check mark next to an object and click **Diagram** at the bottom of the Batch Application Viewer window to display a diagram of batch application objects related to the object.

To place a check mark next to all the objects in a table view pane, choose **Check All** in the **Edit** menu. To uncheck all the selected objects, choose **Uncheck All** in the **Edit** menu.

To place a check mark next to selected objects, select the objects and choose **Check Selected** in the **Edit** menu. To uncheck all the objects, choose **Uncheck Selected** in the **Edit** menu.

Viewing the Flow of Data into or out of a Dataset Select a dataset in the Data Set pane and a step in the Procedure/Program pane, then click the button in the Data Set Flow pane to view the flow of data into or out of the selected dataset (Figure 4-2 on page 4-3). Click the left arrow in the Data Set Flow pane to view the flow of data into the dataset, click the right arrow to view the flow of data out of the dataset.

Viewing Properties Select an object in a table view pane and choose **Properties** in the **View** menu to display a set of tabs with object properties. For usage information, see *Getting Started* in the ATW document set.

Exporting a Table View Report Open a table view and choose **Report** in the **File** menu to export a printed report on the view to HTML, Excel, RTF, Word, or formatted text.

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

Searching Batch Application Table Views

Use the Batch Application Viewer search facility to locate objects in the lefthand column of table views. To use the search facility, enter the text for the search in the field next to the 🔍 button on the tool bar. Batch Application Viewer locates text matches as you type.

Click the 🔍 button or press Enter to navigate to the next matching construct. Click the adjacent ▾ button to view additional choices in a drop-down menu. Choose:

- **Next** to find the next matching object.
- **Wild Next** to use a wildcard pattern to find the next matching object.
- **Find All** to find all matching objects.
- **Wild Find All** to use a wildcard pattern to find all matching objects.

You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).

Working with Batch Application Diagrams

Batch application diagrams show the relationships between jobs, procedures and programs, and data stores (Figure 4-3 on page 4-4). Usage is similar to that for the Diagrammer tool described in Chapter 2.

Generating a Batch Application Diagram

Follow the steps below to generate a batch application diagram.

To generate a batch application diagram:

1 In the appropriate table view pane, place a check mark next to the objects you want to diagram, as described in "Selecting Entries to Diagram" on page 4-6.

2 In the **Diagram** menu, choose the relationships you want to view in the diagram:

  - **Job Dependencies** show dependencies between jobs. Jobs are regarded as dependent if one writes to a dataset and the other reads from the same dataset.
  - **Data Set Usage** shows interactions with datasets.

- **JCL Procedures Usage** shows interactions with procedures.
- **Program Usage** shows interactions with programs.

Tip: Use these choices to control the level of detail in the diagram. The choice is grayed out with a check mark beside it if the relationship must be displayed in the diagram. You can also set these choices after you display the diagram.

3 Edit the color scheme for the relationships if necessary, as described in "Editing the Color Scheme for Batch Application Diagrams" on page 4-8.

4 Click **Diagram** at the bottom of the Batch Application Viewer window. The diagram is displayed in the Diagram pane (Figure 4-3 on page 4-4).

Editing the Color Scheme for Batch Application Diagrams

Diagram objects are color-coded based on the color scheme set in the Diagrammer tool described in Chapter 2. Relationship lines are color-coded based on your settings in the Batch Application View Options window.

To edit the color scheme for relationships in batch diagrams:

1 In the **View** menu, choose **Options**. The Options window opens (Figure 4-4 on page 4-9).

2 In the Diagram Relations pane, click the relationship type whose color you want to edit. The current color of the type is displayed in the **Edge Color** drop-down.

3 Click the arrow beside the drop-down to display a standard Windows color control. Use the Palette tab to select the color of the relationship type from the Windows palette. Use the System tab to match the color of the relationship type with the color of standard Windows elements.

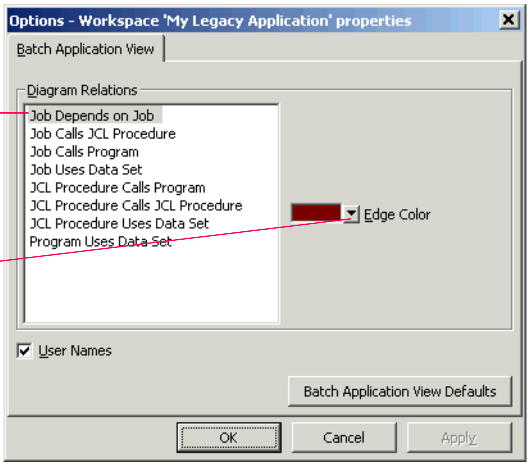
Tip: You can restore the default option settings by clicking **Batch Application Viewer Defaults**, then choosing **Restore De-**

faults in the drop-down menu. Choose **Save To** in the drop-down menu to save the option settings to a file. Choose **Load From** in the menu to restore the option settings from a file.

4A Click **Apply** if you want to save your settings without dismissing the Options window.

4B Click **OK** if you want to save your settings and dismiss the Options window.

Figure 4-4 *Options Window*


Options - Workspace 'My Legacy Application' properties
Batch Application View
Diagram Relations
Edge Color
Batch Application View Defaults
OK
Cancel
Apply

Job Depends on Job
Job Calls JCL Procedure
Job Calls Program
Job Uses Data Set
JCL Procedure Calls Program
JCL Procedure Calls JCL Procedure
JCL Procedure Uses Data Set
Program Uses Data Set

☑ User Names

click to display current color

click to display color options

Creating Job Dependencies in Batch Application Diagrams

Batch Application Viewer treats jobs as dependent if one writes to a dataset and the other reads from the same dataset. Occasionally, you may want to define dependencies between jobs based on other criteria — administrative needs such as scheduling, for example. Batch Application Viewer modifies the diagram only, not the repository.

Creating Job Dependencies To create dependencies between jobs in a batch application diagram, hold down the Alt key, select either job, and drag-and-drop the relationship to the other job. The relationship is displayed in red with three plus (+) signs at either end of the relationship line.

Exporting and Importing Job Dependencies To export job dependencies to a file, for import to diagrams for other projects, choose **Export Dependency Changes** in the **File** menu. A Save As dialog opens, where you can specify the name and folder for the export file.

To import job dependencies from a file, choose **Import Dependency Changes** in the **File** menu. An Open dialog appears, where you can select the file you want to import.

Removing Job Dependencies To remove all user-defined dependencies from a batch application diagram, choose **Clean Up Dependencies** in the **File** menu.

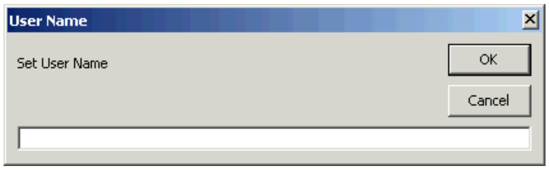
Tip: In certain circumstances — jobs that use cataloged file generation, for example — Batch Application Viewer may create *false dependencies*. You must remove these relationships manually by right-clicking the dependency and choosing **Remove** from the pop-up menu.

Creating User Names for Objects in Batch Application Diagrams

You can give diagram objects friendlier names — so-called *user names.* Batch Application Viewer modifies the diagram only, not the repository.

Creating and Displaying User Names To create a user name for an object in a batch application diagram, select the object and choose **User Name** in the **View** menu. The User Name dialog opens (Figure 4-5). Enter the user name in the text field and click **OK**. To display user names in a diagram, select **User Names** in the Batch Application Viewer Options window (Figure 4-4 on page 4-9).

Figure 4-5 *User Name Dialog*



Exporting and Importing User Names To export user names to a file, for import to diagrams for other projects, choose **Export User Names** in the **File** menu. A Save As dialog opens, where you can specify the name and folder for the export file.

To import job dependencies from a file, choose **Import User Names** in the **File** menu. An Open dialog appears, where you can select the file you want to import.

Removing User Names To remove all user names from a batch application diagram, choose **Clean Up User Names** in the **File** menu.

Copying Batch Application Diagrams to the Clipboard

Choose **Copy to Clipboard** in the **Diagram** menu to copy a batch application diagram to the clipboard in EMF format. You can paste the diagram from the clipboard to a document in a third-party tool such as Word.

Printing Batch Application Diagrams

Choose **Print** in the **Diagram** menu to print a batch application diagram. The Print Preview window opens, where you can change the zoom factor for the diagram to control the number of pages in the print job. Click **OK**.

Exporting Batch Application Diagrams

You can export batch application diagrams to a variety of standard formats. Choose **Save** in the **File** menu to export a diagram to BED, bitmap, JPEG, Visio, Visio XML, DOT, or EMF. For more information on exporting diagrams, see *Getting Started* in the ATW document set.

Note: Visio 2002 must be installed to save a diagram to Visio. Visio 2002 is not required to save to Visio XML.

You can also export a batch application diagram to the ATW WebGen tool by clicking **Create WebGen Page** in the **File** menu. All diagram types are exported to the page. The page is stored in the file *\Work-*

space\Projects*Project*\Webgen\BatchView\Main.htm. Click the hyperlink for a diagram to view it in the browser.

Note: Batch Application Viewer exports an additional diagram to WebGen showing all dependent jobs in the project.

Working with HyperView in Batch Application Viewer

Select an object in a table view or diagram and click **HyperView** at the bottom of the Batch Application Viewer window to view the object in the ATW Hyperview tool. For procedures or data stores referenced by procedures, HyperView navigates to the object in the source code for the procedure. Otherwise, it navigates to the object in the JCL file.

Note: The HyperView pane shows a view-only copy of application source. For HyperView usage information, see *Analyzing Programs* in the ATW document set.

What's Next?

Now that you know how to analyze batch applications in the Batch Application Viewer, let's look at how you perform low-level analysis of PL/I programs in the PL/I Call Diagrammer. That's the subject of the next chapter.

Analyzing PL/I Applications

5

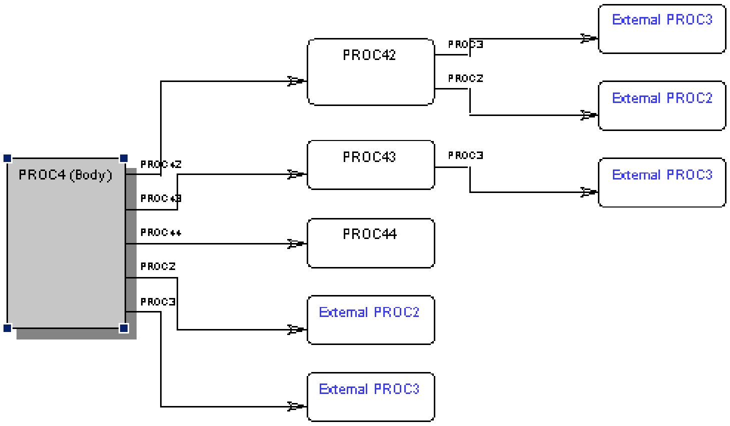
The *PL/I Call Diagrammer* lets you perform low-level analysis of PL/I programs. You can use it to examine call flows for external procedures as you would in the Diagrammer. But you can also use it to examine call flows for internal procedures that the Diagrammer is unable to model.

Understanding PL/I Call Flow Diagrams

Call flow diagrams show the call relationships between PL/I external and internal procedures (Figure 5-1 on page 5-2). The diagrams can model any level of nested procedure. Usage is similar to that for the Diagrammer tool described in Chapter 2.

In each diagram, the name of the called procedure appears on the relationship line. External, internal, and system procedures are color-coded based on your settings in the PL/I Call Diagrammer User Preferences window. For more information, see “Setting PL/I Call Diagram Options” on page 5-3.

Figure 5-1 *PL/I Call Flow Diagram*



Using the PL/I Call Diagrammer

The PL/I Call Diagrammer is implemented in the ATW HyperView tool with a full set of HyperView panes. For Source and Clipper pane usage, and HyperView usage in general, see *Analyzing Programs* in the ATW document set.

Generating PL/I Call Information

Follow the steps below to generate PL/I call information.

To generate PL/I call information:

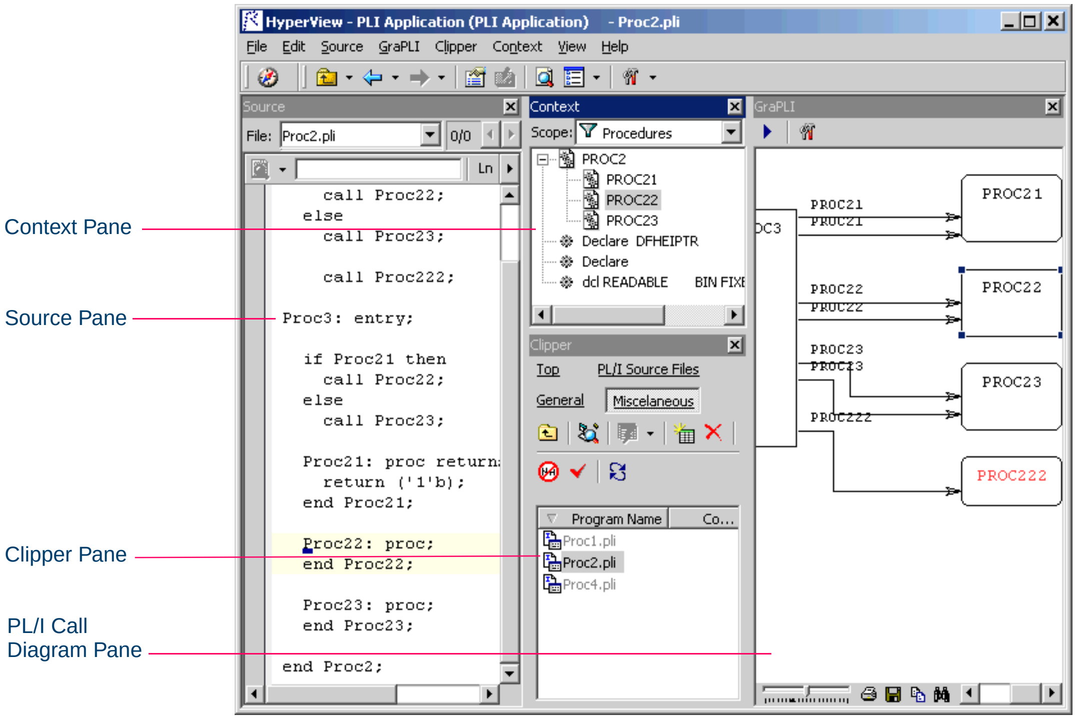
1 In the Repository Browser, select the program you want to analyze and choose **Interactive Analysis** in the **Analyze** menu. The HyperView window opens. In the **View** menu, choose **PL/I Call Diagram**. The PL/I Call Diagrammer window opens (Figure 5-2 on page 5-3).

Tip: Use the Clipper pane to select a different program in the project without having to exit HyperView.

2 The Context pane displays the procedures in the program in hierarchic form. Select the procedure whose relationships you want to

view, then choose **Reconstruct** in the **PLI Call Diagrammer** menu. The PL/I Call Diagram pane displays a diagram that shows each call the procedure makes to an external or internal procedure.

Figure 5-2 *PL/I Call Diagrammer Window (After Drawing Diagram)*



```
      call Proc22;
   else
      call Proc23;

      call Proc222;

Proc3: entry;

   if Proc21 then
      call Proc22;
   else
      call Proc23;

   Proc21: proc return
      return ('1'b);
   end Proc21;

   Proc22: proc;
   end Proc22;

   Proc23: proc;
   end Proc23;

end Proc2;
```

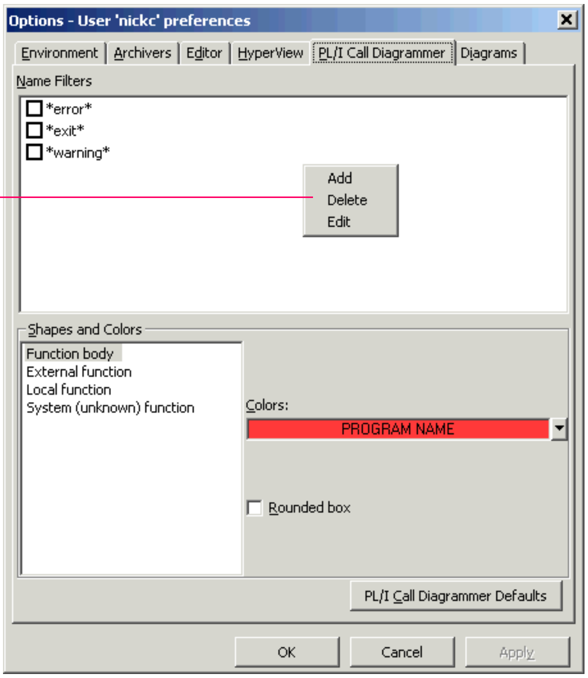
Setting PL/I Call Diagram Options

Use the PL/I Call Diagrammer User Preferences window to filter out unwanted procedures from diagrams and define a color scheme for diagrams.

To set PL/I call diagram User Preferences:

1 In the **View** menu, choose **User Preferences**. The User Preferences window opens. Click the **HyperView** tab. In the Preferences pane, click **PL/I Call Diagrammer**. The PL/I Call Diagrammer Options window opens (Figure 5-3 on page 5-4).

Figure 5-3 *PL/I Call Diagrammer Options Window*



right-click menu

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

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

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

3 In the Shapes and Colors pane, click the object type whose color you want to edit. The current foreground (caption) color of the type is displayed in the **Colors** drop-down.

Click the arrow beside the drop-down and choose **ForeColor** from the pop-up menu if you want to edit the color of the caption for the object type, or **BackColor** if you want to edit the color of the background for the object type.

A standard Windows color control is displayed. Use the **Palette** tab to select the color of the caption or background from the Windows palette. Use the **System** tab to match the color of the caption or background with the color of standard Windows elements.

4 Select **Rounded box** if you want the border of diagram objects to have rounded corners.

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

5A Click **Apply** if you want to save your settings without dismissing the User Preferences window.

5B Click **OK** if you want to save your settings and dismiss the User Preferences window.

Copying PL/I Call Diagrams to the Clipboard

Choose **Copy Diagram** in the **PL/I Call Diagram** menu to copy a PL/I call diagram to the clipboard in EMF format. You can paste the diagram from the clipboard to a document in a third-party tool such as Word.

Printing PL/I Call Diagrams

Choose **Print Diagram** in the **PL/I Call Diagram** menu to print a PL/I call diagram. The Print Preview window opens, where you can change the zoom factor for the diagram to control the number of pages in the print job. Click **OK**.

Exporting PL/I Call Diagrams

You can export PL/I call diagrams to a variety of standard formats. Choose **Save Diagram As** in the **PL/I Call Diagram** menu to export a diagram to BED, bitmap, JPEG, Visio, Visio XML, DOT, or EMF. For more information on exporting diagrams, see *Getting Started* in the ATW document set.

Note: Visio 2002 must be installed to save a diagram to Visio. Visio 2002 is not required to save to Visio XML.

What's Next?

That completes your tour of ATW program and data flow analysis tools. The next chapter shows you how to use the workbench CRUD report and IMS port analysis feature to analyze data operations.

Analyzing Data Operations

6

The workbench parser generates relationships that show the data operations a program performs — Insert, Read, Update, or Delete — and the data objects on which the program operates — files, segments, tables, and so forth. This chapter looks at a report that lets you view these relationships in a convenient format, and a mechanism that lets you trace PSB usage through an entire IMS application call sequence.

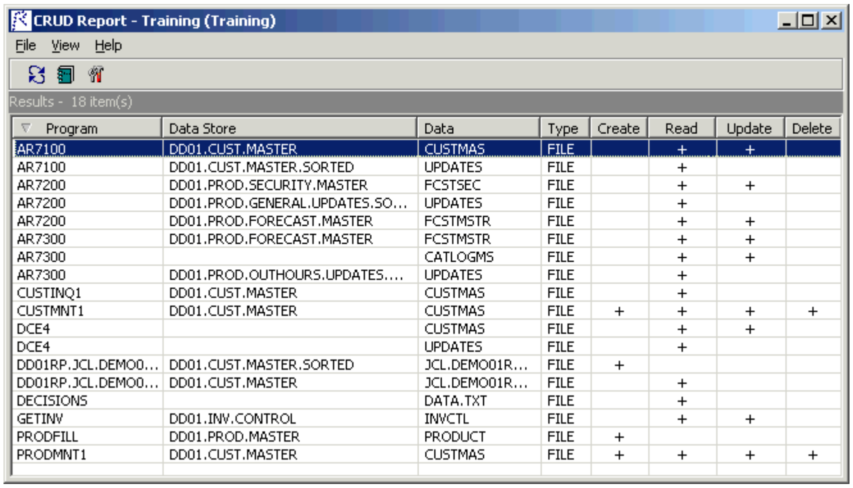
Viewing CRUD Reports

The CRUD report for a project shows the data operations each program in the project performs, and the data objects on which the programs operate.

To generate a CRUD report:

1 In the Repository Browser, select a project and choose **CRUD Report** in the **Prepare** menu. The CRUD Report window opens (Figure 6-1 on page 6-2).

Figure 6-1 *CRUD Report*



Results - 18 item(s)

Program	Data Store	Data	Type	Create	Read	Update	Delete
AR7100	DD01.CUST.MASTER	CUSTMAS	FILE		+	+	
AR7100	DD01.CUST.MASTER.SORTED	UPDATES	FILE		+		
AR7200	DD01.PROD.SECURITY.MASTER	FCSTSEC	FILE		+	+	
AR7200	DD01.PROD.GENERAL.UPDATES.SO...	UPDATES	FILE		+		
AR7200	DD01.PROD.FORECAST.MASTER	FCSTMSTR	FILE		+	+	
AR7300	DD01.PROD.FORECAST.MASTER	FCSTMSTR	FILE		+	+	
AR7300		CATLOGMS	FILE		+	+	
AR7300	DD01.PROD.OUTHOURS.UPDATES....	UPDATES	FILE		+		
CUSTINQ1	DD01.CUST.MASTER	CUSTMAS	FILE		+		
CUSTMNT1	DD01.CUST.MASTER	CUSTMAS	FILE	+	+	+	+
DCE4		CUSTMAS	FILE		+	+	
DCE4		UPDATES	FILE		+		
DD01RP.JCL.DEMO0...	DD01.CUST.MASTER.SORTED	JCL.DEMO01R...	FILE	+			
DD01RP.JCL.DEMO0...	DD01.CUST.MASTER	JCL.DEMO01R...	FILE		+		
DECISIONS		DATA.TXT	FILE		+		
GETINV	DD01.INV.CONTROL	INVCTL	FILE		+	+	
PRODFILL	DD01.PROD.MASTER	PRODUCT	FILE	+			
PRODMNT1	DD01.CUST.MASTER	CUSTMAS	FILE	+	+	+	+

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

2 To select the types of program-to-data object relationships to display, and the data operations to show for each type, choose **Options** in the **View** menu. The CRUD Report Options window opens (Figure 6-2 on page 6-3).

3 In the CRUD Report Options window:

- Place a check mark next to each type of program-to-data object relationship you want to display.
- Place a check mark next to each data operation you want to display.

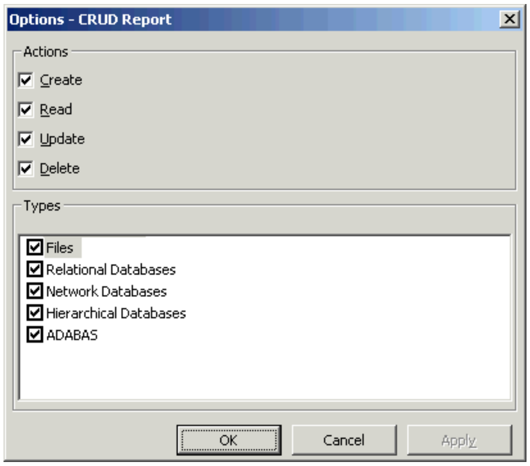
Click **Apply** if you want to save your settings without dismissing the CRUD Report Options window open. Click **OK** if you want to save your settings and dismiss the CRUD Report Options window.

4 Choose **Refresh** in the File menu to refresh the report.

5 Choose **Report** in the **File** menu to export the report to HTML, Excel, RTF, Word, or formatted text.

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

Figure 6-2 *CRUD Report Options Window*



☑ Create
☑ Read
☑ Update
☑ Delete

☑ Files
☑ Relational Databases
☑ Network Databases
☑ Hierarchical Databases
☑ ADABAS

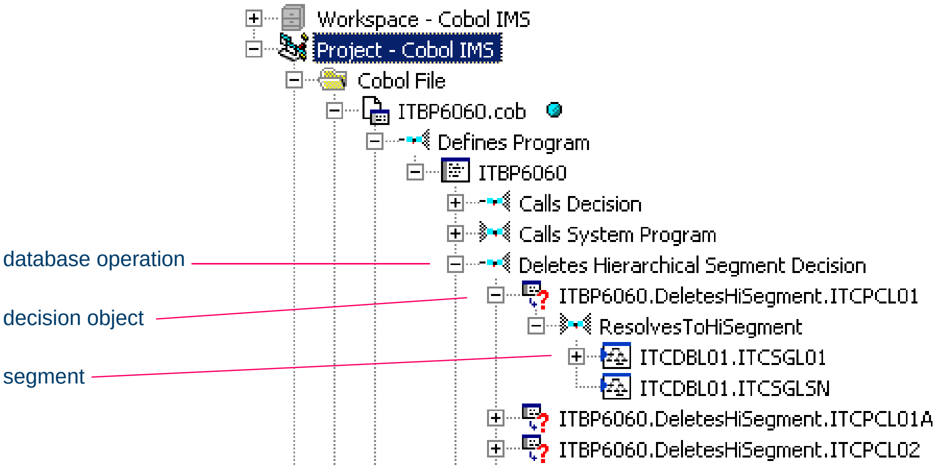
Enabling IMS Port Analysis

It is virtually impossible to determine from program code the database segments or screens an IMS program operates on. Only an application-wide analysis can trace PSB usage through the entire application call sequence.

To determine the types of database operation — insert, read, update, or delete — an IMS program performs, and to list in the browser each of the database segments or screens the operation may be performed on, select a project and choose **IMS Analysis** in the **Prepare** menu.

Figure 6-3 shows typical results. The objects marked with the icon are abstract *decision objects,* indicating that the database operation — in this case, Deletes — has not been resolved to a database segment.

Figure 6-3 *IMS Port Analysis Results*



```
Workspace - Cobol IMS
Project - Cobol IMS
  Cobol File
    ITBP6060.cob
      Defines Program
        ITBP6060
          Calls Decision
          Calls System Program
          Deletes Hierarchical Segment Decision
            ITBP6060.DeletesHiSegment.ITCPCL01
              ResolvesToHiSegment
                ITCDBL01.ITCSGL01
                ITCDBL01.ITCSGLSN
            ITBP6060.DeletesHiSegment.ITCPCL01A
            ITBP6060.DeletesHiSegment.ITCPCL02
```

database operation

decision object

segment

Setup

The IMS port analysis feature assumes that your application identifies IMS “root programs” and corresponding PSBs in a JCL file for a batch application or a PCT file for an online application. A root program is directly invoked by IMS with a list of PCBs as parameters. It can pass these PCBs as parameters in calls to other programs.Verify source files in the following order:

Tip: If you verify an entire project, the workbench parses the files in appropriate order, taking account of the dependencies between file types.

- DBD files (assuming you need full cross-reference information)
- MFS files
- PSB files
- Cobol or PL/I files

Note: For Cobol files, set the **Perform Program Analysis** and **Enable Data Element Flow** project verification options. For PL/I files, set the **Enable Data Element Flow** project verification option

- JCL or PCT files

Note: For PCT files, set the **Treat as IMS System Definition** option in the Workspace Verification options Settings tab.

What's Next?

That's all you need to know to use the workbench CRUD report and IMS port analysis feature to analyze project data operations. Now let's look at how you use the legacy estimation tools to estimate project complexity and effort.

Estimating Complexity and Effort

7

Suppose you are planning to implement a change request for a program and want to know how long it will take to complete the change. ATW *legacy estimation* tools let you compare programs based on weighted values for selected complexity metrics. Based on the comparison, you can develop a credible estimate of the time required to make the requested change.

The complexity metrics used in the calculation are a combination of industry standard and ATW-generated statistics (Table 7-1 on page 7-8). If your own analysis shows that a given program is more or less complex than the weighted calculation would suggest, you can set a *change magnitude* to override the calculated value for the program.

The program might have thousands of source lines, for example, increasing its calculated complexity, while actually being very easy to modify. When you use a change magnitude, your “subjective” estimate of the effort involved — Small, Medium, Large, Extra Large — becomes an input to the effort calculation, along with the weighted values.

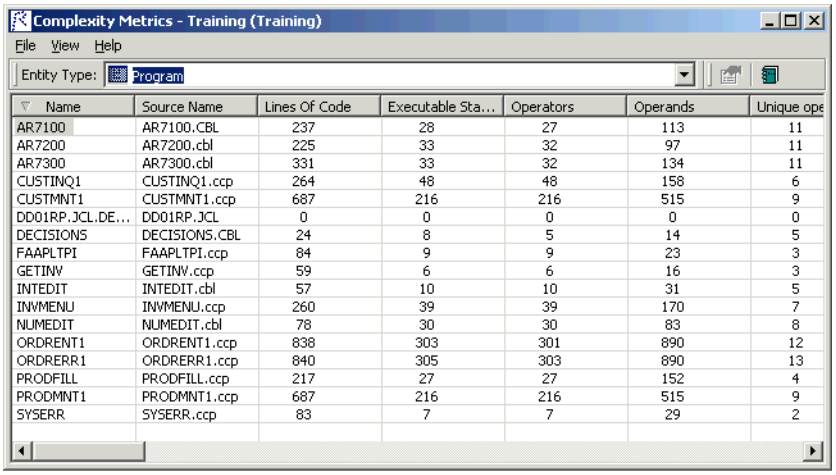
Using the Complexity Metrics Tool

Use the *Complexity Metrics* tool to compare raw complexity values for the objects in your project.

To generate complexity metrics:

1. In the Repository Browser, select the project whose complexity metrics you want to calculate and choose **Complexity** in the **Analyze** menu. A blank Complexity Metrics window opens.
2. In the **Entity Type** drop-down, choose the type of object you want to calculate complexity metrics for. Figure 7-1 shows the results for the program object. The metrics are described in Table 7-1 on page 7-8.

Figure 7-1 *Complexity Metrics Window*



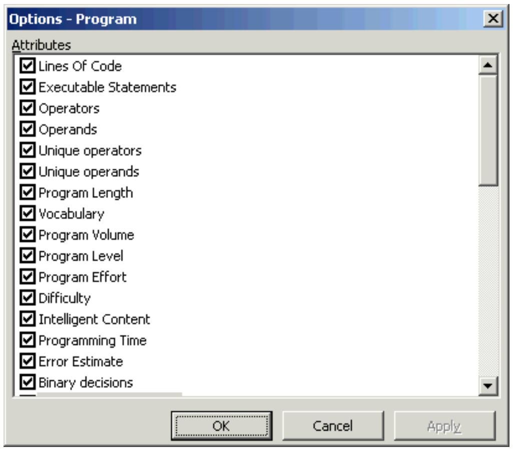
Entity Type: Program

▽ Name	Source Name	Lines Of Code	Executable Sta...	Operators	Operands	Unique ope
AR7100	AR7100.CBL	237	28	27	113	11
AR7200	AR7200.cbl	225	33	32	97	11
AR7300	AR7300.cbl	331	33	32	134	11
CUSTINQ1	CUSTINQ1.ccp	264	48	48	158	6
CUSTMNT1	CUSTMNT1.ccp	687	216	216	515	9
DD01RP.JCL.DE...	DD01RP.JCL	0	0	0	0	0
DECISIONS	DECISIONS.CBL	24	8	5	14	5
FAAPLTPI	FAAPLTPI.ccp	84	9	9	23	3
GETINV	GETINV.ccp	59	6	6	16	3
INTEDIT	INTEDIT.cbl	57	10	10	31	5
INVMENU	INVMENU.ccp	260	39	39	170	7
NUMEDIT	NUMEDIT.cbl	78	30	30	83	8
ORDRENT1	ORDRENT1.ccp	838	303	301	890	12
ORDRERR1	ORDRERR1.ccp	840	305	303	890	13
PRODFILL	PRODFILL.ccp	217	27	27	152	4
PRODMNT1	PRODMNT1.ccp	687	216	216	515	9
SYSERR	SYSERR.ccp	83	7	7	29	2

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

3. To select the metrics to be included in the calculation, choose **Columns** in the **View** menu. The Attributes window opens (Figure 7-2 on page 7-3).

Figure 7-2 *Complexity Metrics — Attributes Window*



☑ Lines Of Code
☑ Executable Statements
☑ Operators
☑ Operands
☑ Unique operators
☑ Unique operands
☑ Program Length
☑ Vocabulary
☑ Program Volume
☑ Program Level
☑ Program Effort
☑ Difficulty
☑ Intelligent Content
☑ Programming Time
☑ Error Estimate
☑ Binary decisions

4 In the Attributes window, place a check mark next to each complexity metric you want to calculate. Click **Apply** if you want to save your settings without dismissing the Attributes window open. Click **OK** if you want to save your settings and dismiss the Attributes window.

5 Select an object and choose **Properties** in the **View** menu to display a set of tabs with object properties. For usage information, see *Getting Started* in the ATW document set.

Using the Effort Estimation Tool

Use the *Effort Estimation* tool to compare source files based on weighted values for selected complexity metrics.

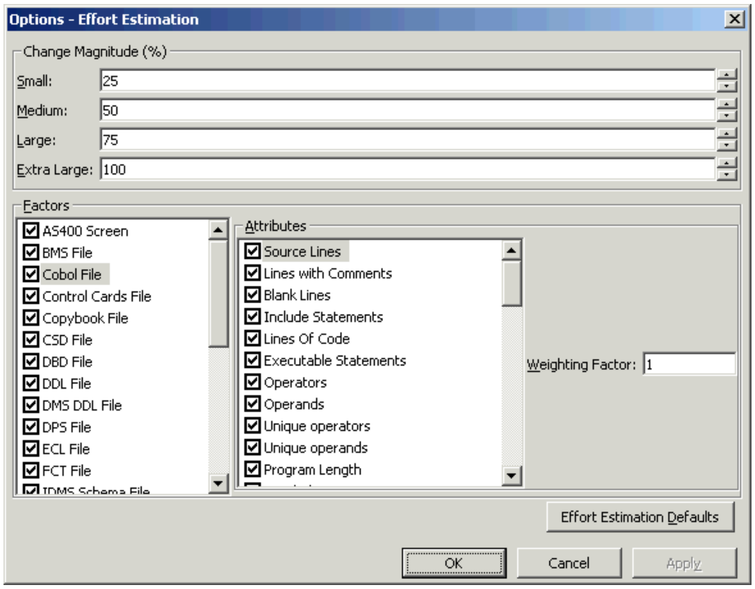
To generate effort estimation statistics:

1 In the Repository Browser, select the project whose effort estimation you want to calculate and choose **Effort** in the **Analyze** menu. A blank Effort Estimation window opens.

2 To select the file types to be included in the calculation, and the complexity metrics for each type, and to set the percentage factor for the change magnitudes used in the calculation, choose **Options** in the

View menu. The Effort Estimation Options window opens (Figure 7-3).

Figure 7-3 *Effort Estimation Options Window*



Options - Effort Estimation

Change Magnitude (%)

Small: 25

Medium: 50

Large: 75

Extra Large: 100

Factors

- ☑ AS400 Screen
- ☑ BMS File
- ☑ Cobol File
- ☑ Control Cards File
- ☑ Copybook File
- ☑ CSD File
- ☑ DBD File
- ☑ DDL File
- ☑ DMS DDL File
- ☑ DPS File
- ☑ ECL File
- ☑ FCT File
- ☑ IDMS Schema File

Attributes

- ☑ Source Lines
- ☑ Lines with Comments
- ☑ Blank Lines
- ☑ Include Statements
- ☑ Lines Of Code
- ☑ Executable Statements
- ☑ Operators
- ☑ Operands
- ☑ Unique operators
- ☑ Unique operands
- ☑ Program Length

Weighting Factor: 1

3 In the Effort Estimation Options window:

- Place a check mark next to each source file type you want to include in the calculation.
- For each source file type to be included in the calculation, place a check mark next to each complexity metric you want to use in the calculation. The metrics are described in Table 7-1 on page 7-8.
- For each complexity metric to be used in the calculation, enter its weight in the **Weighting Factor** field. If you want Cyclomatic Complexity to have twice the weight of Conditional Complexity, for example, set the weighting factor for Cyclomatic Complexity to 2 and the weighting factor for Conditional Complexity to 1.
- To set the percentage factor for the change magnitudes used in the calculation, enter the percentage you want to be applied in the

combo box for each change magnitude value: Small, Medium, Large, and Extra Large. The values shown in Figure 7-3 on page 7-4 are representative. For background on change magnitudes, see “Specifying Change Magnitudes” on page 7-6.

Click **Apply** if you want to save your settings without dismissing the Effort Estimation Options window open. Click **OK** if you want to save your settings and dismiss the Effort Estimation Options window.

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

4 Click the ▶ button on the tool bar to generate effort estimation statistics. The statistics for the selected source file type are displayed. Figure 7-4 on page 7-6 shows the results for Cobol source files.

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

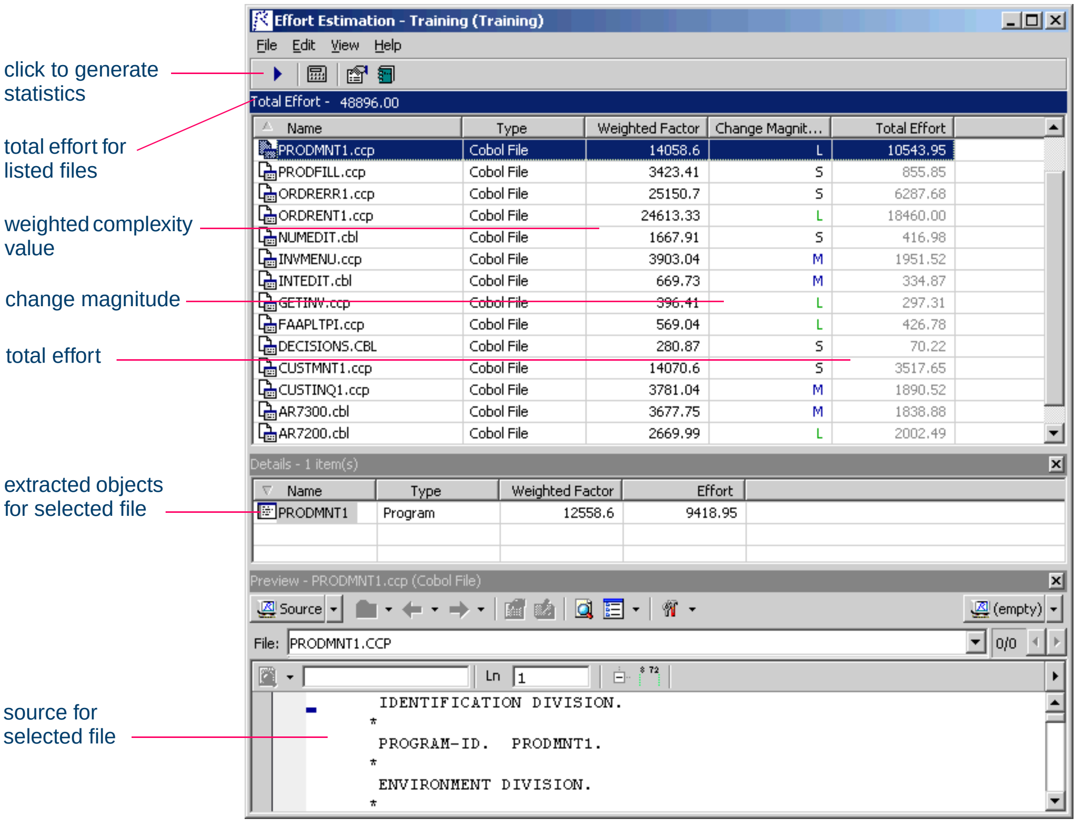
5 Select a source file in the Effort pane to show the effort statistics for each extracted object in the file in the Details pane.

6 Select a source file in the Effort pane to view HyperCode for the file in the Preview pane. The information available depends on the type of object selected — you see only source code for a copybook, for example, but full HyperCode for a program.

Note: “HyperCode” is shorthand for the information displayed in the ATW HyperView tool. For HyperView usage information, see *Analyzing Programs* in the ATW document set.

7 Select a file and choose **Properties** in the **View** menu to display a set of tabs with object properties. For usage information, see *Getting Started* in the ATW document set.

Figure 7-4 *Effort Estimation Window*


click to generate statistics
total effort for listed files
weighted complexity value
change magnitude
total effort
extracted objects for selected file
source for selected file
Effort Estimation - Training (Training)
File
Edit
View
Help
Source
(empty)

Total Effort - 48896.00

Name	Type	Weighted Factor	Change Magnit...	Total Effort
PRODMNT1.ccp	Cobol File	14058.6	L	10543.95
PRODFILL.ccp	Cobol File	3423.41	S	855.85
ORDRERR1.ccp	Cobol File	25150.7	S	6287.68
ORDRENT1.ccp	Cobol File	24613.33	L	18460.00
NUMEDIT.cbl	Cobol File	1667.91	S	416.98
INVMENU.ccp	Cobol File	3903.04	M	1951.52
INTEDIT.cbl	Cobol File	669.73	M	334.87
GETINV.ccp	Cobol File	396.41	L	297.31
FAAPLTPI.ccp	Cobol File	569.04	L	426.78
DECISIONS.CBL	Cobol File	280.87	S	70.22
CUSTMNT1.ccp	Cobol File	14070.6	S	3517.65
CUSTINQ1.ccp	Cobol File	3781.04	M	1890.52
AR7300.cbl	Cobol File	3677.75	M	1838.88
AR7200.cbl	Cobol File	2669.99	L	2002.49

Details - 1 item(s)

Name	Type	Weighted Factor	Effort
PRODMNT1	Program	12558.6	9418.95

Preview - PRODMNT1.ccp (Cobol File)

File: PRODMNT1.CCP 0/0

Ln 1

```
       IDENTIFICATION DIVISION.
      *
       PROGRAM-ID.   PRODMNT1.
      *
       ENVIRONMENT DIVISION.
      *
```

Specifying Change Magnitudes

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

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

You typically set the change magnitude for a file in the HyperView Context pane, but you can also set it (or change it) in the Effort Estimation tool. For Hyperview usage, see *Analyzing Programs* in the ATW document set.

In the Effort Estimation window (Figure 7-4 on page 7-6), select the files whose change magnitudes you want to set and choose **Set Change Magnitude:*magnitude*** in the **Edit** menu, where *magnitude* is S for Small, M for Medium, L for Large, or XL for Extra Large. The effort calculation for the selected files is automatically updated.

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

Exporting Complexity and Effort Reports

You can export complexity and effort reports to a variety of standard formats. Choose **Report** in the **File** menu to export the report to HTML, Excel, RTF, Word, or formatted text.

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

You can also export a report to the ATW WebGen tool by clicking **Create WebGen Page** in the **File** menu.

- The complexity report is stored in the file \Workspace\Projects \Project\Webgen\Complexity\Main.htm.
- The effort report is stored in the file \Workspace\Projects\Project \Webgen\Effort\Main.htm.

Click the hyperlink for a report to view it in the browser.

Supported Complexity Metrics

The following table describes the industry standard and ATW-generated complexity metrics supported by the legacy estimation tools. For details on language-specific metrics, see the *Parser Reference Manual* in the ATW document set.

Table 7-1 *Complexity Metrics*

Metric	Object Types	Description
Absolute Complexity	Program	Number of Binary Decisions divided by the number of statements.
Areas	DMS DDL file, IDMS Schema file	Number of areas.
Binary Decisions	Program	Number of branching conditions in the flow graph with two possible outcomes. Includes statements with implicit condition evaluation (loops, AT END, and so on).
Blank Lines	All	Number of blank lines of source (sequence number area content is ignored).
Call Statements	Program	Number of invocations of external modules.
Columns	DDL file	Number of columns.
Computational Statements	Program	Number of statements performing arithmetic calculations
Conditional Complexity	Program	Number of Binary Decisions plus the number of Unique Operands in Conditional Statements.
Conditional Statements	Program	Number of branching statements with nested statements executed under certain conditions, not including conditional GOTOs.

Table 7-1 *Complexity Metrics* (continued)

Metric	Object Types	Description
Cyclomatic Complexity	Program	**v(G) = e - n + 2**, where v(G) is the cyclomatic complexity of the flow graph (G) for the program in question, e is the number of edges in G, and n is the number of nodes. Quantity of decision logic. The number of linearly independent paths (minimum number of paths to be tested). **v(G) = DE + 1**, where DE is the number of binary decisions made in the program.
Data Elements	Copybook file, Program	Number of declared data items (elementary structures and their fields).
Dead Data Elements	Program	Number of dead data elements. Dead data elements are unused level-1 structures and their members. A level-1 structure is regarded as unused if it and all its members are unused.
Dead Data Elements from Includes	Program	Number of dead data elements in included files. Dead data elements are unused level-1 structures and their members. A level-1 structure is regarded as unused if it and all its members are unused.
Dead Lines	Program	Number of dead lines. Dead lines are source lines containing dead statements or dead data elements.
Dead Lines from Includes	Program	Number of dead lines in included files. Dead lines are source lines containing dead statements or dead data elements.

Table 7-1 *Complexity Metrics* (continued)

Metric	Object Types	Description
Dead Statements	Program	Number of dead statements. A dead statement is a procedural statement that can never be reached during program execution.
Dead Statements from Includes	Program	Number of dead statements in included files. A dead statement is a procedural statement that can never be reached during program execution.
Difficulty	Program	**D = (n1 / 2) * (N2 / n2)**, where n1 is the number of Unique Operators, N2 is the number of Operands, and n2 is the number of Unique Operands.
Entry Points	Program	Number of program entry points.
Error Estimate	Program	**B = E**(2/3) / 3000**, where E is Programming Effort.
Executable Statements	Program	All Cobol Procedure Division statements, plus CONTINUE NEXT STATEMENT. All PL/I statements, except BEGIN, DECLARE, DO (block), END, ENTRY, PACKAGE, and PROCEDURE. All Natural statements, except DEFINE DATA and IGNORE.
Extended Cyclomatic Complexity	Program	Cyclomatic Complexity plus the number of Logical Operators in Conditions. Number of all paths in the program.
Fields of Adabas File Number	Natural DDM file	Number of fields of Adabas file number.

Table 7-1 *Complexity Metrics* (continued)

Metric	Object Types	Description
Foreign Keys	DDL file	Number of foreign keys.
Function Points	Program	Lines of Code divided by K, where K depends on the language: Cobol=77, Natural=52, PL/I=67. Estimate of the number of end-user business functions implemented by the program.
GoTo Statements	Program	Number of GOTO statements, including conditional GOTOs.
Include Statements	Cobol file, Natural Subroutine file, Natural file, PL/I source file	Number of language-specific include statements.
Inner Call Statements	Program	Number of statements that invoke Inner Procedures.
Inner Procedures	Program	Number of structured pieces of code that cannot be invoked from external programs (Cobol paragraphs, inner PL/I procedures, Natural inline subroutines).
Input Fields	Screen	Number of input fields.
Intelligent Content	Program	**I = L * V**, where L is the Program Level and V is the Program Volume. Complexity of a given algorithm independent of the language used to express the algorithm.
IO Statements	Program	Number of statements performing input/output operations.
Lines of Code	Program	Number of lines of code, including copybooks, but not including comments and blank lines.

Table 7-1 *Complexity Metrics* (continued)

Metric	Object Types	Description
Lines with Comments	All	Number of lines of source containing comments, including inline comments placed on lines with statements.
LINK Statements	Program	Number of CICS LINK statements.
Logical Operators in Conditions	Program	Number of binary logical operators used in conditions.
Loop Statements	Program	Number of repetitively executing statements.
Maps	Natural Map	Number of maps.
Nesting Level	Program	Maximum nesting of conditional statements within conditional statements (0 if no conditional statements, 1 if no nesting).
Number of Entries	CSD file, FCT file, PCT file, Valtab file	Number of entries.
Number of Executes	ECL file	Number of executes.
Number of PCBs	PSB file	Number of PCBs.
Number of Processor Calls	ECL file	Number of processor calls.
Number of Steps	JCL file	Number of steps.
Operands	Program	Number of operands (N2). Operands are variables and literals used in operators. Uniqueness of literals is determined by their notation.
Operators	Program	Number of operators (N1). Operators are executable statements and unary and binary operations.
Output Fields	Screen	Number of output fields.

Table 7-1 *Complexity Metrics* (continued)

Metric	Object Types	Description
Parameters	Program	Number of Cobol Procedure Division USING...RETURNING parameters. Number of PL/I PROCEDURE parameters. Number of Natural PARAMETER and PARAMETER USING parameters.
Pointers	Program	Number of data elements declared as pointers.
Primary Keys	DDL file	Number of primary keys.
Program Length	Program	**N = N1 + N2**, where N1 is the number of Operators and N2 is the number of Operands.
Program Level	Program	**L = 1 / D**, where D is the program Difficulty.
Program Volume	Program	**V = N * log2(n)**, where N is the Program Length and n is the program Vocabulary. Minimum number of bits required to code the program.
Programming Effort	Program	**E = V / L**, where V is the Program Volume and L is the Program Level. Estimated mental effort required to develop the program.
Programming Time	Program	**T = E / 18**, where E is the Programming Effort and 18 is Stroud’s Number. Estimated amount of time required to implement the algorithm, in seconds.
Records	DMS DDL file, IDMS Schema file	Number of records.

Table 7-1 *Complexity Metrics* (continued)

Metric	Object Types	Description
Screens	AS400 screen, BMS file, DPS file, MFS file	Number of screens.
Segments	DBD file	Number of segments.
Sets	DMS DDL file, IDMS Schema file	Number of sets.
Source Lines	All	Number of lines of source.
Tables	DDL file	Number of tables.
Unique Operands	Program	Number of distinct operands (n2). Operands are variables and literals used in operators. Uniqueness of literals is determined by their notation.
Unique Operands in Conditions	Program	Number of distinct operands used in conditions.
Unique Operators	Program	Number of distinct operators (n1). Operators are statements and unary and binary operations.
Vocabulary	Program	**n = n1 + n2**, where n1 is the number of Unique Operators and n2 is the number of Unique Operands.
XCTL Statements	Program	Number of CICS XCTL statements.

What's Next?

Now that you know how to estimate project complexity and effort, let's look at how you define an application namespace for your project. That's the subject of the next chapter.

Defining an Application Namespace

8

The Application Namespace tool creates a conveniently organized dictionary that helps you navigate through your system's terminology and modify it as necessary. You can also use the tool to provide pseudo-code versions of application names that are easier for business analysts to understand.

Namespace *mapping* lets you change identifiers in system source code without undermining the system's consistency. If your legacy application uses the identifiers CUSTMR and CSTMR to refer to a customer, for example, you can use the tool to rename both identifiers to CUSTOMER, and propagate the change to all the source files that need it.

Understanding Application Namespaces

When you open the Application Namespace tool, it displays a *dictionary* that lists every *token* in the project. 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.

The dictionary also lists the identifiers that contain each token. To rename the identifiers, you simply change one or more of the tokens they contain. You can apply changes automatically or manually. You can also add business (pseudocode) names for identifiers to the dictionary and provide descriptive comments for constants.

Verifying Renamed Identifiers

Before you propagate name changes to source files, you need to *verify* that the renamed identifiers do not conflict with existing identifiers. The Application Namespace tool can:

- Verify new identifiers successfully.
- Verify new identifiers with non-fatal conflicts — two fields with the same name are members of different structures, for example. The tool places a ⚔ symbol beside identifiers with non-fatal conflicts.
- Verify the new identifiers with fatal errors — a field and a structure have the same name, for example. The tool places a ⚔ symbol beside identifiers with fatal conflicts.

The Application Namespace tool lets you propagate name changes that generate non-fatal conflicts, but not name changes that generate fatal conflicts. You can select an offending identifier in the Names pane to compare its declaration with the declaration for the conflicting identifier in the Conflicts pane.

Propagation Output

When you propagate names changes to source files, the Application Namespace tool stores copies of the updated files in the folder *\Workspace*\Projects*\Project*\Dictionary*\Source File Type*. It does not add the updated source files to the repository. Instead, it leaves it up to you to register and verify the modified files.

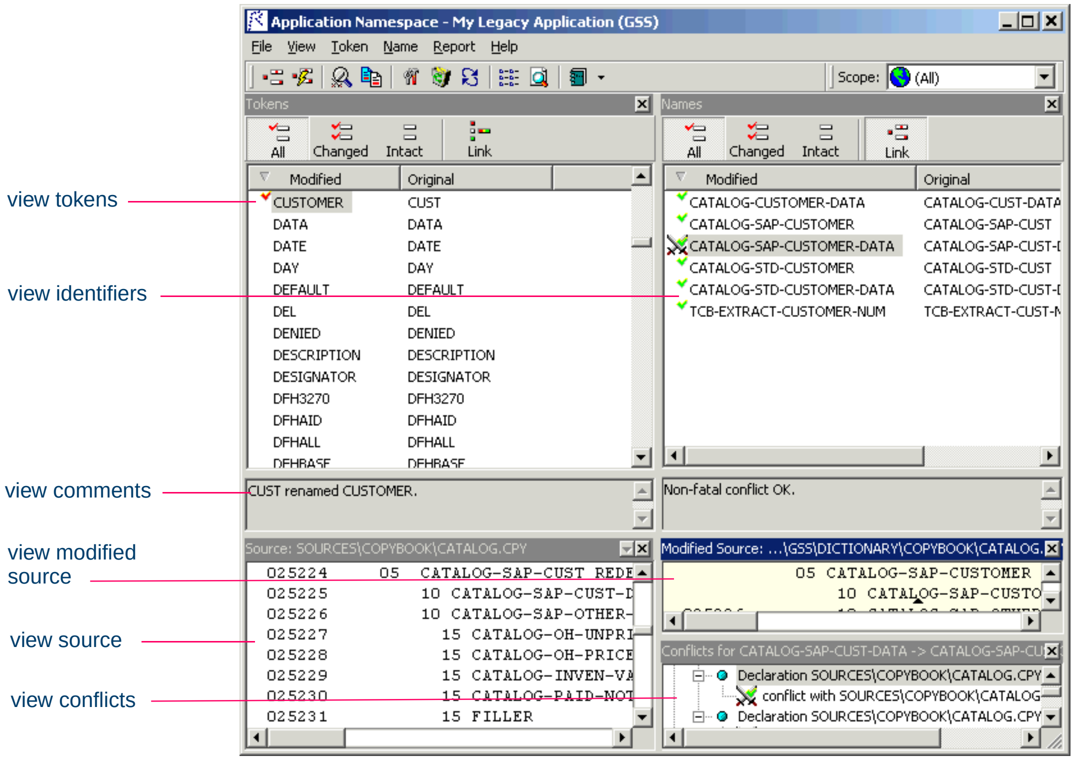
Getting Started in the Application Namespace Tool

Let's look at a sample use of the Application Namespace tool that shows you how to rename identifiers, verify that the renamed identifiers do not conflict with existing identifiers, and propagate the name change to project source files. Assume that you want to replace the token CUST with CUSTOMER in each identifier that uses the token.

To define an application namespace:

1 In the Repository Browser, select the project whose namespace you want to define and choose **Application Namespace** in the **Architect** menu. The Application Namespace window opens, with the dictionary for the project (Figure 8-1).

Figure 8-1 *Application Namespace Window (After Propagation)*



Modified	Original
CUSTOMER	CUST
DATA	DATA
DATE	DATE
DAY	DAY
DEFAULT	DEFAULT
DEL	DEL
DENIED	DENIED
DESCRIPTION	DESCRIPTION
DESIGNATOR	DESIGNATOR
DFH3270	DFH3270
DFHAID	DFHAID
DFHALL	DFHALL
DFHBASE	DFHBASE

Modified	Original
CATALOG-CUSTOMER-DATA	CATALOG-CUST-DATA
CATALOG-SAP-CUSTOMER	CATALOG-SAP-CUST
CATALOG-SAP-CUSTOMER-DATA	CATALOG-SAP-CUST-I
CATALOG-STD-CUSTOMER	CATALOG-STD-CUST
CATALOG-STD-CUSTOMER-DATA	CATALOG-STD-CUST-I
TCB-EXTRACT-CUSTOMER-NUM	TCB-EXTRACT-CUST-N

CUST renamed CUSTOMER.

Non-fatal conflict OK.

Source: SOURCES\COPYBOOK\CATALOG.CPY

```
025224     05  CATALOG-SAP-CUST REDE
025225         10 CATALOG-SAP-CUST-D
025226         10 CATALOG-SAP-OTHER-
025227            15 CATALOG-OH-UNPRI
025228            15 CATALOG-OH-PRICE
025229            15 CATALOG-INVEN-VA
025230            15 CATALOG-PAID-NOT
025231            15 FILLER
```

Modified Source: ...\GSS\DICTIONARY\COPYBOOK\CATALOG.

```
        05 CATALOG-SAP-CUSTOMER
            10 CATALOG-SAP-CUSTO
```

Conflicts for CATALOG-SAP-CUST-DATA -> CATALOG-SAP-CU

- Declaration SOURCES\COPYBOOK\CATALOG.CPY
  - conflict with SOURCES\COPYBOOK\CATALOG
- Declaration SOURCES\COPYBOOK\CATALOG.CPY

2. In the Tokens pane, select the CUST token and choose **Edit** in the right-click menu. In the edit field for the token, enter CUSTOMER and press Enter. Application Namespace places a ✓ symbol next to the token in the Tokens pane.
3. In the **File** menu, choose **Apply Mapping** to rename the identifiers that contain the token. Application Namespace places a ✓ symbol next to the identifiers in the Names pane.

Tip: Choose **Automatic Mapping** in the **File** menu if you want the Application Namespace tool to perform step 3 automatically.

4. In the **File** menu, choose **Verify** to verify that the new identifiers do not conflict with existing identifiers. For background on verification results, see “Verifying Renamed Identifiers” on page 8-2.
5. In the **File** menu, choose **Propagate Changes** to propagate the name change to source files. For background on propagation output, see “Propagation Output” on page 8-2.

Using the Application Namespace Tool

You generate a dictionary for the current project. You can rename identifiers, add business (pseudocode) names for identifiers, and provide descriptive names for constants.

Working with a Project Dictionary

The window for the Application Namespace tool consists of a Tokens pane, Names pane, Source pane, Modified Source pane, Conflicts pane, and Activity Log. You can hide a pane by clicking the close box in the upper righthand corner. Select the appropriate choice in the **View** menu to show the pane again.

Tokens Pane

The Tokens pane displays the tokens in the project in two views:

- The *report view* shows the original and modified names of tokens in a vertically scrolling list. Choose **Report** in the **View** menu to display the report view.

- The *list view* shows the current names of tokens in a horizontally scrolling list. Choose **List** in the **View** menu to display the list view.

Select a token to view the identifiers that contain it in the Names pane and its source code in the Source pane.

Restricting the Views To limit the scope of the display to tokens in a program rather than an entire project, choose the program in the **Scope** drop-down on the Application Namespace tool bar.

Note: You can also limit the display by applying a token filter. See “Setting Application Namespace Options” on page 8-8.

Use the buttons at the top of the Tokens pane to limit the display even further. Click:

- **All** to view all the tokens in the selected project or program.
- **Changed** to view only the modified tokens in the selected project or program.
- **Intact** to view only the unmodified tokens in the selected project or program.
- **Link** to view only the tokens contained in the identifier selected in the Names pane.

Tip: You can use **Link** with *any* of the other three buttons.

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

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

Renaming a Token To rename a token, select the token and choose **Edit** in the right-click menu. Enter the new name in the edit field for the token and press Enter. Application Namespace places a ✓ symbol next to a renamed token.To restore the original name of a token, select the token and choose **Restore** in the **Token** menu.

Adding Comments To add a comment for a token, select it and enter the comment in the comments pane for tokens.

Names Pane

The Names pane displays the original, modified, and business names of identifiers in the project. Select an identifier to view its declaration in the Source pane. Select a renamed identifier to view its modified declaration in the Modified Source pane.

Restricting the View Use the buttons at the top of the Names pane to limit the display of identifiers. Click:

- **All** to view all the identifiers in the selected project or program.
- **Changed** to view only the modified identifiers in the selected project or program.
- **Intact** to view only the unmodified identifiers in the selected project or program.
- **Link** to view only the identifiers that contain the token selected in the Tokens pane.

Tip: You can use **Link** with *any* of the other three buttons.

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

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

Renaming an Identifier To rename an identifier manually, select the identifier and choose **Edit** in the right-click menu. Enter the new name in the edit field for the identifier and press Enter. To map a renamed token to the identifiers that contain it, choose **Apply Mapping** in the **File** menu.

Tip: Choose **Automatic Mapping** in the **File** menu if you want the Application Namespace tool to perform the mapping automatically.

Application Namespace places a ✓ symbol next to a manually renamed identifier and a ✓ symbol next to a mapped identifier. To restore the original name of an identifier, select the identifier and choose **Restore** in the **Name** menu.

Verifying Renamed Identifiers To verify that renamed identifiers do not conflict with existing identifiers, choose **Verify** in the **File** menu. Application Namespace places a ⚔ symbol next to a renamed identifier that generates non-fatal conflicts and a ⚔ symbol next to a renamed identifier that generates fatal conflicts.

Tip: Select an offending identifier in the Names pane to compare its declaration with the declaration for the conflicting identifier in the Conflicts pane.

Propagating Identifier Name Changes To propagate identifier name changes to source files, choose **Propagate Changes** in the **File** menu. The Application Namespace tool lets you propagate name changes that generate non-fatal conflicts, but not name changes that generate fatal conflicts. For background on propagation output, see "Propagation Output" on page 8-2.

Tip: Choose **Files** in the **View** menu to view a list of source files that were modified.

Adding Comments To add a comment for an identifier, select it and enter the comment in the comments pane for identifiers.

Adding a Business Name for an Identifier To add a business name for an identifier, select the identifier and choose **Business Name** in the right-click menu. Enter the new name in the edit dialog for the identifier and click **OK**.

Source Pane

The Source pane displays the declaration for the identifier selected in the Names pane.

Modified Source Pane

The Modified Source pane displays the declaration for the renamed identifier selected in the Names pane.

Conflicts Pane

The Conflicts pane displays the declarations for the renamed identifier selected in the Names pane and any identifiers with which it conflicts.

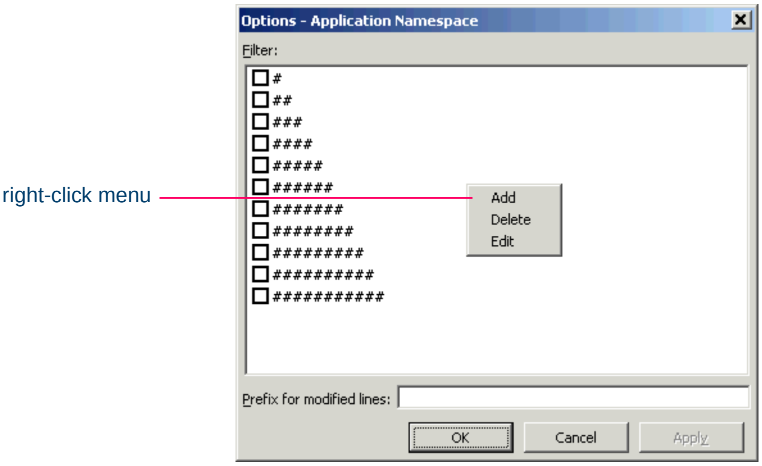
Setting Application Namespace Options

Use the Application Namespace Options window to specify a prefix for modified lines in source code and define a filter that excludes unwanted tokens from the dictionary. By default, the tool defines filters for numeric tokens.

To set application namespace options:

1. In the **View** menu, choose **Options**. The Application Namespace Options window opens (Figure 8-2).

Figure 8-2 *Application Namespace Options Window*



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

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

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

3 In the **Prefix for modified lines** field, enter the text you want to prepend to modified lines in source code.

4A Click **Apply** if you want to save your settings without dismissing the Options window.

4B Click **OK** if you want to save your settings and dismiss the Options window.

Tip: Application Namespace automatically applies the token filter when you click **Apply** or **OK**. You can delete filtered tokens from the application namespace database by choosing **Clean Up** in the **File** menu.

Refreshing a Project Dictionary

You can refresh a project dictionary with any changes you have made to the ATW repository by choosing **Refresh** in the **File** menu. You are prompted to confirm the refresh. Click **Yes**.

Generating and Exporting Application Namespace Reports

The Application Namespace tool offers three reports:

- The Tokens report lists the original and modified names of tokens, any comments you have added for the tokens, and the number of declarations in which the tokens appear. Choose **Tokens** in the **Report** menu to generate the Tokens report.
- The Names report lists the original, modified, and business names of identifiers, any comments you have added for the identifiers, and the number of declarations in which the identifiers appear. Choose **Names** in the **Report** menu to generate the Names report.
- The Files report lists original and modified source files in the selected project or program. Choose **Files** in the **View** menu to generate the Files report.

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

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

You can also export a report to the ATW WebGen tool by clicking **Create WebGen Page** in the **File** menu. All reports are exported to the page. The page is stored in the file *Workspace*\Projects *Project*\Webgen\Dictionary\Main.htm. Click the hyperlink for a report to view it in the browser.

What's Next?

Now that you know how to create an application namespace for your project, let's look at one final project analysis task — using Change Analyzer to understand how a change in the definition or usage of a data field might impact the rest of the application. The next chapter shows you how to do that.

Identifying Classes of Data Items

9

The *Change Analyzer* identifies the *class* of data items used to perform a business function in a legacy application. Among other uses, it lets you answer the kinds of “What if?” questions posed in the recent past by the industry-wide changes for Y2K, Zip+4, and the Euro dollar: “What if I change the type of this variable, or the length of this field — what *other* fields will I also have to change?”

Use change analysis results to prepare project plans and technical specifications. You can generate reports showing source entities that may require modification, lines of code affected, and the like.

Understanding Data Item Classification

Suppose your organization is considering adding support for a new currency and that you are going to have to expand the existing exchange rate data field from 9(5)V9(3) to 9(5)V9(6) to accommodate the currency. You will need to know the data fields that are affected in the database, intermediate fields that may contain or use the exchange rate field in calculations, and so forth.

Seed Fields

Use Change Analyzer to search for the exchange rate field. The object of the search is called a *seed field:*

- If the application uses field names like EX-RATE, EXCH-RATE, EXCHANGE-RATE, and RATE-OF-EXCHG, you would search for data names that contain *EXCH* or *RATE*.
- If you know that some fields already contain the required number of decimal positions and are interested only in those that don't, you might further limit the search by filtering on the PICTURE clause format of the variable, selecting only data items that have a format of 9(5)V9(3).
- You might limit the search even further by choosing fields that have a given initial value.
- If you know there are data fields that will meet the search criteria for name, but are not what you are looking for, you might set up a list of names to exclude, such as INTEREST-RATE and *PRORATE*.

Synonyms

When you execute the search, Change Analyzer returns not only the fields that match the search criteria but any *synonyms* for the fields. A synonym is a data field whose value is related to the value of the matched field — a field whose value is assigned by a MOVE or REDEFINE statement, for example. In these cases, if you increase the size of the matched field, you probably will have to increase the size of the synonym as well.

Seed Lists

As you examine the source code for seed fields and synonyms in Change Analyzer and determine whether a field is affected by a proposed change, you move the fields between *seed lists:*

- The *Working* list contains seed fields or synonyms that you are currently investigating.
- The *Affected* list contains seed fields or synonyms that you have determined would be affected by the proposed change.

- The *Clean* list contains seed fields or synonyms that you have determined would not be affected by the proposed change.

You can generate reports based on these lists, as well as reports showing the affected source code in context.

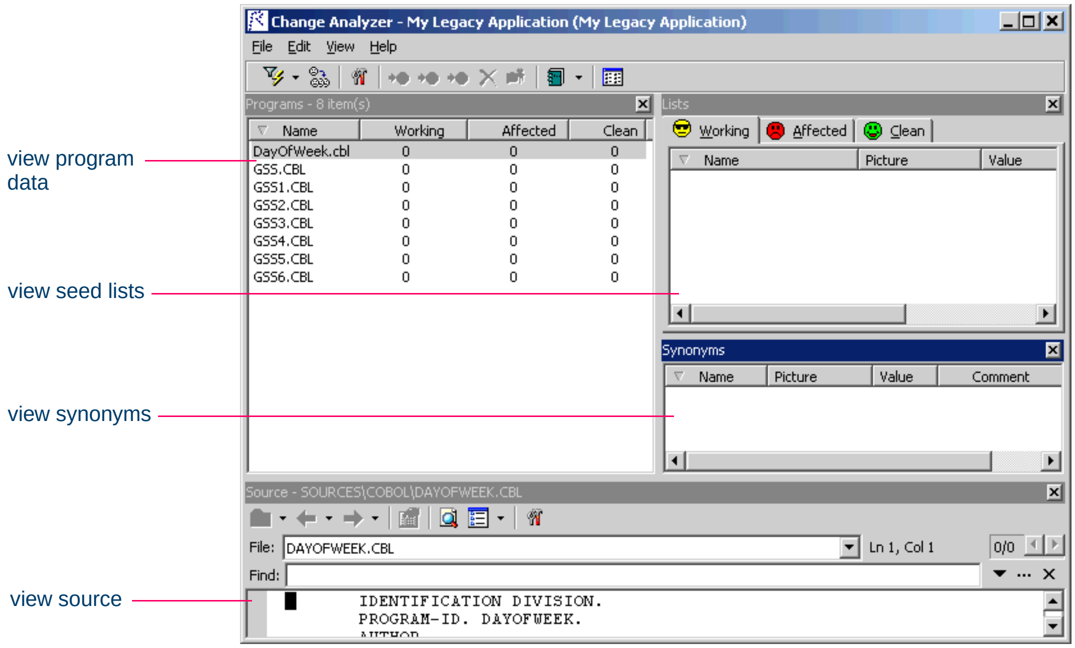
Getting Started in Change Analyzer

Change Analyzer is a complex tool that takes some getting used to. This section describes a sample use of the tool that should help you get oriented in it. Assume that you want to evaluate the impact of changing the YEAR field in your application.

To evaluate change impacts:

1. In the Repository Browser, select the project you want to analyze and choose **Field Change** in the **Analyze** menu. The Change Analyzer window opens (Figure 9-2).

Figure 9-1 *Change Analyzer Window*



Name	Working	Affected	Clean
DayOfWeek.cbl	0	0	0
GSS.CBL	0	0	0
GSS1.CBL	0	0	0
GSS2.CBL	0	0	0
GSS3.CBL	0	0	0
GSS4.CBL	0	0	0
GSS5.CBL	0	0	0
GSS6.CBL	0	0	0

Name	Picture	Value

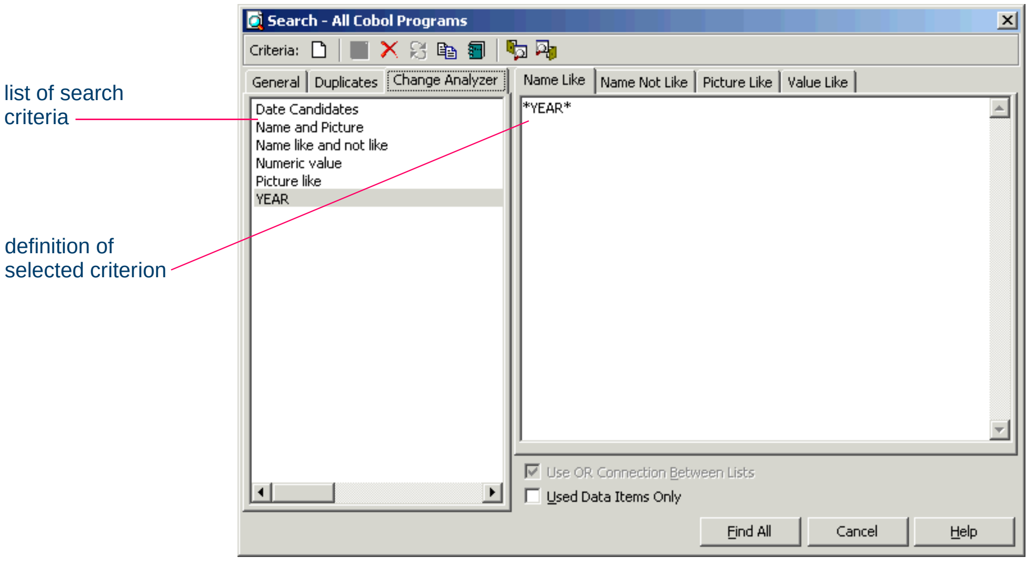
Name	Picture	Value	Comment

File: DAYOFWEEK.CBL
Ln 1, Col 1
0/0
Find:

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. DAYOFWEEK.
       AUTHOR
```

2 In the **File** menu, choose **Apply Filter:To Working**. The Search window opens (Figure 9-2).

Figure 9-2 *Search Window (After Step 4)*



list of search criteria

definition of selected criterion

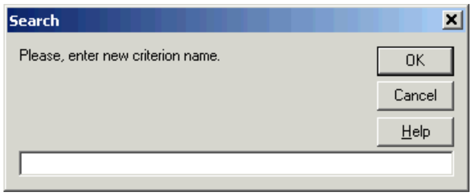
Date Candidates
Name and Picture
Name like and not like
Numeric value
Picture like
YEAR

YEAR

☑ Use OR Connection Between Lists
□ Used Data Items Only

3 In the Search window, click the Change Analyzer tab to view a list of recognized search criteria. Select a search criterion to view its definition in the tabs in the righthand portion of the window. Click the 🗋 button on the tool bar to define a new criterion. The New Search Criterion dialog opens (Figure 9-3).

Figure 9-3 *New Search Criterion Dialog*



Please, enter new criterion name.

4 In the text field, enter YEAR and click **OK**. The YEAR criterion is displayed in the list of recognized criteria in the Change Analyzer tab in the Search window.

5. In the Change Analyzer tab, select the YEAR criterion, enter *YEAR* in the Name Like tab, and click **Find All**. Change Analyzer returns the seed fields for the criterion in the Working list and their synonyms in the Synonyms pane. Select a program in the Programs pane to view its seed fields and their synonyms.

6. In the Working tab, select a seed field to navigate to its source in the Source pane. Examine the source to determine whether the field will be affected by the proposed change:

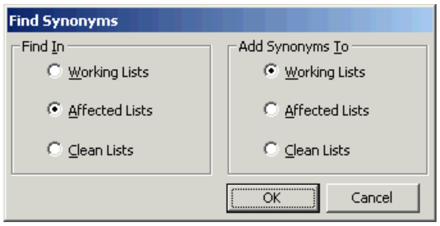
   - If the field will be affected by the change, choose **Move to Affected** in the **Edit** menu. Change Analyzer moves the field to the Affected tab.
   - If the field will not be affected by the change, choose **Move to Clean** in the **Edit** menu. Change Analyzer moves the field to the Clean tab.

7. Repeat step 6 for each seed field in the Working list.

8. Repeat steps 6-7 for each program in the Programs pane.

9. In the **File** menu, choose **Find Synonyms**. The Find Synonyms dialog opens (Figure 9-4).

Figure 9-4 *Find Synonyms Dialog*



Find Synonyms

Find In
○ Working Lists
● Affected Lists
○ Clean Lists

Add Synonyms To
● Working Lists
○ Affected Lists
○ Clean Lists

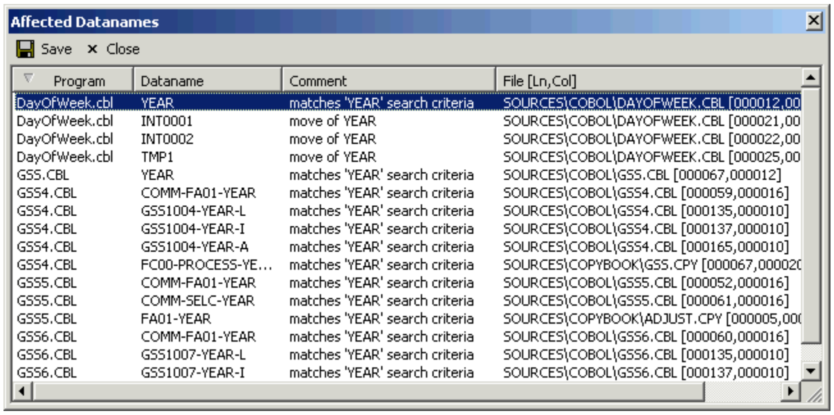
10. In the Find In pane, select **Affected Lists**. In the **Add Synonyms To** pane, select **Working Lists**. Click **OK**. Change Analyzer returns the synonyms for the affected fields in the Working list.

11. Repeat step 6 for each synonym in the Working list.

12. In the **File** menu, choose **Remove Unused:From Affected Lists** to remove fields that are declared but not used from the Affected list.

13 In the **File** menu, choose **Report:Affected Datanames**. Change Analyzer generates the Affected Datanames report (Figure 9-5).

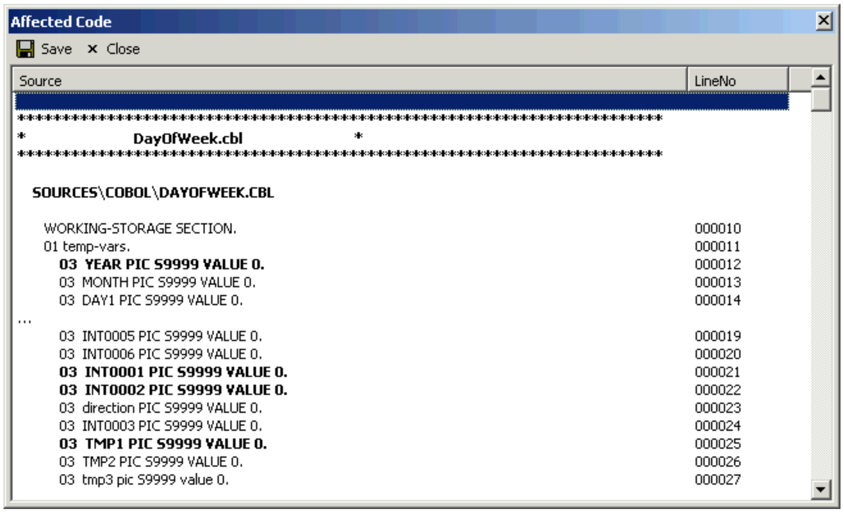
Figure 9-5 *Affected Datanames Report*



Program	Dataname	Comment	File [Ln,Col]
DayOfWeek.cbl	YEAR	matches 'YEAR' search criteria	SOURCES\COBOL\DAYOFWEEK.CBL [000012,00
DayOfWeek.cbl	INT0001	move of YEAR	SOURCES\COBOL\DAYOFWEEK.CBL [000021,00
DayOfWeek.cbl	INT0002	move of YEAR	SOURCES\COBOL\DAYOFWEEK.CBL [000022,00
DayOfWeek.cbl	TMP1	move of YEAR	SOURCES\COBOL\DAYOFWEEK.CBL [000025,00
GSS.CBL	YEAR	matches 'YEAR' search criteria	SOURCES\COBOL\GSS.CBL [000067,000012]
GSS4.CBL	COMM-FA01-YEAR	matches 'YEAR' search criteria	SOURCES\COBOL\GSS4.CBL [000059,000016]
GSS4.CBL	GSS1004-YEAR-L	matches 'YEAR' search criteria	SOURCES\COBOL\GSS4.CBL [000135,000010]
GSS4.CBL	GSS1004-YEAR-I	matches 'YEAR' search criteria	SOURCES\COBOL\GSS4.CBL [000137,000010]
GSS4.CBL	GSS1004-YEAR-A	matches 'YEAR' search criteria	SOURCES\COBOL\GSS4.CBL [000165,000010]
GSS4.CBL	FC00-PROCESS-YE...	matches 'YEAR' search criteria	SOURCES\COPYBOOK\GSS.CPY [000067,00002
GSS5.CBL	COMM-FA01-YEAR	matches 'YEAR' search criteria	SOURCES\COBOL\GSS5.CBL [000052,000016]
GSS5.CBL	COMM-SELC-YEAR	matches 'YEAR' search criteria	SOURCES\COBOL\GSS5.CBL [000061,000016]
GSS5.CBL	FA01-YEAR	matches 'YEAR' search criteria	SOURCES\COPYBOOK\ADJUST.CPY [000005,00
GSS6.CBL	COMM-FA01-YEAR	matches 'YEAR' search criteria	SOURCES\COBOL\GSS6.CBL [000060,000016]
GSS6.CBL	GSS1007-YEAR-L	matches 'YEAR' search criteria	SOURCES\COBOL\GSS6.CBL [000135,000010]
GSS6.CBL	GSS1007-YEAR-I	matches 'YEAR' search criteria	SOURCES\COBOL\GSS6.CBL [000137,000010]

14 In the **File** menu, choose **Report:Affected Code**. Change Analyzer generates the Affected Code report (Figure 9-6).

Figure 9-6 *Affected Code Report*



```
Source                                                                      LineNo

*****************************************************************************
*                     DayOfWeek.cbl                    *
*****************************************************************************

  SOURCES\COBOL\DAYOFWEEK.CBL

    WORKING-STORAGE SECTION.                                                000010
    01 temp-vars.                                                           000011
       03  YEAR PIC S9999 VALUE 0.                                          000012
       03  MONTH PIC S9999 VALUE 0.                                         000013
       03  DAY1 PIC S9999 VALUE 0.                                          000014
...
       03  INT0005 PIC S9999 VALUE 0.                                       000019
       03  INT0006 PIC S9999 VALUE 0.                                       000020
       03  INT0001 PIC S9999 VALUE 0.                                       000021
       03  INT0002 PIC S9999 VALUE 0.                                       000022
       03  direction PIC S9999 VALUE 0.                                     000023
       03  INT0003 PIC S9999 VALUE 0.                                       000024
       03  TMP1 PIC S9999 VALUE 0.                                          000025
       03  TMP2 PIC S9999 VALUE 0.                                          000026
       03  tmp3 pic S9999 value 0.                                          000027
```

Using Change Analyzer

You generate change impact information for the current project. The project must have been verified with the **Enable Data Element Flow** option set in the Verification tab of the Project Options window.

Note: For more information on verification, see *Preparing Projects* in the ATW document set.

Working with Change Impact Information

The window for the Change Analyzer consists of a Programs pane, Lists pane, Synonyms pane, Source pane, and Activity Log. You can hide a pane by clicking the close box in the upper righthand corner. Select the appropriate choice in the **View** menu to show the pane again.

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

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

Programs Pane

The Programs pane displays the programs in the selected project and the distribution of their seed fields in the seed lists. Select a program to view its seed fields in the List pane and their synonyms in the Synonyms pane. You also select a program to view its source in the Source pane.

Lists Pane

The Lists pane displays the Working, Affected, and Clean lists for the program selected in the Programs pane. Click the appropriate tab to view a list. Create projects from a list as described in "Creating Projects in Change Analyzer" on page 9-15, and generate reports as described in "Generating and Exporting Change Analyzer Reports" on page 9-16.

Each tab and its corresponding report contains the following columns:

Tip: Use the Report Control option to restrict the columns displayed in the tab, as described in step 9 on page 9-12.

- The Name column lists the seed fields or synonyms on the list.
- The Length column displays the length of the seed fields or synonyms on the list.
- The Normalized Picture column displays the normalized picture of the seed fields or synonyms on the list.
- The Picture column displays the format of the seed fields or synonyms on the list.
- The Value column displays the initial value of the seed fields or synonyms on the list.
- The Comment field describes the basis for including the seed field or synonym on the list. Select an entry in the Lists or Synonyms pane and choose **Comment** in the **Edit** menu to open a dialog where you can modify the comment.

Applying a Search Filter to a List To apply a search filter to a list, choose **Apply Filter:To *List*** in the **File** menu, then follow the instructions for performing searches in "Searching for Seed Fields in Change Analyzer" on page 9-13.

Selecting Fields Click a field on a list to select it. To select all the fields on a list, click the tab for the list and choose **Select All** in the **Edit** menu.

To select fields based on search patterns, click the tab for the list and choose **Select** in the **Edit** menu. A Select dialog opens, where you can enter the search patterns. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA)

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

Viewing Source Select a field to view its source in the Source pane.

Moving Fields between Lists To move fields between lists, select the fields, then choose **Move to *List*** in the **Edit** menu. For background information, see step 6 on page 9-5.

Removing Unused Fields from Lists To remove fields that are declared but not used from a list, choose **Remove Unused:From *List*** in the **File** menu.

Deleting Seed Fields from a List To delete a seed field from a list, select the field and choose **Delete** in the **Edit** menu. You are prompted to confirm the deletion. Click **Yes**.

Clearing Lists Search results are added to any previous results in a list. To clear the previous results from the list, choose **Clear:*List*** in the **File** menu. To clear all the lists, choose **Clear:All Lists** in the **File** menu.

Synonyms Pane

The Synonyms pane displays the synonyms for the field selected in the Lists pane. Select synonyms as described for list fields in "Selecting Fields" on page 9-8.

Viewing Source Select a synonym to view its source in the Source pane.

Moving Synonyms to Lists To move synonyms to a list, choose **Find Synonyms** in the **File** menu. The Find Synonyms dialog opens (Figure 9-4 on page 9-5), where you can select the list of seed fields to find synonyms for and the list to move the synonyms to.

Source Pane

The Source pane lets you browse the source code for the program selected in the Programs pane and the field or synonym selected in the Lists or Synonyms pane.

Usage is similar to that for the HyperView tool. For HyperView usage information, see *Analyzing Programs* in the ATW document set.

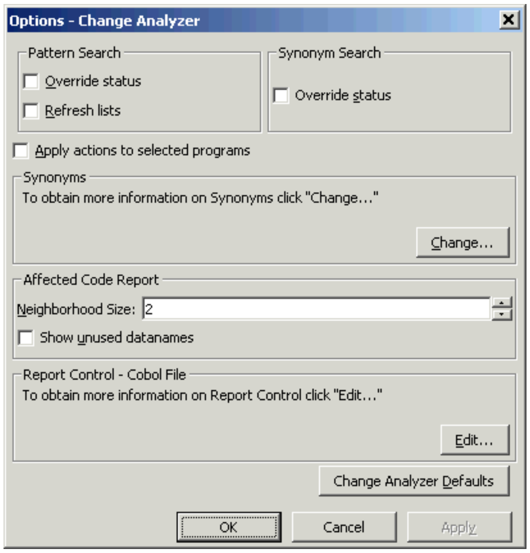
Setting Change Analyzer Options

Change Analyzer options control seed field and synonym pattern searches and the amount of information displayed in the Affected Code report.

To set Change Analyzer options:

1 In the **View** menu, choose **Options**. The Change Analyzer options window opens (Figure 9-7).

Figure 9-7 *Change Analyzer Options Window*



Options - Change Analyzer

Pattern Search
□ Override status
□ Refresh lists

Synonym Search
□ Override status

□ Apply actions to selected programs

Synonyms
To obtain more information on Synonyms click "Change..."

Affected Code Report
Neighborhood Size: 2
□ Show unused datanames

Report Control - Cobol File
To obtain more information on Report Control click "Edit..."

2 In the Pattern Search group box, choose any combination of:

- **Override status** if you want the search results to override the current list assignment of a field. If you do not select this option, Change Analyzer applies the search results only to fields not yet assigned to a seed list.
- **Refresh lists** if you want the search results to replace fields in the target seed list. If you do not select this option, Change Analyzer adds the search results to the fields already assigned to the target list.

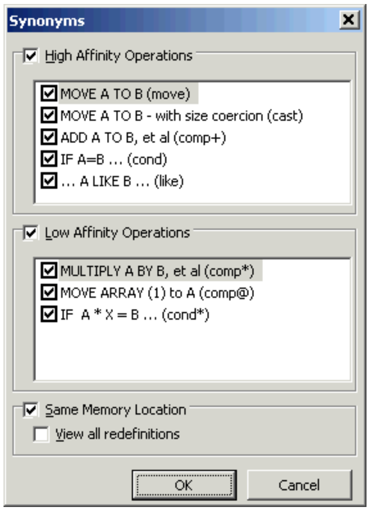
3 In the Synonym Search group box, choose **Override status** if you want the search results to override the current list assignment of a synonym. If you do not select this option, Change Analyzer applies the search results only to synonyms not yet assigned to a seed list.

4 Choose **Apply actions to selected programs** if you want Change Analyzer to act only on the programs selected in the Programs pane.

5 In the Affected Code Report pane, specify in the **Neighborhood Size** combo box the number of lines of unaffected code you want the report to display above and below the line of affected code. (The line of affected code is displayed in **bold** in the report.) Then choose **Show unused datanames** if you want the report to include unused data fields.

6 In the Synonyms group box, click **Change**. The Synonyms Options window opens (Figure 9-8).

Figure 9-8 *Synonym Options Window*



Synonyms

☑ High Affinity Operations

☑ MOVE A TO B (move)
☑ MOVE A TO B - with size coercion (cast)
☑ ADD A TO B, et al (comp+)
☑ IF A=B … (cond)
☑ … A LIKE B … (like)

☑ Low Affinity Operations

☑ MULTIPLY A BY B, et al (comp*)
☑ MOVE ARRAY (1) to A (comp@)
☑ IF A * X = B … (cond*)

☑ Same Memory Location

☐ View all redefinitions

7 Choose any combination of:

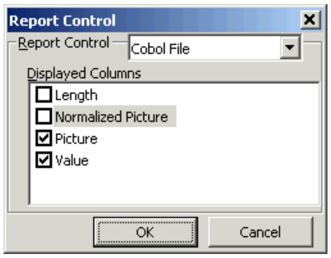
- **High-Affinity Operations** if you want Change Analyzer to identify as synonyms data fields that have a high affinity to seed fields — such as those used in assignment and addition operations. Then choose the types of high-affinity operations in the list box.

- **Low-Affinity Operations** if you want Change Analyzer to identify as synonyms data fields that have a low affinity to seed fields — such as those used in multiplication and division operations. Then choose the types of low-affinity operations in the list box.
- **Same Memory Location** if you want Change Analyzer to identify as synonyms data fields that share a memory location with seed fields. Then choose **View all redefinitions** to identify as synonyms all data fields that share a memory location with seed fields. If you do not select this option, Change Analyzer identifies as synonyms only data fields that are exact redefinitions of a seed field — such as one declared with the same size and offset.

Click **OK** to dismiss the Synonyms Options window and return to the Change Analyzer Options window.

8 In the Report Control group box, click **Edit**. The Report Control Options window opens (Figure 9-9).

Figure 9-9 *Report Control Options Window*



Report Control

Report Control: Cobol File

Displayed Columns

☐ Length
☐ Normalized Picture
☑ Picture
☑ Value

9 In the Report Control group box, select the type of file you are analyzing in the drop-down, then choose the columns you want to be displayed in Change Analyzer lists and their corresponding reports. Click **OK** to dismiss the Report Control Options window and return to the Change Analyzer Options window.

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

10A Click **Apply** if you want to save your settings without dismissing the Options window.

10B Click **OK** if you want to save your settings and dismiss the Options window.

Searching for Seed Fields in Change Analyzer

The Change Analyzer search facility (Figure 9-2 on page 9-4) contains two tabs:

- The General tab opens the HyperView advanced search facility.
- The Change Analyzer tab opens a scoped version of the advanced search facility for Change Analyzer.

Ordinarily, the scoped version of the tool should be sufficient for most searches. If you are already familiar with the advanced search facility, however, you may want to use it instead of the scoped tool. For HyperView usage information, see *Analyzing Programs* in the ATW document set.

Note: Change Analyzer returns only constants and literals found in the Procedure section of Cobol programs.

To search for seed fields:

1 In the **File** menu, choose **Apply Filter: To *List***. The Change Analyzer Search window opens (Figure 9-2 on page 9-4).

2 In the Search window, click the Change Analyzer tab. The Change Analyzer tab displays a list of recognized search criteria. You can define a new criterion as described in step 3 on page 9-4.

3 Select a criterion to edit its definition in the tabs in the righthand portion of the window. Each tab specifies a condition in the definition. The definition can consist of any combination of conditions.

For each condition, enter a list of patterns you want to match, one pattern per line. Select:

- Name Like to specify patterns that are like the name of the field you are searching for.
- Name Not Like to specify patterns that are unlike the name of the field you are searching for.
- Picture Like to specify patterns that are like the format of the field you are searching for.
- Value Like to specify patterns that are like the initial value of the field you are searching for.

You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).

4 Select **Use OR Connection Inside List** if you want the patterns listed in the tabs to be ORed. If you do not select this option, the patterns are ANDed.

5 Select **Use OR Connection Between Lists** if you want the conditions defined in the tabs to be ORed. If you do not select this option, the conditions are ANDed.

6 Select **Used Data Items Only** if you want the search results to consist only of fields that are used in the selected programs. If you do not select this option, search results include fields that are declared but not used.

7 In the Search in pane, click **Name Like** to specify a pattern that is like the name of the program or programs you want to limit the search to. You can use wildcard patterns allowed in LIKE statements by Visual Basic for Applications (VBA).

8 Click **Find All** to execute the search. Change Analyzer returns the seed fields for the selected criterion in the Working list and their synonyms in the Synonyms pane. The results are added to any previous results.

Note: Tool bar usage in the Search window is identical to that for the HyperView advanced search facility. For HyperView usage information, see *Analyzing Programs* in the ATW document set.

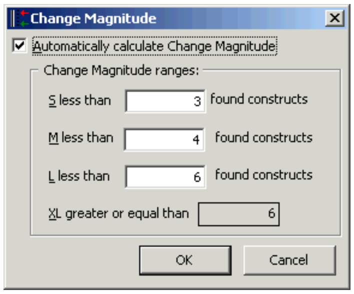
Creating Projects in Change Analyzer

You can create a project directly in Change Analyzer from the results of your analysis. The project contains only source files with fields in a Working, Affected, or Clean list.

To create a project from a list:

1 In the **File** menu, choose **Create Project:From *List***. The New Project dialog opens. Enter the name of the new project in the text field and click **OK**. The Change Magnitude window opens (Figure 9-10).

Figure 9-10 *Change Magnitude Window*



☑ Automatically calculate Change Magnitude

Change Magnitude ranges:

S less than 3 found constructs

M less than 4 found constructs

L less than 6 found constructs

XL greater or equal than 6

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

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

If you want Cobol source files with 6 to 10 fields in a list 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 **O**K. For more information on change magnitudes, see “Specifying the Change Magnitude for a Source File” on page 2-4.

Generating and Exporting Change Analyzer Reports

In the **File** menu, choose **Report:*Report*** to generate Change Analyzer reports:

- The Affected Datanames report (Figure 9-5 on page 9-6) lists the fields in the Affected list. In addition to the information displayed in the Affected list, the report displays the program that uses the field, the source file in which the field is declared, and the line and column numbers of the declaration.
- The Possible Datanames report lists every field in the Working, Affected, and Clean lists. In addition to the information displayed in the Affected Datanames report, the Possible Datanames report displays the disposition of fields in seed lists.
- The Clean Datanames report lists the fields in the Clean list. The report displays the same information as the Affected Datanames report.
- The Affected Code report (Figure 9-6 on page 9-6) displays the source code for the fields in the Affected list in both the program that uses the field and the source file in which the field is declared. The line of affected code is displayed in **bold**, surrounded by the number of lines of neighboring code you specified in step 5 on page 9-11.
- The Metrics report displays the percentage of declarations and uses of affected fields in a program.

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

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

You can also export a report to the ATW WebGen tool by clicking **Create WebGen Page** in the **File** menu. All reports are exported to the page. The page is stored in the file *Workspace*\Projects

\Project\\Webgen\\Diagrammer\\Main.htm. Click the hyperlink for a report to view it in the browser.

What’s Next?

That completes your tour of the ATW project analysis tools! Now you’re ready to start using the program analysis tools — collectively called *HyperView* — to view programs interactively and perform program analysis in stages. *Analyzing Programs* in the workbench documentation set describes HyperView.

Glossary

Activity Log

The Activity Log is a chronological record of your activities in the current Asset Transformation Workbench (ATW) session.

ADABAS

ADABAS is a Software AG relational DBMS for large, mission-critical applications.

Animator

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.

API

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

applet

See Java applet.

Application Analyzer

Application Analyzer is a set of non-invasive interactive tools used to analyze and document legacy systems.

Application Architect

Application Architect uses advanced algorithms to partition code into new components and perform Dead Code Elimination.

Application Namespace tool

The Application Namespace tool creates a conveniently organized dictionary that helps you navigate through your system's terminology and modify it as necessary.

Application Partitioner

Application Partitioner identifies legacy subsystems and partitions them into self-contained projects based on an analysis of repository contents.

Application Profiler

Application Profiler, consisting of WebGen and Profiler, generates a set of HTML views of a legacy application based on the object model created in a previous analysis.

AS/400

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

Asset Transformation Workbench (ATW)

Asset Transformation Workbench (ATW) is a suite of PC-based software products for analyzing, re-architecting, and transforming legacy applications.

Batch Application Viewer

Batch Application Viewer performs low-level analysis of batch processes.

batch refresh

The batch refresh feature lets you register and verify source files in batch mode. Other utilities packaged with the feature let you analyze application complexity, run WebGen, and, if you are licensed to use the Application Architect product, perform Dead Code Elimination.

Bird's Eye pane

The Bird's Eye pane works with the HyperView Source pane to let you quickly identify the location of a code construct relative to the entire program.

BMS

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

BSTR

BSTR is a Microsoft format for transferring binary strings.

business rule

A business rule is a named container that identifies and documents code segments according to their business function. Business rules encapsulate an application's business logic, making the application easier to understand, document, maintain, and test.

Business Rule Manager

Business Rule Manager lets you generate business rules from code segments extracted manually from source or autodetected.

Callie pane

The HyperView Callie pane displays a diagram that shows the flow of control between paragraphs or procedures in a program.

CDML

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

Change Analyzer

Change Analyzer identifies the class of data items used to perform a business function in a legacy application. Among other uses, it lets you answer the kinds of "What if?" questions posed in the recent past by the industry-wide changes for Y2K, Zip+4, and the Euro dollar: "What if I change the type of this variable, or the length of this field — what *other* fields will I also have to change?"

CICS

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

Clipper

The HyperView Clipper tool lets you create lists of candidates for business rule extraction, event injection, and other tasks. Each list captures the results of a different stage of your analysis and serves as input for subsequent tasks.

Cobol

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

COM

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

complexity

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

component

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

Component Maker

The HyperView Component Maker tool lets you “slice out” components from legacy applications — not only component executables but associated Cobol copybooks, PL/I includes, and Natural data areas as well.

computation-based component extraction

Computation-based component extraction lets you build a component that contains all the code necessary to calculate the value of a variable at a particular point in a program — the value of a DayOfTheWeek variable, for example, where it is used to populate a report attribute or screen.

Configuration Manager

Configuration Manager is a tool used to enable Asset Transformation Workbench (ATW) products and configure your workbench for the tools, programming languages, legacy dialects, and character sets in use at your site.

construct

A construct is an item in the parse tree for a source file — a section, statement, condition, variable, or the like. A variable, for example, can be related in the parse tree to any of three other constructs — a declaration, a dataport, or a condition. You view the parse tree for a source file in the HyperView Context pane.

Context pane

The HyperView 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.

copybook

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

CORBA

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

CSD file

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

database schema

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

dataport

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

DB/2

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

DBCS

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

DBMS

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

DDL

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

Dead Code Elimination

Dead code elimination is a type of component extraction that removes unused ("dead") code from a legacy application.

decision resolution

Decision resolution lets you identify and resolve dynamic calls and other relationships that the parser cannot resolve from static sources.

Diagrammer

Diagrammer lets you view the relationships between the objects in a project interactively — programs, files, DDL, Java, screen maps, and more. These relationships describe the ways in which application objects interact. Compare Quick Diagrammer.

DMS

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

domain-based component extraction

Domain-based component extraction "specializes" a program based on the values of one or more variables. The specialized program is typically intended for reuse "in place" — in the original application but under new external circumstances.

DPS

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

ECL

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

effort

Effort is an estimate of the time it will take to complete a task related to a project, based on weighted values for selected complexity metrics.

EJB

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

EMF

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

entity

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

entry point isolation

Entry point isolation extracts a component that contains only the functionality and data definitions required for invocation from the selected entry point.

event injection

Event injection is a type of component extraction that adapts a legacy program to asynchronous, event-based programming models.

Execution Path pane

The HyperView Execution Path pane displays a hierarchical view and diagram of the conditions that determine the flow of control in a program.

external subroutine extraction

External subroutine extraction is a type of structure-based component extraction that replaces a single internal subroutine in a Natural program with an external subroutine.

FCT

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

Flowchart pane

The HyperView Flowchart pane displays a diagram of the flow of control between statements in a paragraph or procedure.

Global Data Flow tool

The HyperView Global Data Flow tool performs low-level analysis of program data flows.

HTML

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

HyperView

HyperView is a set of program analysis tools that let you analyze legacy programs interactively, by examining synchronized, complementary views of the same information — source, context, impacts, and so forth.

IDL

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

IDMS

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

Impact pane

The HyperView Impact pane displays a hierarchical view and diagram of the *impact trace* for a program variable. An impact trace describes how data items interact with each other in a program — exchange values, use each other in computations, and so forth.

Impact Report pane

The HyperView Impact Report pane shows the flow of data from a startup item to every data item that would be affected by its modification. The report is organized in hierarchical form according to the depth of the affected item.

IMS

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

Java

Java is a high-level object-oriented programming language developed by Sun Microsystems.

Java applet

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

JavaBeans

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

JCL

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

JDBC

JDBC stands for Java Database Connectivity, a standard for accessing diverse database systems using the Java programming language.

job

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

job dependencies

Batch Application Viewer treats jobs as dependent if one writes to a dataset and the other reads from the same dataset. Occasionally, you may want to define dependencies between jobs based on other criteria — administrative needs such as scheduling, for example.

logical component

A logical component is an abstract repository object that gives you access to the source files that comprise a component.

MFS

MFS stands for Message Format Service, a method of processing IMS input and output messages.

Missing Copybooks Resolution tool

The Missing Copybooks Resolution tool resolves undefined variables in missing copybooks for Cobol programs verified with the relaxed parsing option.

Model Reference pane

The HyperView Model Reference pane displays the parse tree metamodel in text and diagram form.

name-based partitioning

Name-based partitioning is a partitioning algorithm that assigns source files to projects based on text matching of source file names with specified patterns.

Natural

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

object model

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

object-oriented programming

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

ODBC

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

orphan

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

Orphan Analysis tool

The Orphan Analysis tool lets you analyze and resolve orphan*s*.

parser

The Asset Transformation Workbench (ATW) parser defines the object model and parse tree for a legacy application.

parse tree

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

PCT

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

PL/I

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

PL/I Call Diagrammer

PL/I Call Diagrammer performs low-level analysis of PL/I programs. Use it to examine call flows for internal procedures that the Diagrammer is unable to model.

profile

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

Profiler

Profiler is a Web server-based tool that offers company-wide access to profiles of any repository in your organization. It gives managers, business analysts, testers, and customer support personnel convenient, browser-based access to analyzed legacy code.

project

A project is a logical subdivision of a workspace. You might have a project for the batch portion of your application and another project for the online portion, for example. You can also use a project to collect items for discrete tasks — all the source files affected by a change request, for example.

QSAM

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

Quick Diagrammer

The Quick Diagrammer tool lets you view relationships for selected objects only, rather than an entire project. Compare Diagrammer.

refactoring

Refactoring translates a program into a component with the same functionality and control flow, but a simpler syntax structure.

reference reports

Asset Transformation Workbench (ATW) offers three related reports that you can use to identify missing or unneeded program elements in application source: an unresolved report, an unreferred report, and a cross-reference report.

relationship

The relationships between entities in the repository model for a legacy application describe the ways in which the elements of the application interact.

relationship-based partitioning

Relationship-based partitioning is a partitioning algorithm that assigns source files to projects based on the extent to which the source files are related. If two source files reference the same copybook, for example, they can be regarded as "tightly related," at least as compared with source files that do not reference the same copybook.

relationship weight

A relationship weight determines the importance of that relationship in calculating the connection between source files in relationship-based partitioning.

relaxed parsing

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

repository

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

Repository Browser

The Asset Transformation Workbench (ATW) Repository Browser displays the contents of the current workspace.

Resource Retriever

The Resource Retriever tool lets you identify and restore missing CICS file connectors and transactions in file control tables (FCT) and program control tables (PCT) for Cobol and PL/I programs.

Rules pane

The HyperView Rules pane lets you create business rules from code segments extracted manually from source or autodetected. You can also create business rules from candidates listed in Clipper.

schema

See database schema.

scope

The scope of a diagram determines the objects and relationships it displays. See Diagrammer.

seed field

A seed field is the object of a Change Analyzer search for the class of data items that need to be changed.

Source pane

The HyperView Source pane displays view-only source code for the selected file and included files.

SQL

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

structure-based component extraction

Structure-based component extraction is a type of component extraction that builds a component from a range of inline code — Cobol paragraphs, for example.

synonym

A synonym is a data field whose value is related to the value of the matched seed field — a field whose value is assigned by a MOVE or REDEFINE statement, for example.

system program

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

TIP

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

token

In the Application Namespace tool, 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.

transaction

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

Unknown Statements Resolution tool

The Unknown Statements Resolution tool resolves incorrect or unsupported statements in Cobol programs verified with the relaxed parsing option.

User Interface tool

The User Interface tool lets you analyze the interaction between legacy screens and program logic, and generate an HTML or Java GUI based on the interaction.

VALTAB

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

VSAM

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

WebGen

WebGen generates HTML views of the repositories on your workstation. You can publish the views to Profiler, where they can be accessed by any member of your organization with a browser.

workspace

A workspace is a named container for an application or a portion of an application. Workspaces can be divided into projects.

XML

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

Bibliography

- *IBM Asset Transformation Workbench v1.1 Getting Started (SC31-6877-00)*
- *IBM Asset Transformation Workbench v1.1 Preparing Projects (SC31-6879-00)*
- *IBM Asset Transformation Workbench v1.1 Analyzing Projects (SC31-6880-00)*
- *IBM Asset Transformation Workbench v1.1 Analyzing Programs (SC31-6878-00)*
- *IBM Asset Transformation Workbench v1.1 Profiling Projects (SC31-6881-00)*
- *IBM Asset Transformation Workbench v1.1 Creating Components (SC31-6876-00)*
- *IBM Asset Transformation Workbench v1.1 Parser Reference (SC31-6882-00)*
- *IBM Asset Transformation Workbench v1.1 Architecture Reference (SC31-6898-00)*

Notices

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

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

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

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

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

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

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

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

IBM Corporation

P.O. Box 12195, Dept. TL3B/B503/B313

3039 Cornwallis Rd.

Research Triangle Park, NC 27709-2195

U.S.A.

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

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

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

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

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

Trademarks

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

Table 1. Trademarks

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

The following terms are trademarks of other companies:

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

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

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

Index

A

Application Namespace tool
- adding business names for identifiers 8-7
- generating and exporting reports 8-9
- getting started 8-3
- overview 8-1
- propagating name changes 8-7
- refreshing a dictionary 8-9
- renaming identifiers 8-6
- renaming tokens 8-5
- setting options 8-8
- verifying renamed identifiers 8-7

B

Batch Application Viewer
- copying diagrams 4-11
- creating job dependencies 4-9
- editing color scheme 4-8
- exporting diagrams 4-11
- generating batch application information 4-2
- generating diagrams 4-7
- overview 4-1
- printing diagrams 4-11
- renaming objects 4-10
- searching in table views 4-7
- working with table views 4-5

C

Change Analyzer
- creating a project 9-15
- generating and analyzing reports 9-16
- getting started 9-3
- moving synonyms to lists 9-9
- overview 9-1
- searching for seed fields 9-13
- setting options 9-10

Complexity Metrics tool
- exporting reports 7-7
- generating complexity metrics 7-2
- overview 7-1
- supported complexity metrics 7-8

CRUD report 6-1

D

data operations 6-1, 6-3

Diagram Editor
- copying a diagram 2-22
- editing diagram layout 2-17
- editing objects 2-19
- editing relationships 2-20
- exporting a diagram 2-23
- opening a diagram 2-14
- printing a diagram 2-23
- saving a diagram 2-22
- searching for objects 2-15
- selecting objects 2-17

Diagrammer
- copying diagrams 2-6
- defining diagram scopes 2-9
- editing color scheme of diagrams 2-7
- exporting diagrams and reports 2-13, 2-26
- generating diagrams 2-4
- overview 2-1
- printing diagrams 2-6
- Quick Diagrammer 2-23
- searching for objects 2-13

diagrams
- batch application 4-1
- editing 2-14
- global data flows 3-1
- PL/I call flows 5-1
- relationship flows 2-1

dictionary
- adding business names for identifiers 8-7
- generating and exporting reports 8-9
- overview 8-1
- propagating name changes 8-7
- refreshing 8-9
- renaming identifiers 8-6
- renaming tokens 8-5
- verifying renamed identifiers 8-7

E

Effort Estimation tool
- exporting reports 7-7
- generating effort estimation statistics 7-3
- overview 7-1
- specifying change magnitudes 7-6
- supported complexity metrics 7-8

G

Global Data Flow tool
- copying diagrams 3-1, 3-13
- exporting diagrams 3-13
- generating diagrams 3-5
- printing diagrams 3-13
- setting options 3-9

I

IMS port analysis 6-3

J

JCL
- verification requirements for IMS port analysis 6-4

L

legacy estimation tools
- exporting reports 7-7
- generating complexity metrics 7-2
- generating effort estimation statistics 7-3
- overview 7-1
- specifying change magnitudes 7-6
- supported complexity metrics 7-8

P

PL/I Call Diagrammer
- copying diagrams 5-5
- exporting diagrams 5-6
- generating diagrams 5-2
- overview 5-1
- printing diagrams 5-5
- setting options 5-3

Program Control Table (PCT) 6-4

project, creating in Change Analyzer 9-15

Q

Quick Diagrammer
- changing the diagram scope 2-24
- defining scopes 2-25
- exporting multiple diagrams 2-26
- generating diagrams 2-23
- including source files in a project 2-26
- mirroring the diagram scope 2-25
- overview 2-23

R

relationship flow diagrams
- copying 2-6
- defining scopes 2-9
- editing color scheme 2-7
- exporting 2-13, 2-26
- generating 2-4
- overview 2-1
- printing 2-6
- searching for objects 2-13

reports, diagram-based 2-13, 2-26

S

scope
- defining for application namespace 8-5
- defining in relationship flow diagram 2-9, 2-25

synonyms 9-2

T

tokens 8-1

Product Number: 5724-L54

(1P) P/N:5724-L54

SC31-6880-00