Scenario Manager User Guide

Release 6.2.5
April 2015

ORACLE®
FINANCIAL SERVICES

ORACLE®

Scenario Manager User Guide

Release 6.2.5
April 2015

Part Number: E62969_09

Contents

List of Figures

List of Tables

List of Tables

About this Guide

This guide explains how to use all of the features and functionality of the Scenario Manager application (referred to herein as the Scenario Manager).

This chapter contains the following sections:

- Who Should Use this Guide
- Scope of this Guide
- How this Guide is Organized
- Where to Find More Information
- Conventions Used in this Guide

Who Should Use this Guide

The *Scenario Manager User Guide,* Release is designed for use by the Data Miners. The Data Miner is responsible for adjusting the pattern logic or thresholds of existing scenarios and employing data mining techniques to define new patterns and scenarios. This includes both mining data to detect new scenarios and mining data to identify patterns that indicate known scenarios.

Scope of this Guide

This guide defines the features, functionality, and step-by-step procedures for features within the Scenario Manager including its tools:

- DataSet Editor
- Sequence Scenario Editor
- Rule Scenario Editor
- Link Analysis
- Job Editors
- Threshold Manager
- Augmentation

How this Guide is Organized

The *Scenario Manager User Guide,* contains the following chapters:

- Chapter 1, *About Scenario Manager*, defines the features and functionality of the Scenario Manager.

- Chapter 2, *Common Features*, defines the features, functionality, and processes that are common throughout the Scenario Manager.

- Chapter 3, *DataSet Editor*, defines the DataSet Editor features and functionality within the Scenario Manager.

- Chapter 4, *Sequence Scenario Editor*, defines the Sequence Scenario Editor features and functionality within the Scenario Manager.

- Chapter 6, *Rule Scenario Editor*, defines the Rule Scenario Editor features and functionality within the Scenario Manager.

- Chapter 5, *Link Analysis*, defines the Link Analysis feature and functionality of each of the dataset editors within the Scenario Manager.

- Chapter 7, *Threshold Manager*, defines the Threshold Manager features and functionality within the Scenario Manager.

- Chapter 8, *Augmentation*, defines the augmentation tool features and functionality within the Scenario Manager.

- Chapter 9, *Job Editors*, defines the Job Editors' features and functionality within the Scenario Manager.

- Appendix A, *Scenario Design Guidelines*, describes technical guidelines for the design of scenarios and their components.

- Appendix B, *Naming Conventions*, provides naming conventions for creating datasets and scenario components.

- Appendix C, *Scenario Manager to User Interface Mapping*, provides an example of mapping elements of a scenario and pattern roles to the Web Application User Interface (UI).

- The *Index* provides an alphabetized cross-reference list that helps you locate information quickly.

Where to Find More Information

For more information about Oracle Financial Services products, refer to the following documents:

- *Administration Guide*

- *Installation Guide-Stage 3*

- *Alert Management User Guide*

To find more information about Oracle Financial Services and our complete product line, visit our web site at www.oracle.com/financialservices.

Conventions Used in this Guide

Table 1 lists the conventions of this guide.

Table 1. Conventions used in this Guide

Convention	Meaning
Italics	• Names of books, chapters, and sections as references • Emphasis
Bold	• Object of an action (menu names, field names, options, button names) in a step-by-step procedure • Commands typed at a prompt • User input
Monospace	• Directories and subdirectories • File names and extensions • Process names • Code sample, including keywords and variables within text and as separate paragraphs, and user-defined program elements within text
<Variable>	• Substitute input value

About Scenario Manager

This chapter provides a high-level description of the dataset editor within the Scenario Manager that is used for detection. This chapter focuses on the following topics:

- Introduction
- Accessing the Scenario Manager

Introduction

Firms that handle large quantities of data want to analyze data for specific behaviors. The Scenario Manager enables a business to define how these behaviors manifest themselves in their data and to detect possible occurrences of these behaviors.

Elements Used in the Detection Process

Our software uses sophisticated pattern recognition techniques to identify behaviors of interest, or *scenarios*, which are indicative of potentially interesting behavior. These behaviors can take multiple representations in a firm's data. A *pattern* is a specific set of detection logic and match generation criteria for a particular type of behavior. Using the Scenario Manager, you can create one or more patterns to implement a scenario. Patterns access data by datasets, which are views that are defined in the Scenario Manager.

By applying patterns to business data, the application detects data records that satisfy the logic and criteria, which a pattern defines. If one or more data records meet a scenario's pattern of behavior, then a *match* is created. Records that contribute to the exhibition of the behavior are associated to the match as *matched records* and display in the Web Application as building blocks. In addition, patterns store *bindings,* which are relevant data that display as highlights in the Web Application. The entity that is responsible for the behavior of interest is considered the *focus* of the match. For example, a focus can be an account, execution, client bank, employee, investment advisor, or group of accounts.

The application generates an *alert* to package one or more matches for analysis and action in the Web Application. If Oracle Financial Services discovers multiple matches for the same focus, processing combines the matches to create a single alert and facilitate the analysis of the found behavior.

To facilitate analysis of alerts, the application groups scenarios that represent related business problems into *scenario classes*.

Defining the Detection Tools

The Scenario Manager includes tools that you can use to perform data detection. Business firms use these tools to locate and analyze behaviors of interest to detect unusual activity within their business. These tools include the following:

- **DataSet Editor:** Select data sources, filter data, add derived attributes, and join data sources to create virtual tables of data. This data enables you to search for and analyze behaviors of interest.
- **Sequence Scenario Editor:** Define a behavior that consists of events in a predetermined order. You can use these events to thread multiple data streams together to find behaviors of interest.

- **Rule Scenario Editor:** Access pattern elements that you need to create when using this editor for detecting behaviors of interest.

- **Link Analysis:** Use the data mining technique to find networks of entities related in an interesting or unusual manner.

- **Job Editors:** Define a collection of instructions for executing the scenarios defined in the tools to detect and generate alerts.

- **Threshold Manager:** Add a threshold or threshold set to a pattern or dataset to define tunable values. A threshold provides flexibility for changing variable values. Changes to values can be made by the Threshold Editor in the Administration Tools without having to change the values defined in the Scenario Manager at the dataset or pattern level.

  Note: You cannot use the Admin Tools to create threshold sets.

- **Augmentation:** Add additional data to existing matches (such as account number or the source of a deposit) so that a firm can analyze their matches in familiar terms that are understandable (that is, adding the name of a trader rather than displaying the TRADER_ID attribute in the display).

Accessing the Scenario Manager

To access the Scenario Manager, you provide login information and then select a domain in which to work. You can configure the Scenario Manager to display an editor other than the default DataSet Editor. Contact your system administrator to change the editor that displays when you open the Scenario Manager.

To access the Scenario Manager, follow these steps:

1. Execute the `kddstart.bat` file.

   The system displays the Login dialog box.

2. Type your user ID and password (with a Data Miner role and associated privileges) in the **Login** and **Password** fields.

3. Click **Login** to display the Select a Domain dialog box.

4. Select the desired domain in the Domains list.

5. Click **OK**.

   By default, the DataSet Editor dialog box displays the datasets for the domain in the Attributes area. Refer to the section *About the DataSet Editor*, on page 9, for more information about the editor.

Common Features

This chapter defines the features, functionality, and processes that are common throughout the Scenario Manager application. This chapter focuses on the following topics:

- Common Menu Bar Options
- Keyboard Shortcut Keys
- Select DataSet Dialog Box
- Status Bar Messages

Common Menu Bar Options

The menu bar provides access to all menus within the Scenario Manager, specifically the Windows and Help menus. The following definitions apply to *all* editors within the Scenario Manager.

Windows Menu

The **Windows** menu contains the following options, in the order they are displayed on the menu:

- **DataSets:** Opens the DataSet Editor (refer to Chapter 3, *DataSet Editor*, on page 9, for more information).
- **Network:** Opens the Link Analysis Network Definition Editor (refer to Chapter 5, *Link Analysis*, on page 67, for more information).
- **Rule Patterns:** Opens the Rule Scenario Editor (refer to Chapter 6, *Rule Scenario Editor*, on page 101, for more information).
- **Sequence Patterns:** Opens the Sequence Scenario Editor (refer to Chapter 4, *Sequence Scenario Editor*, on page 39, for more information).
- **Thresholds:** Opens the Threshold Manager (refer to Chapter 7, *Threshold Manager, on page 131*, for more information).
- **Jobs:** Opens the following job editors:
    - Rule Matcher
    - Sequence Matcher
    - Link Analysis

    (Refer to Chapter 9, *Job Editors*, on page 161, for more information.)

Help Menu

The **Help** menu contains the **About** option, which provides the following information:

- Software version

- Build date of software

- Current domain

- Your free and total bytes of memory

- Java Virtual Machine version (JVM)

Keyboard Shortcut Keys

Table 2 lists the keyboard shortcut that the Scenario Manager provides to access or perform many functions.

Table 2. Keyboard Shortcut Keys and Functions

Shortcut Key	Function
Ctrl+N	Create New Scenario
Ctrl+O	Open Existing Scenario
Ctrl+S	Save
Ctrl+W	Close
Ctrl+Q	Exit
Ctrl+E	Export
Ctrl+G	Generate Report
Ctrl+I	Import
Ctrl+X	Cut
Ctrl+C	Copy
Ctrl+V	Paste
CTRL-click	Select multiple items from list
SHIFT-click	Select a range of items in a list

Although each of the functions is not available throughout all of the tools within the Scenario Manager, these shortcut keys are consistent if the function is active.

Select DataSet Dialog Box

Each editor of the Scenario Manager provides a dataset that you can use to perform a specific function. For example, when creating a scenario, you must always select each dataset against which to run the detection pattern.

To open a Select Dataset dialog box, follow these steps:

1. From the **Dataset** menu of the Dataset Editor, select **Open**.

   The Select Dataset dialog box displays.

You can select each dataset in the Select DataSet dialog box (Figure 1).
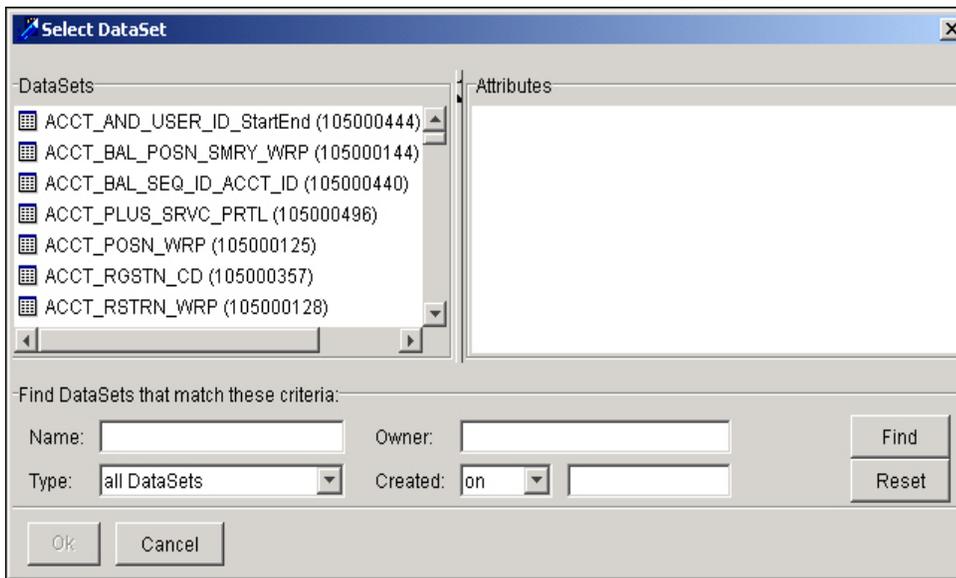
Figure 1.   Select DataSet Dialog Box

Table 3 describes the components of the Select DataSet dialog box.

Table 3. Select DataSet Dialog Box Components

Component	Function	Reference
DataSets	Lists all available datasets, based on the specified domain.	
Attributes	Displays the attributes associated with the selected dataset in the DataSets area	
Find DataSets	Enables you to define the search criteria for a dataset as follows: ● **Name**: Dataset name. ● **Owner**: Owner of the dataset. ● **Type**: 　■ **All DataSets**: Search all datasets. 　■ **Realized DataSets**: Search for realized or *derived* datasets. **Note:** Selecting either All DataSets or Realized DataSets displays the same output. 　■ **Wrapper DataSets:** Search for wrapper datasets. ● **Created**: This specifies whether to search for datasets before, on, or after the date you enter in the Date Entry field, as mm/dd/yyyy.	*Chapter 3, DataSet Editor*, on page 11.

Searching for and Selecting a DataSet

To search for and then select a DataSet, follow these steps:

1. Specify any of the following search criteria in the Select DataSet dialog box: **Name**, **Type**, **Owner**, or **Created**.

2. Click **Find** to begin the search.

   A list of datasets that meet the search criteria display in the DataSets area, if the search finds a match.

3. Select dataset name from the DataSets area.

   The system highlights the dataset and its attributes display in the Attributes area.

4. Click **OK**.

   Note: When selecting a dataset in the Sequence Scenario or Rule Scenario Editor, the Dataset selector presents only datasets that do not contain thresholds from other scenarios.

Status Bar Messages

Each tool in the Scenario Manager displays instructional text in the status bar at the bottom of an editor window. These instructions assist you in using the specific editor by providing the next step to take in the procedure you are performing.

A sample Sequence Scenario Editor with a message in the status bar is shown (Figure 2): `Please enter a short name for the scenario.`
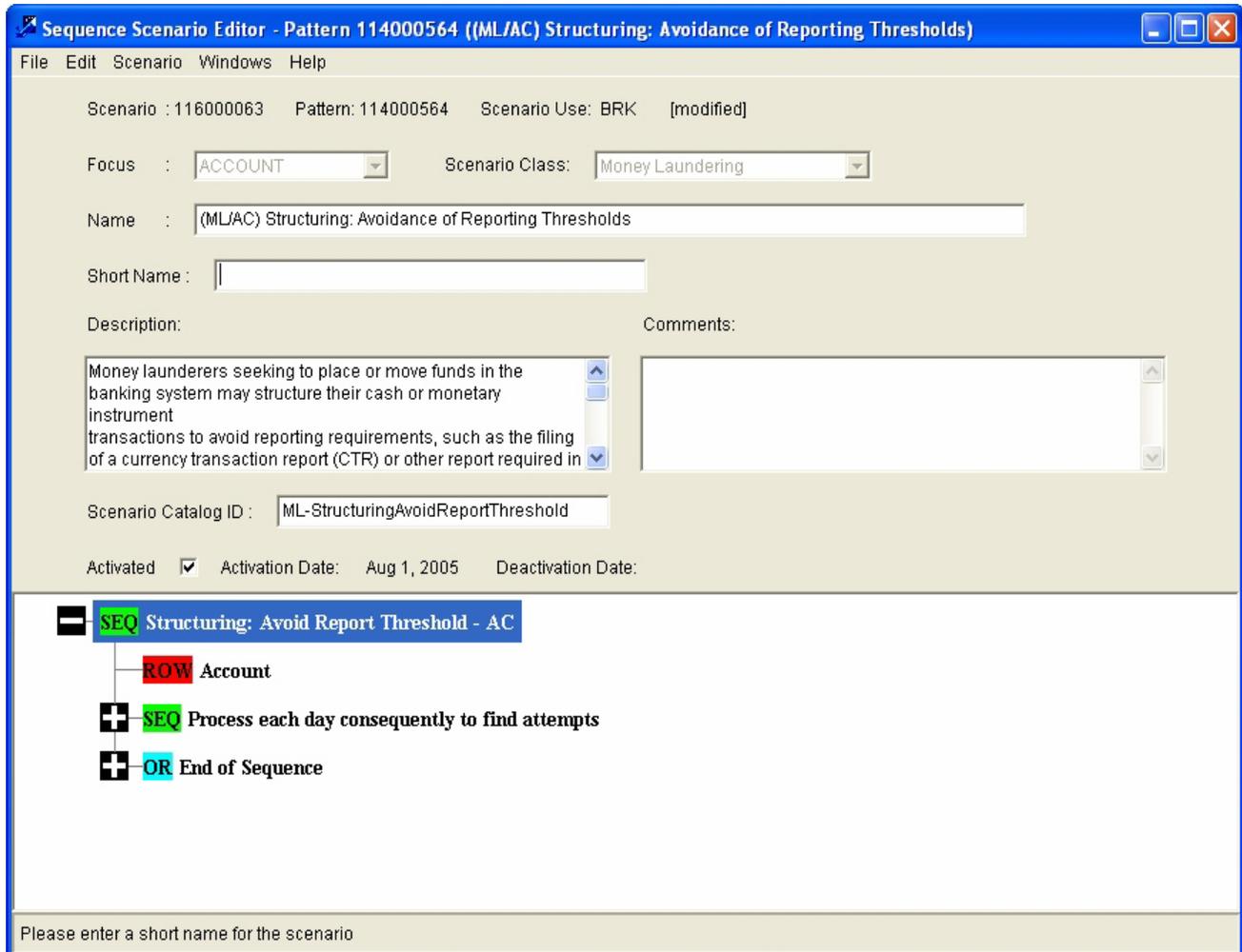
Figure 2. Sample Sequence Scenario Editor with Status Bar Message

CHAPTER 3   *DataSet Editor*

This chapter defines the features and functionality of the DataSet Editor Tool. This chapter focuses on the following topics:

- About the DataSet Editor
- Components of the DataSet Editor
- Using the DataSet Editor
- Importing a DataSet
- Exporting a DataSet
- Generating a DataSet Report
- DataSet Editing Design Tips

About the DataSet Editor

The DataSet Editor produces views, which are virtual tables based on an underlying SQL definition. The dataset provides a common interface for accessing data from a variety of sources. Datasets consist of a collection of data elements, each with a set of attributes that fully characterize that element. Within the DataSet Editor, you use graphical icons to define these elements—to select attributes, join tables, filter records, and derive elements. Access and construction of those data elements depends on the specific type of dataset. Two types of datasets are available:

- **Wrapper:** Accesses an external data source, such as a database table. A wrapper dataset provides attribute definitions and other descriptive information, however, does not specify how the elements are constructed or accessed. When a request is made for the wrapper dataset's elements, the request is forwarded to the external source, and the results are properly formatted for the client. Wrapper dataset is a mirror image of a table. For example, if there is an Account table then the wrapper for this table will be ACCT.

- **Derived:** Combines or modifies the contents of other datasets. A derived dataset takes one or more source datasets as input and applies a set of transformations to create a new set of elements. Source datasets can come from wrappers around external sources or other derived datasets.

    A derived dataset specifies the definition, selection, and combination of element attribute values through a sequence of transformation operators. For example, there are two datasets, Dataset A and Dataset B. If Dataset B is derived from Dataset A then, Dataset B is the derived dataset.

DataSet Editor Terminology

Table 4 provides a list of terms used to define the concepts of the DataSet Editor.

Table 4. DataSet Editor List of Terms

Term	Definition
Dataset	Set of data that is derived from a business firm for the purpose of finding behaviors of interest.
DataSet Editor	Tool that the Scenario Manager provides to create datasets that are then used with other Scenario Manager editors to create scenarios.
Derived Dataset	Entity that combines or modifies the contents of other datasets. Derived datasets take one or more source datasets as inputs and applies a set of transforms to create a new set of elements. Source datasets can be datasets that wrap around external sources or other derived datasets.
Transforms	Transforms that the DataSet Editor provides: ● **DataSet Source:** Identifies a table or view used in the dataset. ● **Join:** Combines multiple source datasets. ● **Filter:** Creates a restrictive clause to limit the records in the dataset. ● **Derive:** Enables derivation of a new attribute from an existing attribute.
Wrapper Dataset	Dataset that accesses external data sources such as database tables. However, wrapper datasets do not specify element construction or access.

Components of the DataSet Editor

The DataSet Editor includes the following areas (Figure 3):

- Menu Bar

- Attributes Area

- Transforms Area

- SQL Area

- Comments Area
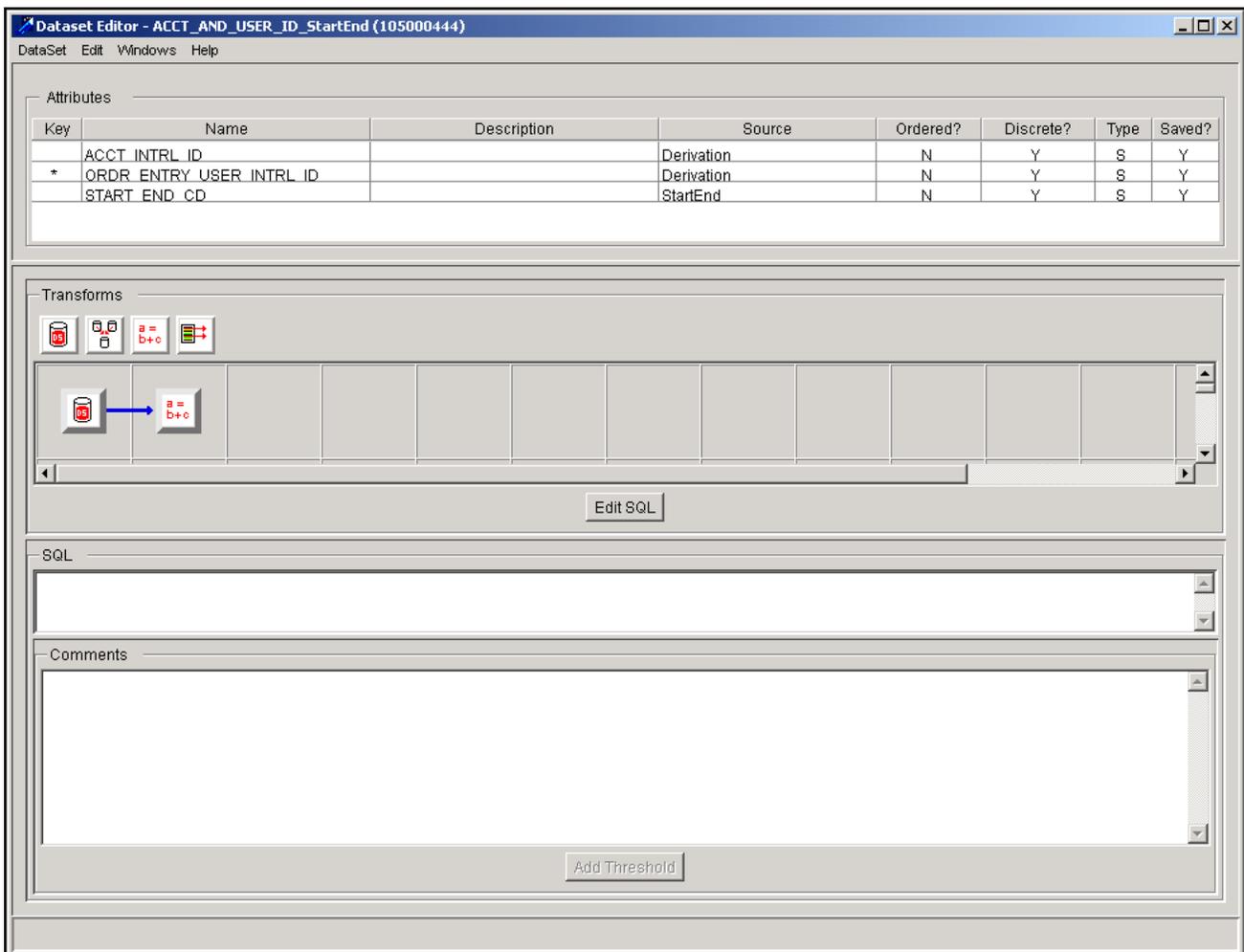
Figure 3. DataSet Editor

Menu Bar

The menu bar contains commands and functions in the following menu options:

- DataSets Menu

- Edit Menu

- Windows Menu

- Help Menu

The following sections describe the DataSets and Edit menu options. Refer to section *Common Menu Bar Options*, on page 3, for information about the Windows and Help options.

DataSets Menu

The DataSets menu includes the following options.

- **New:** Select the DataSet Editor. It essentially selects New from the DataSet menu (that is, it is unnecessary to click this menu option).

  However, if you are working in the DataSet Editor and have any unsaved changes and select New from the DataSet menu, the system displays a warning message that you have unsaved data.

- **Open:** Opens the Select DataSet dialog box, in which you can search for and select a dataset. Refer to section *Select DataSet Dialog Box*, on page 5, for more information.

- **Revert:** Reverts to the original dataset prior to saving any changes.

- **Save:** Saves any changes to the current dataset.

- **Save As:** Saves the current dataset with a unique name that you provide.

- **Delete:** Deletes a dataset from the Scenario Manager.

  Note: Only when other datasets or patterns do not use it. A message displays: *Yes, the dataset is deleted. This is not recommended, as it will destabilize the scenario(s) that were using the dataset.*

- **Properties:** Displays the current dataset name, table name where the dataset resides, owner of the dataset, and creation date of the dataset. It also enables to add or remove domains and set the format of the current dataset.

  - **General** tab: Provides information about the dataset, such as, name of the dataset, table used, owner, and creation date. If you change the dataset name on this tab, the system saves the dataset with the new name.

  - **Domains** tab: Enables you to select a domain to which a dataset belong. The domain code is *MTS*(Mantas), which allows separating data from different products. It is possible to create a new domain code at installation time, if required.

- **Import:** Imports a dataset in XML format. Refer to section *Importing a DataSet*, on page 35, for more information.

- **Create Wrapper:** Creates a wrapper dataset. Refer to section *Creating a Wrapper Dataset*, on page 18, for more information.

- **Export:** Exports a dataset to an XML format. Refer to section *Exporting a DataSet*, on page 33, for more information.

- **Generate DataSet Report:** Generates an HTML file that contains detailed information about the dataset. Refer to section *Generating a DataSet Report*, on page 34, for more information.

- **Close:** Closes the DataSet Editor.

- **Exit:** Exits the Scenario Manager.

Edit Menu

The Edit menu includes the following options.

- **Cut:** Removes the dataset selection from the DataSet Editor.

- **Copy:** Copies the existing dataset selection.

- **Paste:** Inserts the cut or copied selection at the insertion point, which replaces any highlighted text in the editor.

- **Delete:** Deletes a selected transform icon from the **Transforms** area.

- **Properties:** Displays the properties for the selected transform or attribute characteristics. Refer to section *Modifying Attribute Properties*, on page 31, for more information.

Attributes Area

The Attributes area displays attributes that the current dataset defines. An attribute value can be inherited from another dataset or computed in a Derive transform based on the values of other attributes.

The Attributes area contains the following columns:

- **Key:** Indicates that the attribute is the DataSet key that uniquely identifies a record in the PDM table. (Database naming conventions use the suffix SEQ_ID for fields that are unique record identifiers.) Key attributes dictate the type of building block that displays in the Web Application; only one attribute can be the DataSet key.

- **Name:** Displays the attribute name. The name must start with a letter, followed by up to thirty letters, digits, or underscore characters.

  Note: In the current implementation, you should avoid using an SQL-reserved word (for example, *where*) as an attribute name, which can result in an error. Attribute names must be unique within a given dataset, although you can use the same attribute name in different datasets.

- **Source:** Identifies the source dataset. This dataset is normally another dataset. When the source is identified as Derivation, the attribute was created from other attributes within the current dataset.

- **Type:** Shows the range of values to which the attribute can be bound:

  - **D** (Date)

  - **F** (Float)

  - **I** (Integer)

- **S** (String)

- **T** (Time)

Transforms Area

The Transforms area displays the transforms associated with the current dataset (if derived) (Figure 4).
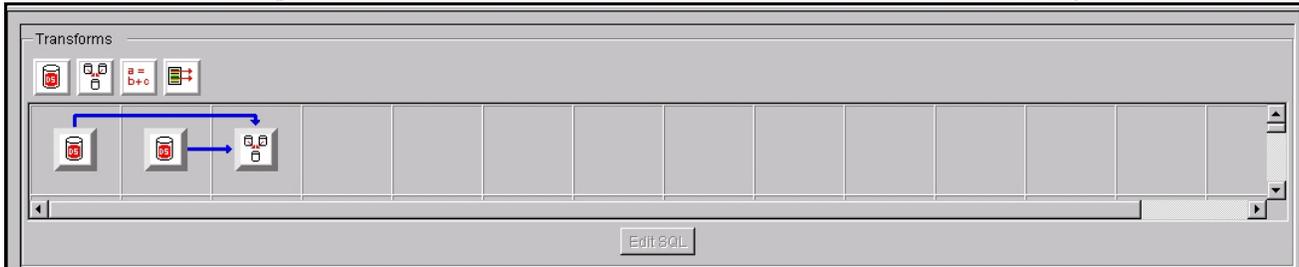
Figure 4. Transforms Area

An icon identifies each transform. You can add or modify transforms by adding one or more transform icons into the Transforms area and then linking them.

The system builds or modifies a new dataset from an existing dataset by placing one or more transforms in the grid, connecting them, and defining or establishing criteria in the transformation edit dialog boxes, as needed. The transform that represents the dataset source must always be present in the grid.

Note: Because of restrictions in the transformation language, some elements must be the first transform in the series. In other words, the input must be external data sources, not other transforms. Conversely, some transforms require other transforms as input and cannot be the first transformation in the series. To define transforms within a dataset, you must create at least one dataset. The system builds the new dataset by adding transforms to the grid, linking the transforms, and defining transform properties or variables.

Join is the only transform that requires establishment of more than one source dataset. Therefore, it is possible to use all transforms within a single dataset. You can use more than one Dataset Source, Derive New Attributes, or Filter transform within one dataset.

The following defines transforms:

DataSet Source: Specifies a dataset to use as input to the current dataset.

Join DataSets: Specifies linking between the selected sources (tables and views).

Derive New Attributes: Enables derivation of a new attribute from existing attributes.

Filter Elements: Enables creation of a filter, or *restrictive* clause, where an operator and a specified value follow a specified attribute.

For example,

APPRAISED_VALUE >= 200000

As your cursor moves over a transform, a tooltip appears below. When you click a transform, an icon displays in the grid in the first available square (from left to right).

You can move transforms by dragging the icon to a desired grid box. Releasing the left mouse button displays the icon button in the designated box.

Oracle Financial Services Transforms Syntax

Table 5 defines the Oracle Financial Services transforms syntax for use in graphical mode.

Table 5. Transforms Syntax Definitions

Meaning	Syntax	Keyboard Keys/Proper Syntax
Relational Operators		
not operator	!	SHIFT-1
equals	=	=
not equal	!=	SHIFT-1 =
greater than	>	SHIFT-period
greater than or equal to	≥	SHIFT-period-=
less than	<	SHIFT-comma
less than or equal to	≤	SHIFT-comma-=
Mathematical Operators		
addition	+	SHIFT-=
subtraction	-	--
multiplication	*	SHIFT-8
division	/	Forward slash
exponentiation	^	SHIFT-6

Table 5. Transforms Syntax Definitions (continued)

Meaning	Syntax	Keyboard Keys/Proper Syntax
null comparison	isNULL<attribute>	is-Caps lock-NULL
Other Conditions		
When evaluating expressions together	( )	SHIFT-9 <expressions> SHIFT-0
variable names	$	SHIFT-4 <variable>
strings	' '	SHIFT-single quote <string> SHIFT-single quote

When working in the Transforms area, you are working in *graphical mode*. When working in the SQL area, you are working in *SQL mode* (refer to section *SQL Area*, on page 16, for more information).

Transforms Usage Tips

Use the following guidelines when working in the Transforms area:

- To display a transform operator in the first available cell within the grid in the Transforms area, click the transform icon.

  - To link transforms, hold the Shift key and then click and hold the left mouse button on top of the first icon. Then drag the mouse to the second icon to form the link.

  - When linking the transformation operators and the line between the two transforms displays red, you cannot link the transform operators.

  - If the line is green, however, you can link the transform operators.

  - When you release the Shift key and the left mouse button, a line connects the two icons.

- To save a new dataset, you must select attributes for the dataset.

- When you use the Derive New Attributes transform operator, the system automatically adds the attributes to your dataset when you successfully complete the edit dialog boxes associated with each function.

Refer to *Appendix A, Scenario Design Guidelines*, on page 175, for other tips and guidelines.

SQL Area

The SQL area, or *SQL editor*, enables you to view and modify the SQL code for your dataset by clicking the **Edit SQL** button. Before going into SQL mode, you must save the dataset in graphical mode.

You can work between the Transforms area and the SQL area. If you created the dataset using the transforms, you can then manually create a more sophisticated definition for your dataset by using the SQL area.

SQL Mode Usage Tips

Use the following guidelines when defining a dataset in SQL mode:

- Save your dataset before using the **Edit SQL** button.

- You can use the DataSet Source and Derive New Attributes icons in the Transforms area to change the attributes in your datasets.

- Use all features of the SQL language, including advanced SQL clauses in the editor such as union, group by, having, and exists or not exists.

- The system does not validate modified SQL code so you must ensure that SQL code you specify is accurate and error-free.

- You must ensure that you define all datasets and attributes in graphical mode that you are using in SQL mode. Also ensure that definitions that you create in SQL mode override definitions in graphical mode.

- The text DSVIEWxxx (that is, dataset name) must appear only in the SQL text area where you intend to use it. This text should not appear in a comment in the SQL area.

Refer to *Appendix A, Scenario Design Guidelines*, on page 175, for other tips and guidelines.

Comments Area

While creating a dataset using transforms and the SQL editor to modify the dataset, the system places any specified comments in the **Comments** text area.

The Comments button is not enabled by the system till the time you have used the SQL editor.

Note: It is recommended that you structure your comments by using your login ID, date of the comment, and information relevant only to the dataset to which the transform applies. Other users can then read the comments and follow the structure of your dataset.

The **Join**, **Derive New Attributes**, and **Filter** transforms have their own Comments text area, which you can access from the individual transforms.

Using the DataSet Editor

The following sections provide step-by-step instructions for using the DataSet Editor:

- Accessing the DataSet Editor
- Creating a Wrapper Dataset
- Creating a Derived Dataset
- Modifying a DataSet
- Reverting a DataSet
- Deleting a DataSet

The following sections provide these instructions as well as access of the DataSet Editor.

Note: If a dataset used in augmentation contains thresholds, views used in augmentation, then use the values of the thresholds from the scenario's base threshold set. The values of the thresholds in the base set may be different from the values in the detection process.

Accessing the DataSet Editor

By default, opening the Scenario Manager also opens the DataSet Editor (refer to section *Accessing the Scenario Manager*, on page 2, for more information).

Note: The editor that first displays when you open the Scenario Manager is configurable; see your system administrator to change the editor that displays when you open the Scenario Manager.

Creating a Wrapper Dataset

The DataSet Editor enables to import table definitions to create a *wrapper dataset* that provides access to the underlying database table. Creating a wrapper dataset involves selecting the database table, which changes the attribute properties assigned during the import, as needed, and saves it.

Every table in the database that is part of a derived dataset must have a wrapper dataset built for it.

The system creates the wrapper dataset with an attribute corresponding to each column in the table. The attribute type is based on the corresponding column type. The attributes of the new dataset are displayed in the Attributes area.

You cannot add transforms to an imported dataset, hence transforms are disabled.

To create a wrapper dataset in the DataSet Editor, follow these steps:

1. Select **DataSet** and then **Create Wrapper**.

   The Import Table dialog box displays a list of available databases (Figure 5).

Figure 5. Import Table Dialog Box

2. Double-click the database that contains the appropriate table.

   The Import Table dialog box displays the selected database in the **Look in** field and lists available tables.

3. Select the table for which you want to create the wrapper dataset.

4. Click **Ok** to import the selected table.

Creating a Derived Dataset

You create a derived dataset to combine or modify the contents of other datasets. A derived dataset takes one or more source datasets as input and applies a series of transformations to create a set of data records for use in behavior detection. Transformations (or transforms) describe the combination, alteration, and modification of inputs to produce the resulting dataset. Some transformations restrict or filter the existing elements in the source sets, others construct new elements by adding or modifying element attributes.

Before using the transforms, you must create at least one dataset. You build the new dataset by adding transforms to the grid, linking the transforms, and defining transform properties or variables. Only the Join transform requires establishment of more than one source dataset. Therefore, you can use all transforms within a single dataset. You can also use some transforms, such as the Filter Elements transform, more than once.

Note: If a dataset used in augmentation contain thresholds, views used in augmentation then use the values of the thresholds from the scenario's base threshold set. The values of the thresholds in the base set may be different from the values used in the detection job.

The following sections provide the basic instructions on creating a derived dataset:

- To Add a DataSet Source
- Joining DataSets
- Filtering a DataSet

- Creating a Derived Attribute Within a DataSet

- Saving a DataSet

- Modifying Dataset Properties

- Modifying Attribute Properties

The above procedures assume that you are using all transforms, as these sections present instructions for using each transform separately.

Note: The Dataset Editor displays an exception on the console when a thread is terminated during processing.

To Add a DataSet Source

To add each dataset source, follow these steps:

1. Click the [icon] icon in the Transforms area (Figure 4).

   The system adds the icon to the first available cell in the grid area.

2. Double-click the **DataSet Source** icon.

   The Edit DataSet dialog box displays (Figure 6).
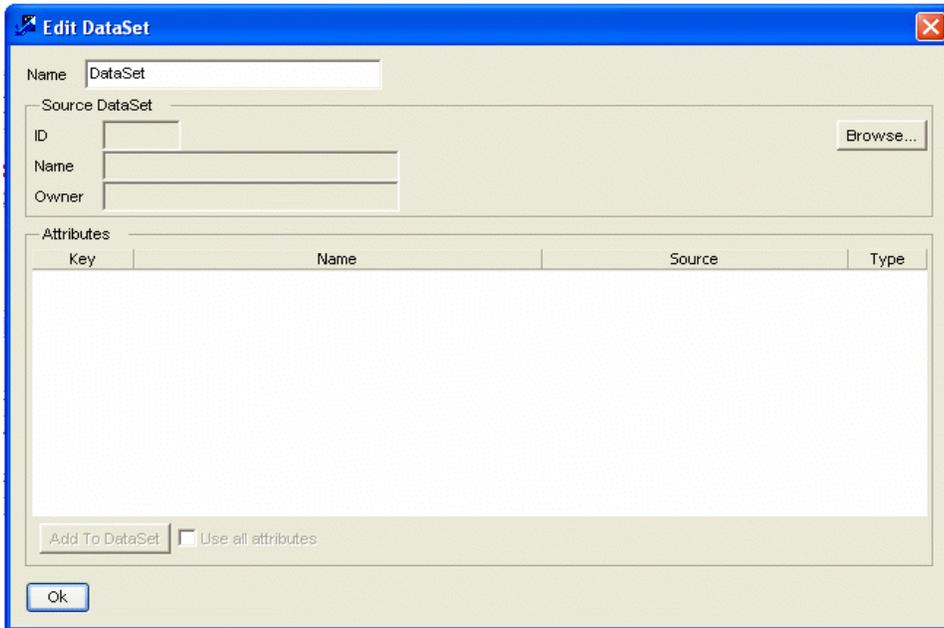
Figure 6. Edit DataSet Dialog Box

3. Click **Browse** in the Edit DataSet dialog box to select a source dataset.

   The Select DataSet dialog box displays.

4. Select the desired dataset from the Datasets area.

5. Click **OK**.

The Edit DataSet dialog box displays the DataSet ID, Name, Owner, and Attributes.

6. Do one of the following:

   - Hold down the **CTRL** key and click the desired attributes.

   - Select the **Use all attributes** option.

7. Click **Add to DataSet** to add the selected attributes to the open dataset.

8. Click **OK**.

   The attributes display in the Attributes area.

Joining DataSets

The Join DataSets transform enables you to join multiple datasets, which you do in the Edit Join dialog box (Figure 7).
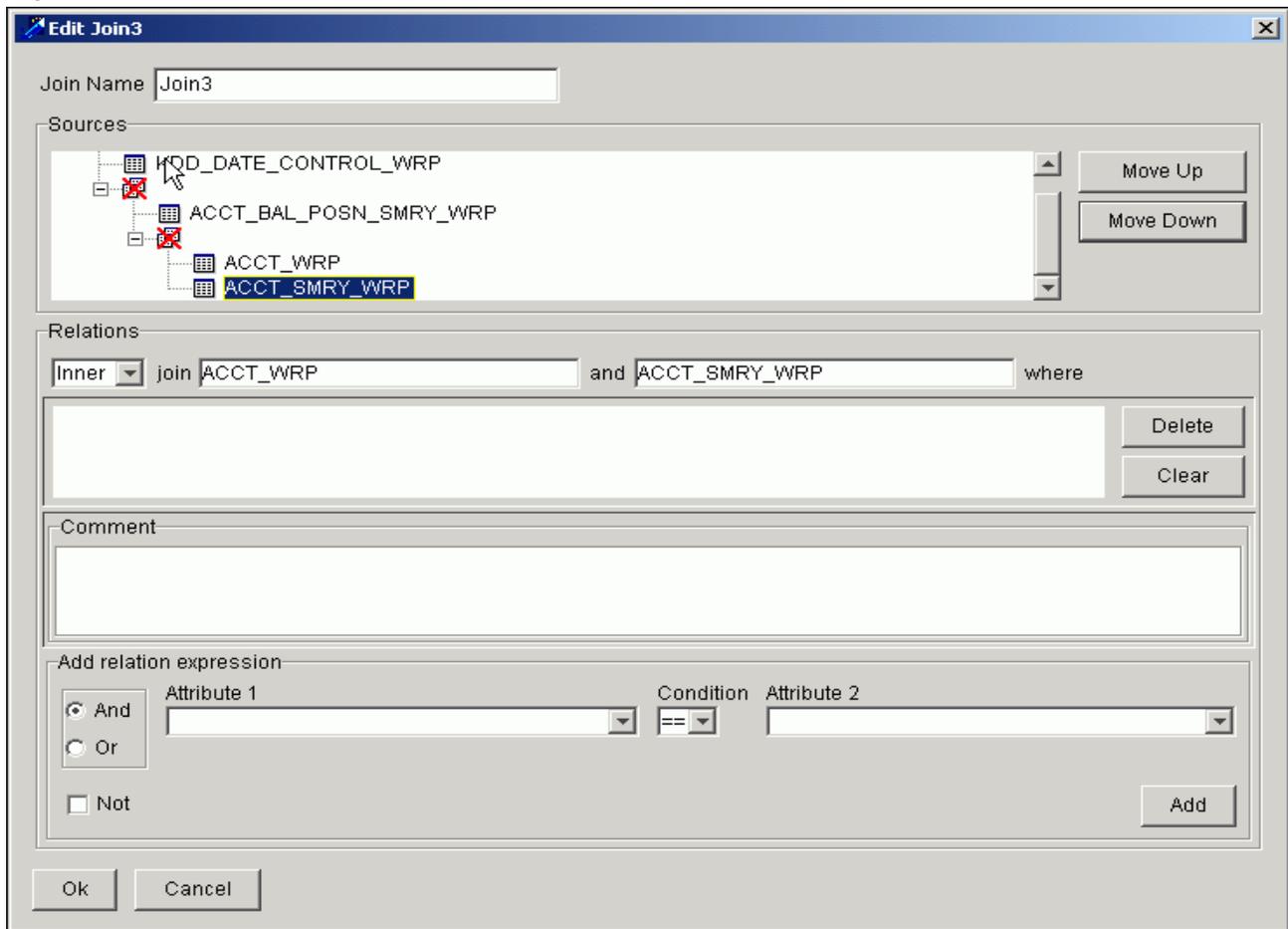
Figure 7. Edit Join Dialog Box

The elements in the new dataset contain attributes selected from each source dataset. You must select at least two datasets for the join.

The Edit Join dialog box contains the following areas:

- **Sources:** Indicates where the datasets of the join are ordered.

- **Relations:** Specifies where you define the type of join.

- **Add relation expression:** Specifies additional or alternative join relation expressions.

The following sections describe each of these areas.

Note: You may define the joins of the datasets in any order, however, you must define joins for all of the listed source datasets.

Sources Area

The Sources area contains the source datasets. The red **X** in Sources indicates that a join is not yet defined between the datasets.

Use the **Move Up** and **Move Down** buttons to move a dataset to a higher or lower position in the list.

Note: The **Move Up** and **Mode Down** buttons are inactive unless you have three or more datasets.

When you select the source dataset at the top, the dataset title and the next listed dataset title appear in the Relations area (on either side of the word *and*). If only three source datasets are available, the second dataset title box in the Relations area contains the remaining two datasets (that appear together at a single level). To complete the Join relation expression, you must define a Join relation expression between the top dataset and the other two datasets.

Relations Area

The Relations area includes the following components:

- **[Type]**: Enables you to select the type of join (Inner, Left, and Right).

  - An Inner Join indicates that only those elements that are shared in both sources.

  - Left and Right are Outer Joins. An Outer Join indicates that you want the elements even if they are not (to the left or right) in both source datasets. (Not all databases currently support a Full Outer Join.)

  Text boxes to the right of the join type indicate the tables that a Join expression links together. The text box underneath the list specifies the Join condition.

- **Delete:** Deletes the selected Join relation expressions.

- **Clear:** Clears all Join relation expressions established for the Join of the datasets.

- **Comment:** Enables to add comments about the Join in this text area.

Add Relation Expression Area

After selecting the source dataset, specify the Join relation expression. A Join relation expression can include any of the following components:

- **And or Or** option: Use to specify whether a Join condition is required (And) or is an alternative (Or) to the previous expression.

- **Not** option: Use this option to negate a Join relation expression

- **Attribute 1**: Contains the list of attributes from the dataset on the left of the relation expression.

- **Condition**: Shows the type of operator used for comparing Join fields.

- **Attribute 2**: Contains the list of attributes from the dataset on the right of the relation expression.

- **Add:** Adds the Join relation expression to the Relations area.

Click **Ok** to save the Join transform and return to the DataSet Editor.

To join datasets, follow these steps:

1. Click the [icon] icon in the Transforms area.

   The system adds the icon to the first available cell in the grid area.

2. Link the **DataSet** icons to the **Join** icon.

   A blue arrow displays between joined transforms to show they are linked (refer to section *Transforms Usage Tips*, on page 16, for more information).

3. Double-click the **Join** icon.

   The Edit Join dialog box displays with each dataset. A red **X** displays above each dataset to indicate that you have not established the Join between datasets.

4. Define the join, within the Relations area, between the datasets (refer to section *Joining DataSets*, on page 21, for more information).

5. Click **OK**.

6. Repeat the procedure to build the remaining Join.

You may add more datasets to the Join at any time before saving the dataset. When you add a new dataset, however, the Join relation expressions between the original source datasets no longer exist. You must therefore redefine the joins between all the source datasets. When you access the Edit Join dialog box after adding another source dataset, the newly added dataset appears at the bottom of the Sources tree. The last two datasets always appear together on one level.

To define the Join between datasets (this procedure is a subset of joining the datasets procedure), follow these steps:

1. Select an attribute from the drop-down list under **Attribute 1** in the Add relation expression area.

2. Select the appropriate **Condition** operator after selecting the first attribute.

   With Inner joins, you typically use all operators, however, use only the equal (=) condition operator with any Outer joins.

3. Select an attribute from the drop-down list under **Attribute 2** after selecting the appropriate condition operator.

4. Do one of the following:

   a. Click **Add** after selecting the second attribute. The Join relation expression displays in the Relations area.

      The red **X** no longer appears in the sources diagram since a Join was defined for these datasets.

   b. Click **Delete** to remove selected join relation expressions.

   c. Click **Clear** to clear all the join relation expressions established for joins on the two datasets.

5. Do one of the following:

   a. Select the **And** option to join the identified datasets where the first Join relation expression and the next apply.

b. Select the **Or** option to join the identified datasets where the first Join expression or the next applies.

c. Select the **Not** check box to negate a Join relation expression.

Optional: Add any comments about defining a Join to the Comment area.

6. Click **Ok** to accept the joined relation after completing the Join and return to the DataSet Editor with the established dataset Joins.

Note: Clicking **Ok** before creating join relation expressions for all of the source datasets displays a warning message.

Filtering a DataSet

The *Filter Elements* transform enables you to indicate that the system is to filter one or more elements from the input source on their respective attribute values. You do this in the Edit Filter dialog box (Figure 8).
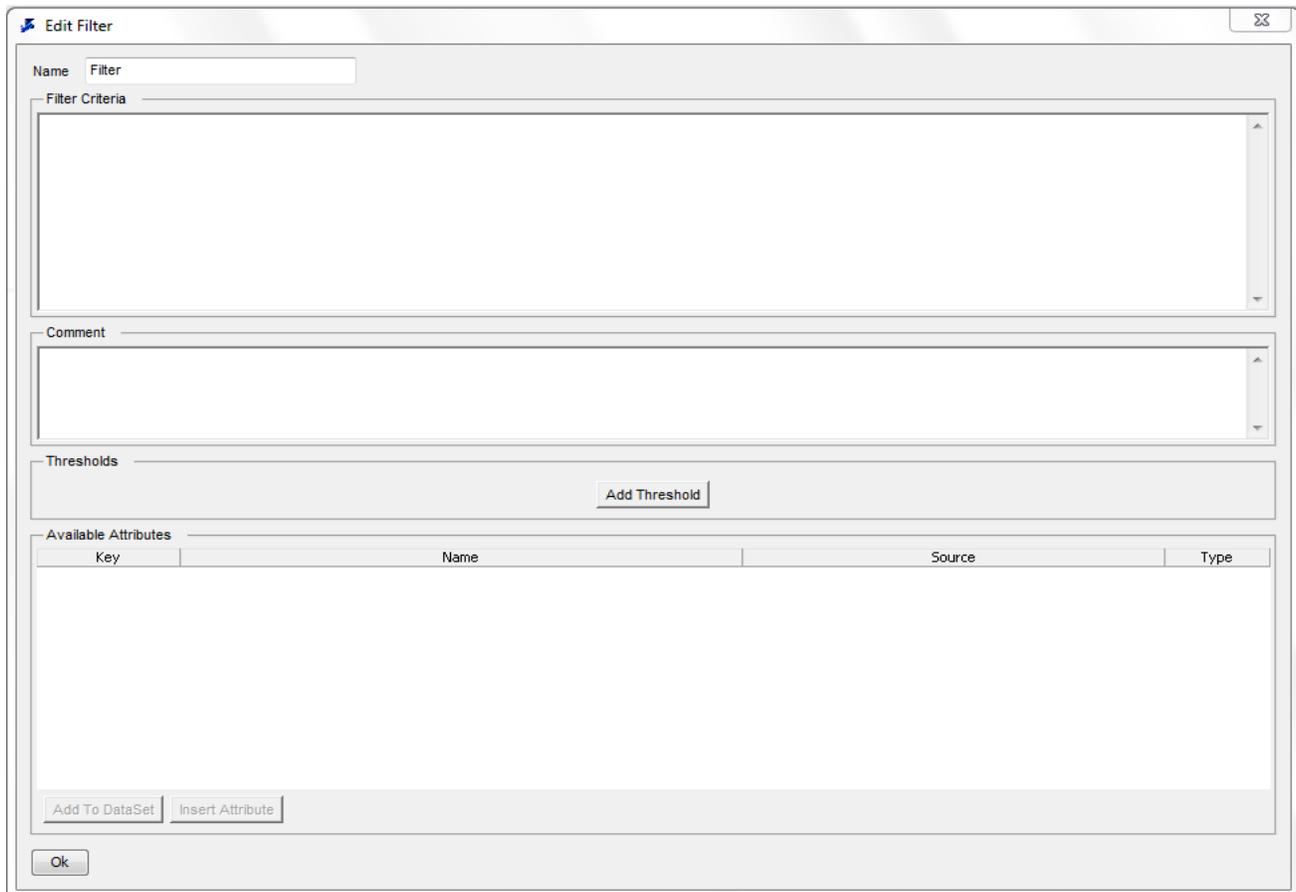
Figure 8. Edit Filter Dialog Box

You may link the Filter Elements icon in the grid to the DataSet Source icon or to any other icons (as long as a source dataset has been established).

The components of the Edit Filter dialog box include the following areas:

● **Name:** Identifies the filter.

- **Filter Criteria:** Contains the filter that you create. This is where you add attributes, operators, and attribute values to build the filter.

- **Comment:** Enables you to enter a comment related to the filter (not the dataset or other transform).

  Click **Add Threshold** button to open the Threshold Manager dialog box, in which you can select a threshold for use in the filter criteria.

- **Available Attributes:** Displays a list of available attributes to use in the filter. This area has two buttons, which become enabled after you make a selection:

  - **Add to Dataset:** Adds an attribute to the dataset.

  - **Insert Attribute:** Inserts an attribute into the Filter Criteria area. The entry includes both the source dataset name and the system name for the attribute.

  Click **OK** to save the filter transform.

Guidelines for Creating Filters

The following sections provide guidelines for creating filters when using equality operators, attribute values, and functional and logical operators.

Relational Operators

Table 6 lists the equality operators that you can use to build your filter.

Table 6. Relational Operators

Operator	Definition
= =	Equal To
! =	Not Equal To
<	Less Than
< =	Less Than or Equal To
>	Greater Than
> =	Greater Than or Equal To

Attribute Values

After entering an operator, you must enter the attribute value or another attribute as follows:

- **Date Attribute:** Single quotation mark, day of month, hyphen, three alpha characters for month, hyphen, four-digit year, and single quotation mark (that is, 'dd-Mmm-yyyy')

  Example: `'18-May-2006'`

- **Float Attribute:** Series of decimal digits, decimal point, and optional digits

  Example: `61.19`

- **Integer Attribute:** Series of decimal digits

  Example: `60`

- **String Attribute:** Single quotation mark, case-sensitive system attribute value, and single quotation mark

  Example: `'Rural'`

> **Note:** If the attribute value contains an apostrophe, that single quote is duplicated when using the value in this context. For example, if the value to be filtered is O'Quinn for the string attribute of Claimant Name, the value 'O' 'Quinn' appears within the Filter Criteria area.

> **Note:** You can use the **like** SQL operator in filters (for example, `ACCT_TYPE like 'IRA%'`).

Functional and Logical Operators

To establish multiple filter conditions, use standard mathematical operators (that is, +, -, *, or /), logical operators (for example, AND, OR, or NOT), and parentheses to define the required logic.

To filter a dataset, follow these steps:

1. Click the icon in the Transforms area.

   The system adds the icon to the next available square in the Transforms area grid.

2. Link the **Join** icon to the **Filter Elements** icon.

3. Double-click the **Filter Elements** icon.

   The Edit Filter dialog box displays.

4. Enter the name of the filter in the **Name** field.

5. Select the desired attribute and click **Insert Attribute** in the Available Attributes area.

   The system adds the attribute to the Filter Criteria field.

6. Click the cursor in the **Filter Criteria** area after the attribute to begin building the filter.

7. Follow these steps:

   ● Enter the equality operator.

   ● Enter the attribute value, another attribute, or a threshold.

   ● Enter the attribute value preceded by the applicable equality operator.

8. Type `[space]` *and* or *or* to start the next constraint.

9. Click **Ok** after entering the required filters.

Creating a Derived Attribute Within a DataSet

To edit the derivation, follow these steps:

1. Click the Editing/Creating a derived attribute.

2. Click icon in the dataset editor.

   This adds button to the Transforms area.

3. Double click the button

   The Edit Derivation dialog box displays.

The Edit Derivation dialog box that you use when creating a new derived attribute within a dataset from other attributes displays (Figure 9).
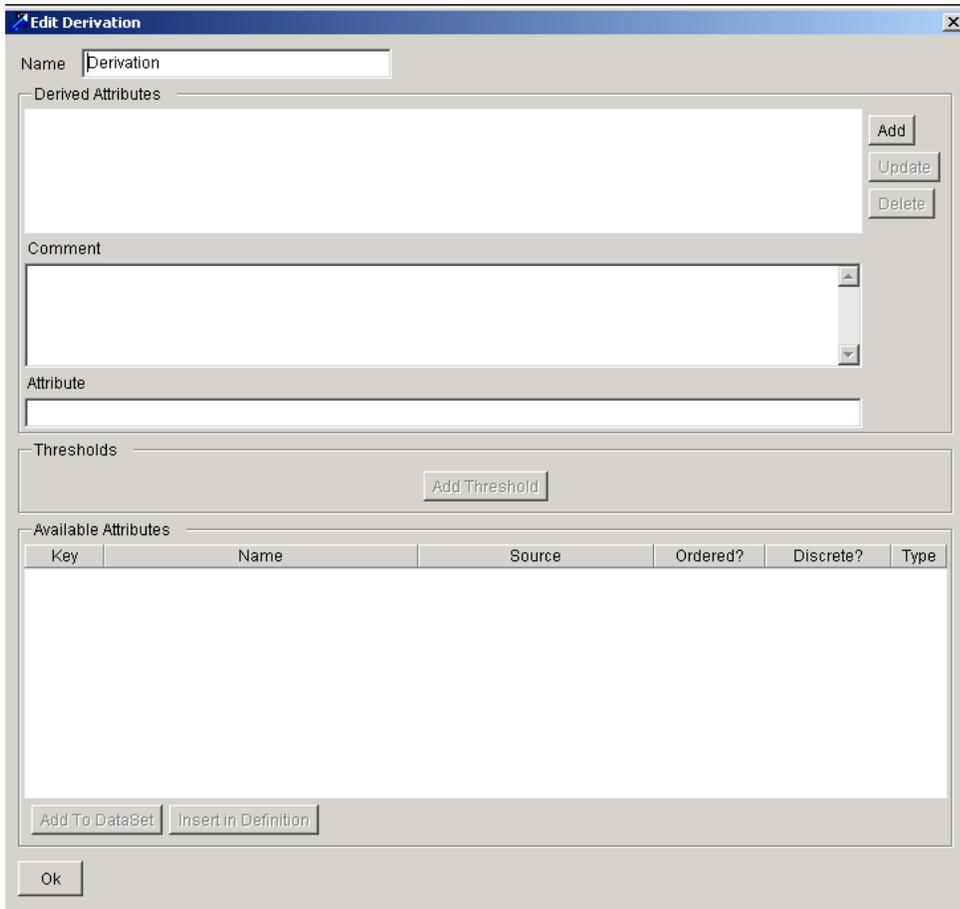
Figure 9. Edit Derivation Dialog Box

Components of the Edit Derivation dialog box include the following:

- **Name:** Identifies the derivation label (not the name of the actual derivation).
- **Derived Attributes:** Enables you to build the derivation in this area, which contains the following:
    - **Add:** Adds the attribute from the Attribute field to the Derived Attributes area.
    - **Update:** Updates the selected derived attribute.
    - **Delete:** Deletes the desired derived attribute.
    - **Comment:** Provides a comment related to the derivation (not the dataset or other transform).

- ■ **Attribute** area: Provides the name of the derivation. When entering the name of your derivation, follow these naming conventions:

    - ◆ Start with a letter

    - ◆ Enter up to 40 alphanumeric characters

    - ◆ Use underscores, if desired

    - ◆ Do not use spaces

- ● **Thresholds** area: Opens the Threshold Selector dialog box, in which you can select a threshold for use in the derivation.

- ● **Available Attributes** area: Displays available attributes to use in the derivation. The columns of this area include the following: Key, Name, Source, and Type. This areas has two buttons:

    - ■ **Add to Dataset:** Adds the attribute to the dataset after selecting attribute in the Available Attributes area.

    - ■ **Insert in Definition:** Provides a shortcut for adding an attribute to the Attribute field. Select an attribute in the Available Attributes area and then click **Insert in Definition**. The entry includes the source dataset name, separated by a period, and then the system name for the attribute. For example:

      `ACCT_WRP.OPTN_APPRVL_LVL_CD`

    The **Add to DataSet** and **Insert Attribute** buttons are enabled after you make a selection in the Available Attributes area.

- ● **OK:** Saves the derivation transform.

To create a derived attribute within a dataset, follow these steps:

1. Click the icon in the **Transforms** area.

   The system adds the icon to the next available square in the Transforms area grid.

2. Double-click the **Edit Derivation** icon.

   The Edit Derivation dialog box displays.

   Optional: Enter the name of the derivation in the **Name** field.

3. Type the name of the attribute of the derivation in the **Attribute** field and then type the Assign operator (=); for example:

   `d_OPTN_APRVL_FL=`

4. Write the formula of your derived attribute by doing either of the following:

   - ● Click **Insert in Definition** to add attributes.

   - ● Click **Add Thresholds** to add a threshold.

   Use the appropriate functional operators with your formula (refer to section *Functional and Logical Operators*, on page 26 for more information).

5. Click **Add**.

   The string displays in the **Derived Attributes** field.

6. Click **OK**.

When you use the **Derive New Attributes** transform operator, the system automatically adds the attributes to your dataset when you successfully complete each editor dialog box associated with each transform.

Note: You can use SQL **Case** statements in deriving new elements.

Saving a DataSet

You must save a dataset that you create by importing or defining a derived dataset. You can then use this saved dataset as the source for new datasets using the transform grid, where applicable. You must also save the dataset if you make any modifications to it or any attribute properties. You can save a dataset in either of two ways:

- **Save**: When you save a newly created wrapper or derived dataset for the first time, the standard Save DataSet dialog box displays, which prompts you for the name you want to use for your dataset. Type the name of the new dataset (up to 32 alphanumeric characters) in the Name.

  On an existing dataset, this save function saves any modifications you made to that dataset with the same properties, however, the Save DataSet dialog box does not display. The Status Bar indicates the save is in progress and the dataset is saved once completed.

- **Save As**: When you select save as function, the Save DataSet dialog box displays, which enables you to save a new dataset by entering a new name, up to 32 alphanumeric characters.

To save a dataset, follow these steps:

1. From the **DataSet** menu using the DataSet Editor, select **Save** or **Save As**.

   The Save DataSet dialog box displays, unless this is a previously saved dataset and you selected the **Save** option.

2. Enter the name of the data set you want to save.

3. Click **Ok**.

Modifying a DataSet

To modify a dataset, its transforms, or SQL, you must open the dataset first. You can then change specific properties of an attribute, delete an attribute from the dataset, and delete a transform from the dataset. You can also remove changes after making them; however, before saving the dataset using a revert option, once open, you can follow the same procedures identified previously within this chapter to make your modifications and save the updated dataset. In addition, you can modify the following after creating and saving a dataset:

- **Dataset properties** (refer to section *Modifying Dataset Properties*, on page 30, for more information)

- **Attribute properties** (refer to section *Modifying Attribute Properties*, on page 31, for more information)

Modifying Dataset Properties

You can change dataset properties through the DataSet Properties dialog box (Figure 10).

Figure 10. DataSet Properties Dialog Box

The Dataset Properties dialog box has two tabs: General and Domains.

- **General:** Contains general information about a dataset, if applicable. This feature can be helpful if you are viewing a dataset that another individual created. The General tab includes:

  - **Name:** Name of the dataset.

  - **Table:** Name of the database table that corresponds to the dataset.

  - **Owner:** User ID of the creator or owner.

  - **Created:** Date the dataset was created.

- **Domains:** Includes domain information:

  - **Domains:** Lists the domains that access the dataset.

  - **Add:** Adds a domain.

  - **Delete:** Deletes a domain.

To modify dataset properties, follow these steps:

1. From the **DataSet** menu of the DataSet Editor, select **Properties**.

The DataSet Properties dialog box displays (Figure 10).

2. Edit the **Name** field.

3. Click the **Domains** tab to view or change the product designation.

4. Click **OK** to save your changes.

Modifying Attribute Properties

Select an attribute and click **Properties** in the Edit menu The Attribute Properties dialog box displays (Figure 11).

Figure 11. Attribute Properties Dialog Box

The following components allow you to modify the attribute properties:

- **ID**: Displays the system-generated value *–1* until you save the attribute.

- **Name**: Displays the system-generated name.

- **Raw Type**: Select the base type of the attribute (the default is *String*).

- **DataSet Key**: Select this check box if the attribute uniquely identifies each element.

Click **Ok** to return to the Dataset Editor dialog box with your changes.

After saving the derived attribute, the ID changes to a system-generated number. A subsequent display of this dialog box displays the system-assigned number.

To modify attribute properties, follow these steps:

1. Select an attribute from the **Attributes** area.

2. From the **Edit** menu, select **Properties**, or double-click the selected attribute.

   The [Attribute] Properties dialog box displays (Figure 11).

3. Select the **DataSet Key** check box if the attribute uniquely identifies each element.

   This enables the **DataSet Contents**.

4. Enter a name for the dataset element in the **DataSet Contents**.

5. Click **OK**.

   The DataSet Editor dialog box displays the updated property values reflected in the attribute listing.

Reverting a DataSet

You can use the **Revert** option to return to previous settings of a dataset if you did not save the dataset with modifications. When you save a dataset with changes, you cannot revert to a previous version. The ability to revert to previous settings is limited to a removal of *all* changes, not a removal of partial or selected changes.

To revert dataset elements, follow these steps:

1. From the **DataSet** menu, select the **Revert** option from an opened dataset that you have not saved, however, have made changes to.

   A Confirm dialog box displays.

2. Click **Yes** to remove any modifications you made and return the dataset to its last saved state.

Deleting a DataSet

Deleting a dataset removes it from the Scenario Manager; however, you cannot delete a data set if it has been used for a job, or if it has children—meaning, if other data sets rely upon it.

To delete a dataset, follow these steps:

1. From the **DataSet** menu in the DataSet Editor, click **Open**.

2. Select a dataset from the Select DataSet dialog box.

3. Click **OK**.

4. From the **DataSet** menu, click **Delete**.

   The Delete Confirmation dialog box displays.

5. Click **Yes** to delete the dataset.

Warning: Dataset once deleted cannot be retrieved.

Exporting a DataSet

You can export any dataset that you create in the DataSet Editor to an eXtensible Markup Language (XML) file. You can then move or copy the dataset from one environment to another.

To export a dataset, follow these steps:

1. From the **DataSet** menu of the Dataset Editor, select **Open**.

   The Select DataSet dialog box displays.

2. Select the dataset you want to export from the Select DataSet dialog box.

3. Click **Ok**.

   The selected dataset displays in the **DataSet Editor**.

4. From the **DataSet** menu, select **Export**.

   The Export DataSet dialog box displays (Figure 12).
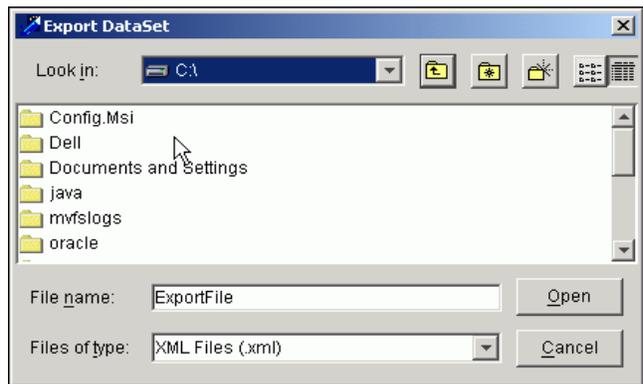
Figure 12. Export DataSet Dialog Box

5. Click the **Look In** field drop-down arrow to view available directories.

6. Click the ![icon] icon to change the folder location within the selected directory to locate your file.

7. Type the file name in the **File Name** field.

   Typically, a file listing displays. However, you should verify that the file is not in the specified directory by typing the name and clicking **Open**.

8. Click **Open**.

   The system saves the exported dataset to the specified location.

   Note: It is recommended that you add the `.xml` extension to the file name, as the system does not automatically do this for you.

Generating a DataSet Report

You can generate a dataset report that enables you to easily view the attribute names in the dataset, the SQL code of the dataset, and the name of any parent datasets. In addition, this report provides scenario information related to the dataset, such as scenario names and any associated thresholds. An example report displays (Figure 13).

Figure 13. DataSet Report Sample

The system generates the dataset report into an HTML format that you can then open and print for use.

To generate a dataset report, follow these steps:

1. From the **DataSet** menu of the Dataset Editor, select **Generate DataSet Report**.

   The DataSet Report dialog box displays.

2. Click the **Look In** field drop-down arrow to view the available directories.

3. Click the [icon] icon to change the folder location within the selected directory to locate your file.

4. Type the file name in the **File Name** field.

5. Click **Open**.

   When naming the file, you must save the file with an HTML extension (for example, `dataset_report.html`).

6. Click **Report**.

The system saves your report to the selected directory.

Importing a DataSet

When you open an Oracle Financial Services dataset definition file to import, the system imports the file as a dataset with a system-generated title (for example, KDD_IMPORT70). The DataSet Editor initially reads the file, determines attribute types, and loads data from the file. If importing a file without headers, the system uses a generic descriptor to replace missing headers (for example, *field*).

For an imported dataset, you can change attribute values such as Key and Type.

To import a dataset, follow these steps:

1. From the **DataSet** menu of the Dataset Editor, select **Import**.

   The Import DataSet dialog box displays (Figure 14).
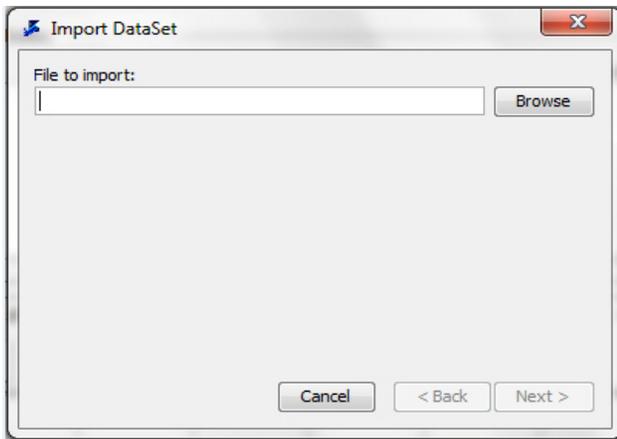
Figure 14. Import DataSet Dialog Box

2. Click **Browse**.

   The Select Import File dialog box displays (Figure 15).
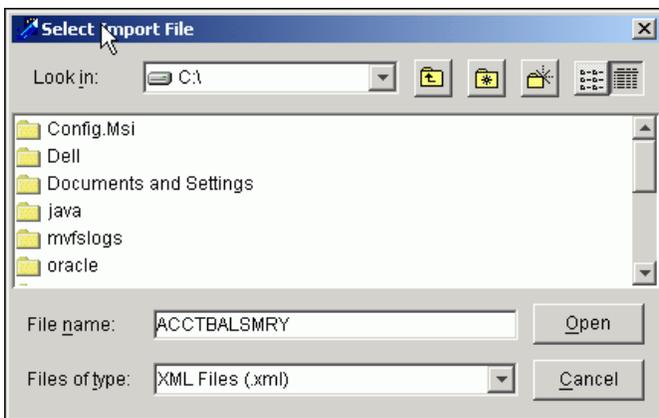
Figure 15. Select Import File Dialog Box

3. Click the **Look In** field drop-down arrow to view the available directories.

4. Click the ![icon] icon to change the folder location within the selected directory to locate your file.

5. Type the file name in the **File Name** field.

   Typically, a file listing displays. However, you should verify that the the file is not in the specified directory by typing the name and clicking **Open**.

6. Click **Open**.

   The Import DataSet dialog box displays again and shows the specified file in the **File to Import** field.

7. Click **OK**.

   The name of the file to be imported displays in the **Old** column in the Import DataSet dialog box.

8. Click the file name in the **Old** column.

   This enables the **By ID**, **By Name**, and **Browse** buttons.
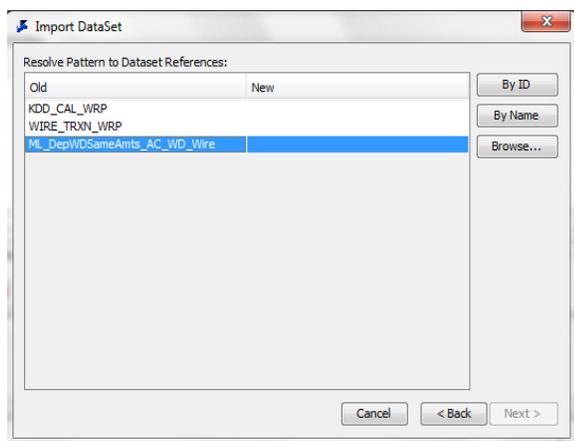
Figure 16. Import DataSet Dialog Box

9. Click **Browse** to select the desired dataset.

   The Select Table dialog box displays.

10. Double-click the domain name under which the desired dataset resides. Then double-click the database name where the dataset resides.

11. Click the desired dataset table.

12. Click **OK**.

   The Import DataSet dialog box displays the selected dataset table name in the **New** column.

13. Click **Next**, then **Finish** to import the dataset.

14. From the File menu, click **Save** or **Save As** to be sure you save your imported dataset with a new, unique name.

DataSet Editing Design Tips

The following design tips provide methods in which you can optimize dataset performance:

- Specify the correct data type (generally String, Float, or Integer) when creating derived attributes. The most common mistake is to specify a number as a string; using string comparisons, 100 is less than 99.

- Specify a valid dataset key that uniquely identifies each record in the dataset; end the name with *SEQ_ID*. If the dataset generates matches that display as building blocks, use one of the key attributes for the dataset key.

- Avoid creating any derived attribute that can cause a *divide by zero* error.

- Filter out null values in the fields that you want to use in the pattern.

- Never allow a null focal entity key.

Refer to *Appendix A, Scenario Design Guidelines*, on page 175, for more information about scenario design tips and guidelines.

Sequence Scenario Editor

This chapter defines the Sequence Scenario Editor features and functionality within the Scenario Manager. This chapter focuses on the following topics:

- About the Sequence Scenario

- Sequence Scenario Editor Components

- Using the Sequence Scenario Editor

- About the Pattern Report

- Sequence Scenario Design Tips

About the Sequence Scenario

There are a variety of detection modules for identifying different types of behaviors, referred to as *scenarios*. Scenarios are clustered into scenario classes, which represent sets of related behaviors that are generally handled in a similar manner. For each scenario that a solution set provides, one or more patterns are specified. These patterns define data relationships that can help to identify instances of the behavior of interest. Using a detection module, business data is searched for records that satisfy the patterns of behavior that a scenario defines. When one or more data records match a scenario's pattern of behavior, a match is created. The data records that triggered the match are associated with it for analysis and audit purposes. These are referred to as matched records in the application.

The Sequence Scenario Editor provides access to the pattern elements that you need to create when using this editor for detecting behaviors of interest. It also provides access to all required features and functionality for creating sequence patterns.

Sequence Scenario Editor Terminology

Table 7 provides a list of terms used to define the concepts of the Sequence Scenario Editor.

Table 7. Sequence Scenario Editor List of Terms

Term	Definition
Augmentation	Process that is executed after the completion of a detection job to supplement the information stored and displayed with each match. Augmentation can be added through bindings or matches. Two augmentation mechanisms are available: - **Binding Augmentation:** Adds information for display in the alert highlights. - **Match Augmentation:** Adds references to additional matched records that are shown on the Alerts Details page on the Web Application.

Table 7. Sequence Scenario Editor List of Terms (continued)

Bound Variable	Variables captured in a pattern. Four types of bindings are available: ● **Bind Before:** Variable is captured *before* checking the constraints ● **Bind After:** Variable is captured *after* checking the constraints. ● **Rebind Before:** Previously bound variable is incremented or updated, and completed *before* checking of constraints. ● **Rebind After:** Previously bound variable is incremented or updated, and completed *after* checking of constraints.
Focus	Entity upon which an alert is based.
Pattern	Defined set of detection logic that is used to search data for behaviors of interest.
Scenario	Entity that represents a behavior of interest within a business.
Scenario Class	Group or category of scenarios that represents related behaviors.
Sequence Scenario Editor	Tool used within the Scenario Manager that enables Data Miners to build and modify sequence scenarios.

Sequence Scenario Editor Components

The Sequence Scenario Editor dialog box has the following components (Figure 17):

- Menu Bar

- Scenario Elements

- Pattern Editor Area
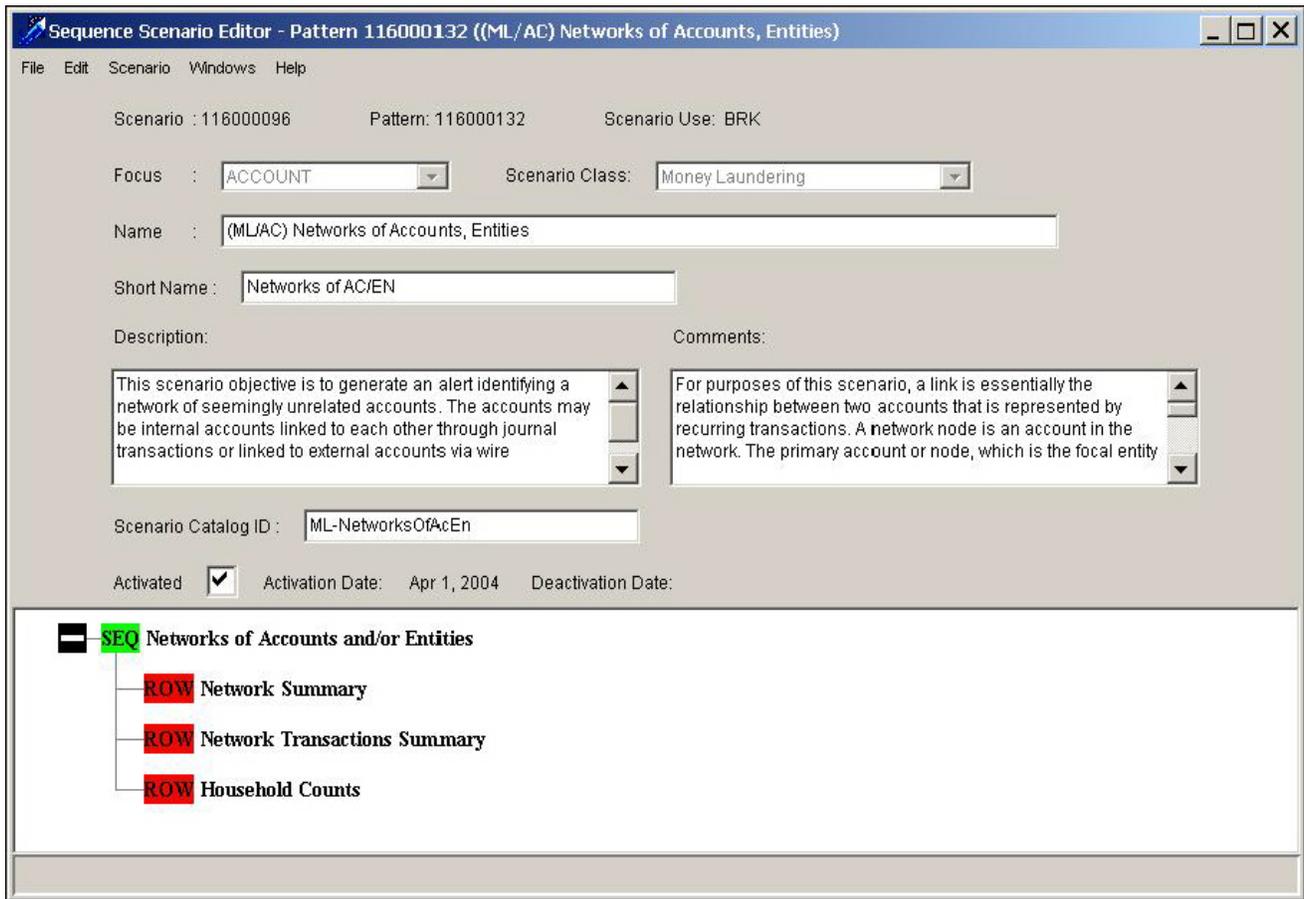
Figure 17. Sequence Scenario Editor

The following sections describe these components.

Menu Bar

The menu bar contains commands and functions as options that you can select in the File, Edit, Scenario, Windows, and Help menus. The following sections describe the Edit and Scenarios menus. Refer to Chapter 2, *Common Features*, on page 3, for information about the Windows, and Help menus.

File menu

File menu contains two options:-

- **Close**: It will close the current window.

- **Exit** : It will close the Scenario Manager application.

Edit Menu

The Edit menu contains the following options:

Note: These options apply only to the Pattern Editor and are not available for any other field in the Sequence Scenario Editor.

- **Cut:** Deletes the selected pattern component.

- **Copy:** Copies the selected pattern component.

- **Paste:** Pastes a previously-copied or cut pattern component before or after the selected component.

- **Delete:** Deletes the selected pattern component.

- **Modify:** Opens the Properties dialog box for the pattern component you have highlighted, which enables to you make and save changes as desired.

Scenario Menu

The **Scenario** menu contains the following options:

- **Create New Scenario:** Creates a new sequence pattern.

- **Open Existing Scenario:** Opens a scenario.

- **Copy Scenario:** Copies the active pattern into a new scenario.

- **Copy Pattern:** Copies a pattern from within a scenario.

- **Save:** Saves the current scenario with any changes you have made.

- **Save As:** Saves a copy of the open pattern with a unique ID.

- **Revert:** Reverts to the saved version of a pattern within a scenario.

- **Export:** Exports a sequence pattern to an XML file.

- **Import:** Imports a sequence pattern from a previously-exported scenario (in XML format).

- **Print:** Prints the selected scenario pattern.

- **DataSet:** Adds, deletes, or reorders datasets associated with the sequence pattern.

- **Add Sequence:** Adds a sequence to the pattern before, after, or as a child to the selected pattern element.

- **Add Row:** Adds a row to the pattern before, after, or as a child to the selected pattern element.

- **Add Alternation:** Adds an *or* operator to the sequence pattern. You can add the *or* operator before, after, or as a child to the selected pattern element.

- **Edit Roles:** Defines the variables, format, and display type of Scenario attributes.

- **Edit Augmentation Rules:** Creates augmentation rules to supplement the information stored and displayed with each match. (Refer to Chapter 8, *Augmentation,* on page 145, for more information.)

- **Generate Pattern Report**: Generates an HTML file that contains detailed information about the sequence pattern. Refer to section section *Guidelines for Generating a Pattern Report*, on page 64, for more information.

- **Edit Threshold Analyzer Roles**: Defines the Threshold Analyzer Roles. Refer to section section *Defining Threshold Analyzer Role*, on page 62, for more information.

Scenario Elements

The Scenario Elements area contains the information that you need to define a sequence scenario (Figure 18).

Figure 18. Sequence Scenario Editor—Scenario Elements Area

The Scenario Elements area contains the following fields:

- **Focus:** Indicates the focal entity for a scenario; the focal entity key ID (for example, an account ID) must be captured in the pattern.

- **Scenario Class:** Specify the scenario class for this scenario. A scenario class groups scenarios together based on similar behaviors of interest.

- **Name:** Defines the long version of the scenario name, which is not displayed on the Web Application.

- **Short Name:** Specifies the abbreviated name for the scenario, which is shown on the Web Application.

- **Scenario Class**: Specify the scenario class for this scenario. A scenario class groups scenarios together based on similar behaviors of interest.

- **Description**: Specify information that is pertinent to the scenario, this serves as the description in a generated pattern report.

- **Comments:** Specifies additional information that you want to include in a scenario.

- **Scenario Catalog ID:** Identifies the unique ID of a scenario.

- **Activated:** Displays the current status of the open scenario. Activated status is the default; to deactivate, click the check box next to the *Activated* field.

Pattern Editor Area

The Pattern Editor area contains the information that you need to define a pattern for a sequence pattern (Figure 19).
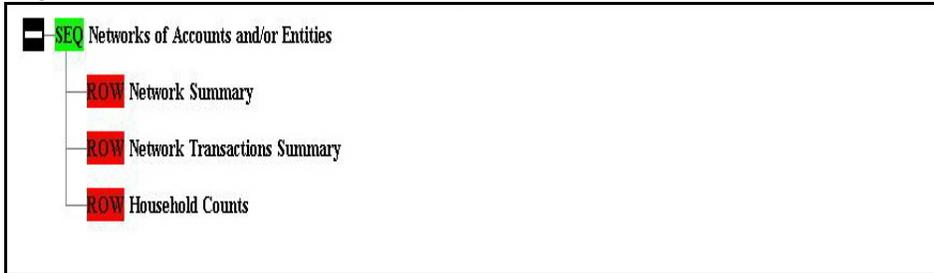
Figure 19. Sequence Scenario Editor-Pattern Editor Area

The Pattern Editor area contains the following elements:

- **Sequence:** Serves as a header for a group of events in the pattern. Every pattern starts with a sequence. Additional sequence blocks can be used for sub-series of activities that must happen in a particular order.

- **Row:** Specifies an event in sequence pattern.

- **Alternation:** Provides an *or* operator for the pattern. Use this when more than one events/sequence can satisfy the conditions of the pattern optionally.

Using the Sequence Scenario Editor

Features in the Sequence Scenario Editor enable you to:

- Create a scenario (refer to section *Creating a Scenario*, on page 45, for more information).

- Define scenario usage (refer to section *Defining Scenario Usage Information*, on page 46, for more information).

- Delete a sequence scenario (refer to section *Defining Basic Elements of a Sequence Scenario*, on page 47, for more information).

- Specify a dataset for a sequence scenario (refer to section *Selecting a Dataset for a Sequence Scenario*, on page 47, for more information).

- Define a scenario pattern (refer to section *Defining the Scenario Pattern*, on page 50, for more information).

Creating a Scenario

The Sequence Scenario Editor provides access to all functionality required for creating or modifying scenarios.

Follow the below steps to create a scenario.

- Defining Scenario Usage Information

- Defining Basic Elements of a Sequence Scenario

- Selecting a Dataset for a Sequence Scenario

- Editing Tags

- Using DataSet Attributes for Sorting

- Defining the Scenario Pattern

- Defining General Tab Properties for a Row

- Defining Constraints Tab Properties for a Row

- Defining Bindings Tab Properties for a Row

- Defining Pattern Roles

- Saving a Scenario

Note: You must perform these sequential procedures in the order that they appear.

Defining Scenario Usage Information

When you Create a New Scenario through the Scenario menu from the Sequence Scenario Editor dialog box, you must first specify the type of environment in which the new scenario is used. Following is a brief description of each possible option (available options are based on the scenarios you have chosen to include in your application and can include any or all of these):

- **BRK (Break mode):** Standard for production scenarios; all changes are tracked (no pattern can be modified without creating a new ID).

- **EXP (Experimental mode):** Used for development; enables a data miner to save changes without generating a new ID.

- **SYS (System):** Not used by scenario developers; applies to system scenarios. The **SYS** scenario code is used for user defined alerts. It is displayed in the scenario manager because it is a part of set of possible scenario use codes and is thus in the metadata. However, it is never used for scenarios as it is only attached to alerts that were defined by a user and not generated by scenarios. Only BRK and EXP should be used in the Scenario Manager when creating scenarios.

Scenarios are automatically changed from EXP to BRK mode when they are extracted using the Scenario Migration Utility. Scenario mode can also be adjusted by updating the KDD_SCNRO.SCNRO_USE_CD table. However, we do not recommend using this manual adjustment.

To define scenario usage information, follow these steps:

1. From the **Windows** menu, click **Sequence Patterns**.

   The Sequence Scenario Editor displays (Figure 17).

2. From the **Scenario** menu of the Sequence Scenario Editor, select **Create New Scenario**.

   The Select Use dialog box displays (Figure 20).

Figure 20. Select Use dialog box

3. Highlight your choice and click **Select**.

   A blank Sequence Scenario Editor dialog box displays (Figure 17).

Defining Basic Elements of a Sequence Scenario

To create a new sequence scenario, you must define some information in a specified order to successfully create a scenario (refer to section *Scenario Elements*, on page 43, for more information).

To define the basic elements of a sequence scenario, follow these steps:

1. From the **Windows** menu, select **Sequence Patterns**.

   The Sequence Scenario Editor displays (Figure 17).

2. From the Scenario menu of the Sequence Scenario Editor, select **Create New Scenario**.

3. Define the scenario usage for the scenario (refer to section *Defining Scenario Usage Information*, on page 46, for more information).

4. Select the focal entity of your scenario from the **Focus** drop-down list

5. Select the scenario class under which you want your scenario classified from the **Scenario Class**.

6. Type a unique name for the scenario in the **Name** field.

7. Type a unique short name for the scenario in the **Short Name** field.

8. Provide a description of the scenario (up to 255 characters) in the **Description** field.

9. Add any additional comments in your scenario in the **Comments** field.

10. Type a Catalog ID for the scenario in the Scenario Catalog ID field.

The Activation Information area displays that the scenario is Activated (the default setting). To change the status, select the check box next to the **Activated** field. The Name, Short Name, Description, Scenario Catalog ID, and Comments fields are grayed out so that you cannot enter any data (Figure 17).

Selecting a Dataset for a Sequence Scenario

Refer to section *About the DataSet Editor*, on page 9, for additional information about datasets that you can create. To define a dataset for your sequence scenario successfully, you must select a dataset and then define the sort order of the attributes within the dataset.

The following sections describe these procedures:

To select a dataset for a sequence scenario, follow these steps:

1. From the **Windows** menu, select **Sequence Patterns**.

   The Sequence Scenario Editor displays (Figure 17).

2. From the **Scenario** menu of the Sequence Scenario Editor, select **Open Existing Scenario**.

3. Select a scenario from the Scenario/Pattern dialog box.

4. From the **Scenario** menu of the Sequence Scenario Editor, select **Dataset**.

   The Sequence Dataset Editor dialog box displays (Figure 21).
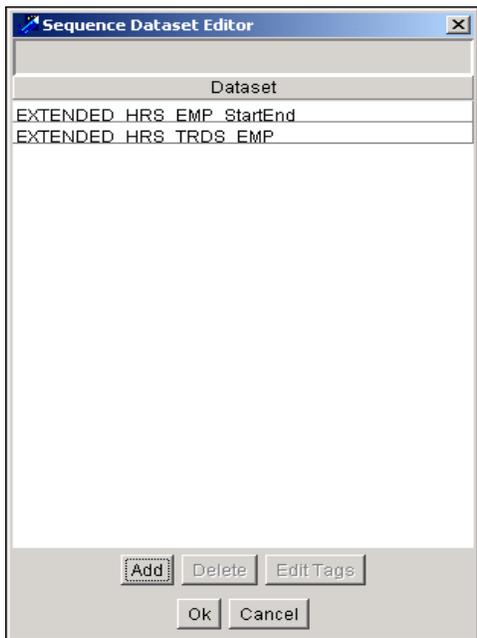
Figure 21. Sequence Dataset Editor Dialog Box

This dialog box lists the datasets that compose the current sort. You can filter the list of datasets to include:

- Wrapper datasets only (all wrapper datasets are realized)

- Wrapper datasets and any realized, derived datasets

- All datasets (wrapper and derived)

5. Click **Add** in the dialog box to view available datasets.

6. Click the dataset you want to add to your sort.

7. Click **Ok**.

   The selected dataset name displays in the Sequence Dataset Editor dialog box.

You can now define the sort for your pattern through the Edit Tags function.

Editing Tags

Editing tags involves defining the sort order of datasets through the use of attributes within one or more datasets. You can use multiple datasets to define sort as long as all of the datasets contain at least one *like* attribute. The *like* refers to "same type". In order to define a sort order between datasets, each dataset must have a column used in the sorting that have a same type across all datasets. For example, if you were defining the sort order across three datasets and wanted to sort by date, the selected sort attribute for each of the datasets must have a common type (for example, DATE). If one of the dataset do not have a DATE column that the user wanted to sort by, then such a sort could not be performed across the datasets. If you use only one dataset, the system defines the sort order by your selected attribute. If, however, you use multiple datasets, the sort order is based on the dataset order that you establish in your sequence pattern.

To edit tags, follow these steps:

1. From the **Windows** menu, select **Sequence Patterns**.

   The Sequence Scenario Editor displays (Figure 17).

2. From the Scenario menu of the Sequence Scenario Editor, select **Open Existing Scenario**.

3. Select a scenario from the Scenario/Pattern dialog box

4. From the **Scenario** menu in the editor, select **DataSets**.

   A list of datasets displays that you previously added to your sort.

5. Select a dataset name.

6. Click **Edit Tags**.

   The Pattern DataSet Editor dialog box displays with blank fields (Figure 22).
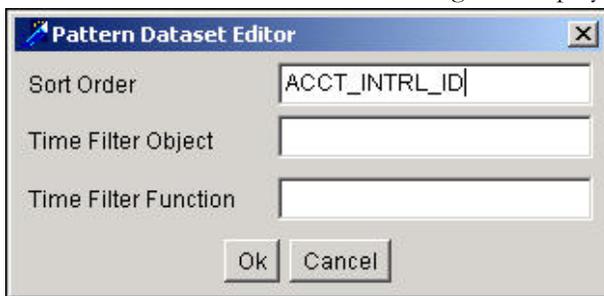
Figure 22. Pattern Dataset Editor Dialog Box

7. Type the name of the attributes that you want to use for your sort in the **Sort Order** field.

   If you use more than one attribute, separate each one by a comma with no spaces.

   Note: The application does not use the Time Filter Object and Time Filter Function fields.

8. Click **Ok**.

You can now define your scenario pattern.

Using DataSet Attributes for Sorting

Pattern datasets identify the datasets that you want to use in your pattern. They allow you to specify how to sort your datasets. Using the Edit Tag function to define sort order, use the following guidelines:

- You can specify multiple attributes (comma-separated) when defining the sorting sequence.

- You can intermingle multiple datasets based on the sorting attributes; the dataset with the fewest sorting attributes drives the extent of inter-dataset mixing. The sorting algorithm uses the lowest common denominator approach. For example, if you have two datasets with the following attributes and the following sort orders:

  DATASET1: INT1, FLOAT1, STRING1

  SORT ORDER: INT1, FLOAT1, STRING1

  DATASET2: INT2, FLOAT2, STRING2

  SORT ORDER: INT2, FLOAT2

  The sorting will only sort by INT1/INT2 and FLOAT1/FLOAT2 leaving STRING1/STING2 since, a similar attribute does not exist in DATASET2 defined sort order.

- Use the same data type for sorting attributes in each dataset.

- Arrange the datasets so that the system handles tie-breaks correctly; invert the list if you want to run the job in reverse. In cases where a sorting attribute has the same value in two datasets, the order of retrieval is based on the dataset order.

Defining the Scenario Pattern

To define your scenario patterns successfully, complete the procedures that the following sections describe:

- Defining the Sequence of a Scenario

- Defining the Alternation for a Scenario

- Adding Rows to a Scenario

Defining the Sequence of a Scenario

You can configure two types of sequences through the Sequence Scenario Editor: Outer Sequences and Sub-sequences. Whether you create a new scenario or open an existing one, the top of the pattern in the Pattern Editor field is *SEQ* [DESCRIPTION]. This is considered the Outer Sequence for a scenario. Any subsequent sequences that you add are referred to as Sub-sequences, since they are considered *children* to the Outer Sequence. For each type of sequence, you need to define specific properties as you add them.

Outer Sequence Properties

You can define six properties of an Outer Sequence through the Sequence Properties dialog box (Figure 23).
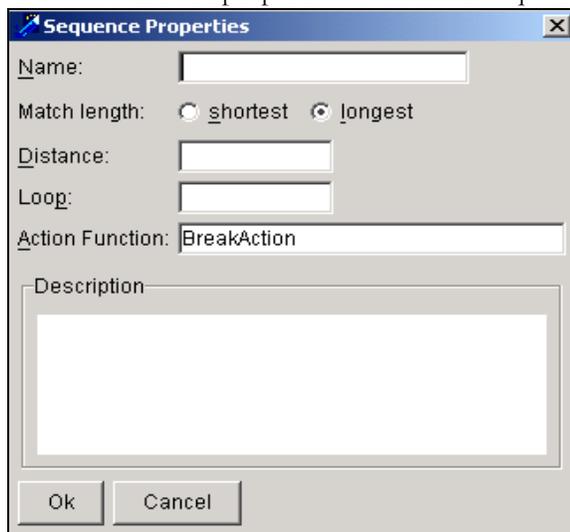
Figure 23. Sequence Properties Dialog Box

These properties include the following:

- **Name:** Defines the description that appears in the Sequence Scenario Editor as the sequence name.

- **Match Length:** Indicates a match of longest or shortest possible series of events.

  - **Shortest:** Match with fewest matching events**.**

  - **Longest:** Match with most matching events (the default).

Only one match is created from any starting event. Multiple matches that start at the same point indicate whether the system is to return the shortest or longest match.

- **Distance:** Is not currently used.

- **Loop:** Indicates the number of times the sequence can be found. The loop count can be a specific number (for example, two or six) a bound range (for example, one to ten), or an open-ended range (for example, one or more).

  Proper syntax for this field is:

  number1 - number2

  Examples of valid entries are:

  - **2:** Sequence must be matched exactly twice.

  - **1-5:** Sequence can be found between one and five times.

  - **-10:** Sequence can be found up to 10 times.

- **Action Function:** Displays the action you want the system to perform if a match is found for a sequence. Currently, the only option for this field is **BreakAction**, which is the default value. The possible matching functions behavior are as follow:

  - **BreakAction:** Creates a match for the longest (or shortest) match starting with each event.

  - **MultiSupportBreakAction:** Creates matches only if BreakAction would create N or more matches. Specify N by binding a variable SUPPORT (through GUI) to a number. Creates multiple matches.

  - **SingleSupportBreakAction:** Creates a match only if BreakAction would create N or more matches, and groups the matches together into one match.

- **Description:** Provides additional descriptive information about the sequence.

- **OK**: Click to confirm

- **Cancel**: Click to cancel the action

Sub-sequence Properties

When you double-click a secondary (child) sequence name within a scenario, the Sequence Properties dialog box displays. In this dialog box, you can define the properties. Two required properties for a sub-sequence include:

- Loop
- Description

Refer to section *Outer Sequence Properties*, on page 50, for definitions of each.

Note: This process assumes that you completed the procedures for *Defining Basic Elements of a Sequence Scenario*, on page 47, and created a dataset. Refer to Chapter 3, *DataSet Editor,* on page 9, for procedures for creating datasets.

To define a Sequence or Sub-sequence, follow these steps:

1. From the **Windows** menu, select **Sequence Patterns**.

   The Sequence Scenario Editor displays (Figure 17).

2. From the Scenario menu of the Sequence Scenario Editor, select **Create New Scenario or Open Existing Scenario**.

3. Select a scenario from the Scenario/Pattern dialog box if **Open Existing Scenario** is used. For Create New Scenario, refer to section *Creating a Scenario*, on page 45.

4. Double-click the **SEQ [DESCRIPTION]** string in the **Pattern Editor** field, in the Sequence Scenario Editor.

   The Sequence Properties dialog box displays (Figure 23).

5. Type the string name in the **Name** field as you want it to display in the **Pattern Editor** field.

6. Click **Ok**.

   The Pattern Editor field in the Sequence Scenario Editor dialog box, now contains the name that you specified in Step 2 (Figure 17).

7. Click **Scenario**, highlight **Add Sequence**, and then select **As Child** to add another sequence to your pattern.

8. Double-click the sequence you just added.

9. Type the desired name in the Pattern Editor in the **Description** field.

10. Define the remaining optional fields (Distance, Loop, and Prefix) as desired.

11. Add Sub-sequences, as desired; complete all of the above steps for each.

Refer *Sub-sequence Properties*, on page 51, for information about field requirements.

Note: The depreciated **Distance** parameter is no longer used. It appears in the editor only to provide compatibility with previous versions of the Scenario Manager..

Defining the Alternation for a Scenario

An Alternation provides an '`or`' operator for the pattern within a sequence. This is used when more than one event or sub-sequence satisfies the conditions of a pattern. An alternation is always a child to a sequence block or another alternation, and is composed of a description that is used in the Editor.

To define the Alternation for a scenario, follow these steps:

1. Highlight the sequence to which you want to add a child alternation (you can also add an alternation to another alternation).

2. Select the **Scenario** menu and highlight **Add Alternation**.

3. Click **Before**, **After**, or **As Child** to the selected sequence or alternation.

   The **OR [DESCRIPTION]** string displays in the Pattern Editor dialog box.

4. Double-click the string.

5. In the Pattern Editor dialog box, type the desired name in the **Description** field.

Adding Rows to a Scenario

A row represents the events in a pattern. Access the properties of a row via three tabs: General, Constraints, and Bindings. The following sections defines each of these tabs.

To add a row to a scenario, follow these steps:

1. From the **Windows** menu, select **Sequence Patterns**.

   The Sequence Scenario Editor displays (Figure 17).

2. From the Scenario menu of the Sequence Scenario Editor, select **Create New Scenario** or **Open Existing Scenario**.

3. Select a scenario from the **Scenario/Pattern** dialog box if **Open Existing Scenario** is used. For Create New Scenario, refer to section *Creating a Scenario*, on page 45.

4. Click the string in the **Pattern Editor** field in the Sequence Scenario Editor to which you want to add a row to highlight the string.

5. From the **Scenario** menu, select **Add Row**, then select **Before, After or As Child**.

   The string **ROW [DESCRIPTION]** displays in the Pattern Editor field.

6. Repeat these steps for each row that you want to add.

   You can now define general information, constraints, and bindings for each added row.

Defining General Tab Properties for a Row

The **General** tab shows basic properties of a pattern row (Figure 24).
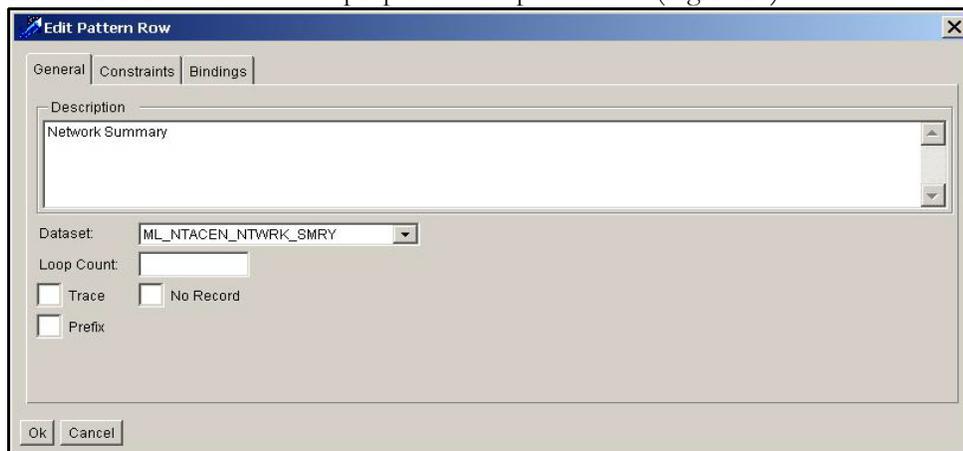
Figure 24. Edit Pattern Row Dialog Box-General Tab

The **General** tab contains the following components:

- **Description:** Displays the name of the row that display in the Editor.

- **DataSet:** Specifies the dataset in which the event must occur.

- **Loop Count:** Indicates the number of times the event may occur in succession. The default value is 1 if the value is left null. Proper syntax is the same as the expressions used for sequences (for example, 1-2 means one or two, 0- implies zero or more).

- **Trace:** Is not used and must remain unselected.

- **No Record:** Indicates whether the key ID for this event is recorded, which results in the Web Application including a building block.

  By default, the system records all rows. Checking this box keeps the record from appearing on the Alert Details page of the Web Application.

- **Prefix:** Indicates application of a bidirectional search. This technique is infrequently used and often not efficient.

- **OK**: Click to confirm

- **Cancel**: Click to cancel the action

To define general information about a row, follow these steps:

1. From the **Windows** menu, select **Sequence Patterns**.

   The Sequence Scenario Editor displays (Figure 17).

2. From the Scenario menu of the Sequence Scenario Editor, select **Create New Scenario** or **Open Existing Scenario**.

3. Select a scenario from the **Scenario/Pattern** dialog box if **Open Existing Scenario** is used. For Create New Scenario, refer to section *Creating a Scenario*, on page 45.

4. Double-click the row you want to define in the Sequence Scenario Editor.

   The Edit Pattern Row dialog box displays, showing the General Tab fields (Figure 24).

5. Type a description of the row in the **Description** field.

6. Click the drop-down menu in the **DataSet** field and select the source dataset for the attributes used in this row pattern. This drop-down menu lists all applicable datasets.

7. Type the loop parameter in the **Loop** field to define the number of times you want a sequence to run. The default value is 1.

8. Do *not* click the **TRACE** check box (this is not used and must remain unselected).

9. Click the **No Record** check box to indicate that you do *not* want the data record that satisfied this row condition to be recorded if a match occurs. If recorded, it results in a building block which the **Alerts Display** page contains.

10. Select the **Prefix** check box to enable bidirectional searching. It allows you to look at the data both before and after an event of interest (for example, a large and possibly market-moving event being placed). This is a performance issue because essentially you have to make two passes through the data stream, one forward and one backward. You can accomplish the similar behavior without using this feature by using the time functions to artificially shift the time.

    Note: We do not recommend doing this as it is an inefficient and time-consuming way to perform a search. It is only used when you need to look at data both before and after an event of interest.

Defining Constraints Tab Properties for a Row

The Constraints Tab properties identify the constraints or conditions applied to the row. This is a data display and entry field. You can type entries or use the **Add Variable** and **Add Attribute** buttons to select variables and expressions for use in this field. When you select a variable or expression in either dialog box, it appears at the cursor point in the text box.

Proper Syntax Values for Constraint

Table 8 provides the proper syntax for defining constraints in a row.

Table 8. Row Constraints Syntax

Parameter	Syntax
Relational Operators	==, !=, >, ≥, <, ≤
Mathematical Operators	+, -, *, /
Logical Operators	&&, ||, !
Identification of null strings	isnull(*<attribute>*)

Displaying Sequence Scenario Row Constraints

Identify accounts with concentrated positions in a single security via type. Following is an example of a row constraint:

```
HI_CNC_SCRTY_VAL_AM ³ @CONC_POSN_AM &&
HI_CNC_SCRTY_EQTY_PT ³ @CONC_PT &&
MRGN_EQTY_PT f @EQTY_PT
Expression of negation of this condition can be:
HI_CNC_SCRTY_VAL_AM < @CONC_POSN_AM ||
HI_CNC_SCRTY_EQTY_PT < @CONC_PT ||
MRGN_EQTY_PT > @EQTY_PT
```

or

```
!(HI_CNC_SCRTY_VAL_AM ³ @CONC_POSN_AM &&
HI_CNC_SCRTY_EQTY_PT ³ @CONC_PT &&
MRGN_EQTY_PT f @EQTY_PT)
```

Use the following guidelines when defining constraints:

- Capture strings in double quotes.

- Use parentheses for expressions that need to be evaluated together.

- Prefix variable names with a dollar sign ($).

Prefix threshold names with an at sign (@).

To define constraints for a row, follow these steps:

1. In the Edit Pattern Row dialog box, click the Constraints tab (Figure 24).

   The Edit Pattern Row or Constraints Tab dialog box displays (Figure 25).
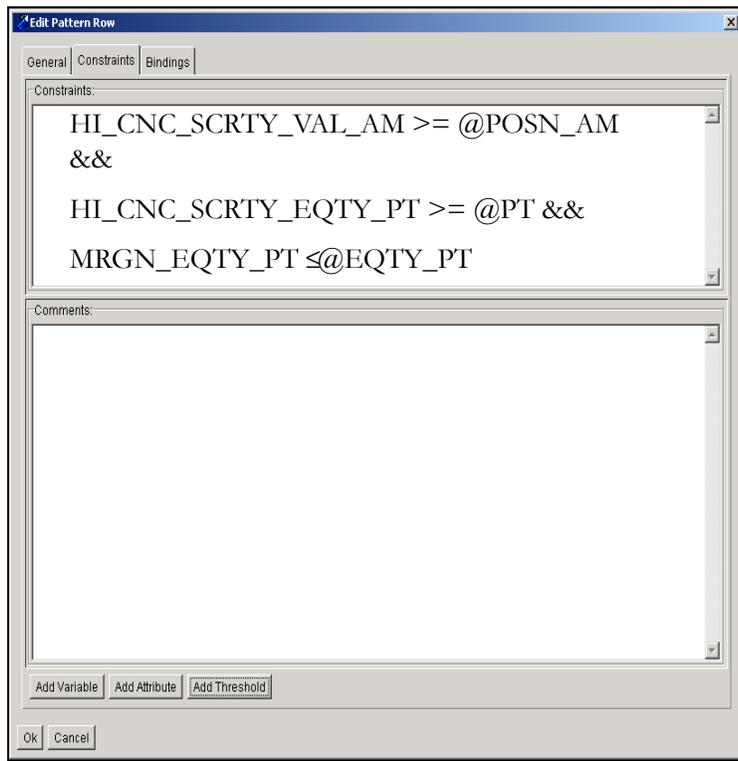
Figure 25. Edit Pattern Row Dialog Box-Constraints Tab

2. Click the **Add Variable** and **Add Attribute** buttons to select variables and expressions.

3. Select the name of the variable or attribute in the applicable dialog box.

4. Click **OK** to add.

5. If you added a value incorrectly, highlight the string you want to remove and press the **Delete** button on your keyboard.

Note: The Comments section provides users or developers to document their work for their benefit or for others benefit.

Defining Bindings Tab Properties for a Row

Use the **Bindings** tab of the Edit Pattern Row dialog box to define any variables you want to capture in a specified pattern (Figure 26).
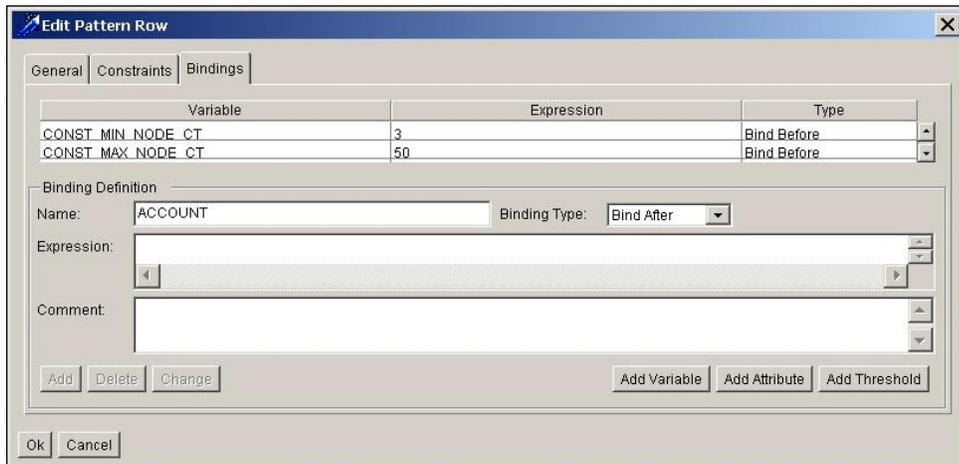
Figure 26. Edit Pattern Row Dialog Box - Bindings Tab

Standard uses of bound variables can include:

- Setting constants to refer to in pattern constraints.

- Storing information from one record to compare against another.

- Tracking running counts or aggregate amounts across multiple records.

- Capturing information for display in highlights.

Bindings Tab Components

The Bindings tab includes the following components:

- **Variable:** Identifies a variable to be bound in the current row. This is a data display and entry field.

- **Expression:** Specifies an expression to which the current variable is bound. This field can be either a fixed value (for example, 400000), a dataset attribute, a computational formula, or a conditional expression. The expressions can also use other bound variables and thresholds if they are already defined.

- **Type:** Indicates that the current variable is bound before or after the constraint is matched.

- **Binding Definition** area: Includes the following fields:

   ■ **Name:** Identifies the binding.

   ■ **Binding Type:** Indicates that the current variable is bound before or after the constraint is matched. Also indicates whether this is the first time this variable is being bound or whether it has been bound before (Rebind). Options for selection of this drop-down menu include the following:

      ◆ **Bind Before:** Captures the variable before checking constraints.

      ◆ **Bind After:** Captures the variable after checking constraints.

      ◆ **Rebind Before:** Increments or updates a variable that was previously bound; the system does this before checking constraints.

◆ **Rebind After:** Increments or updates a variable that was previously bound; the system does this after checking constraints.

● **Expression**: Enables you to add or edit a binding expression.

● **Comment**: Enables you to add or edit optional comments about the selected binding.

● **Add:** Adds the current variable to the variable list.

● **Delete:** Deletes the current variable.

● **Change:** Changes the values stored for the currently selected variable.

● **Ok:** Applies any modifications to the row.

● **Cancel:** Cancels any modifications and returns you to the Sequence Scenario Editor.

● **Add Variable**: Opens the Variable Selector dialog box, in which you can use another binding to define the current binding.

● **Add Attribute:** Opens the Attribute Selector dialog box, in which you can add attributes from the current dataset for use in defining the current binding.

● **Add Threshold:** Opens the Threshold Selector dialog box, in which you can add a threshold from the current scenario for use in defining the current binding.

Proper Syntax Values for Conditional Bindings

Proper syntax for a conditional binding is:

```
cond(<condition 1>;<value 1>;<condition 2>;<value 2>; ... ;<default value>)
```

For example, the variable AC_CT can be rebound to the following expression:

```
cond(ACCT_INTRL_ID==$ACCOUNT; $AC_CT; $AC_CT + 1)
```

In this example, the variable AC_CT is set equal to its current value if the condition is satisfied (that is, if the attribute ACCT_INTRL_ID equals the variable $ACCOUNT and is set to (AC_CT+1)).

Guidelines for Defining Bound Variables

Use the following tips when using bound variables:

● Initialize variables with a Bind Before or Bind After before using them in a Rebind statement.

● Ensure that the pattern passes through the row in which the necessary variables are bound; this may necessitate binding of variables on both sides of an Alternation.

● Initialize variables to a value of the appropriate data type (for example, use a number if the value is intended to be numeric).

Refer to Appendix A, *Scenario Design Guidelines*, on page 175, for more information about design tips and guidelines.

To Define Bindings in a Row

To define bindings in a row, follow these steps:

1. Click the Bindings tab.

   The Edit Pattern Row dialog box, which contains the Bindings tab fields displays (Figure 26).

2. Type the name of the variable to be bound to the selected row in the **Name** field. The specified attribute must be in all of the previously selected datasets.

3. Click the drop-down in the **Binding Type** field to select whether you want the row to be bound before or after matching the constraint (refer to section *Defining Bindings Tab Properties for a Row*, on page 56, for more information).

4. Type additional information you want to associate to the selected row in the **Comment** field.

5. Click **Add**.

   You have successfully added binding conditions to your row.

Defining Pattern Roles

Pattern roles specify the bound variables that are saved after creating a match. The system stores saved variables in the KDD_BREAK_BINDING table. It stores all instructions defined in the Pattern Roles Editor in the KDD_PTTRN_ROLE table.

To use variables for highlights, augmentation, or scoring, you must add them to Pattern Roles via the Pattern Roles Editor (Figure 27).
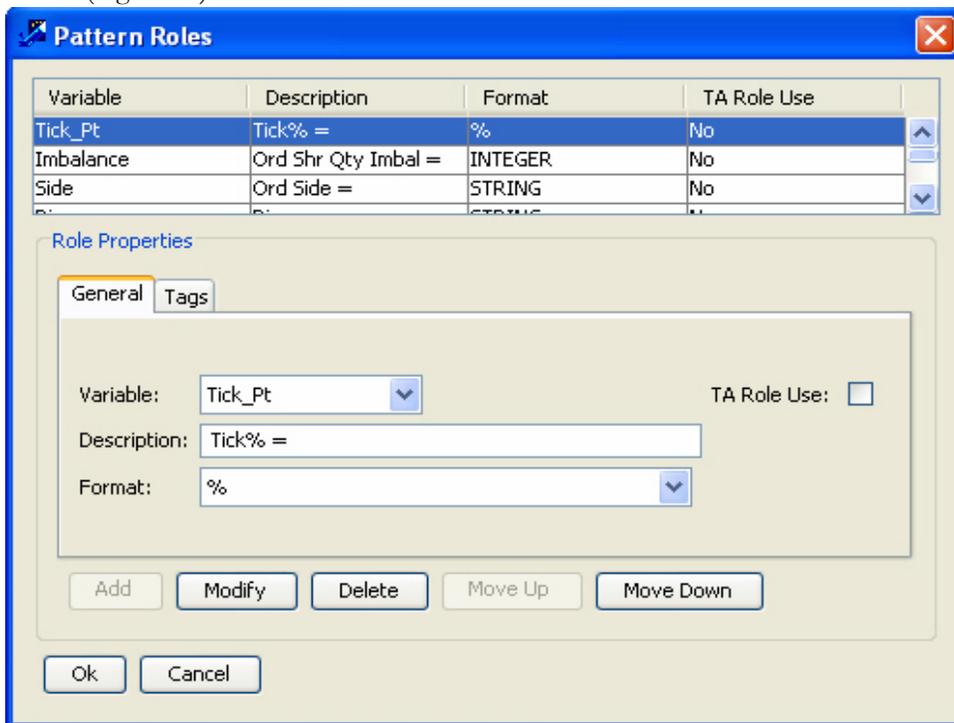
Figure 27. Pattern Roles Dialog Box

Defining highlights requires a format string; all highlight variables appear on the **Alerts Display** page in the order that you define them.

Pattern Role Components

When defining patterns, specify the following:

- **Variable:** Defines the variable to display as an alert.

- **Description:** Specifies the text you want associated with a specific role.

- **Format:** Identifies the highlights from other pattern roles and defines the display of the pattern roles in the Web Application UI.

- **TA Role Use: S**pecifies if a pattern role variable should be available for the threshold analyzer role.

Format Options for Pattern Roles

Table 9 describes the most commonly used format options for this field (case-sensitive).

Table 9. Pattern Role Format Field Options

Format Option	Description
STRING	Does not apply formatting to the value returned from the database. This format is also useful for numbers that should not have thousands separators, for example, numeric account numbers.
INTEGER	Applies thousands separators to whole numbers.
REAL2	Applies thousands separators to real numbers, truncating to two digits of fractional data.
REAL5	Applies thousands separators to real numbers, truncating to five digits of fractional data.
$.00	Applies thousands separators and a currency symbol, e.g., $, to a number. Fractions are displayed. For example, a positive number is displayed as $1,234.56 and a negative number is displayed as ($1,234.56).
$	Applies thousands separators and a currency symbol to a number. Fractions are rounded to the nearest whole number. For example, positive numbers are displayed as $1,234 and negative numbers are displayed as ($1,234).
$K	Applies thousands separators and a currency symbol to a number. Fractions are rounded to the nearest whole number. If the number is 1,000,000 or greater, the system rounds the number to the nearest thousand, and replaces the final three digits with the letter *K*. For example, positive numbers display as $1,234 or $2,346K for $2,345678.90, while negative numbers display as ($1,234) or ($2,346K) for -$2,345678.90.
CURRENCY	Applies the appropriate currency format (base currency of installation) to a value.
MMDDYY	Displays the short form of the date, for example, 02/28/01.
MMDDYYYY	Displays the long form of the date is displayed, for example, 02/28/2001.
DATETIME	Displays the short date format and time format in the 24-hour clock, for example, 02/28/01 23:59:59.
HHMMSS	Displays the long time format in 24-hour clock. The expected input is an integer that represents a number of seconds. For example, the display of 93599 seconds is 25:59:59.
HHMMSS_PT	Displays the long Time format in 24-hour clock with milliseconds precision. For example, the display of 103001500 will be as 10:30:01:500

Table 9. Pattern Role Format Field Options (continued)

Format Option	Description
%	Displays the data as a percentage.
ALT_CUR	Used for cases where all highlights/roles are of the same currency, but NOT base currency. The format for the amount(s) should be ALT_CUR. Then, the scenarios will record an additional binding with binding name Alt_Currency_Code that holds the currency code. The highlights generator would then/take find the value from the Alt_Currency_Code binding and use it to format all of the bindings with format ALT_CUR. For example: *OrderPrice* role with format ALT_CUR and Alt_Currency_Code binding with value as ISSUE_CRNCY_CD from pattern, let's say value is EUR. Thus *OrderPrice* will be displayed as xxx EUR. Here xxx is the value coming from pattern.
REF_CUR	Used for cases where all highlights/roles are not of the same currency (including base currency) or different bindings having different currencies to display, use a format code REF_CUR for the amount, and then have an additional binding with the same name, but with _CUR on the end and contain the currency to be used. For example, define *TradePrice* and *ActivityPrice* roles with format REF_CUR and TradePrice_CUR, and ActivityPrice_CUR binding to hold the Currency to be used for the above, say EUR and USD respectively. Thus they will be displayed as *TradePrice : xxx EUR & ActivityPrice : zzz USD*. Here xxx and zzz are the values coming from pattern.

Guidelines for Defining Pattern Roles

With a new sequence scenario pattern where no information is provided, the Pattern Roles dialog box is blank. With an existing sequence scenario pattern, the Pattern Roles dialog box includes a list of Pattern Roles that Variable, Description, TA Role Use, and Format Identify.

Use the following guidelines in the Pattern Roles editor:

- Add variables to pattern roles if using them for highlights, augmentation, or scoring.

- Providing a format string results in use of the variable as a highlight.

- Arrange the variables in the order they should appear in the highlights using the **Move Up** and **Move Down** buttons.

- Create a Pattern Roles entry for the focal entity binding. This variable has the same name as the scenario focus type (for example, ACCOUNT).

Refer to Appendix A, *Scenario Design Guidelines*, on page 175, for more information about design tips and guidelines.

To define pattern roles, follow these steps:

1. From the **Windows** menu, select **Sequence Patterns**.

   The Sequence Scenario Editor displays (Figure 17).

2. From the Scenario menu of the Sequence Scenario Editor, select **Create New Scenario** or **Open Existing Scenario**.

3. Select a scenario from the Scenario/Pattern dialog box if **Open Existing Scenario** is used. For Create New Scenario, refer to section *Creating a Scenario*, on page 45.

4. From the **Scenario** menu in the Sequence Scenario Editor dialog box, select **Edit Roles**.

   The Pattern Roles dialog box displays (Figure 27).

5. Click the drop-down menu in the **Variable** field and select the variable that you want to display as an alert.

6. Type the text you want associated with this role in the **Description** field.

7. Click the drop-down menu in the **Format** field to identify the highlights from other pattern roles, and to define the display of the pattern roles on the Alerts Display page (refer to section *Format Options for Pattern Roles*, on page 60, for more information).

   Optional: Select the **TA Role Use** check box if you want to assign pattern role variable to a threshold analyzer role.

8. Click **OK**.

Defining Threshold Analyzer Role

You can define Threshold Analyzer Role from the Sequence Scenario Editor..

To define Threshold Analyzer Role using Sequence Scenario Editor, follow these steps:, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Sequence Patterns**.

   The Sequence Scenario Editor window displays (Figure 17).

2. From the **Scenario** menu of the Sequence Scenario Editor, select **Open Existing Scenario**.

3. From the **Scenario** menu, select **Edit Roles**.

   The Pattern Roles dialog box displays (Figure 27).

4. In the Pattern Role dialog box, follow these steps:

   a. Select the desired variable from the **Variable** drop-down list.

   b. Type the text you want associated with this role in the **Description** field.

   c. Click the arrow of the **Format** drop-down list to identify the highlights from other pattern roles, and to define the display of the pattern roles on the Web Application pages.

      Optional: Select the **TA Role Use** check box if you want to assign pattern role variable to threshold analyzer role.

   d. Click **Modify**.

5. Click **Ok** to close the Pattern Roles dialog box.

6. From the **Scenario** menu of the Sequence Scenario Editor, select **Edit Threshold Analyzer Roles**.

   The Threshold Analyzer Roles dialog box displays (Figure 54).

7. In the Threshold Analyzer Roles dialog box, follow these steps:

   a. Select the desired variable from the **Variable** drop-down list.

   b. Modify the text you want associated with this role in the **Description** .

   c. Select the format from the **Format** drop-down list.

    d. Select the field position from the **Variable Field Name** drop-down list. By default, the value specified is blank.

    e. Select the value from the **Use as Axis** drop-down list.

    f. Type the unit in the **Display Units**.

    g. Select the numeric value from the **Round Digits Count** drop-down list.

    h. Click **Add**.

8. Click **Ok** to close the Threshold Analyzer Roles dialog box.

Saving a Scenario

After completing configuration of scenario, make sure to save the information.

To save a scenario, follow these steps:

1. From the **File** menu, select **Close**.

    A Warning dialog box displays (Figure 28).

Figure 28. Save Scenario Dialog Box

2. Click **Save** and close the scenario.

    The system saves the scenario with the name you defined in the **Name** field on the Sequence Scenario Editor dialog box (Figure 17).

    Note: To close and discard the new scenario (or any entered information), click **OK.** To cancel the operation and keep the scenario open, click **Cancel**.

You have completed all of the procedures required to create a sequence scenario successfully.

Copying a Scenario

To copy a scenario, follow these steps:

1. From the **Scenario** menu, select **Copy Scenario**.

2. Click **Save As** and then provide a new name for the scenario.

Caution: When copying a scenario, the following message displays when a scenario contains a threshold or dataset that contains thresholds:

You are attempting to copy a scenario that uses a dataset(s) that includes a threshold(s). If you copy this scenario and alter the thresholds, you also alter the threshold(s) for any other scenario that uses this dataset(s). In the Administration Tool's Threshold Editor, you can only modify the threshold value from the original scenario.

Use extreme caution when copying a scenario with this characteristic, because you create dependencies among scenarios. Changing thresholds affects both the original and copied scenarios.

Copying the Pattern

To copy a pattern, follow these steps:

1. From the Scenario menu, select **Copy** Pattern.

2. Click **Save As** to save a copy of the pattern in same scenario.

About the Pattern Report

You can use the Sequence Scenario Editor and the Rule Scenario Editor to generate and print details of the selected pattern and its scenario using the Generate Pattern Report function. You can use this information for analysis or troubleshooting purposes.

The report is in HTML format and contains pertinent information that describes a scenario and pattern, including:

- Scenario information
- Pattern information
- Schematic of the pattern logic
- Description of the pattern logic
- Constraints for each row or rule
- Bindings for each row or rule
- Saved pattern roles
- Thresholds
- Threshold Sets
- Threshold Usage
- Augmentation rules
- Description
- Comments
- Datasets (SQL that defines each dataset in the pattern)
- Sort order of the pattern used in the dataset

Guidelines for Generating a Pattern Report

Use the following guidelines when generating pattern reports:

- Name a generated scenario report using elements from the scenario. A desired format may be *scenario name_pattern ID*, since pattern ID is a unique parameter within any given scenario.
- Print in landscape mode to ensure that the formatting is accurate and that the printed report is readable.

- Do not override the default HTML file extension.

To Generate a Pattern Report

To generate a pattern report, follow these steps:

1. From the **Scenario** menu of the Sequence Scenario Editor, select **Open Existing Scenario**.

2. Select the desired scenario to open.

   The details of the scenario display in the Sequence Scenario Editor (Figure 17).

3. From the **Scenario** menu, select **Generate Pattern Report**.

   The Scenario/Pattern Report dialog box displays.
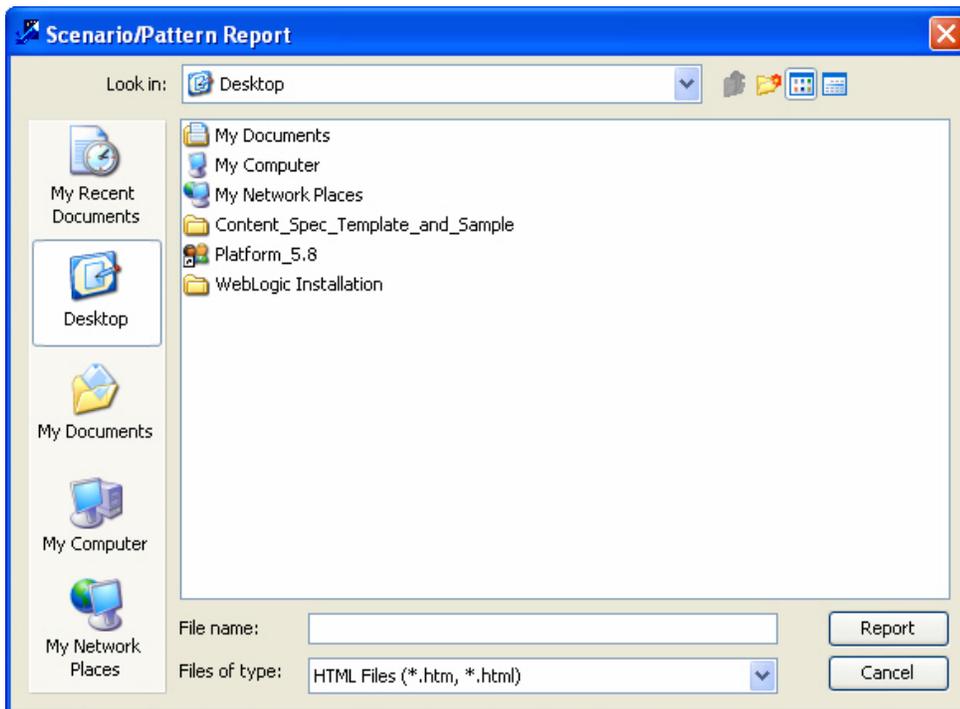
Figure 29. Scenario/Pattern Report Dialog Box

4. Define the location where you want to save the generated pattern report by navigating using the **Look In** field.

5. Type the desired file name in the **File Name** field.

6. Click **Report**.

   The system creates the report and saves it to the specified directory. You can then open and print the report through your browser.

Sequence Scenario Design Tips

The following tips can be helpful when working with scenarios:

- Select an unusual occurrence to serve as the initial event in a pattern whenever possible. Doing this reduces the number of match states (partial matches) that are created during a run.

- Limit the rows retrieved in the datasets used by a pattern. Specifically, if certain records cannot satisfy the pattern criteria, apply logic to filter them out up-front. If certain detection conditions can be checked in the dataset, it is generally more efficient to allow the database to filter them out rather than applying this logic in the pattern. It is also helpful to determine the appropriate lookback period range that is necessary to identify the behaviors of interest. Thus, the patterns do not look through a larger volume of historical data than is really necessary.

- Use dataset sorting to the maximum extent possible. To optimize performance, sort records that are used together to form an alert into adjacent positions in the dataset. For example, if a pattern is looking for a behavior within one account, sort the datasets it uses so that the records are clustered together by account. Doing this minimizes the volume of data through which the pattern must search to find a match, and reduces the number of partial matches that are held in memory at any one time.

- Refer to Appendix A, *Scenario Design Guidelines*, on page 175, for more information about guidelines and design tips.

CHAPTER 5 — *Link Analysis*

This chapter defines the Link Analysis features and functionality within the Scenario Manager.

This chapter focuses on the following topics:

- About Link Analysis
- Link Analysis Procedures within the Scenario Manager
- About the Network Definition Editor
- Using the Network Definition Editor
- About the Link Analysis Job Editor
- Using the Link Analysis Job Editor
- Using Sequence Matcher or Rule Matcher Job Editor to Define Data Match Criteria

About Link Analysis

Link Analysis is a data mining technique used to find networks of related entities that appear to be exhibiting some behavior of interest to the business. For example, a network can be a set of brokerage accounts linked together by some chain of internal funds transfers, a set of corporations linked together by wire transfers among themselves, a set of people related as beneficiaries in a set of wire transfers, or a set of people who share a common address, deposit, or name.

It is useful for detecting money laundering and fraud because criminals often work in fraud rings. Fraud rings often contain a series of linked entities that are not normally linked together for business. Link Analysis enables you to detect these networks.

As with any data mining technique, you follow these steps:

1. Identify the behavior in which you are interested. Link Analysis is an appropriate data mining technique if the behavior of interest is based upon a group of related entities.

2. Determine the category of possible related entities and the manner in which they are related. For example, various categories of accounts may exchange money through wire transfers.

3. Identify the characteristics of this group, or network, that are interesting or indicative of the behavior.

4. Determine the patterns that distinguish networks that are exhibiting the behavior of interest, from those that are not.

This thought process prepares you to use the Scenario Manager to create *Network Definitions* and *Patterns* that detect the behavior of interest in your data.

Link Analysis Terminology

Table 10 provides a hierarchically ordered list of terms that describe the concepts of Link Analysis.

Table 10. Link Analysis Glossary

Term	Definition
node	Business entity of interest (for example, institution, bank, tax ID, employee, customer, account, address, or household) involved in a link.
link	Specific occurrence of an association between two nodes (for example, a wire transfer from one customer to another customer, or to the address of a beneficiary). Zero (0), one (1), or more links can exist between two nodes.
relationship	Association between two nodes (for example, the two nodes are related in one or more transactions). Only a zero (0) or one (1) relationship can exist between two nodes.
network	Collection of nodes that links connect.
primary node	Node in the network that has the most links to other distinct nodes.
network definition	Type of relationship that defines a network, the characteristics that describe a network, and the logic to derive those characteristics.
source link dataset	Dataset that represents a set of logic used to find links and nodes and the relationship between them.
Network Characteristics dataset	Dataset that represents the characteristics or description of a network—the results of the definable characteristics. Only one Network Characteristics dataset can exist for each Network Definition.
input characteristic dataset	Dataset that specifies the logic to derive the characteristics stored in the Network Characteristics dataset. You can use more than one input characteristic dataset in a Network Definition.
standard characteristics	Properties of a network; for example, the number of nodes or average link weight. Within the Link Analysis process, fourteen standard characteristics are automatically captured and stored in the KDD_NTWRK table for a network.
definable characteristics	Properties or descriptors of a network that the Data Miner derives. They are based either on the network itself or on the business data associated with the network. The number of definable characteristics for a network is unlimited as long as a table in the database contains the results.

Link Analysis Phases and Tools

Conceptually, you can divide the Link Analysis technique into five phases. Each phase of Link Analysis consists of steps that uses different tools. Refer to the high-level steps of *Phases One through Three* and the tools used Figure 30 on page 70.

These phases allow you to follow these steps:

1. Specify the logic to find specific types of links between specific types of nodes and indicating which of these types to combine into networks (refer to section *Phase One-Building the Logic to Find Links and Combine into Networks*, on page 70, for more information).

   Optional: Specify the interesting characteristics of the resulting networks and the logic to derive these characteristics (refer to section *Phase Two-Defining Network Characteristics*, on page 72, for more information).

   Note: You can create a Network Definition with or without definable characteristics. If you create a Network Definition without definable characteristics, you do not need to perform tasks within Phase 2.

2. Create a Link Analysis Job to detect networks (refer to section *Phase Three-Creating the Link Analysis Job*, on page 75, for more information).

3. Specify patterns to evaluate networks (refer to section *Phase Four-Specifying Scenario Patterns or Rule Patterns to Generate Matches on Interesting Networks*, on page 76, for more information).

4. Create a Sequence Matcher or Rule Matcher job to execute the patterns and generate matches (refer to section *Phase Five-Creating a Sequence Matcher or Rule Matcher Job to Evaluate Networks and Generate Matches*, on page 76, for more information).

(Refer to section *Link Analysis Procedures within the Scenario Manager*, on page 77, for instructions on the specific steps of a phase.)

- Within these phases, use the Dataset Editor to create the following:

  - Source Link dataset that specifies the logic to detect links.

  - Network Characteristics dataset to capture the characteristics that distinguish the networks. (Creating a Network Characteristics dataset is optional, however, it is dependent upon Input Characteristic dataset).

  - Input Characteristic datasets that specify the logic to populate the characteristics captured in the Network Characteristics dataset. (Creating an Input Characteristic dataset is optional, however, it is dependent upon a Network Characteristics dataset).

- Use the Network Definition Editor to create a network definition that maps together:

  - One or more Source Link datasets that contain the links for construction of networks.

  - Network Characteristics dataset that captures interesting characteristics for the resulting networks.

  - Input Characteristic dataset that contains the logic to derive each characteristic (that is, column) in the Network Characteristics dataset.

- Use the Link Analysis Job Editor to create a job that executes a Network Definition to detect and characterize networks.

- Use the Sequence Pattern or Rule Pattern Editors to create sequence patterns or rule patterns that specify the logic and evaluate the characteristics of a network to detect networks that exhibit a behavior of interest.

- Use the Sequence Matcher or Rule Matcher Job Editors to create jobs that execute the sequence patterns or rule patterns to generate matches focused on the networks that exhibit the behavior of interest.
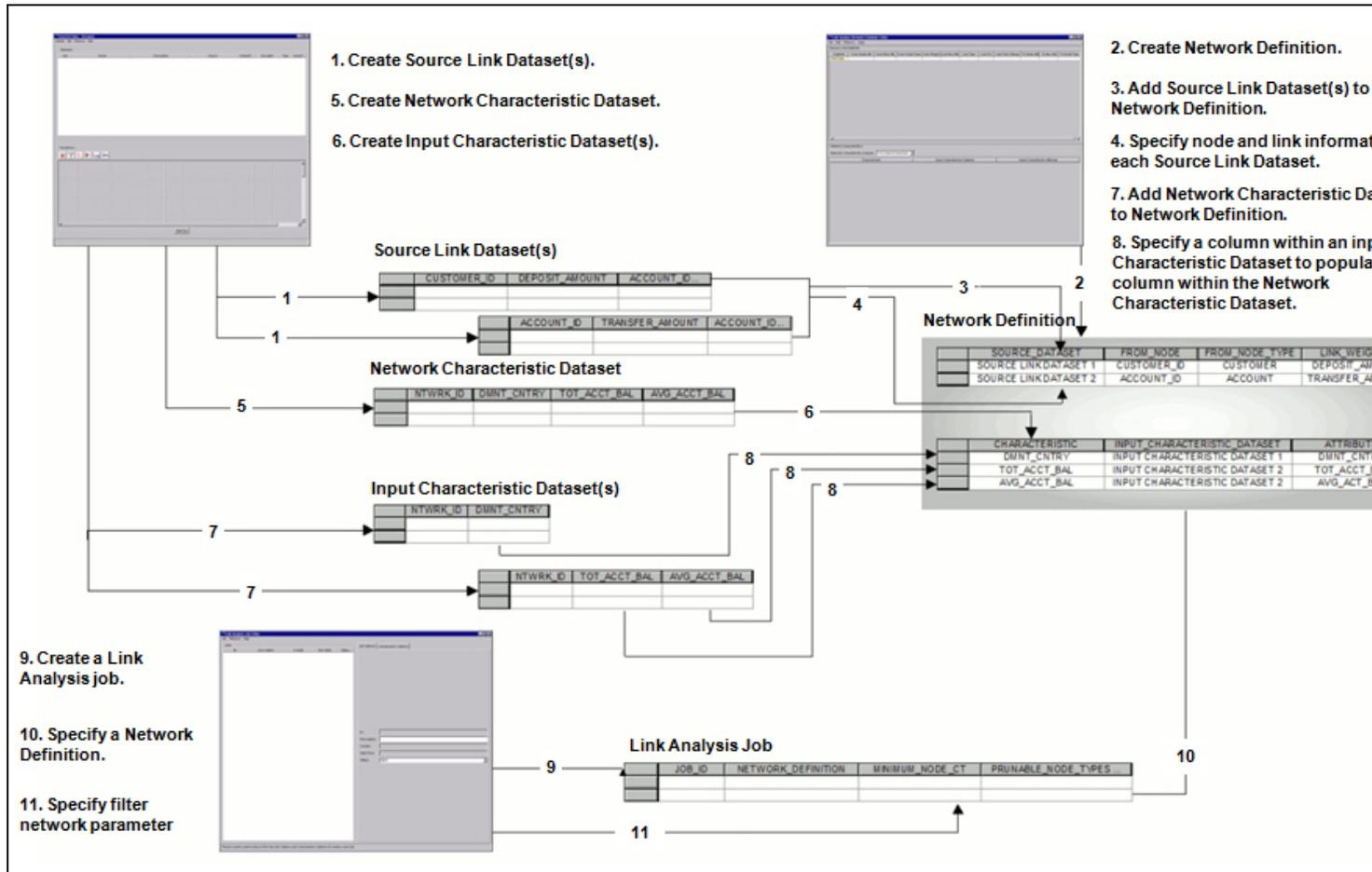
Figure 30. Link Analysis Steps

Phase One-Building the Logic to Find Links and Combine into Networks

Phase One specifies the logic to find specific types of links between specific types of nodes, and determines which of these types to combine into networks. Table 11 describes the steps and includes the Developer Tool used for the completion of the steps of Phase One.

Table 11. Phase One Steps and Tools

Step		Developer Tool
1	Create Source Link dataset.	DataSet Editor
2	Create new Network Definition.	Network Definition Editor
3	Add Source Link dataset to the Network Definition.	Network Definition Editor
4	Specify node and link information in each source link dataset.	Network Definition Editor

Step 1: Identify the entities in which you are interested and the kind of relationships that connect them. Specify this information by using the DataSet Editor to create Source Link datasets. Each Source Link dataset captures the logic to find a specific relationship (that is, a link) between two specific types of entities (that is, nodes). You must create a

separate Source Link dataset for each type of link or types of nodes that are grouped into networks. Each Source Link dataset must also contain basic information about the link and each of the nodes (for example, From Node key, From Business key, To Node key, or To Business key).

Note: The From Node key, From Business key, To Node key, and To Business key refer to a *business key*. A business key is a data field that the business uses to identify an Entity.

Step 2: Create a Network Definition to combine the pieces of your template together. Use the Network Definition Editor to create a Network Definition. A Network Definition represents a template that is used to detect networks.

Step 3: Use the Network Definition Editor to add the Source Link dataset to the Network Definition, which specifies the type of links and entities that should be evaluated to detect networks. Adding multiple Source Link datasets to a Network Definition enables the creation of networks that contain links and nodes of multiple types. This is useful in detecting complex networks.

Step 4: Use the Network Definition Editor to specify the node and link information within each Source Link dataset and to determine the columns in each dataset that represents each component of the link. For example, identify the columns in each Source Link dataset that represent the IDs of the From Node and the To Node. If appropriate, you will identify the Link Weight, a measure of the value or importance of the link. Refer to section *Components of the Network Definition Editor*, on page 78, for additional information about the columns and their definitions of the Network Definition Editor.

Phase Two-Defining Network Characteristics

Phase Two specifies the interesting characteristics of the resulting networks and the logic to derive these characteristics. (This is an optional phase-you do not have to specify definable Network Characteristics to create a Network Definition.) Table 12 defines the steps and includes the Developer Tool used for the completion of the steps of Phase Two.

Table 12. Phase Two Steps and Tools

Step		Developer Tool
5	Create Network Characteristics dataset.[1]	DataSet Editor
6	Create Input Characteristic dataset.	DataSet Editor
7	Add Network Characteristics dataset to the Network Definition.	Network Definition Editor
8	Map a column within an Input Characteristic dataset to populate each column of the Network Characteristics dataset.	Network Definition Editor

[1] You may need to create a new table in the database to store the Network Characteristics dataset (refer to section *Network Characteristic Table*, on page 73).

Step 5: Use the DataSet Editor to create a Network Characteristics dataset that specifies the *definable characteristics* that you wish to capture to describe the networks. Characteristics can be based on the source information represented by the nodes and links that comprise the network (for example, total dollar volume of transactions associated with customers in the network or number of account nodes with balances > USD 100,000) or based on the actual network nodes and links. Each column in a Network Characteristics dataset represents a characteristic of a network. Only one Network Characteristics dataset can exist per Network Definition.

In addition, the system captures standard characteristics based on data inherent to the network. Refer to section *Captured Standard Characteristics*, on page 73, for a listing of the standard characteristics.

Step 6: Use the DataSet Editor to create an Input Characteristic dataset to specify the filter and derivation logic (that is, SQL) necessary to populate each of the characteristics (that is, columns) contained in the Network Characteristics dataset.

Step 7: Use the Network Definition Editor to specify the characteristics you want to capture for a network by adding a Network Characteristics dataset to the Network Definition.

Step 8: Use the Network Definition Editor to map each network characteristic (that is, column) in the Network Characteristics dataset to a characteristic (that is, column) in an Input Characteristic dataset that contains the logic to derive the characteristic. This fully populated Network Definition represents both the location of the data to be analyzed and the results to be captured.

Network Characteristic Table

For definable characteristics, you must create a table external to the Scenario Manager prior to the creation of the Network Characteristics dataset. The Network Characteristics dataset is the wrapper dataset that provides access to this table.

The table must contain a column titled NTWRK_ID as its primary key, as well as a column for each characteristic you would like to capture to describe the networks. Table 13 provides a sample Network Characteristic table.

Table 13. Sample Network Characteristic Table

Column Name	Column Type	Nullable	Foreign Key	Primary Key	Description
NTWRK_ID	Number(10)	No	KDD_NTWRK	Yes	The network ID
AVG_ACCT_BAL	Number(16,6)	Yes		No	The average account balance
TOT_ACCT_BAL	Number(16,6)	Yes		No	The total account balance

Captured Standard Characteristics

Table 14 describes the characteristics captured automatically for every network. You can use these standard characteristics for patterns and for display in the Oracle Financial Services product. The KDD_NTWRK table stores these characteristics for all Networks. Column names are IDs of the network, network definition, and network characteristic dataset. In addition, two job-related column names are information about the job ID and the time the job ran.

Table 14. Standard Characteristics in the KDD_NTWRK Table

Column Name	Description
NODE_CT	Number of nodes in the network
PRMRY_NODE_ID	ID of the Primary Node in the network
PRMRY_NODE_LINK_CT	Number of nodes to which the Primary Node is linked
PRMRY_NODE_TOTAL_MEAS	Primary Node total measure (sum of the weight of the links associated with the Primary Node, both incoming and outgoing links)
PRMRY_NODE_IN_MEAS	Primary Node incoming measure (sum of the weight of links with directionality into the Primary Node)
PRMRY_NODE_OUT_MEAS	Primary Node outgoing measure (sum of the weight of links with directionality away from the Primary Node)
LINK_CT	Number of links in the network
AVG_LINK_WT	Average weight of the links in the network
MAX_LINK_WT	Maximum weight of a link in the network
FIRST_LINK_TS	Earliest timestamp of a link in the network
LAST_LINK_TS	Latest timestamp of a link in the network
PRMRY_NODE_IN_CT	Number of links with directionality into the Primary Node
PRMRY_NODE_OUT_CT	Number of links with directionality away from the Primary Node

Table 14. Standard Characteristics in the KDD_NTWRK Table (continued)

Column Name	Description
PRMRY_NODE_NO_DIRN_CT	Number of links associated with the Primary Node with no directionality
PRMRY_NODE_BUS_ID	Business ID of the Primary Node
NTWRK_ID	Sequence ID that uniquely identifies a network.
NTWRK_DEFN_ID	ID of the Network Definition that produced the network
NTWRK_CHRST_DSET_ID	Dataset ID that contains the definable characteristics for the network
RUN_ID	ID of the run that produced the network
CREATE_TS	Timestamp of the run that produced the network

Relationship Information

Information about all links between each set of nodes composes a relationship. This information describes the aggregate association between the nodes and is automatically captured for each link in a network. Table 15 provides the column name and definition of the captured relationship information. Column names are not relationship information but IDs of the linked nodes and the link summary ID.

Table 15. Relationship Info in the KDD_LINK_SUMMARY Table

Column Name	Description
TOT_LINK_WEIGHT	Absolute sum of the link weight of all links between the two nodes, regardless of the directionality
TOT_LINK_CT	Total count of all links between the two nodes, regardless of the directionality
LINK_WEIGHT_INTO_FIRST_NODE	Absolute sum of the link weight of all links between the two nodes that have directionality to the node represented by the FIRST_NODE_ID
LINK_CT_INTO_FIRST_NODE	Count of the links between the two nodes that have directionality to the node represented by the FIRST_NODE_ID
LINK_WEIGHT_INTO_SCND_NODE	Absolute sum of the link weight of all links between the two nodes that have directionality to the node represented by the SCND_NODE_ID
LINK_CT_INTO_SCND_NODE	Count of the links between the two nodes that have directionality into the node represented by SCND_NODE_ID
LINK_WEIGHT_NO_DIRN	Absolute sum of the link weight of all links between the two nodes that have no directionality
LINK_CT_NO_DIRN	Count of the links between the two nodes that have no directionality
AGGR_LINK_DIRN	Summarization of the directionality of the links between the two nodes (that is, NONE, ONE_WAY, or BOTH)
LINK_SUMMARY_ID	Unique internal ID of the link summary
FIRST_NODE_ID	ID of the first node in the pair
SCND_NODE_ID	ID of the second node in the pair

Relationship Type Information

Additional relationship type information is automatically captured for each relationship in a network. This information summarizes each link type for the links of that type between nodes.

Table 16 provides the column name and definition of the relationship type summary information captured. Column names are not link summary information but IDs of the linked nodes and the link type summary ID.

Table 16. Relationship Type Info in the KDD_LINK_TYPE_SUMMARY Table

Column Name	Description
LINK_TYPE_CD	Type of the links being summarized
TOT_LINK_WEIGHT	Absolute sum of the link weight of all links between the two nodes, regardless of directionality
TOT_LINK_CT	Total count of all links between the two nodes, regardless of directionality
LINK_WEIGHT_INTO_FIRST_NODE	Absolute sum of the link weight of all links between the two nodes that have directionality to the node that the FIRST_NODE_ID represents
LINK_CT_INTO_FIRST_NODE	Count of the links between the two nodes that have directionality to the node that the FIRST_NODE_ID represents
LINK_WEIGHT_INTO_SCND_NODE	Absolute sum of the link weight of all links between the two nodes that have directionality to the node represented by the SCND_NODE_ID
LINK_CT_INTO_SECOND_NODE	Count of the links between the two nodes that have directionality to the node that the SCND_NODE_ID represents
LINK_WEIGHT_NO_DIRN	Absolute sum of the link weight of all links between the two nodes that have no directionality
LINK_CT_NO_DIRN	Count of the links between the two nodes that have no directionality
AGGR_LINK_DIRN	Summarization of the directionality of the links between the two nodes (that is, NONE, ONE_WAY, or BOTH)
LINK_TYPE_SUMMARY_ID	Unique internal ID of the link type
FIRST_NODE_ID	ID of one of the nodes (first) in the pair
SECOND_NODE_ID	ID of the other node (second) in the pair

Phase Three-Creating the Link Analysis Job

Phase Three creates a Link Analysis Job to detect networks. The job contains several parameters that filter the data evaluated for inclusion in networks, as well as filters for the resulting networks. Table 17 defines the steps and includes the Developer Tool used for the completion of the steps.

Table 17. Phase Three Steps and Tools

Step		Developer Tool
9	Create a Link Analysis job.	Link Analysis Job Editor

Table 17. Phase Three Steps and Tools

10	Specify a Network Definition.	Link Analysis Job Editor
11	Specify filter parameters.	Link Analysis Job Editor

Step 9: Use the Link Analysis Job Editor to create the job that generates and characterizes networks based on a Network Definition created in the Network Definition Editor.

Step 10: Use the Link Analysis Job Editor to specify a Network Definition that contains both the source link dataset to be evaluated for networks and a Network Characteristics dataset to be populated with the characteristics of the discovered networks.

Step 11: Use the Link Analysis Job Editor to specify parameters for filtering links based on a lookback period, filtering networks based on a new network lookback period, and the minimum number of nodes. Refer to section *Link Analysis Options Tab*, on page 95, for more information about filtering and pruning.

Phase Four-Specifying Scenario Patterns or Rule Patterns to Generate Matches on Interesting Networks

After creating a Network Definition, use the Scenario Pattern or Rule Pattern editors to specify patterns that evaluate the characteristics of networks to identify, and generate matches, for networks that appear to be exhibiting a behavior of interest. Refer to Chapter 4, *Sequence Scenario Editor*, on page 39, for information about using these editors.

Phase Five-Creating a Sequence Matcher or Rule Matcher Job to Evaluate Networks and Generate Matches

After creating the scenario or rule patterns, use the Sequence Matcher Job Editor or the Rule Matcher Job Editor to create a job that executes the pattern to generate matches. Refer to Chapter 9, *Job Editors*, on page 161, for information about using these editors.

Link Analysis Procedures within the Scenario Manager

The continuing sections of this chapter describes how to perform steps defined in each phase outlined in previous sections. These instructions are not in the order and are not defined in each phase, however, they are grouped together by the tools used to perform them (Figure 30):

● **DataSet Editor** (refer to section *Using the DataSet Editor with Link Analysis*, on page 77, for more information).

● **Network Definition Editor** (refer to section *About the Network Definition Editor*, on page 78, for more information).

● **Link Analysis Job Editor** (refer to section *About the Link Analysis Job Editor*, on page 92, for more information).

Note: You can complete steps 1, 5, and 6 in the DataSet Editor *prior* to performing Steps 2, 3, 4, 7, and 8 in the Network Definition Editor (in Figure 30). Completing the steps in a grouped fashion (by tool), enables you to complete all the steps involved in creating a Network Definition with characteristics. This avoids having to open and close each tool, as the data is not synchronized or refreshed between the DataSet and Network Definition Editors.

Using the DataSet Editor with Link Analysis

To create the Source Link dataset, the Input Characteristic dataset, and the Network Characteristics dataset, use the DataSet Editor tool. Refer to Chapter 1, *About Scenario Manager*, on page 1, for information about creating a dataset. Requirements for each type of dataset include the following:

● **Source Link dataset:** Requires that the Oracle Financial Services key of the link is the primary key (for example, SEQ_ID).

   Note: An *Oracle Financial Services Key* is a Oracle Financial Services-specific data field that uniquely identifies an Entity. Oracle Financial Services products generate the value of this field during data ingestion.

● **Input Characteristics dataset:** Requires the NTWRK_ID as its primary key, data type Number (10), and the same data types and nullable status per column as the Network Characteristics dataset.

   Note: The Network Definition Editor does not check the nullable status between the Characteristic column and the Input Characteristic Attribute column in the Network Characteristics area.

● **Network Characteristics dataset:** Requires the NTWRK_ID as its primary key, data type Number (10), and the same data types and nullable status per column as the Source Link dataset.

   Note: In a network definition, you should use a dataset that is associated with a scenario (either because it contains a threshold or is dependent on a dataset that contains a threshold) only if that definition is used in the same scenario.

About the Network Definition Editor

The following section describes the components and menu options in the Network Definition Editor.

Components of the Network Definition Editor

The Network Definition Editor displays Figure 31.
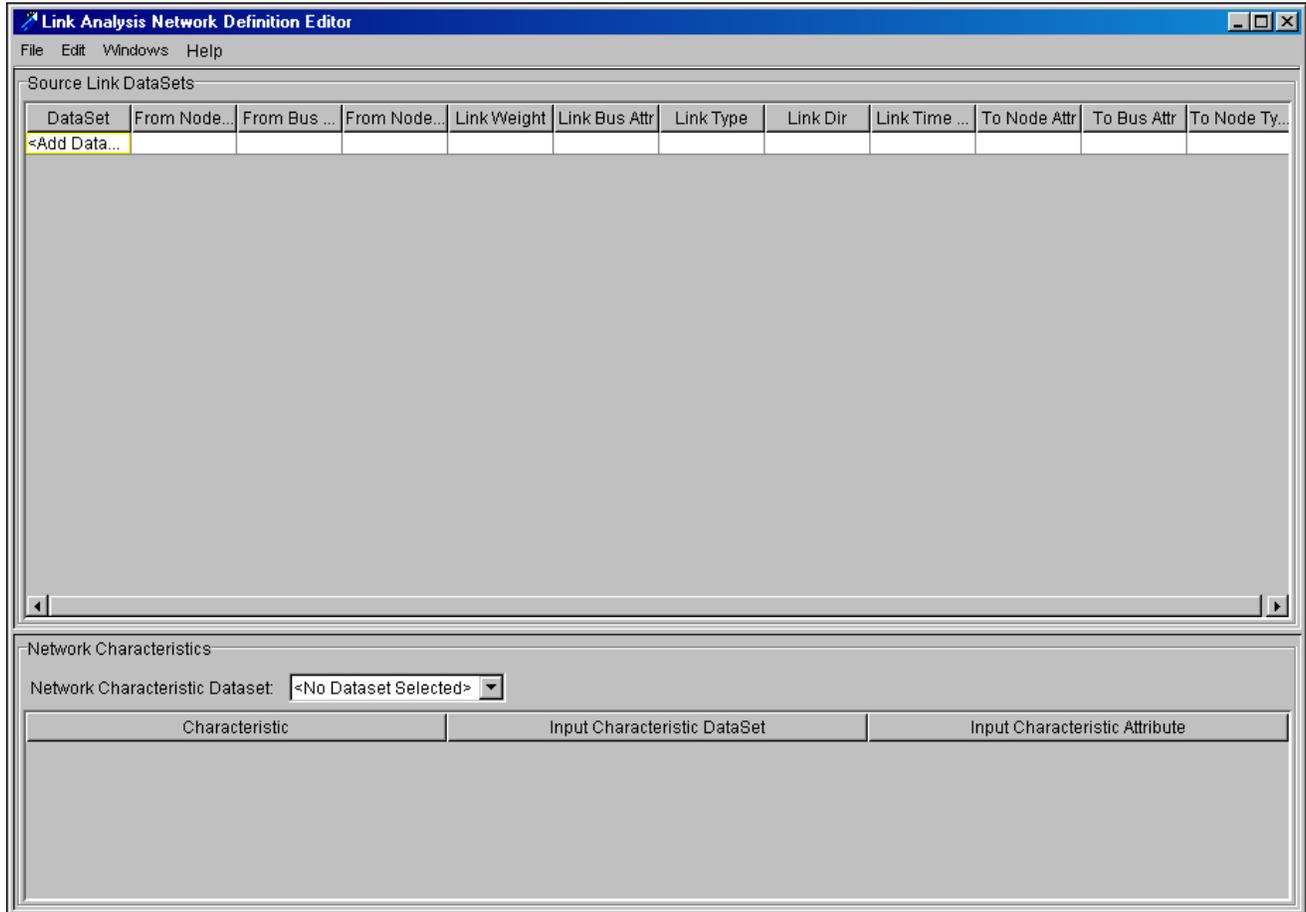
Figure 31. Network Definition Editor

The Network Definition Editor contains three components:

- **Menu Bar:** Provides access to other tools, but primarily provides functionality for the Network Definition Editor (for example, saving and deleting).

- **Source Link DataSets:** Specifies the link and node information within the Source Link dataset, and groups one or more datasets into a Network Definition.

- **Network Characteristics:** Specifies and maps each characteristic (that is, column) in the Network Characteristics dataset to a characteristic (that is, column) in an Input Characteristic dataset.

Network Definition Editor Menu Bar

The Network Definition Editor menu bar consists of four menus:

● **File:** Provides the basic options used to create, modify, and delete a Network Definition.

● **Edit:** Provides additional functionality for the Network Definition Editor that is not accessible from within the areas of the Network Definition Editor.

● **Windows:** Provides access to other tools.

● **Help:** Provides the About option that displays system properties for the Scenario Manager (for example, version of software, date of the build, and domain).

The following sections describe these menus. Refer to Chapter 2, *Common Features*, on page 3., for descriptions of the Windows and Help menu options.

File Menu

The File menu contains the following menu options:

● **New:** Entering the Network Definition Editor actually selects New from the File menu (that is, it is unnecessary to click this menu option). However, if you are working in the Network Definition Editor and select New from the File menu, the system displays a warning message if you have unsaved data. Click one of the following buttons in the Warning dialog box:

   ■ **Save:** Saves the Network Definition you were creating or modifying and then clears the columns in the Source Link DataSets area to the default values so that you may begin work on a new Network Definition.

   ■ **OK:** Saves the data and clears the columns in the Source Link DataSets area to the default values.

   ■ **Cancel:** Returns to the Network Definition without saving data or creating a new Network Definition.

● **Open:** Opens an existing Network Definition.

   ■ If you are creating or modifying a Network Definition and have unsaved data, the system displays a warning message that you have unsaved data. You may click one of the following buttons in the Warning dialog box:

      ◆ **Save:** Saves the Network Definition you were creating or modifying and then displays the Network Definition Selector dialog box displays (refer to section *Opening an Existing Network Definition*, on page 87, for more information).

      If you select Save and the system prompts you with a save error dialog box, you must correct your Network Definition before you can save. You must then select the **Open** option from the File menu again.

      ◆ **OK:** Does not save the data and then displays the Network Definition Selector dialog box (refer to section *Opening an Existing Network Definition*, on page 87, for more information).

      ◆ **Cancel:** Returns you to the Network Definition you were creating or modifying without saving your data or opening an existing Network Definition.

   ■ If you were *not* creating or modifying a Network Definition, the Network Definition Selector dialog box displays (refer to section *Opening an Existing Network Definition*, on page 87, for more information).

● **Save:** Saves an existing Network Definition that has been modified and that you own. The system saves the Network Definition with the same Network Definition name and ID. Refer to Table 18 for rules on saving a Network Definition.

- **Save As:** Saves a new Network Definition or saves a copy of an existing Network Definition as another Network Definition (that is, copies an existing Network Definition to create a new Network Definition, which includes a new Network ID). Refer to Table 18 for rules on saving as in the Network Definition Editor.

- **Delete:** Deletes a Network Definition. You may only delete Network Definitions that you own. For details of this task refer to section *Deleting a Network Definition*, on page 90, for more information.

- **Close:** Closes the Network Definition Editor (refer to section *Closing the Network Definition Editor and Exiting the Scenario Manager*, on page 90, for more information).

- **Exit:** Closes the Scenario Manager (refer to section *Exiting the Scenario Manager*, on page 91, for more information).

Table 18 lists the rules that the system enforces when you attempt to save, or save as, a Network Definition.

Table 18. Network Definition Save and Save As Rules

Rules
You must select a value for the *From Node Attr* column for all of the datasets in the Source Link DataSets area. If you do not, the system displays an error dialog box with the message *Please select an attribute in the From Node Attr column for each Source Link DataSet(s) where you have also selected a From Node Type with an identifying attribute of Key (for example, ACCOUNT - Key Attr).*
You must select a value for the *From Bus Attr* column for all of the datasets in the Source Link DataSets area. If you do not, the system displays an error dialog box with the message *Please select an attribute in the From Bus Attr column for each dataset in the Source Link DataSets area.*
You must select a value in the *From Node Type* column of all datasets displayed in the Source Link DataSets area. If you do not, the system displays an error dialog box with the message *Please select the type in the From Node Type column for each dataset in the Source Link DataSets area.*
You must select a value in the Link Type column of all datasets displayed in the Source Link DataSets area. If you do not, the system displays an error dialog box with the message. *Please select a type of Link Type column for each dataset in the Source Link DataSets area.*
You must select a value in the To Node Attr column of all datasets displayed in the Source Link DataSets area. If you do not, the system displays an error dialog box with the message *Please select an attribute in the To Node Attr column for each Source Link DataSet(s) where you have also selected a From Node Type with an identifying attribute of Key (for example, ACCOUNT - Key Attr).*
You must select a value in the To Bus Attr column of all datasets displayed in the Source Link DataSets area. If you do not, the system displays an error dialog box with the message *Please select an attribute in the To Bus Attr column for each dataset in the Source Link DataSets area.*
You must select a value in the To Node Type column of all datasets displayed in the Source Link DataSets area. If you do not, the system displays an error dialog box with the message *Please select a type in the To Node Type column for each dataset in the Source Link DataSets area.*
You cannot select the same values in the *From Node Attr* column and the *To Node Attr* column of any row in the Source Link DataSets area. If you do, the system displays an error dialog box with the message *The From Node Attr and To Node Attr columns cannot have the same attribute. Please select a different attribute in either the From Node Attr column or the To Node Attr column for each dataset in the Source Link DataSets area.*

Table 18. Network Definition Save and Save As Rules (continued)

Rules
You cannot select the same values in the *From Bus Attr, To Bus Attr*, and the *Link Bus Attr* columns of any row in the Source Link DataSets area. If you do, the system displays an error dialog box with the message *Please select different attributes in the From Bus Attr, To Bus Attr, and Link Bus Attr columns for each dataset in the Source Link DataSets area.*
You must select a dataset in the *Input Characteristic DataSet* column and an attribute in the *Input Characteristic Attribute* column for each characteristic in the *Characteristic* column displayed in the Network Characteristics area. If you do not, the system displays the error dialog box with the message *Please select a dataset and attribute to populate each characteristic displayed in the characteristic matrix.*
You cannot save a Network Definition with the same name as an existing Network Definition. If you do, the system displays an error dialog box with the message "A network definition editor already exists with the entered name. Please enter a unique name."

File Menu Shortcut Keys

Table 19 provides the list and description of shortcut keys on the **File** menu.

Table 19. File Menu Shortcut Keys

Menu Option	Shortcut Keys	Description
New	CTRL+N	Creates a new Network Definition.
Open	CTRL+O	Opens an existing Network Definition.
Save	CTRL+S	Saves a Network Definition.
Close	CTRL+W	Closes the Network Definition Editor.
Exit	CTRL+Q	Exits the Scenario Manager.

Edit Menu

The Edit menu contains the following menu options:

- **Remove DataSet:** Enables you to select and remove a Source Link dataset from the Source Link DataSets area (refer to section *Removing a Source Link DataSet from a Network Definition*, on page 89, for more information).

- **Add Link Types:** Enables you to add a new link type (refer to section *Adding a Link Type*, on page 88, for more information).

- **Add Node Types:** Enables you to add a new node type (refer to section *Adding a Node Type*, on page 88, for more information).

Source Link DataSets Area

The Source Link DataSets area enables you to add Source Link datasets to a Network Definition and to map the columns within each Source Link dataset (Figure 32).
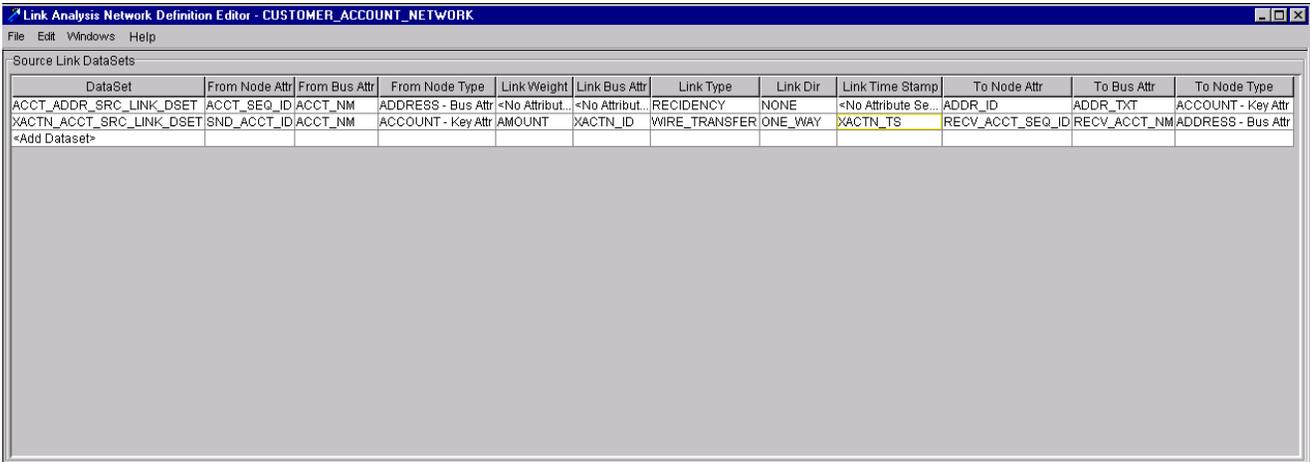

Figure 32. Source Link DataSets Area

The Source Link DataSets area columns are resizable, as is the entire Source Link DataSets area. Vertical and horizontal scroll bars display as needed.

Each row in the Source Link DataSets area contains the following drop-down lists:

● **DataSet**: Lists all realized datasets with integer or float attributes as the primary key, in the current domain that you may select for inclusion into your Network Definition. The default option is **<Add DataSet>**.

■ The system populates all the columns with the dataset default values after you select a Source Link dataset. Until then, the columns are blank and non-editable.

■ The last empty row of the DataSet drop-down list column is always enabled so that you may add another Source Link dataset.

■ The rows within the Source Link DataSets area are sorted in ascending alphabetical order based on the dataset name (when you have opened an existing dataset).

■ The DataSet column excludes all datasets chosen as the Network Characteristics dataset or as an Input Characteristic dataset.

● **From Node Attr:** Provides a list all attributes associated with the selected dataset that are of integer or float type. This attribute describes the key of the from node (for example, ACC_SEQ_ID or SNDACTT_SEQ_ID as illustrated in Figure 30 on page 70). The default option is **<Select Attribute>** until you select a From Node attribute and save it within the Network Definition. This field is either mandatory or optional depending on your selection in the From Node Type column.

● **From Bus Attr:** Provides a list of attributes associated with the selected dataset. This attribute describes the business key of the from node and is used for display purposes or to identify the node (for example, ACCT_NAME). The default option is **<Select Attribute>** until you select a From Bus attribute and save it within the Network Definition.

● **From Node Type:** Provides the available node types defined within the current domain and the corresponding identifying attribute of the node type (for example, ADDRESS -Bus Attr or CUSTOMER_ACCOUNT - Key Attr). The default option is **<Select Type>** until you select a From Node

Type attribute and save it within the Network Definition. In addition, you can add node types (refer to section *Adding a Node Type*, on page 88, for more information).

- **Link Weight:** Provides the names of each integer and real attribute within the selected dataset, excluding those attributes with SEQ_ID in the name (for example, AMOUNT). The default value is **<No Attribute Selected>** until you select a Link Weight attribute and save it within the Network Definition. An entry in this column is optional.

- **Link Bus Attr:** Provides list of attributes associated with the selected dataset. This attribute describes the business key of the Link Weight column and is used for display purposes (for example, XACTN_ID). The default value is **<No Attribute Selected>** until you select a Link Bus attribute and save it within the Network Definition. An entry in this column is optional.

- **Link Type:** Provides available link types defined within the current domain for the Oracle Financial Services product (for example, RESIDENCY or WIRE_TRANSFER). The default value is **<Select Type>** until you select a Link Type attribute and save it within the Network Definition. In addition, you can add link types (refer to section *Adding a Link Type*, on page 88, for more information).

- **Link Dir:** Provides the ONE_WAY and NONE options. These options represent the direction of the link. The default value is ONE_WAY until you select a Link Dir type and save it within the Network Definition.

- **Link Time Stamp**: Provides the names of each date attribute within the selected dataset (for example, XACTN_TS). If you do not specify a Link Time Stamp value, the system uses the current date for all links contained in Source Link dataset. The default value is **<No Attribute Selected>** until you select a Link Time Stamp attribute and save it within the Network Definition. An entry in this column is optional.

  Note: If you specify a Link Time Stamp attribute, any links that have a null value for that column are assigned the current date as the Link Time Stamp value.

- **To Node Attr:** Provides the list of attributes associated with the selected dataset that are of integer or float type. The column represented by this attribute should be the Oracle Financial Services key (for example, SEQ_ID) for the to node (for example, ADDR_ID or RECVACCT_SEQ_ID). The default value is **<Select Attribute>** until you select a To Node attribute and save it within the Network Definition. This field is either mandatory or optional depending on your selection in the To Node Type column.

- **To Bus Attr:** Provides the names of each attribute within the selected dataset. This attribute describes the business key of the to node and is used for display purposes or to identify the node (for example, ADDR_TEXT or RECV_ACCT_NAME). The default value is **<Select Attribute>** until you select a To Bus attribute and save it within the Network Definition.

- **To Node Type:** Provides the available node types defined within the current domain and the corresponding identifying attribute of the node type (for example, ADDRESS - Bus Attr or CUSTOMER_ACCOUNT - Key Attr). The default value is **<Select Type>** until you select a To Node Type attribute and save it within the Network Definition. In addition, you can add node types (refer to section *Adding a Node Type*, on page 88, for more information).

Network Characteristics Area

In the Network Characteristics area, you can define the relationship between the Network Characteristics dataset and an Input Characteristic dataset by mapping one attribute in the Network Characteristics dataset and one attribute in one Input Characteristic dataset (Figure 33).

Note: Using the Network Characteristics area of the Network Definition Editor is optional. Use this area only when you want to include definable characteristics.

Figure 33. Network Characteristics Area

The Network Characteristics area columns are resizable and vertical and horizontal scroll bars display as needed. The columns of the Network Characteristics area include the following:

- **Network Characteristics DataSet**: Provides the dataset name of each realized dataset that is associated with the Oracle Financial Services domain, contains a NTWRK_ID as its key attribute, and is not already selected as a Source Link dataset (for example, RELATED_ACCT_NTWRK_CHRST). The default value is **<**No DataSet Selected**>**.

- **Characteristic** column: Automatically populates when you select a Network Characteristics dataset. This column is not editable (for example, AVG_ACCT_BAL or TOT_ACCT_BAL). Each attribute (column) of the Network Characteristics dataset is in a separate row, excluding the NTWRK_ID column.

- **Input Characteristic DataSet**: Provides the name of each realized dataset that is associated with the Oracle Financial Services domain, contains a NTWRK_ID as its key attribute, and is not already selected as a Source Link dataset or Network Characteristics dataset (for example, INPUT_NTWRK_CHAR). The default option is **<**Select DataSet**>**.

- **Input Characteristic Attribute**: Provides the name of each column, excluding the NTWRK_ID column, of the selected dataset in the Input Characteristic DataSet column of the same row. This is of the same column type (for example, varchar or date) as the attribute in the Characteristic column (for example, AVG_ACCT_BAL or TOT_ACCT_BAL). The default option is **<**Select Attribute**>**.

Using the Network Definition Editor

This section provides instructions for performing the following procedures:

- Creating a Network Definition
- Modifying a Network Definition
- Deleting a Network Definition
- Closing the Network Definition Editor and Exiting the Scenario Manager

Creating a Network Definition

This section provides the steps necessary to create a Link Analysis scenario:

1. Accessing the Network Definition Editor.

2. Creating a Network Definition (without Definable Characteristics).

   Optional: Creating a Network Definition (with Definable Characteristics).

Accessing the Network Definition Editor

To display the Link Analysis Network Definition Editor, click Networks in the Windows menu of the Scenario Manager.

Creating a Network Definition (without Definable Characteristics)

To create a Network Definition *without* definable characteristics, follow these steps:

1. Access the Network Definition Editor.

2. From the **DataSet** drop-down list, select the desired dataset from the list of available Source Link datasets.

   The system displays the name of the Source Link dataset in the first row. Each additional Source Link dataset selected displays in the next row.

3. Repeat Steps #2 and #3 to add more Source Link datasets.

4. Select the appropriate values from the remaining columns (for each row if you have selected more than one Source Link dataset).

5. Click **Save As**..

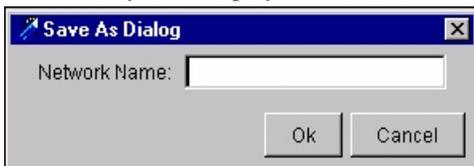   The system displays the Save As dialog box (Figure 34).

Figure 34. Save As Dialog Box

6. Type the name of the Network Definition in the **Network Name** field.

7. Click **Ok** to save the Network Definition.

Creating a Network Definition (with Definable Characteristics)

To create a Network Definition *with* definable characteristics, follow these steps:

1. Access the Network Definition Editor.

2. From the **DataSet** drop-down list, select the desired dataset from the list of available Source Link datasets.

   The name of the Source Link dataset displays in the first row. Each additional Source Link dataset selected displays as the next row.

3. Repeat Steps #2 and #3 to add more Source Link datasets.

4. Select the appropriate values from the remaining columns (for each row if you selected more than one Source Link dataset).

5. From the **Network Characteristics** drop-down list, select the desired dataset from the list of available Network Characteristics datasets.

   The system creates a characteristic row for each attribute in the Network Characteristics dataset (except for the NTWRK_ID attribute).

6. Map a characteristic to an Input Characteristic dataset by clicking the first row of the Input Characteristic DataSet drop-down list.

   This system displays the available Input Characteristic datasets.

7. Click the dataset in which the attribute used to populate the characteristic exists.

8. Map each characteristic to an Input Characteristic attribute by clicking the first row of the Input Characteristic Attribute drop-down list.

   This system displays the available Input Characteristic attributes.

9. Click the attribute that specifies the logic to populate the characteristics.

10. Repeat Steps #7 through #11 for each characteristic row.

11. From the **File** menu, select **Save As**.

    The system displays the Save As dialog box.

12. Type the name of the Network Definition in the **Network Name**.

13. Click **OK**.

   The system saves the Network Definition.

Modifying a Network Definition

The following subsection provide instructions on how to modify an existing Network Definition:

- Opening an Existing Network Definition

- Adding a Link Type

- Adding a Node Type

- Removing a Source Link DataSet from a Network Definition

Opening an Existing Network Definition

To open an existing Network Definition, follow these steps:

1. From the **File** menu of the Link Analysis Network Definition Editor, select **Open**.

   The Network Definition Selector dialog box displays (Figure 35).
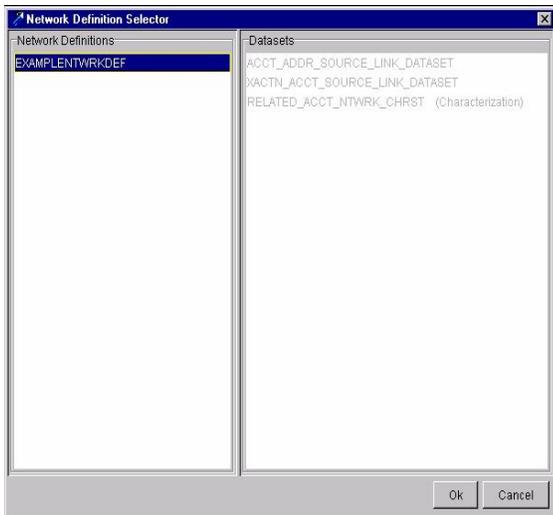
Figure 35. Network Definition Selector Dialog Box

2. Select the desired Network Definition in the Network Definitions area.

   The Network Definitions area populates with each Network Definition within the domain. The Network Characteristics dataset associated to the selected Network Definition display in the DataSets area (for example, Sample). In addition, the DataSets area lists the Source Link datasets and identifies them with the *Characterization* text (for example, RELATED_ACCT_ NTWRK_CHRST).

3. Click **OK**.

   The system opens the selected Network Definition in the Link Analysis Network Definition Editor.

Adding a Link Type

To add a link type, follow these steps:

1. From the **Edit** menu of the Link Analysis Network Definition Editor, select **Add Link Types**.

   The Link Type Editor dialog box displays with all existing Link types for the domain (Figure 36).
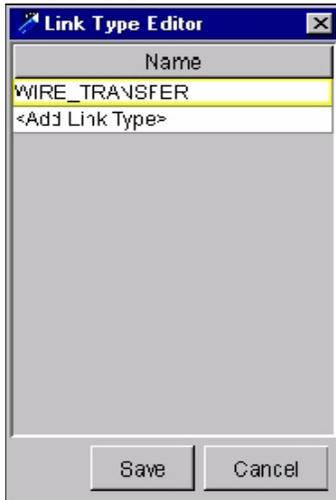

Figure 36. Link Type Editor

2. Type the link name into the **<Add Link Type>** text box (for example, WIRE_TRANSFER).

3. Press **ENTER**.

   The system adds the link type and displays the **<**Add Link Type**>** row.

4. Click **Save**.

   The system closes the Link Type Editor dialog box and adds the new link type to the Link Type drop-down list in the Source Link DataSets area.

   Note: You cannot save a link type with the same name as an existing link type, nor can you edit an existing link type.

Adding a Node Type

To add a node type, follow these steps:

1. From the **Edit** menu of the Link Analysis Network Definition Editor, select **Add Node Types**.

   The Node Type Editor dialog box displays with all existing Node Types for the domain (Figure 37).
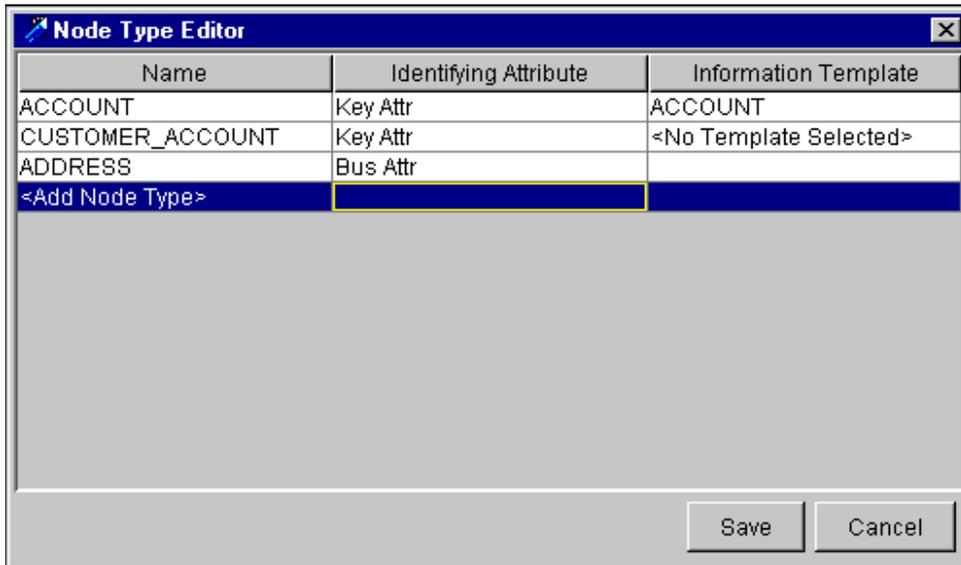
Figure 37. Node Type Editor Dialog Box

2. Type the node name into the **<Add Node Type>** text box
   (for example, ADDRESS).

3. Press **ENTER**.

   The system adds the node type and displays the **<**Add Node Type**>** row.

4. Click inside the **Identifying Attribute** drop-down list to select either the **Bus Attr** (Business Attribute) or default **Key Attr** (Key Attribute) options to define which column in the Source Link DataSet that serves as the unique ID for nodes of this type.

   Note: If you select the Bus Attr option you may not select an option in the **Information Template** drop-down list.

   Optional: Click inside the **Information Template** drop-down list to select an available option.

   The values in the **Information Template** drop-down list are the product's focus types (for example, Account, Client Bank, or Household. Selecting an Information Template provides the mechanism for the system to display the appropriate data, based on the focus, within the Oracle Financial Services application (through the visualization function).

5. Click **Save**.

   The system closes the Node Type Editor dialog box and adds the new node type to the Node Type drop-down list in the Source Link DataSets area.

   Note: You cannot save a node type with the same name as an existing node type, nor can you edit an existing node type.

Removing a Source Link DataSet from a Network Definition

To remove a Source Link dataset from the Link Analysis Network Definition Editor, follow these steps:

1. Select the appropriate Source Link dataset in the Link Analysis Network Definition Editor from which you want to remove the Source Link dataset.

2. From the Edit menu, select **Remove DataSet**.

   The Confirmation dialog box displays.

3. Click **OK**.

   The Source Link dataset is removed from the Link Analysis Network Definition Editor.

Deleting a Network Definition

To delete a Network Definition, follow these steps:

1. From the **File** menu, select **Delete.**

   The Confirmation dialog box displays.

2. Click **OK**.

   The system deletes the Network Definition and returns you to an empty Link Analysis Network Definition Editor.

   Note: When you delete a Network Definition, you are deleting only the selected Network Definition and *not* the Source Link dataset(s) associated to the deleted Network Definition.

Closing the Network Definition Editor and Exiting the Scenario Manager

The following sections provide instructions on how to close the Network Definition Editor and how to exit the Scenario Manager.

Closing the Network Definition Editor

To close the Network Definition Editor however, not the Scenario Manager, follow these steps:

1. From the **File** menu, select **Close**.

   The system closes the Network Definition Editor and returns you to the DataSet Editor.

   If you are creating or modifying a Network Definition and do not save your updates, the system displays the Warning dialog box (Figure 38).
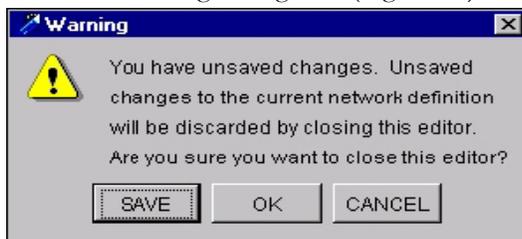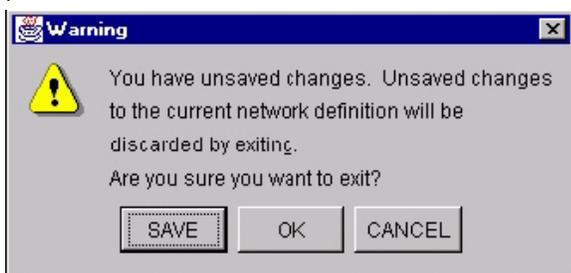

Figure 38. Close Warning Dialog Box

2. Click **Save**.

   The system saves your work and then closes the Network Definition Editor.

3. Click **OK**.

The system saves the modification and closes the Network Definition Editor without saving your changes.

4. Click **Cancel.**

The system remains in the Network Definition Editor.

Exiting the Scenario Manager

To exit the Scenario Manager, follow these steps:

1. From the **File** menu, click **Exit**.

The system exits the Scenario Manager.

If you were creating or modifying a Network Definition and did not save your updates, the system displays the Warning dialog box (Figure 39)

.

Figure 39. Exit Warning Dialog Box

2. Click **SAVE**.

The system saves your work and then exits the Scenario Manager.

3. Click **OK**.

The system saves the modification and closes the exit the Scenario Manager without saving your changes.

4. Click **CANCEL.**

The system remains in the Scenario Manager.

About the Link Analysis Job Editor

Use the Link Analysis Job Editor to define the parameters within a Link Analysis Job. The following sections define the components of the Link Analysis Job Editor and describe how to create, modify, and delete a Link Analysis job.

Components of the Link Analysis Job Editor

The Link Analysis Job Editor is composed of four components:

- **Link Analysis Job Editor**: Provides access to other tools, but mainly provides functionality for the Link Analysis Job Editor (for example, saving and deleting).

- **Job**: Provides a job selection list for each Link Analysis job associated with the current domain.

- **Job Options**: Enables you to enter information specific to a Link Analysis job.

- **Link Analysis Options**: Enables you to create or modify a Link Analysis job—pruning nodes or specifying a minimum number of nodes.

Link Analysis Job Editor Menu Bar

The Link Analysis Job Editor Menu Bar consists of three options:

- **Job:** Provides the basic options used to create, copy, and delete a Link Analysis job.

- **Windows:** Provides access to other tools.

- **Help:** Provides the About option which describes system properties for the Scenario Manager (for example, version of software, date of the build, and domain).

The following section describes the **Job** menu options. Refer to Chapter 2, *Common Features*, on page 3., for descriptions of the **Windows** and **Help** options.

Job Menu

The Job menu contains the following options:

- **New:** Entering the Link Analysis Job Editor actually selects New in the Job menu (that is, it is unnecessary to click this menu option). However, if you are working in the Link Analysis Job Editor with unsaved changes and select New from the Job menu, the system displays a warning message that you have unsaved data. Click one of the following buttons in the Warning dialog box:

  - **Save:** Saves the Link Analysis job that you are creating or modifying, and clears the columns in the Job Options tab to the default values so that you may begin work on a new Link Analysis job.

  - **OK:** Does not save the data and clears the columns in the Job Options tab to the default values.

  - **Cancel:** Returns to the Link Analysis job without saving your data or creating a new Link Analysis job.

- **Save:** Enables you to save an existing Link Analysis job that has been modified, or a new job. You must complete the following fields on the Link Analysis Options tab before the Save option is available on the File menu:

  a. Select an entry in the **Select a Network Definition** drop-down list.

  b. Type a valid entry into the **Lookback Period** field.

  c. Type a valid entry into the **New Network Lookback Period** field.

The remaining fields on the Link Analysis Options tab are optional. Refer to section *Creating a Link Analysis Job*, on page 98, for details on creating and saving a Link Analysis job.

- **Copy:** Copies an existing Link Analysis job after you select it (refer to section *Copying an Existing Link Analysis Job*, on page 98, for more information).

- **Delete:** Deletes a Link Analysis job. You may only delete Link Analysis Job that you own (refer to section *Deleting a Link Analysis Job*, on page 99, for more information).

- **Close:** Closes the Link Analysis Job Editor and returns to the DataSet Editor. For details of this task, as the steps to exit the Link Analysis Job Editor are the same for exiting the Network Definition Editor, refer to *Closing the Network Definition Editor, on page 90*, for more information.

- **Exit:** Closes the Scenario Manager (refer to section *Exiting the Scenario Manager*, on page 91, for more information).

Jobs Area

The Jobs area lists all Link Analysis jobs for the selected domain and enables you to select jobs from this list for editing, copying, or deleting (Figure 40).


Figure 40. Jobs Area

The Jobs area includes the following columns of information:

- **ID:** Contains the ID of the Link Analysis job.

- **Description:** Contains a description of the Link Analysis job.

- **Creator:** Contains the user ID of the person that created the job.

- **Run Start:** Specifies the date the Link Analysis job was last executed.

- **Status:** Displays the status of the Link Analysis job.

Job Options Tab

The Job Options tab enables you to enter a description of the Link Analysis job and define its status (Figure 41).
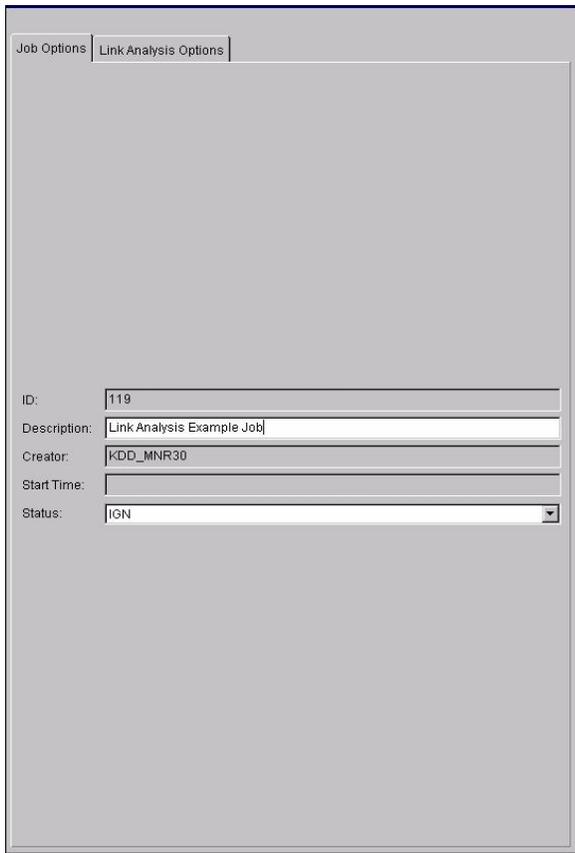
Figure 41. Job Options Tab

The components of the Job Options Tab include the following:

- **ID:** Non-editable.

  - When creating a new job, the text box is null; when you save the job, the system assigns an ID to the job.

  - When modifying an existing job, the text box contains the ID assigned to the selected job.

- **Description:** This field enables you to describe the Link Analysis job.

  - When creating a new job, the text box is null.

  - When modifying an existing job, the text box contains the description of the selected job.

- **Creator:** Non-editable.

  - When creating a new job, the text box is null; when you save the job, the system assigns a User ID to the job.

  - When modifying an existing job, the text box contains the user ID of the person that created the job.

- **Start Time:** Non-editable.

  - When creating a new job, the text box is null.

  - When modifying an existing job, the text box contains the date the job was last executed.

- **Status:** This field enables you to define the status of the Link Analysis job.

  - When creating a new job, the drop-down list contains the options: NEW (the default) or IGN (Ignore).

  - When modifying an existing job that has been executed or is currently being executed, the drop-down list contains the options that represent the status of the job. If the status is ERR, the RES option is added to the drop-down list.

Link Analysis Options Tab

Use the Link Analysis Options tab to filter the links evaluated in the construction of Networks and saved, resulting networks (Figure 42).

Figure 42. Link Analysis Options Tab

The components of the Link Analysis Options tab include the following:

- **Select a Network Definition**: Provides the names for each Network Definition associated with the Oracle Financial Services domain.

  - When creating a new job, the default is **<Select Network Definition>**.

  - When modifying or copying an existing job, the drop-down list contains the Network Definition of the selected job.

- **Lookback Period:** Number of days to look back from the current day to determine the range of links to be examined for network construction. The value must be a positive integer (≥ 1).

  - When creating a new job, the text box is null.

  - When modifying or copying an existing job, the text box contains the Lookback Period of the selected job.

- **New Network Lookback Period:** Number of days to look back from the current day to determine the date ranges for new networks. The New Network Lookback Period cannot be greater than the Lookback Period. All constructed networks that do not have a link with a timestamp in this range, after pruning, are considered old and are not processed further. To obtain optimal processing, set this to the frequency of the Link Analysis job executions. The value must be a non-negative integer (≥ 0).

  - When creating a new job, the text box is null.

  - When modifying or copying an existing job, the text box contains the New Network Lookback Period of the selected job.

Caution: If the Link Time Stamp column is null, all rows from the Source Link dataset are considered for networks. With nothing to reduce the result set of the query, large numbers of networks and nodes or links can be created. In addition, every time the job is run, the same networks are recreated.

- **Minimum Number of Nodes:** Specifies the threshold of which all constructed networks with fewer nodes than that specified are discarded. If the network contains fewer nodes than the value entered in this text box, the system prunes to entire network. The value must be a positive integer ($>=1$). The system discards small networks after pruning. Selecting a minimum number of nodes for the Link Analysis job is optional.

  - When creating a new job, the text box is null.

  - When modifying or copying an existing job, the text box contains the Minimum Number of Nodes of the selected job.

- **Select One or More Prunable Node Types:** Specify the types of nodes that are pruned from the network if they are only loosely related.

  Pruning is the process of eliminating uninteresting nodes from a network by removing nodes of specific types that are loosely related to the network (that is, related to only one other node in the network). A check box displays for each value specified as the From Node Type or To Node Type of each Source Link dataset in the Network Definition. If you do not specify a node type, the job does not consider any nodes for pruning. Selecting a prunable node type for the Link Analysis job is optional.

  - You must select a Network Definition in the Select a Network Definition area before you can select a node to prune.

  - Select the check boxes when you create a new job.

  - Select the check boxes associated with the desired job when modifying or copying an existing job.

> ■ Select zero (0), one (1), or more node types for pruning.

- **Status Bar:** Displays error messages for invalid entries and also shows message when change is made to either or both tabs - Job Options and Link Analysis Option. The following situations produce error messages in the Status Bar:

  > ■ Entering a value in the Lookback Period field that is not an integer or is an integer less than one (1).

  > ■ Entering a value in the Lookback Period field that is less than the value in the New Network Lookback Period field.

  > ■ Entering a value in the New Network Lookback Period that is not an integer or is an integer less than zero (0).

  > ■ Entering a value in the New Network Lookback Period that is greater than the value in the Lookback Period field.

  > ■ Entering a value in the Minimum Number of Nodes field that is not an integer or is an integer less than one (1).

Link Analysis Job Constraint

The database account that you use to run a Link Analysis job must have Insert and Update permissions to the table specified as the Network Characteristics dataset for the selected Network Definition.

Link Analysis Job Status or Termination

A job status of RUN indicates a job is currently in progress. In the KDD_RUN table, the RUN_MSG_TX column indicates which task within a job is currently running. A job status of FIN indicates successful execution.

A Link Analysis module terminates with an error condition if a metadata error occurs. A metadata error is just a generic error that refers to any issue with the data that might cause the job to terminate. For example, if the Link Analysis job references a dataset id for a dataset that does not exist a metadata error will be thrown.

A Link Analysis job run terminates either when all necessary information has been copied, or an error occurs. If an error occurs, the system records an ERR status in the KDD_RUN table and reports it to the looper process. The system then associates the ERR status with the entire job. To determine success or failure of a job and/or its associated run, you must query the KDD_JOB and KDD_RUN tables in the database.

Refer to Chapter 9, *Job Editors*, on page 161, for additional information about jobs.

Using the Link Analysis Job Editor

To follow these steps within the Link Analysis Job Editor, these sections assume you have completed the required steps in the Network Definition Editor. In addition, the instructions in these sections do not define how to run the Link Analysis job, but rather how to create, modify, copy an existing, or delete a Link Analysis job. Refer to Chapter 9, *Job Editors*, on page 161, for instructions on running a job.

Performing Link Analysis Procedures

This section covers the following procedures:

- Creating a Link Analysis Job
- Modifying an Existing Link Analysis Job

- Copying an Existing Link Analysis Job

- Deleting a Link Analysis Job

Creating a Link Analysis Job

To create a Link Analysis job, follow these steps:

1. From the **Windows** menu, select **Jobs** and then **Link Analysis**.

2. The Link Analysis Job Editor displays.

   Optional: Complete the **Description** and **Status** fields of the Job Options tab.

3. Click the Link Analysis Options tab to display it.

4. Select the desired Network Definition from the **Select a Network Definition** drop-down list.

   The system populates the Select One or More Prunable Node Types with the available nodes.

   Selecting a prunable node type for the Link Analysis job is optional.

5. Select the check box of one or more nodes to prune.

6. Complete the **Lookback Period**, **New Network Lookback Period**, and **Minimum Number of Nodes** fields.

   Selecting a minimum number of nodes for the Link Analysis job is optional.

7. Check the **Status Bar** at the bottom of the window for any entry errors into these fields.

8. From the **Job** menu, click **Save**.

   The **Save** option is not available if an error occurs.

   If no errors occur, the system saves the new Link Analysis job and displays the job in the Jobs area.

Modifying an Existing Link Analysis Job

To modify an existing Link Analysis job, follow these steps:

1. In the Link Analysis Job Editor, select the desired job to modify from the Jobs area.

   The system populates the Link Analysis Job Editor fields with the values from the selected job.

   Note: If the Link Analysis job is not created by you, then you cannot modify the job.

2. Modify the values of the copied job as needed.

3. Check the **Status Bar** for any entry errors into these fields.

4. From the **Job** menu, select **Save**.

   The **Save** option is not available if an error occurs.

   If no errors occur, the system saves the new Link Analysis job and displays the job in the Jobs area.

Copying an Existing Link Analysis Job

To copy an existing Link Analysis job to create a new job, follow these steps:

1. In the Link Analysis Job Editor, select the desired job to copy from the Jobs area.

2. From the **Job** menu, select the **Copy** option.

The system populates the Link Analysis Job Editor fields with the values from the selected job but provides a new job ID and you become the owner.

3. Modify the values of the copied job as needed.

4. Check the **Status Bar** for any entry errors into these fields.

5. From the **Job** menu, select **Save**.

   The **Save** option is not available if an error occurs.

   If no errors occur, the system saves the new Link Analysis job and displays the job in the Jobs area.

Deleting a Link Analysis Job

To delete a Link Analysis job, follow these steps:

1. In the Link Analysis Job Editor, select the desired job to delete from the Jobs area.

   Note: If the Link Analysis job is not created by you, then you cannot delete the job.

2. From the **Job** menu, select **Delete**.

   The system displays a confirmation dialog box.

3. Click **OK**.

   The system closes the dialog box and removes the job from the Jobs area.

Using the Rule Scenario Editor and the Sequence Scenario Editor to Identify Networks

After creating a Network Definition, use the Sequence Patterns or Rule Patterns Editor to specify patterns that identify interesting networks and are then used in generating matches. Refer to Chapter 4, *Sequence Scenario Editor*, on page 39 or Chapter 6, *Rule Scenario Editor*, on page 101, for information about using these editors.

Using Sequence Matcher or Rule Matcher Job Editor to Define Data Match Criteria

After creating the sequence or rule patterns, use the Sequence Matcher or Rule Matcher Job Editor to generate the matches. Refer to Chapter 9, *Job Editors*, on page 161 for information about using these editors.

CHAPTER 6

Rule Scenario Editor

This chapter defines the Rule Scenario Editor features and functionality that are in the Scenario Manager. This chapter focuses on the following topics:

- About the Rule Scenario
- Components of the Rule Scenario Editor
- About the Components of a Rule
- Using the Rule Scenario Editor
- About Pattern Reports
- Rule Scenario Design Tips

About the Rule Scenario

A rule pattern includes a primary dataset and optionally one or more secondary datasets. The primary dataset returns the records to be initially considered for the secondary datasets for matches (for example, all accounts with more than USD 50,000 in trades in the last month). Each secondary dataset enforces additional constraints on records that are related to the same focal entity identified in the primary dataset (for example, all trades in excess of USD 10,000 involving the account that the primary dataset identifies). The dataset may specify a range of records for the focus that must satisfy the constraint (for example, the system generates a match only if at least 10 trades in excess of USD 10,000 occur that involve identification of the account in the primary dataset).

A match is generated if the focal entity satisfies the constraints of the primary dataset, each of the secondary datasets, and the range on each secondary dataset. Flexibility of the secondary datasets enables a pattern to detect matches and augment analysis data (for example, highlights, building blocks, or scoring parameters) in a single application.

The Rule Scenario Editor provides access to the pattern elements that you need to create when using this editor for detecting behaviors of interest. The editor also provides access to all required features and functionality for creating rule patterns.

When creating a rule, consider the following guidelines:

- A dataset must be selected when defining a primary or sub-rule.
- The focus of the selected dataset links the primary rule and sub-rules together within a rule pattern.

Rule Scenario Editor Terminology

Table 20 provides a list of terms used to define the concepts of the Rule Scenario Editor.

Table 20. Rule Scenario Editor List of Terms

Term	Definition
Checkpoint	Specifies additional constraints that are applied only to the bindings defined within the primary rule and any sub-rules. Checkpoints do not have associated datasets.
Primary Rule	Specifies the matches based on a dataset and any additional constraints that are applied to the matches that the primary rule finds. You can apply only one primary rule within a rule pattern.
Rule Matching	Uses *if-then* Boolean structure for finding behaviors of interest.
Rule Scenario Editor	The tool used within the Scenario Manager that enables you to define, access, and maintain existing rule constraints for patterns in a search.
Sub Rule	Specifies additional pattern elements that are applied to matches found by the primary rule and that further defines the data matches. Many sub-rules can be included in a rule pattern.

Components of the Rule Scenario Editor

The Rule Scenario Editor contains the following components (Figure 43):

- Menu Bar

- Scenario Navigation Tree Area

- Scenario Elements Area

- Pattern Editor Area
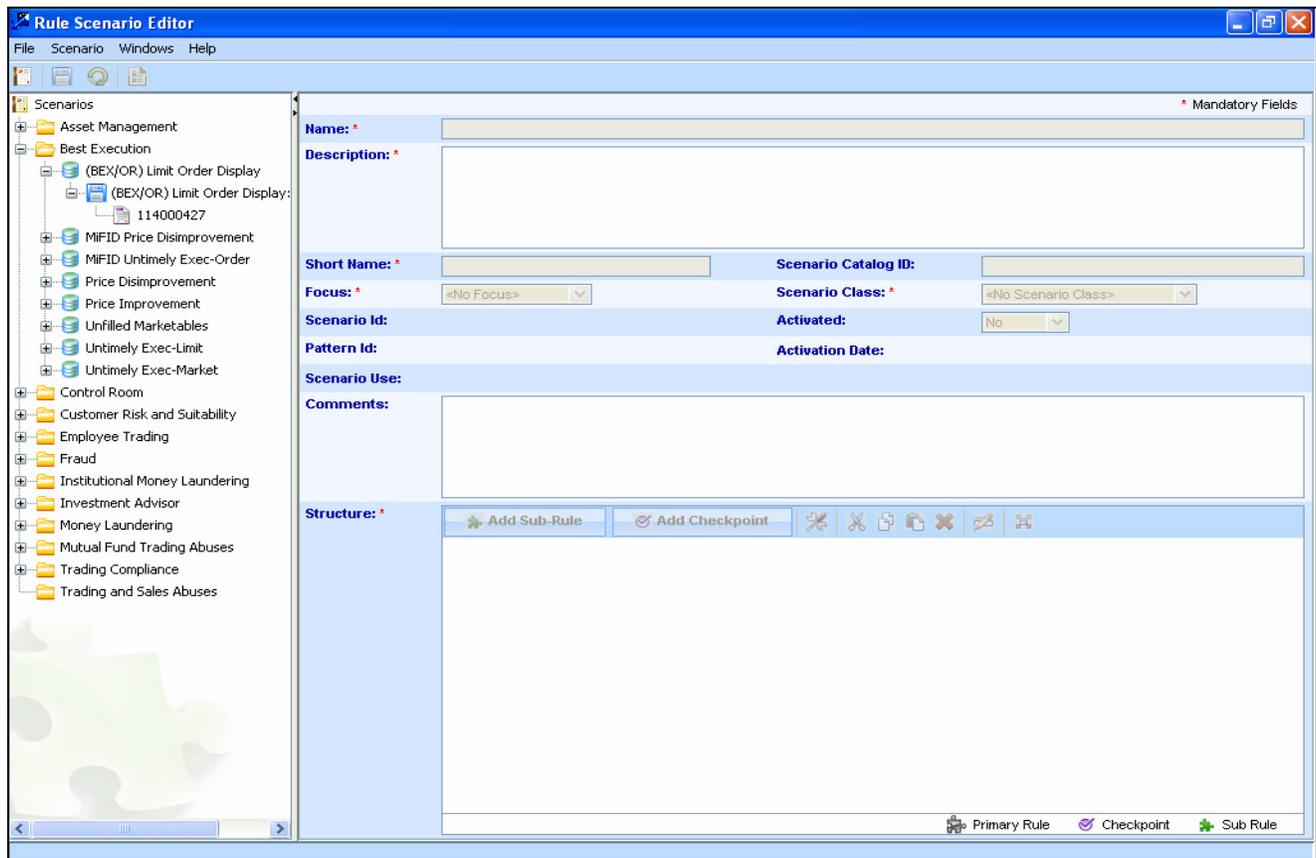
Figure 43. Rule Scenario Editor

Menu Bar

The Menu bar contains available commands, functions, and options. The following sections describe the Scenario menu options. Refer to Chapter 2, *Common Features*, on page 3, for information about the **Windows**, and **Help** menu options.

File menu

File menu contains two options:-

- **Close**: It will close the current window.

- **Exit** : It will close the Scenario Manager application.

Scenario Menu

The Scenario menu contains the following options:

- **Create New Scenario:** Creates a new scenario.

- **Copy Scenario:** Copies a scenario and the current pattern (not all patterns in a scenario are copied).

- **Copy Pattern:** Copies the current pattern and creates a new ID for that pattern within the same scenario. It does not copy the entire scenario or any other patterns within the current scenario.

- **Save:** Saves the current scenario with any changes you have made.

- **Save As:** Saves a copy of the open pattern with a unique ID.

- **Revert:** Reverts to the saved version of a pattern within a scenario. You can use this function when you have modified a pattern, however, you want to discard your changes and populate the component fields with the previous values.

- **Export:** Exports a rule pattern to an XML file.

- **Generate Pattern Report:** Creates an HTML file that contains detailed information about the current scenario pattern (refer to section *Generating a Pattern Report*, on page 129, for more information).

- **Import:** Imports a rule pattern from a previously exported scenario (in XML format).

- **Edit Augmentation Rules:** Enables you to add additional information (not used for data detection) to the Web Application pages to enable a business user to analyze data with in-house terminology that applies to their environment. Such values can include, however, are not limited to, Account Number or Stock Name. Refer to Chapter 8, *Augmentation*, on page 145, for additional information.

- **Edit Threshold Analyzer Roles**: Defines the Threshold Analyzer Roles. Refer to section *Defining Elements of a Threshold Analyzer Roles*, on page 124, for more information.

Scenario Navigation Tree Area

The scenario navigation tree area contains hierarchical relationships among Scenario Class, Scenario Short Name, Scenario Name, and Pattern (Figure 44)
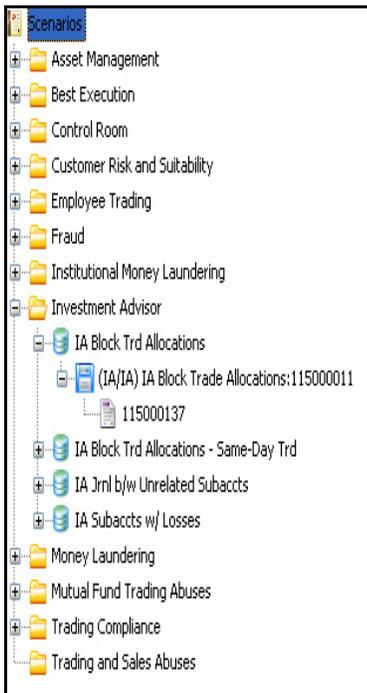
.

Figure 44. Rule Scenario Editor Navigation Tree Area

You can click the plus sign (+) or minus sign (–) next to an item in the tree to display or hide a list beneath the item. The tree contains the following items:

- **Scenarios:** Specify the list of Scenario Class. This is a top order in the hierarchy.

- **Scenario Class:** Specify the scenario class for this scenario. A scenario class groups scenarios together based on similar behaviors of interest.

- **Scenario Short Name:** Specify the abbreviated name for the scenario; the Web Application includes this element. A scenario short name groups pattern together based on similar behavior of interest.

- **Scenario Name**: Specify the actual name of the scenario.

- **Pattern:** Specify the pattern name.

Scenario Elements Area

The Scenario Elements area contains all required information for defining a rule scenario (Figure 45).

**Figure 45. Rule Scenario Editor Scenario Elements Area**

The Scenario Elements area contains the following elements:

- **Name:** Define the long version of the scenario name; the Web Application does not include this element.

- **Description:** Specify information that is pertinent to the scenario, this serves as the description in a generated pattern report.

- **Short Name:** Specify the abbreviated name for the scenario; the Web Application includes this element.

- **Scenario Catalog ID:** Identify the unique ID of a Oracle Financial Services scenario.

- **Focus:** Select the focal entity for a scenario; the focal entity key ID (for example, an account ID) must be captured in the pattern.

- **Scenario Class:** Specify the scenario class for this scenario. A scenario class groups scenarios together based on similar behaviors of interest.

- **Activated**: View the current status of the open scenario. The default is Activated; to deactivate a scenario, select the drop-down list next to the Activated field.

- **Comments:** Add any additional information you want to include about a scenario. This is typically used for detailed technical information, and displays only in the editor.

Pattern Editor Area

Use the Pattern Editor area to define a pattern that searches data for matches based on a dataset and any additional constraints (sub-rules or checkpoints) within the pattern (Figure 46).

Figure 46. Rule Scenario Editor Pattern Editor Area
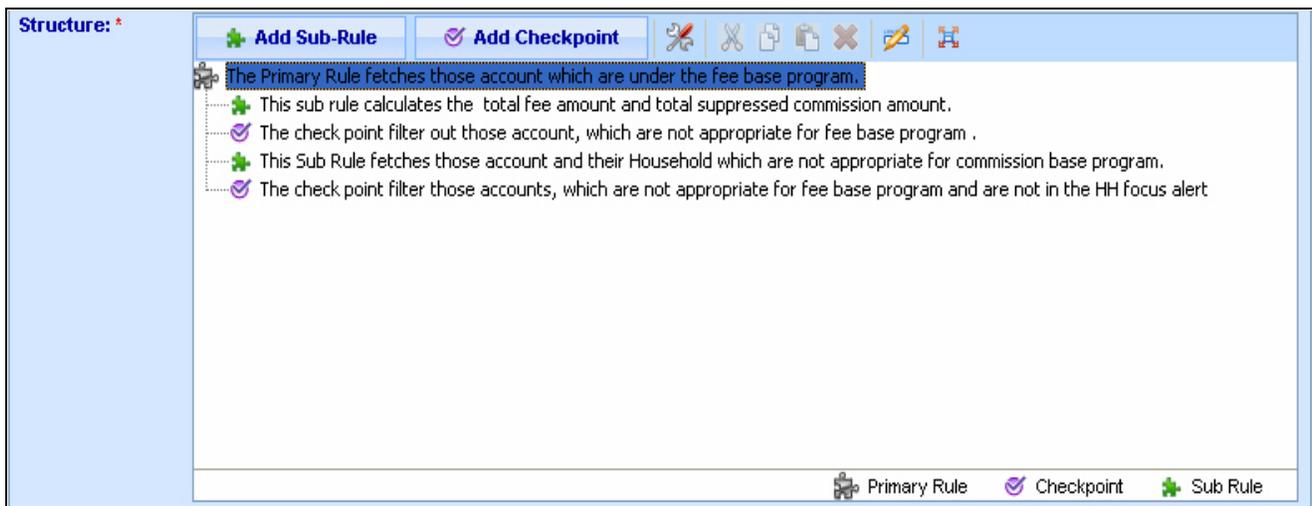
In the Pattern Editor area, you can define the elements of a primary rule, sub-rules, and checkpoints that create the rule pattern.

The Pattern Editor area contains the following elements:

- **Primary Rule:** Identifies the initial filter within a pattern; only one primary rule per rule pattern is allowed (refer to section *Primary Rule*, on page 108, for more information).

- **Sub-Rules:** Defines additional pattern elements that apply to matches that the primary rule detects. You can apply many sub-rules to a rule pattern; sub-rules are optional (refer to section *Sub-Rules*, on page 108, for more information).

- **Checkpoints:** Enables you to add additional constraints that must be satisfied to generate a match. Checkpoints evaluate variables captured in the primary rule or a sub-rule. A checkpoint does not have a dataset associated with it, and checkpoints are optional (refer to section *Checkpoints*, on page 108, for more information).

Pattern Editor Toolbar

The Pattern Editor area contains the following toolbar elements:

- **Add Sub-Rule:** Adds sub-rules to a rule scenario pattern.

- **Add Checkpoint:** Adds checkpoints to a rule scenario pattern.

- **Modify:** Enables to you make and save changes to a pattern component using the Edit Rule dialog box.

- **Cut:** Allows you to remove and copy the selected pattern component.

- **Copy:** Copies the selected pattern component.

- **Paste:** Pastes a previously-copied or cut pattern component before or after the selected component.

- **Delete:** Deletes the selected pattern component.

- **Edit Roles:** Defines the variables, format, and display type of scenario attributes.

- **Maximize/Restore:** Expand or Contract the Pattern Editor area.

About the Components of a Rule

Three components compose a rule pattern: primary rule, sub-rules, and checkpoints. The following sections describe each component.

Primary Rule

Use a primary rule within a rule pattern to:

- Select a dataset to be evaluated.

- Define constraints against the associated records.

- Specify the focal entity attribute.

- Define variables or bindings.

- Indicate whether dataset rows are recorded as matches.

- Indicate whether records that represent the same focus are clustered into a single, potential match or are listed as separate matches.

- Define thresholds within rule constraints and variable or bindings.

The conditions of the primary rule must be satisfied so the detection process can move forward and evaluate the sub-rules and checkpoints.

Sub-Rules

Use a sub-rule within a rule pattern to:

- Select a dataset for evaluation.

- Define constraints against the associated records.

- Specify attributes within the dataset that represent the focus as the primary rule.

- Define variables or bindings (sub-rules calculate variable sums, counts, minimums, maximums, and averages).

- Indicate whether dataset rows are recorded as matches.

- Define thresholds and variable or bindings instantiated in the primary rule, preceding sub-rules, and preceding checkpoints within rule constraints and variable or bindings.

Checkpoints

Checkpoints evaluate the variables within the rules and sub-rules in a pattern. Any findings based on data found by the rule and sub-rule constraints based on the elements and any calculations specified within the sub-rule rather than the dataset, because no dataset is defined within a checkpoint. Checkpoint constraints that are not met for a specific match are stopped and the search continues to the next match.

When matches are found, potential matches that do not match the rule pattern are not searched, and the search continues with the next condition in the rule pattern.

Comparing Primary Rules, Sub-Rules, and Checkpoints

Table 21 compares the rule scenario building components.

Table 21. Comparisons of Rule Scenario Building Components

Primary Rule	Sub-Rule	Checkpoint
Only one primary rule is defined per rule pattern.	Defines many possible sub-rules within a rule pattern.	Defines many possible checkpoints within a rule pattern.
Processes data to create matches.	Processes data to sort any unmatched data from the matches found in the primary rule.	Processes data to eliminate any unmatched data from the matches found in the primary rule and sub-rules.
Defines only one dataset for the primary rule.	Identifies a dataset unique to the primary rule for each sub-rule.	Does not define a dataset for a checkpoint.
Compares data to constraints within the specified dataset.	Compares matches of the primary rule and sub-rule. If they match, then compares potential matches to constraints within the primary rule and sub-rules within the pattern.	Compares data to constraints. If no matches are created, the checkpoint constraint continues to search for the next match.
Calculates sums, counts, minimums, maximums, and averages.	Calculates sums, counts, minimums, maximums, and averages.	Compares data to constraints, and does not create matches if a checkpoint constraint fails.

About Focus Clustering

Refer to section *Defining Elements for a Rule*, on page 111, for information about setting focus clustering.

Using Aggregate Bindings With Focus Clustering
You can use aggregate bindings in primary rules and sub-rules if you select focus clustering. Aggregate bindings perform computations across data records that are evaluated within a primary rule or sub-rule. Aggregate bindings calculate maximums, minimums, sums, averages, and counters. An attribute is defined for each calculated binding value, and the results of the aggregate binding calculations reflect values as they relate to the selected attributes.

The following expressions are used for performing calculations by applying aggregate bindings to a rule. The argument (arg) represents the selected attribute (predefined in the scenario) for defining your binding calculations:

- **Max(arg):** Finds the maximum attribute value for all records that meet the constraint.

- **Min(arg):** Finds the minimum attribute value for all records that meet the constraint.

- **Sum(arg):** Adds the attribute values for all records that meet the constraint.

- **Avg(arg):** Calculates averages of the attribute values for all rules that meet the constraint.

- **CountDistinct(arg):** Counts the number of unique entities found that meet the constraint.

- **Count():** Counts the total number of entities found that meet the constraint.

- **Count All():** Counts the total number of entities found regardless of the constraint.

> **Note:** You can use conditional expressions only for sum and average functions.

Refer to section section *To add bindings to a primary or sub-rule, follow these steps:*, on page 118, for information about defining expressions for rules.

Table 22 provides sample values for the function of aggregate bindings.

Table 22. Sample Values of Aggregate Bindings

Trades for Account 123	Amount of Trade (Attribute= AMT)	Trader (Attribute=TRADER)	$500 Constraint Met (Y/N)
Trade 1	100	Joe	N
Trade 2	1,000	Taylor	Y
Trade 3	10,000	Taylor	Y

Results are:

Min(AMT): $ 1,000

Max(AMT): $ 10,000

Sum(AMT): $ 11,000

Avg(AMT): $ 5,500

CountDistinct(TRADER): 1

Count(): 2

CountAll(): 3

Using the Rule Scenario Editor

The Rule Scenario Editor enables you to create a rule pattern by completing the procedures that this section describes.

Defining Elements for a Rule

You create the elements of a rule by specifying the parameters that compose the rule. Follow the listed procedures to create a rule pattern (not necessarily in this order).

- Creating a New Scenario

- Accessing the Rule Scenario Editor

- Defining Elements for a Rule

- Adding Constraints to a Primary or Sub-Rule

- Adding Bindings to a Primary or Sub-Rule

- Adding a Checkpoint to a Rule Pattern

- Adding Constraints to a Checkpoint

- Adding Bindings to a Checkpoint

- Creating Pattern Roles within a Rule Pattern

- Defining Elements of a Pattern Role

- Creating Augmentation Rules for a Pattern

- Saving a Scenario

Creating a New Scenario

To create a new scenario using the Rule Scenario Editor, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Patterns.**

   The Rule Scenario Editor window displays.

2. From the **Scenario** menu of the Rule Scenario Editor, select **Create New Scenario**.

   Alternatively: Click the **Create New Scenario** button from the toolbar.

   The Select Use window displays.

3. Select the environment type of the new scenario. For example, BRK, EXP, or SYS.

4. Click **Select**.

   The right panel displays the scenario fields in the Scenario Elements area.

5. Type the information in the Scenario Elements area.

6. A primary rule displays **[NO DESCRIPTION]** string in the Pattern Editor area.

7. Enter information of the primary rule.

   Note: A scenario must have atleast one primary rule.

8. Click **Save**.

Accessing the Rule Scenario Editor

Before you can create rule scenarios, you need to access the Rule Scenario Editor.

To access the Rule Scenario Editor, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Patterns**.

   The Rule Scenario Editor window displays.

2. From the scenario navigation tree area, navigate to the desired pattern element.

3. Double-click the pattern element or click the **Modify** button.

   The Rule Scenario Editor window enables and displays **PRIMARY RULE [DESCRIPTION]** string in the Pattern Editor area.

   Now you can define the pattern elements of the primary rule.

Defining Elements for a Rule

You can define rule element values in the fields of the Edit Rule dialog box, under Constraints and Bindings tabs (Figure 47).
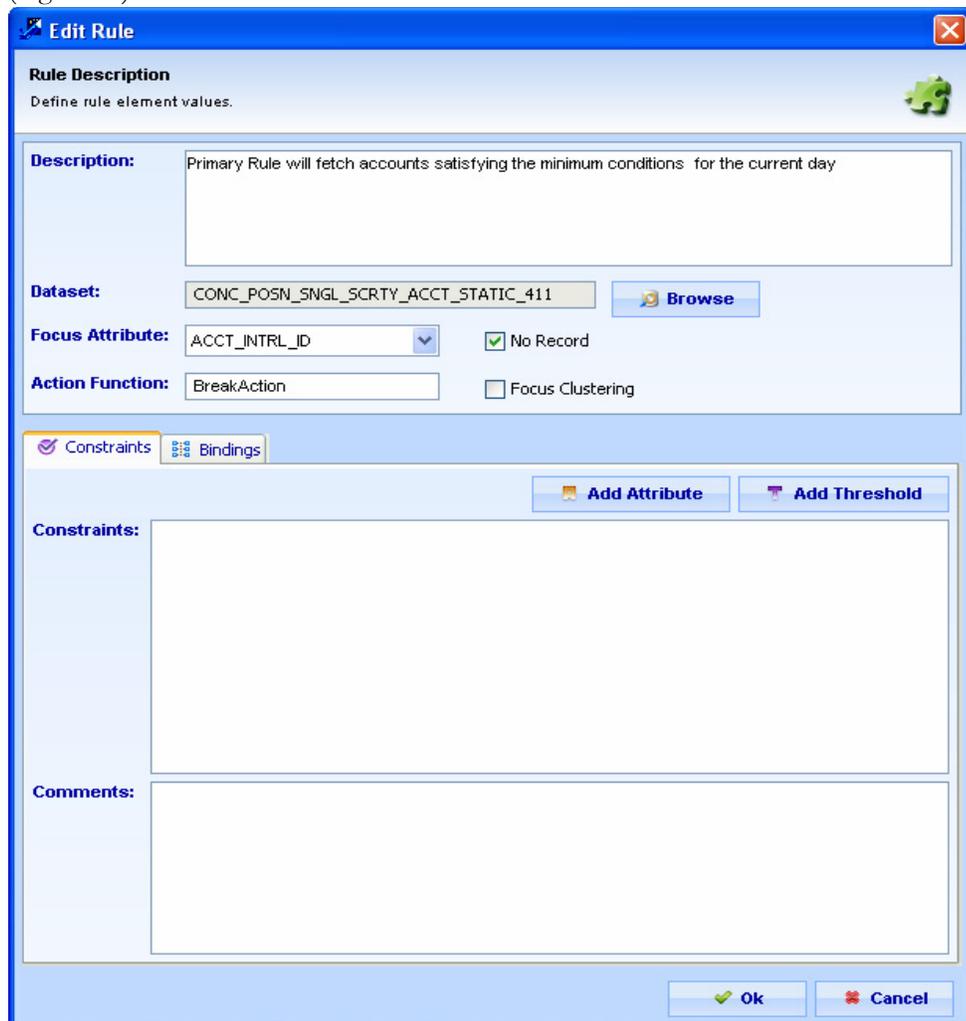
Figure 47. Edit Rule Dialog Box

The following sections describe the features for each of these fields and tabs.

Rule Description: Primary Rule

The Primary Rule Description enables you to specify element values to the rule pattern.

Figure 48. Edit Rule Description Fields—Primary Rule

The Primary Rule Description contains the following elements:

- **Description:** Enables you to specify the purpose of a rule or any desired information that relates to a rule.

- **Dataset:** Enables you to select a source dataset that contains the attributes that you want to use in the rule pattern.

- **Focus Attribute** drop-down**:** Enables you to select the focus attribute for a rule.

- **Action Function:** Contains the BreakAction field.

- **No Record:** Enables you to set a rule so that its key ID is not recorded and is not shown as a building block on the Web Application pages. The default setting (the No Record check box is not selected) records the rule.

- **Focus Clustering:** Enables you to specify whether evaluation of all records for a particular focal entity should occur collectively in the detection process. Under the default setting (not selected), the detection process evaluates the records from the primary rule one at a time.

Rule Description: Sub- Rule

The Sub-Rule Description enables you to specify element values for any sub-rules you add to the rule pattern (Figure 49).

Figure 49. Edit Rule Description Fields—Sub-Rule

The Primary Rule Description contains the following elements:

- **Description:** Enables you to specify the purpose of a rule or any desired information that relates to a rule.

- **Dataset:** Enables you to select a source dataset that contains the attributes that you want to use in the rule pattern.

- **Focus Attribute:** Enables you to select the focus attribute for a rule.

- **No Record:** Enables you to set a rule so that its key ID is not recorded and shown as a building block on the Web Application pages. The default setting (not selected) records the rule.

To define the elements for a primary or sub-rule, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Pattern**.

   The Rule Scenario Editor window displays (Figure 43).

2. From the scenario navigation tree area, navigate to the desired pattern element.

3. Double-click **Primary Rule [DESCRIPTION]** in the Pattern Editor area, or click **Modify**.

   Alternatively: Double-click **Sub-Rule [DESCRIPTION]** in the Pattern Editor area, or click **Modify**.

   The Edit Rule dialog box displays (Figure 48).

4. Type a description of the rule in the **Description** field.

5. Click **Browse** to open the Select DataSet dialog box.

   a. In the **DataSet** field, select the source dataset for the attributes used in this rule pattern. The **DataSet** list box displays all applicable datasets.

   b. Click **Ok** to close the Select DataSet dialog box.

   Optional: Select the **No Record** check box to indicate that you do *not* want to record the selected rule ID if a match is found for the entire sequence.

   Optional (Not applicable for Sub-Rule): Select the **Focus Clustering** check box to turn this feature on. Refer to *About Focus Clustering, on page 109*, for additional information.

6. Click **Ok**.

Adding Constraints to a Primary or Sub-Rule

The constraints of a rule specify the exact criteria specified within a rule that must be met for a match to occur. It can include any attributes from the selected dataset, or any variables or thresholds captured on previous rules.

Constraints Tab

The Constraints tab enables you to specify constraints for a primary and any sub-rules you add to a rule pattern (Figure 50).
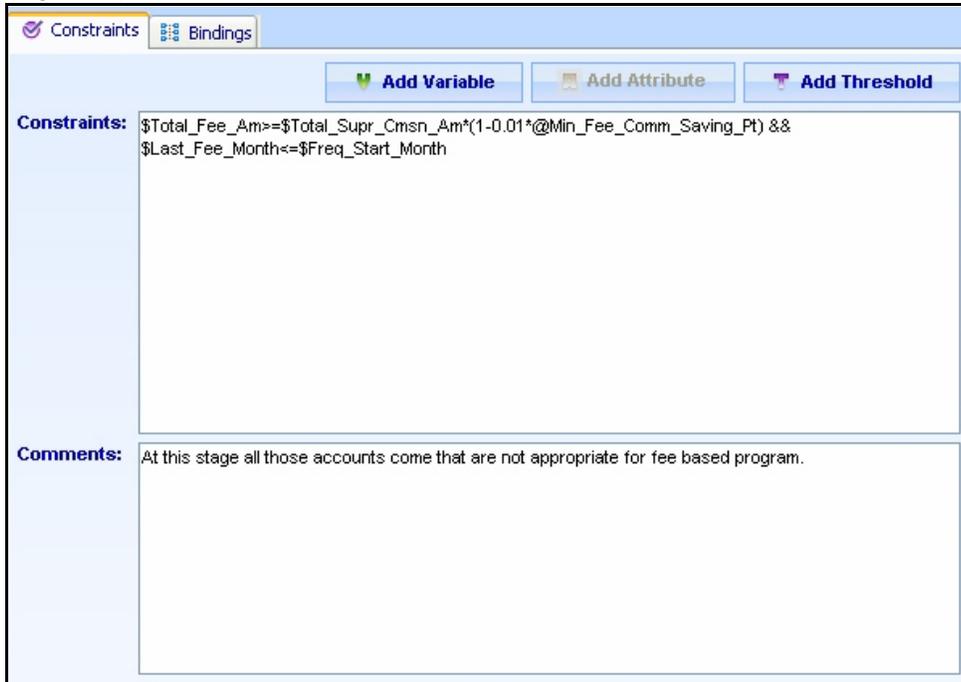

Figure 50. Edit Rule Dialog Box—Constraints Tab

The Constraints tab contains the following elements:

- **Constraints:** Specifies the exact criteria that the rule search must meet for matches to result. This fields requires a C-like syntax.

- **Comments:** Provides additional information about the constraint defined within the rule.

- **Add Variable:** Indicates the variables defined earlier in the pattern used to build your constraint. When clicked, a list displays all variables defined in the current rule or previous rules.

- **Add Attribute:** Specifies the attribute within the specified dataset used to build your constraint. When clicked, a list displays all related attributes from which you can select.

- **Add Threshold:** Specifies the thresholds within the scenario used to build your constraint. When clicked, a list displays all thresholds defined for the scenario.

To add constraints to a primary or sub-rule, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Patterns**.

   The Rule Scenario Editor window displays (Figure 43).

2. From the scenario navigation tree area, navigate to the desired pattern element.

3. Double-click **Primary Rule [DESCRIPTION]** in the Pattern Editor area, or click **Modify**.

   Alternatively: Double-click **Sub-Rule [DESCRIPTION]** in the Pattern Editor area, or click **Modify**.

The Edit Rule dialog box displays (Figure 48).

4. Click the Constraints tab.

   The Constraints tab displays (Figure 50).

5. Specify a constraint by doing any of the following:

   a. Click the **Add Attribute** button to select attributes you want as part of a constraint.

   b. *Not applicable for Primary Rule:* Click the **Add Variable** button to select variables you want as part of a constraint.

   c. Click the **Add Threshold** button to select thresholds you want as part of a constraint.

6. Click **Ok**.

   The Constraints displays attributes, variables, and thresholds in the order added.

   Optional: Type any details you want to add to the constraint in the **Comments** field.

7. Click **Ok**.

   The constraint string displays in the **Constraints**.

Adding Bindings to a Primary or Sub-Rule

Bindings are variables captured in a pattern; they are used for:

- Storing information from one record to compare against another

- Tracking running counts or aggregate amounts across multiple records

- Defining highlights

Bindings Tab

Use the **Bindings** tab to specify bindings for any rules and sub-rules that you add to the rule pattern (Figure 51).
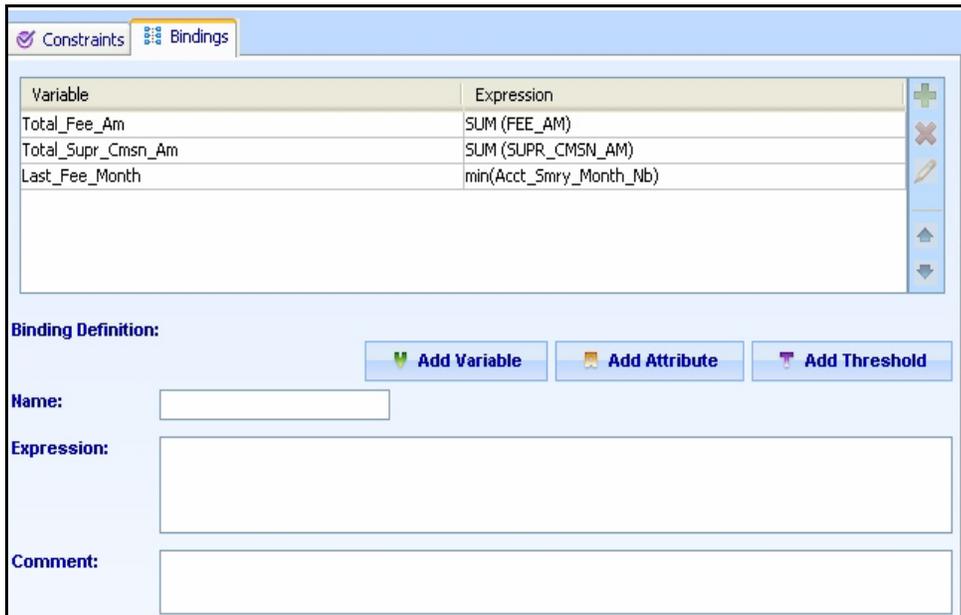
Figure 51. Edit Rule Dialog Box—Bindings Tab

Variables can use conditional (if/then/else) logic. Proper syntax for a bound threshold expression is:

cond(<condition>;<value>;<default value>)

You can also string together conditional statements, as follows:

cond(<condition 1>;<value 1>;<condition 2>;...;<condition n>;<default value>)

Note: Use conditional expressions only within sum and average aggregate functions.

The Bindings tab contains the following elements:

- **Variable and Expression:** Lists all variables and their definitions (expressions) that a rule defines.

- **Name:** Enables you to specify the attribute to add to a rule (the attribute must be in all of the datasets you previously selected).

- **Expression:** Specifies the bindings expression you want to add to a rule.

- **Comment:** Specifies any additional information about the bindings.

- **Add Variable**: Opens the Variable Selector dialog box, in which you can use another binding to define the current binding.

- **Add Attribute:** Opens the Attribute Selector dialog box, in which you can add attributes from the current dataset for use in defining the current binding.

- **Add Threshold:** Opens the Threshold Selector dialog box, in which you can add a threshold from the current scenario for use in defining the current binding.

- **Add:** Saves any new bindings to a rule.

- **Delete:** Deletes a rule binding.

- ✏️ **Modify:** Saves any changes you made to an existing rule binding.

- 🔼 **Move Up:** Scroll binding upward.

- 🔽 **Move Down:** Scroll binding downward.

When you create a new rule scenario (through the Rule Scenario Editor Scenario menu), the Primary Rule for the scenario displays in the Pattern Editor area.

To add bindings to a primary or sub-rule, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Patterns**.

   The Rule Scenario Editor window displays (Figure 43).

2. From the scenario navigation tree area, navigate to the desired pattern element.

3. Double-click **Primary Rule [DESCRIPTION]** in the Pattern Editor area, or click **Modify**.

   Alternatively: Double-click **Sub-Rule [DESCRIPTION]** in the Pattern Editor area, or click **Modify**.

   The Edit Rule dialog box displays (Figure 48).

4. Click the Bindings tab.

   The Bindings tab displays (Figure 51).

5. Type the name of the variable to be bound to the selected rule in the **Name** field.

6. Specify a binding by doing any of the following:

   a. Click the **Add Attribute** button to select attributes you want as part of a binding expression.

   b. *Not applicable for Primary Rule:* Click the **Add Variable** button to select variables you want as part of a binding expression.

   c. Click the **Add Threshold** button to select thresholds you want as part of a binding expression.

   The attributes, variables, and thresholds display in the Expressions field in the added order.

7. Click **OK**. Added bindings display in the **Bindings** list box.

   Optional: Type any additional information you want to associate to the selected rule in the **Comment** field.

Adding a Checkpoint to a Rule Pattern

When defining checkpoints in a rule pattern through the Edit Rule dialog box, follow the same process as for adding sub-rules with the following exceptions:

- Checkpoints search for primary rule and sub-rule bindings. They do not search from their own datasets as they do not have a dataset.

- Checkpoints do not allow you to add variables or focus attributes when defining checkpoints since all searches are based on the primary and sub-rule constraints.

- Use the Edit Rule dialog box for Checkpoint adding checkpoints to a rule.

The Edit Rule dialog box for Checkpoint enables you to specify constraints and bindings for any checkpoint that you add to the rule pattern (Figure 52).
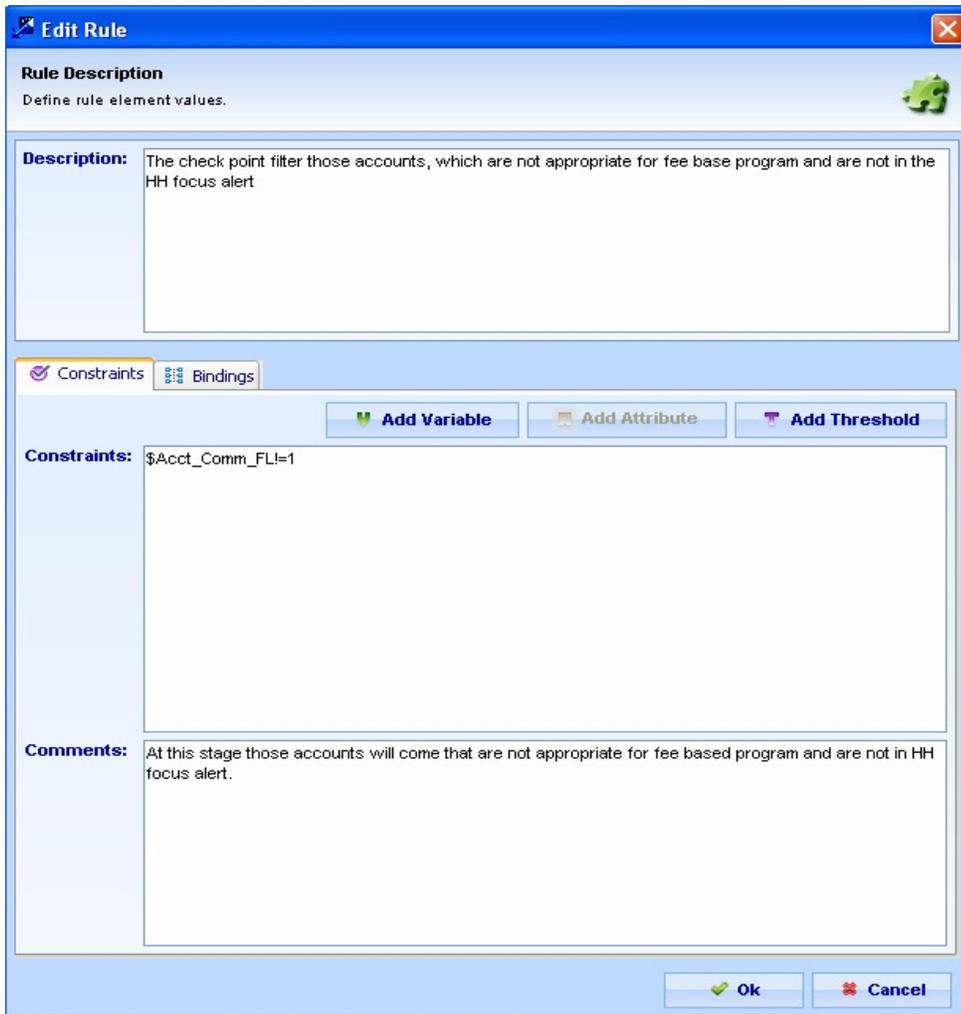

Figure 52. Edit Rule Dialog Box—Checkpoint

The Edit Rule dialog box for Checkpoint contains the **Description** field, in which you can specify a description for a checkpoint. The specified value displays in the Pattern Editor area of the Rule Scenario Editor dialog box (**Checkpoint [**_value_**]**).

To add a checkpoint to a rule pattern, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Patterns**.

    The Rule Scenario Editor window displays (Figure 43).

2. From the scenario navigation tree area, navigate to the desired pattern element.

3. Select the primary rule in the Pattern Editor area of the Rule Scenario Editor.

    The **Add Checkpoint** button enables.

4. Click **Add Checkpoint**..

The new rule displays as **Checkpoint [NO DESCRIPTION]** in the Pattern Editor area.

5. Double-click the **Checkpoint [NO DESCRIPTION]** string.

   The Edit Rule dialog box for Checkpoint displays showing the Checkpoint you want to name (Figure 52).

6. Type the description of the checkpoint in the **Description** field.

7. Click **Ok**.

   The Rule Scenario Editor window displays the specified description in the Pattern Editor area.

You are now ready to define constraints and bindings for the added checkpoint. Refer to section *Adding Constraints to a Checkpoint*, on page 120 and section *Adding Bindings to a Checkpoint*, on page 120, for more information about using these functions.

Adding Constraints to a Checkpoint

The constraints of a checkpoint specify the exact criteria that must be met for a match to occur. It can include any binding or variable specified in a previous rule or sub-rule and checkpoint.

Note: Adding constraints to a checkpoint is same as adding constraints to a primary or sub-rule. Refer to section section *Adding Constraints to a Primary or Sub-Rule*, on page 114, for instructions on how to add a constraint to a checkpoint.

Adding Bindings to a Checkpoint
The bindings of a checkpoint are variables captured in a pattern. They are used to store information from one record to compare against another. In addition, they track running counts or aggregate amounts across multiple records and define highlights.

Note: Adding bindings to a checkpoint is same as adding bindings to a primary or sub-rule. Refer to section section *Adding Bindings to a Primary or Sub-Rule*, on page 116, for instructions on how to add a binding to a checkpoint.

About Pattern Roles

Use pattern roles to specify the bound variables that are saved after a match is created.

You must add variables to pattern roles via the Pattern Roles Editor so that they are saved after matches are created, and can be used in highlights, augmentation, or scoring.

Creating Pattern Roles within a Rule Pattern

Use the following guidelines when defining pattern roles:

- Use pattern roles to determine the bound variables that are saved after a match is created.

- Check that other variables are written to the log files, however, not saved in the KDD tables after a job is complete.

- Add variables to pattern roles if they are used for highlights, augmentation, or scoring.

- Use the variable as a highlight if a format string is available.

- Arrange the variables in the order they should display in the highlights using the **Move Up** or **Move Down** buttons.

Defining Elements of a Pattern Role

When creating a pattern role, you need to define several elements, which are described in the following sections.

Pattern Roles General Tab

The General tab enables you to specify elements for defining a role in your pattern (Figure 53).
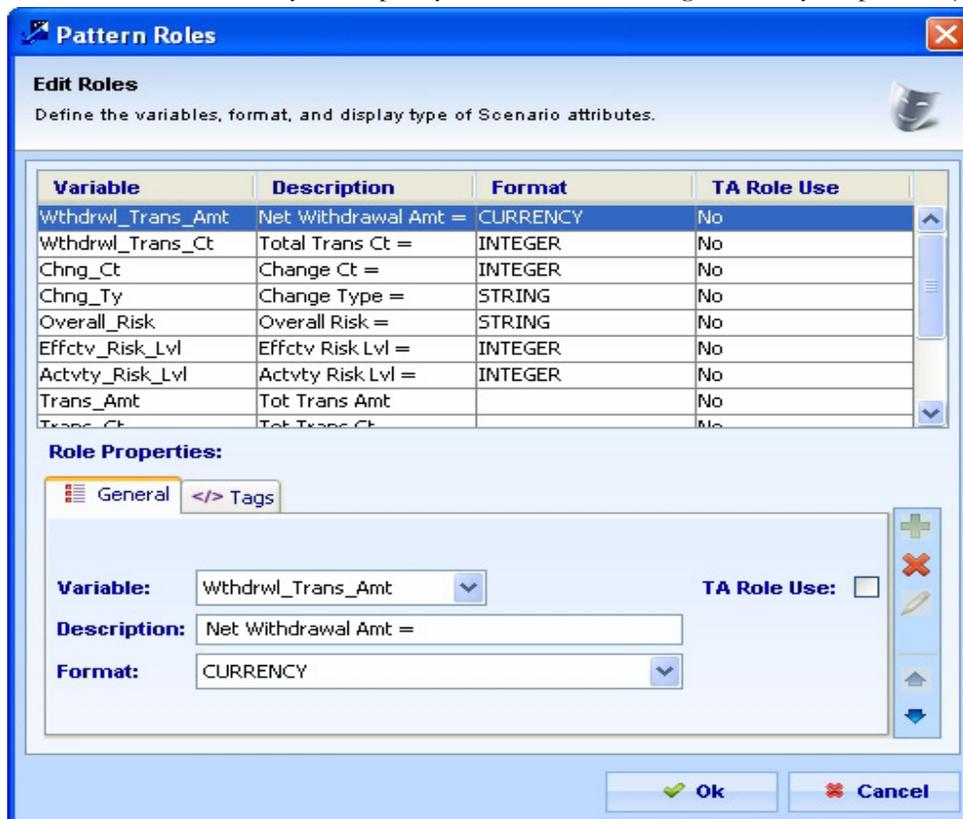
Figure 53. Pattern Roles Dialog Box—General Tab

The Pattern Roles General tab contains the following elements:

- **Pattern Roles** grid: Lists all roles defined for a rule scenario.

- **Variable**: Enables you to select the variables you want to save to the KDD_BREAK_BINDING table. Clicking the column header orders the list of roles alphabetically by variable name.

  Note: If you select a variable that has been defined for other patterns of the same scenario as the current pattern, and that variable has been designated for use as a Threshold Analyzer Role, Scenario Manager automatically selects the **TA Role Use**.

- **Description**: Enables you to specify any additional information to include in a role. Clicking the column header orders the list of roles alphabetically by description.

- **Format**: Enables you to select the desired format and to define whether or not to display the specified variable. If you select the **<NOT DISPLAYED>** option in the drop-down list, the variable does not display in the Web Application.

- **TA Role Use:** Enables you to specify if a pattern role variable should be available for the threshold analyzer role in Threshold Analyzer Roles dialog box. By default, the **TA Role Use** check box is clear.

- **Add**: Saves the specified pattern role information.

- **Modify**: Applies changes to a pattern role.

- **Delete**: Deletes a pattern role. Clicking the **Delete** button removes the variable from the KDD_PTTRN_ROLE table, and the associated variable from the KDD_TA_ROLES table on the condition that no other patterns for a scenario should have a pattern role record for that particular variable.

- **Move Up**: Changes the order of a role by moving the selected role up.

- **Move Down**: Changes the order of a role by moving the selected role down.

- **OK**: Saves, changes, and closes the dialog box.

- **Cancel**: Cancels any changes and retains the previously saved role information.

Table 23 defines the most commonly used pattern role format field options.

Table 23. Pattern Role Format Field Options

Format Option	Description
<NOT DISPLAYED>	Does not display the variable in the Web Application, when selected.
STRING	Does not apply formatting to the value returned from the database. This format is also useful for numbers that should not have thousands separators, for example, numeric account numbers.
INTEGER	Applies thousands separators to whole numbers.
REAL2	Applies thousands separators to real numbers, truncating to two digits of fractional data.
REAL5	Applies thousands separators to real numbers, truncating to five digits of fractional data.
$.00	Applies thousands separators and a currency symbol, for example, $, to a number. Fractions are displayed. For example, a positive number displays as $1,234.56 and a negative number displays as ($1,234.56).
$	Applies thousands separators and a currency symbol to a number. Fractions are rounded to the nearest whole number. For example, positive numbers display as $1,234 and negative numbers display as ($1,234).

Table 23. Pattern Role Format Field Options (continued)

Format Option	Description
$K	Applies thousands separators and a currency symbol to a number. Fractions are rounded to the nearest whole number. If the number is 1,000,000 or greater, the number is rounded to the nearest thousand, and the final three digits are replaced with the letter "K." For example, positive numbers display as $1,234 or $2,346K for $2,345678.90 and negative numbers display as ($1,234) or ($2,346K) for -$2,345678.90.
CURRENCY	Applies the appropriate currency format (including base currency) to a value.
MMDDYY	Displays the short form of the date (for example, 05/18/04).
MMDDYYYY	Displays the long form of the date (for example, 05/18/2004).
DATETIME	Displays the short date format and time format in the 24-hour clock (for example, 05/18/04 23:59:59).
HHMMSS	Displays the long time in 24-hour format. The expected input is an integer that represents a number of seconds. For example, 93599 seconds is displayed as 25:59:59.
%	Displays the data as a percentage.
ALT_CUR	Used for cases where all highlights/roles are of the same currency, but NOT base currency. The format for the amount(s) should be ALT_CUR. Then, the scenarios will record an additional binding with binding name Alt_Currency_Code that holds the currency code. The highlights generator would then/take find the value from the Alt_Currency_Code binding and use it to format all of the bindings with format ALT_CUR. For example: *OrderPrice* role with format ALT_CUR and Alt_Currency_Code binding with value as ISSUE_CRNCY_CD from pattern, let's say value is EUR. Thus *OrderPrice* will be displayed as xxx EUR. Here xxx is the value coming from pattern.
REF_CUR	Used for cases where all highlights/roles are not of the same currency (including base currency) or different bindings having different currencies to display, use a format code REF_CUR for the amount, and then have an additional binding with the same name, but with _CUR on the end and contain the currency to be used. For example, define *TradePrice* and *ActivityPrice* roles with format REF_CUR and TradePrice_CUR, and ActivityPrice_CUR binding to hold the Currency to be used for the above, say EUR and USD respectively. Thus they will be displayed as *TradePrice : xxx EUR & ActivityPrice : zzz USD*. Here xxx and zzz are the values coming from pattern.
HHMMSS_PT	Displays the long Time format in 24-hour clock with milliseconds precision. For example, the display of 103001500 will be as 10:30:01:500

To define elements of a pattern role, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Patterns**.

   The Rule Scenario Editor window displays (Figure 43).

2. From the scenario navigation tree area, navigate to the desired pattern element.

3. In the Pattern area, click **Edit Roles**.

   The Pattern Roles dialog box displays (Figure 53).

4. Select the desired variable from the **Variable** drop-down list.

5. Type the text you want associated with this role in the **Description** field.

6. Click the arrow of the **Format** drop-down list to identify the highlights from other pattern roles, and to define the display of the pattern roles on the Web Application pages (refer to Table 23 for details about the options for this field).

7. Select the **TA Role Use** check box if you want to assign pattern role variable to a threshold analyzer role.

8. Click **Ok**.

Creating Tags for Pattern Roles

The tags function is not required and is typically not used.

Defining Elements of a Threshold Analyzer Roles

The Threshold Analyzer Roles dialog box enables you to define elements for defining a role in your pattern. These elements are then used in the Threshold Analyzer utility for further analysis. Refer to *Administration Tools User Guide,* for more information on Threshold Analyzer utility.

Figure 54. Threshold Analyzer Roles Dialog Box

The Threshold Analyzer Roles dialog box contains the following elements:

- **Threshold Analyzer Roles**: Lists all roles defined for a rule scenario. The grid does not display roles, if you had not defined threshold analyzer role for the scenario earlier. However, the grid displays the role, if you had already defined variables for the scenario, even if for a different pattern than the current one.

   In addition, on selecting a particular role, the associated fields populates in the Role Properties section.

- **Role Properties**: The Role Properties section enables you to define variables. It contains the following elements:

  - **Variable**: Pre-populates the variables defined for all patterns for the current scenario, where you have selected the **TA Role Use** check box in the Pattern Role window.

  - **Display Name**: Pre-populates the description as defined in the Pattern Role window for the selected variable, excluding the "=" sign. You can modify the description text up to 100 characters in length.

  - **Format**: Pre-populates the format defined in the Pattern Role window for the selected variable. If you change threshold analyzer role for which a format was defined for the pattern role, then you must also select the format currently specified for the pattern role from the **Format** drop-down list.

  - **Round Digits Count**: Enables you to select value based on the specified numeric type format in the **Format** drop-down list. The **Round Digit Count** drop-down list is disabled, if the format specified is a non-numeric type.

  - **Display Units**: Enables you to enter value (for example, Hundreds, Thousands, and so forth) based on the specified numeric type format in the **Format** drop-down list. The **Display Units** is disabled, if the format specified is a non-numeric, or a numeric with percentage type, respectively.

  - **Variable Field Name**: Enables you to select a numeric or non-numeric value based on the following:

    - Existence of the Threshold Analyzer data table for the current scenario class and the pattern-binding variable as a TA Role variable within that class

    - Existence of the Threshold Analyzer data table for the current scenario class and but no corresponding pattern-binding variable as a TA Role variable within that class

    - No Threshold Analyzer data table exists for the current scenario class

    ```
    If there is a defined Threshold Analyzer data table for the current scenario's
    class AND if the pattern binding variable exists within the KDD_TA_ROLE table for
    a scenario of the same class the drop-down will pre-populate with only the
    FIELD_NM_TX defined for that variable and shall not be editable.
    ```

    ```
    If there is a defined Threshold Analyzer data table for the current scenario's
    class AND the pattern binding variable does not yet exist within the KDD_TA_ROLE
    for a scenario of the same class, the drop-down list will include:
    ```

    - <blank> - default

    - <Create New>

    - List of Column Names from the Threshold Analyzer data table applicable to the current scenario's class (for example, columns defined for the KDD_TA_ML_DATA table).

    ```
    If there is no Threshold Analyzer data table defined for the current scenario's
    class, the options will include:
    ```

    - <blank> - default

    - Number Field Position 1 thru Number Field Position 20 (show 1 thru 20 as separate options)

    - Character Field Position 1 thru Character Field Position 10 (show 1 thru 10 as separate options)

    ```
    If the specified format is of a numeric type, this drop-down list will display
    only the blank option and:
    ```

    - The Number Field Position 1 - 20 options (where list is populated for scenario class with no TA data table)

◆ The applicable TA data table columns of type NUMBER (28, 8)

`If the specified format is of a non-numeric type, this drop-down list will display only the blank option and:`

◆ The Character Field Position 1 - 10 options (where list is populated for scenario class with no TA data table)

◆ The applicable TA data table columns of type VARCHAR2 (100).

■ **Use as Axis**: Enables you to select the value (for example, blank, Yes, or No) based on the specified format as numeric in the **Format** drop-down list. On selecting *Yes*, this field appears in the Threshold Analyzer utility UI, where it indicates variable in the variables list for X/Y axis graph. Refer to *Administration Tools User Guide*, for more information.

● **Add**: Saves the specified threshold analyzer role information. The information is stored in the `KDD_TA_ROLES` table.

● **Modify**: Applies changes to a threshold analyzer role information.

● **Delete**: Deletes a threshold analyzer role. Clicking the **Delete** button removes the record associated with the threshold analyzer role, without impacting the pattern role.

● **Move Up**: Changes the order of a role by moving the selected role up.

● **Move Down**: Changes the order of a role by moving the selected role down.

● **OK**: Saves, changes, and closes the Threshold Analyzer Roles dialog box.

● **Cancel**: Cancels any changes and retains the previously saved role information.

To define elements of a threshold analyzer role using the Rule Scenario Editor, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Patterns**.

   The Rule Scenario Editor window displays (Figure 43).

2. From the scenario navigation tree area, navigate to the desired pattern element.

3. In the Pattern area, click **Edit Roles**.

   The Pattern Roles dialog box displays (Figure 53).

4. In the Pattern Role dialog box, follow these steps:

   a. Select the desired variable from the **Variable** drop-down list.

   b. Type the text you want associated with this role in the **Description** field.

   c. Click the arrow of the **Format** drop-down list to identify the highlights from other pattern roles, and to define the display of the pattern roles on the Web Application pages.

      Optional: Select the **TA Role Use** check box if you want to assign pattern role variable to threshold analyzer role.

   d. Click **Modify**.

5. Click **Ok** to close the Pattern Roles dialog box.

6. From the **Scenario** menu of the Rule Scenario Editor, select **Edit Threshold Analyzer Roles**.

   The Threshold Analyzer Roles dialog box displays (Figure 54).

7. In the Threshold Analyzer Roles dialog box, follow these steps:

a. Select the desired variable from the **Variable** drop-down list.

b. Modify the text you want associated with this role in the **Display Name**.

c. Select the format from the **Format** drop-down list.

d. Select the field position from the **Variable Field Name** drop-down list. By default, the value specified is blank.

e. Select the value from the **Use as Axis** drop-down list.

f. Type the unit in the **Display Units** .

g. Select the numeric value from the **Round Digits Count** drop-down list.

h. Click **Add**.

8. Click **OK** to close the Threshold Analyzer Roles dialog box.

Creating Augmentation Rules for a Pattern

In the Rule Scenario Editor, you can create Match and Binding Augmentation rules for a pattern. Match Augmentation adds building blocks to matches and Binding Augmentation adds highlights to matches. Refer to Chapter 8, *Augmentation,* for information about using the Augmentation tool.

Saving a Scenario

After completing all procedures to configure your scenario, you need to ensure that you save the scenario information.

To save a scenario, click **Save** in the **Scenario** menu. The system saves the scenario with the same name you entered in the Name field in the Rule Scenario Editor dialog box (Figure 43).

Refer to section section *Using the Sequence Scenario Editor*, on page 45, for additional information about modifying and deleting a scenario

Copying a Scenario

To copy a scenario, follow these steps:

1. From the **Scenario** menu of the Rule Scenario Editor, select **Copy Scenario**.

   A new copy of the scenario displays.

2. Click **Save As** and then provide a new name for the scenario.

Caution: When copying a scenario, the following message displays when a scenario contains a threshold or dataset that contains thresholds:

You are attempting to copy a scenario that uses a dataset(s) that includes a threshold(s). If you copy this scenario and alter the thresholds, you also alter the threshold(s) for any other scenario that uses this dataset(s). In the Administration Tool's Threshold Editor, you can only modify the threshold value from the original scenario.

Use extreme caution when copying a scenario with this characteristic, because you create dependencies among scenarios. Changing thresholds affects both the original and copied scenarios.

Copying the Pattern

To copy a pattern, follow these steps:

1. From the Scenario menu, select **Copy** Pattern.

2. Click **Save As** to save a copy of the pattern in same scenario.

About Pattern Reports

You can use the Sequence Scenario Editor and the Rule Scenario Editor to generate and print details of the selected pattern and its scenario using the *Generate Pattern Report* function. You can use this information for analysis or troubleshooting purposes.

The Rule Pattern Report is generated based on the current pattern within a rule scenario (not all patterns in a scenario). You need to generate a separate pattern report for each pattern.

This information is valuable from a Data Miner's perspective because it enables you to analyze scenario information efficiently in one place versus having to navigate throughout the entire tool to view all of the separate pieces of a scenario.

You can also use this report when a business user makes changes to scenario elements on the business side and then reports any issues or problems. Using the pattern report, you can compare the changes that the user made to the original scenario information to help solve the issue.

This section contains step-by-step procedures for generating a pattern report. The report generates an HTML report that contains pertinent information that describes a scenario and pattern, including:

- Scenario information

- Pattern information

- Schematic of the pattern logic

- Description of the pattern logic

- Constraints for each row or rule

- Bindings for each row or rule

- Saved pattern roles

- Thresholds

- Threshold Sets

- Threshold Usage

- Augmentation rules

- Description

- Comments

- Datasets (SQL that defines each dataset in the pattern)

Generating a Pattern Report

Use the following guidelines when generating pattern reports:

- Name the generated scenario report using elements from the scenario. A desired format may be scenario *name_pattern ID*, since pattern ID is a unique parameter within any given scenario.

- Print in landscape to ensure that the formatting is accurate and the printed report is readable.
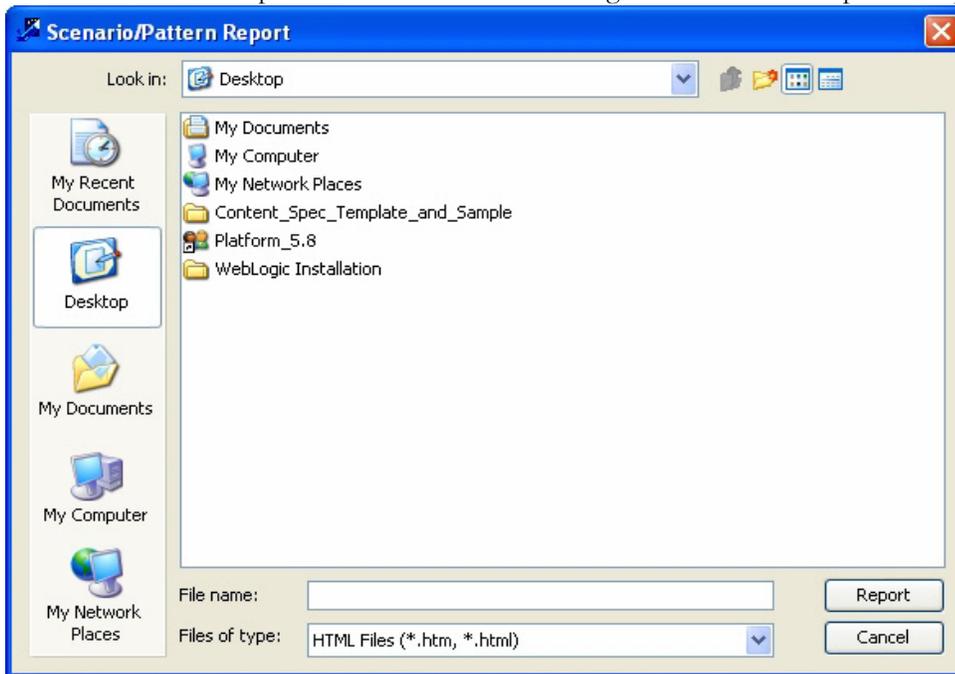
Figure 55. Scenario or Pattern Report Dialog Box

To generate a pattern report, follow these steps:

1. From the **Windows** menu of the DataSet Editor, select **Rule Patterns**.

   The Rule Scenario Editor window displays (Figure 43).

2. From the scenario navigation tree area, navigate to the desired pattern element.

3. Double-click the pattern ID in the scenario navigation tree area.

   The pattern information displays in the Rule Scenario Editor window.

4. From the **Scenario** menu of the Rule Scenario Editor, select **Generate Pattern Report**.

   Alternatively: Click from the toolbar.

   The Scenario/Pattern Report dialog box displays (Figure 55).

5. Select the location where you want to save the generated pattern report by navigating via the **Look In** field.

6. Type the desired filename in the **File Name** field.

7. Click **Report**.

The system creates the report and saves it to the specified directory. You can then open and print the report through your IE browser.

Rule Scenario Design Tips

The following tips can be helpful when working with rule patterns:

- Limit the rows retrieved in the datasets that a pattern uses. Specifically, if certain records cannot satisfy the pattern criteria, it is best to apply logic to filter them out beforehand. If certain detection conditions can be checked in the dataset, it is generally more efficient to enable the database to filter them out rather than applying this logic in the pattern. You should also determine the appropriate lookback period range that is necessary to identify the behaviors of interest, to prevent the patterns from looking through a larger volume of historical data than is really necessary.

- Apply the most restrictive pattern conditions in the primary rule. Rule patterns access data incrementally, retrieving records only for the entities that satisfy the conditions of the primary rule. Applying the most limiting conditions beforehand helps to reduce the total number of records that are retrieved and evaluated in the detection process.

- Use periodic checkpoints to terminate the search at the point when the conditions for a match can no longer be satisfied. Rule patterns include explicit checkpoints that must be satisfied in order for the detection process to move forward. If the conditions in a checkpoint are not met for a particular entity, the detection process does not continue to retrieve data for that entity. Developers should apply checkpoints in patterns wherever they can substantially reduce the population of records being considered for a match.

Refer to Appendix A, *Scenario Design Guidelines, on page 175*, for more information about design tips and guidelines.

CHAPTER 7 — *Threshold Manager*

This chapter defines the features and functionality of Threshold Manager of the Scenario Manager. This chapter focuses on the following topics:

- About the Threshold Manager
- Threshold Manager Components
- Using the Threshold Manager

About the Threshold Manager

A *threshold* is a parameter used in a detection process that a user can adjust by using the Administration Tools. A *threshold set* is a collection of threshold values that are used together in alert generation jobs for a variety of applications.

The Threshold Manager enables you to create thresholds and threshold sets for use in patterns and datasets, and to specify the data type, definition, and adjustable range for each threshold. You can also adjust the current value of a threshold.

Note: Before creating a threshold, you must first create a scenario. Refer to Chapter 4, *Sequence Scenario Editor*, on page 39, for more information about creating a scenario.

Threshold Manager Components

You can access the Threshold Manager by selecting the **Thresholds** option in the Scenario Manager **Windows** menu (Figure 56).
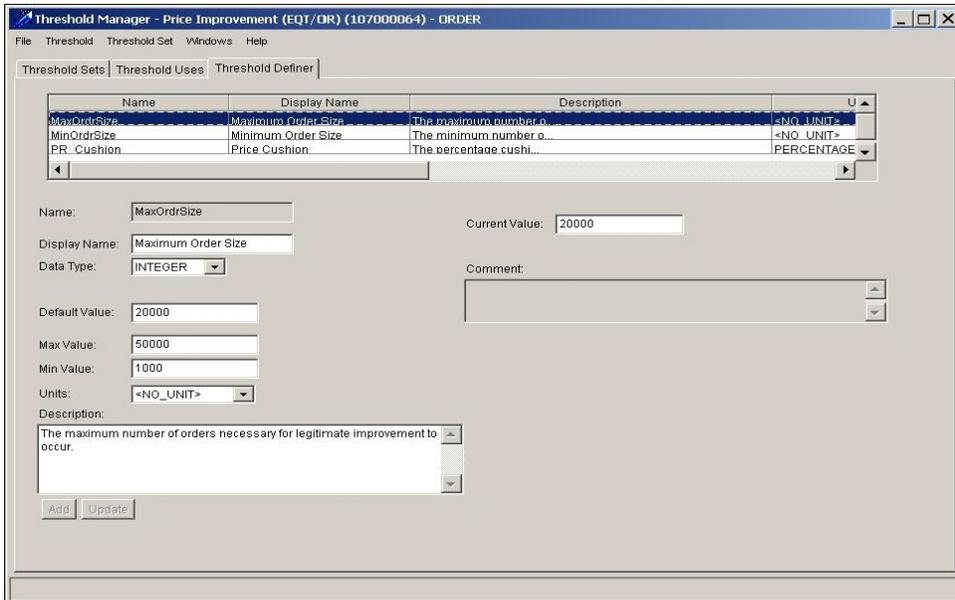
Figure 56. Threshold Manager Window

The Threshold Manager includes the following components:

- Menu Bar
- Threshold Tabs

The following sections describe these components.

Menu Bar

The Menu Bar contains the following menus:

- File Menu
- Threshold Menu
- Threshold Set Menu
- Windows Menu
- Help Menu

File Menu

The File menu contains the following options:

- **Open Scenario:** Opens a scenario for which you want to define or modify thresholds.
- **Close:** Closes the Threshold Manager.
- **Save:** Saves changes that you made in the Threshold Manager.

- **Revert:** Reverts to the previously saved thresholds. You can use this option if you do not commit any changes with the **Save** option.

- **Threshold Report..:** Displays a portion of a scenario report that describes threshold information.

- **Exit:** Closes the Scenario Manager.

Threshold Menu

The Threshold menu contains the following options:

- **New:** Enables you to create a new threshold.

- **Delete:** Enables you to delete a threshold.

Threshold Set Menu

The Threshold Set menu contains the following options:

- **New:** Enables you to create a new threshold set from the base set.

- **Delete:** Enables you to delete a threshold set.

Windows Menu

The Windows menu contains the following options:

- **DataSets:** Launches the DataSet Editor

- **Networks:** Launches the Link Analysis Network Definition Editor

- **Rule Patterns:** Launches the Rule Scenario Editor

- **Sequence Patterns:** Launches the Sequence Scenario Editor

- **Jobs:** Accesses the following editing tools:

  - **Rule Matcher:** Launches the Rule Matcher Job Editor

  - **Sequence Matcher:** Launches the Sequence Matcher Job Editor

  - **Link Analysis:** Launches the Link Analysis Job Editor

Help Menu

The Help menu contains the About option, which provides information about the current version of the Scenario Manager application.

Threshold Tabs

The Threshold Manager contains configured fields in the following tabs:

- **Threshold Sets:** Enables use of the same scenario with different threshold values at runtime.

- **Threshold Uses:** Displays information about the patterns and datasets that employ the thresholds.

- **Threshold Definer:** Allows specification of display name, description, relevant values (maximum, minimum, current, and sample), data types, and units.

Using the Threshold Sets Tab

The Threshold Sets tab provides the contents of the Base Threshold Set and any existing sets (which you access in the Threshold Sets drop-down menu), which is shown in Figure 57. You can use this base set to create an additional (derived) threshold set and use it at runtime in the detection process; the new set also contains the thresholds that the base set defines. For each threshold set that you create, you need to define a new job in the appropriate job editor.
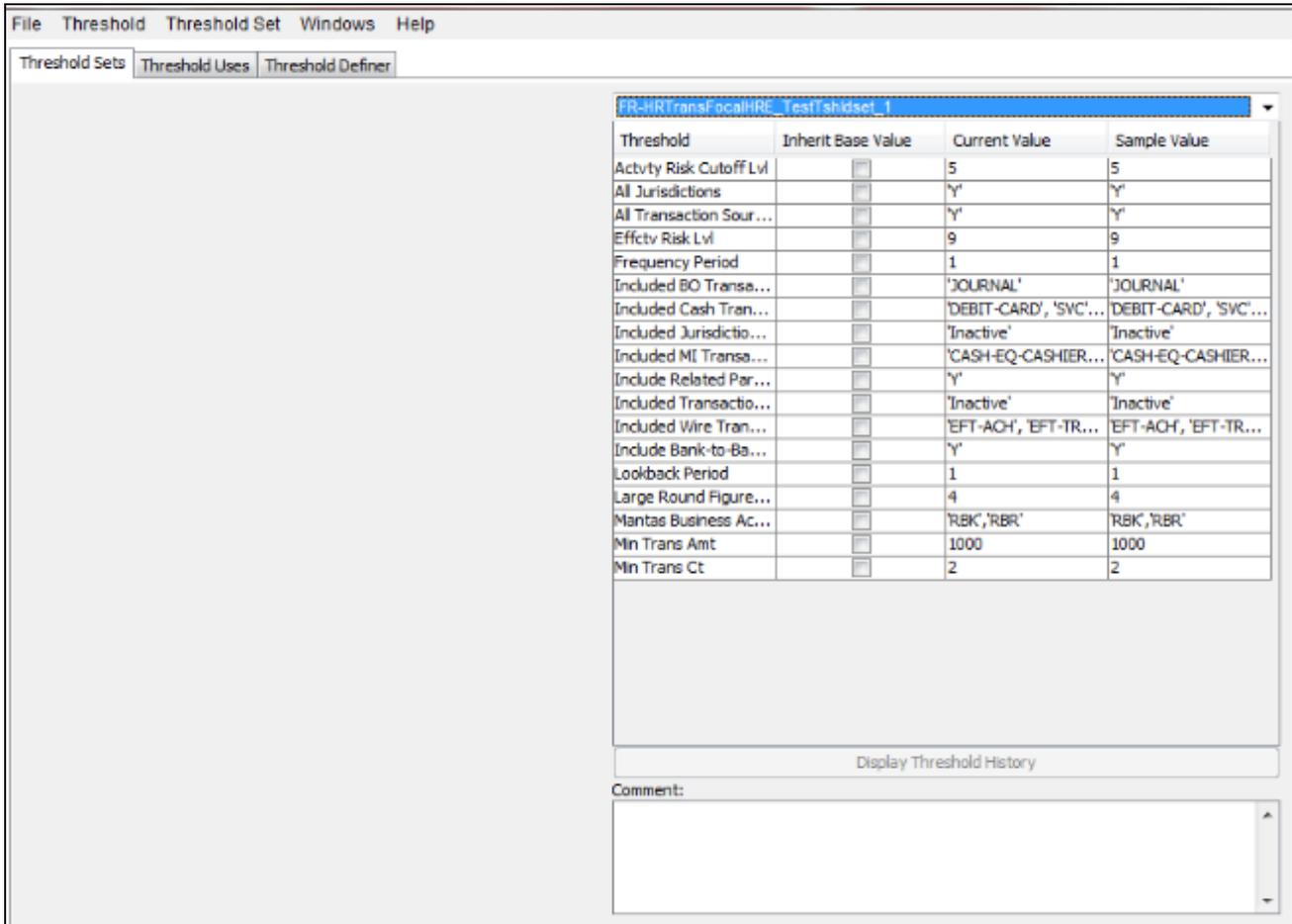
Figure 57. Threshold Sets Tab

Note: You can create a threshold set for specific requirements for each jurisdiction in which a scenario is applied. Threshold sets maybe created for several reasons, and jurisdiction is one of those reasons.

The Threshold Sets tab contains the following components:

- **Threshold Set Description:** Allows you to select and modify threshold values within a Threshold Set to be used at runtime.

- **Display Threshold History:** Displays the Threshold History dialog box which provides the history of the selected threshold.

When you select a Threshold in the Threshold Set Description area for modification, the Display Threshold History button is enabled. Selection of this button displays the Threshold History dialog, which provides history details of the selected threshold. The Threshold History dialog box displays (Figure 58).
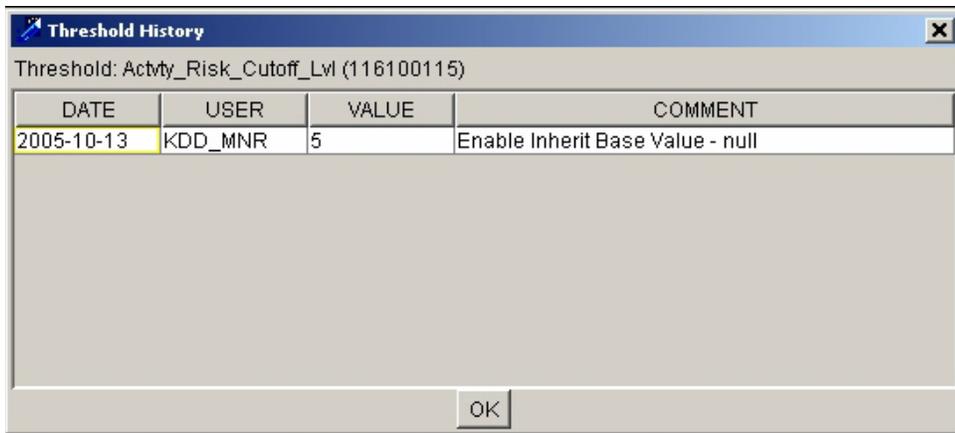
Figure 58. Threshold History Dialog Box

Using the Threshold Sets Description Area

The Threshold Set Description area describes the thresholds in the selected Threshold Set. Table 26 describes the fields in this area.

Table 24 describes the fields in the Threshold Sets tab.

Table 24. Threshold Sets Description Area Fields

Field	Description
Threshold Sets	Drop-down list that allows you to select an existing threshold set
Threshold	Name of the threshold
Inherit Base Value	Indicator that a threshold has the same values as that of the corresponding base threshold
Current Value	Current value of the threshold within the threshold set
Sample Value	Sample value of the threshold within the threshold set
Comment	Additional comments that you want to add about the threshold (up to 4,000 characters)

Using the Threshold History Dialog Box

The Threshold History dialog box displays the history details of the selected threshold when you select the Display Threshold History button in the Threshold Sets tab.

Table 25 describes the fields in the Threshold History dialog box.

Table 25. Threshold History Dialog Box Fields

Field	Description
Threshold	Displays the name of the selected threshold.
Date	Displays date the threshold was created.
User	Displays name of the user that created the threshold.
Value	Displays current value of the threshold.
Comment	Displays comments added to the threshold.

Using the Threshold Uses Tab

The Threshold Uses tab provides information about the current use of thresholds for a scenario (Figure 59).
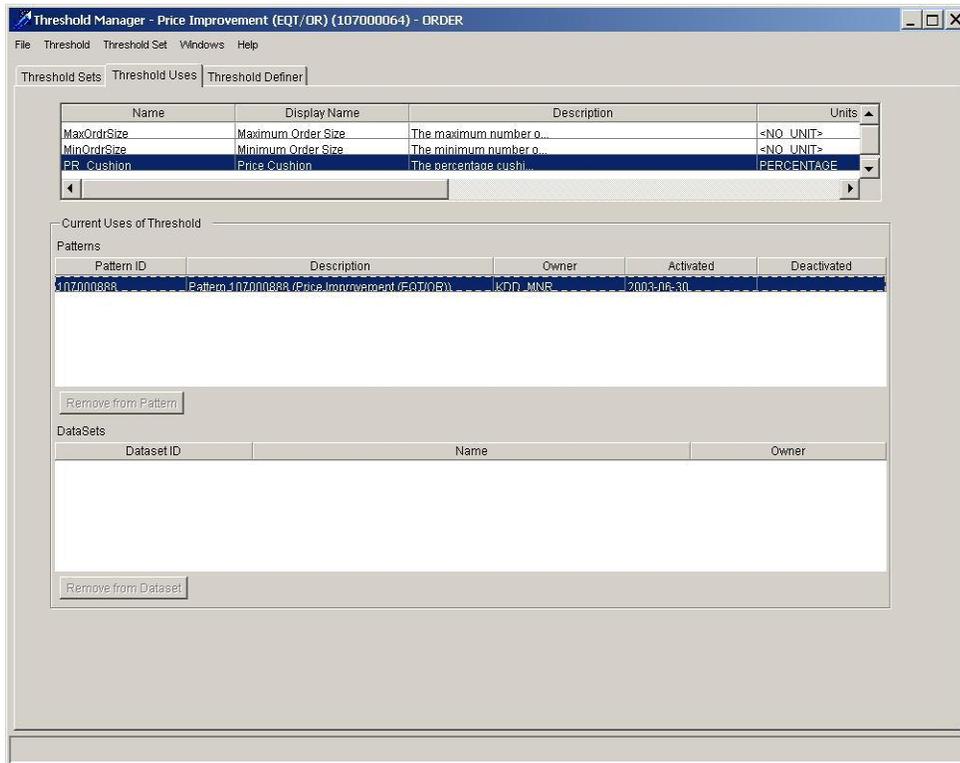
Figure 59. Threshold Uses Tab

The Threshold Uses tab contains the following components:

- **Threshold List**: Contains the names of the active thresholds for a scenario.
- **Current Uses of Threshold**: Describes the patterns and datasets that use a particular threshold.

Using the Threshold List

The **Threshold List** lists and describes the threshold in the selected threshold set. Table 26 describes the fields in this area.

Table 26. Threshold List Fields

Field	Description
Name	Name of the threshold that the threshold set contains
Display Name	Actual display representation of the threshold. This field has a 255-character constraint
Description	Description of the threshold
Units	Unit of measure for the variable defined in a threshold as it applies to the defined data type
Data Type	Data representation (for example, integer or string) of the threshold
Sample	Base value that is defined for a threshold
Current	Current value of the threshold in the base threshold set that the scenario or dataset uses
Min	Minimum threshold value
Max	Maximum threshold value

Using the Current Uses of Threshold Area

The **Current Uses of Threshold** area provides you with information about the patterns and datasets that use a particular threshold. The following sections describe the areas that contain these elements.

About Patterns

The Patterns area lists all patterns in ascending order (based on Pattern ID) in which the selected threshold is used. Table 27 describes the fields in this area.

Table 27. Patterns Area Fields

Field	Description
Pattern ID	ID of the pattern.
Description	Description of the pattern.
Owner	Owner of the pattern.
Activated	Date the pattern was last activated.
Deactivated	Date the pattern was last deactivated.

In the Pattern area, clicking the Remove from Pattern button enables you to remove a selected threshold from an associated pattern and replace it with the current value of the threshold in the Base Threshold Set. If this function is not active, you can remove the threshold only in the appropriate pattern editor.

Refer to the section section *Removing Thresholds from a Pattern and Dataset*, on page 142, for more information.

About DataSets

The DataSets area lists all datasets in ascending order (based on DataSet ID) in which the selected threshold is used.

The DataSets contain the following fields:

- **DataSet ID:** ID of the dataset

- **Name:** Name of the dataset

- **Owner:** Owner of the dataset

Note: If the datasets that augmentation uses contain thresholds and views used in augmentation then use the values of the thresholds from the scenario's base threshold set. The values of the thresholds in the base set may be different from the values used in the detection job.

The DataSets area also contains the Remove from DataSet button, which enables you to remove a selected threshold from an associated dataset and replace it with the current value of the threshold in the Base Threshold Set. If this function is not active, you can only remove the threshold in the DataSets Editor.

Note: You can remove a threshold from a threshold set only if the scenario is not in break (BRK) mode.

Refer to the section section *Removing Thresholds from a Pattern and Dataset*, on page 142, for more information.

About the Threshold Definer Tab

The Threshold Definer tab displays the active, base version of the thresholds associated with the selected scenario (Figure 60).
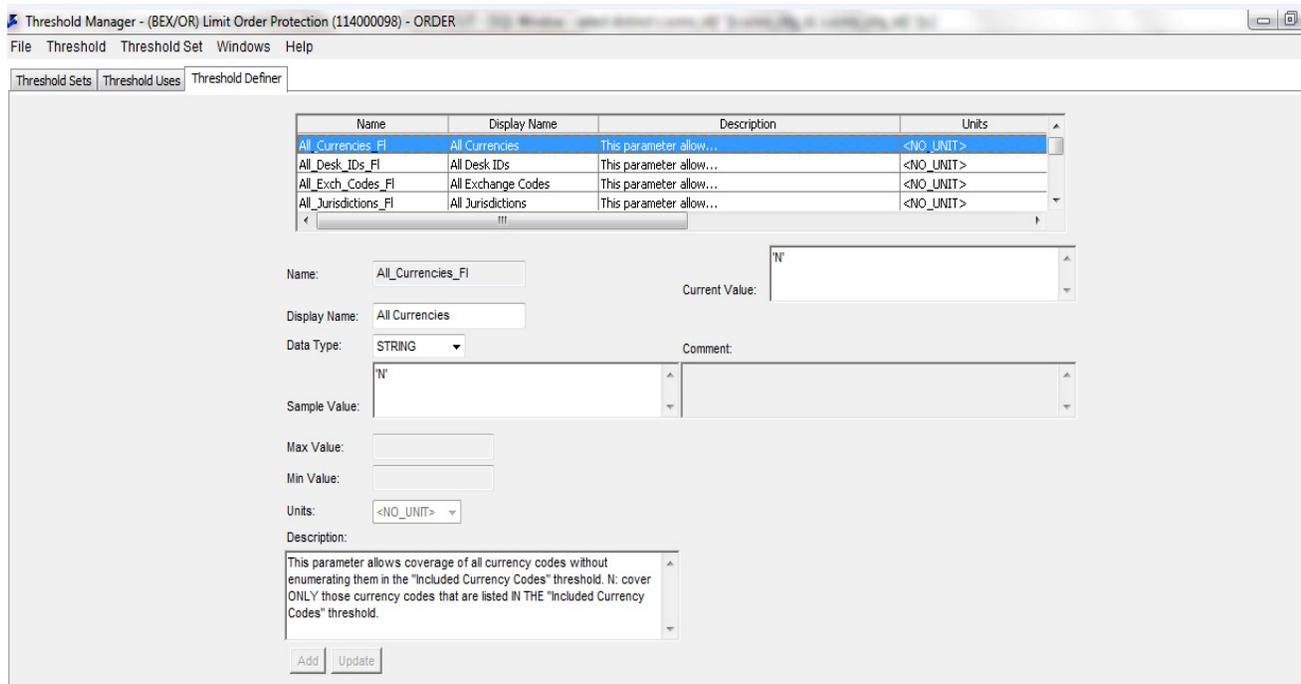
Figure 60. Threshold Definer Tab

The Threshold Definer tab contains read-only fields in the Threshold List Area, and modifiable threshold fields that correspond directly to the fields in the Threshold Description area. When you select a threshold in the Threshold Description area, the values for this threshold fill the modifiable fields automatically. You can then use these values to modify the selected threshold or to create a new threshold. Refer to the section section *Defining Thresholds*, on page 141, for more information.

Using the Threshold List Area

The Threshold List area contains the following fields:.

Table 28. Threshold List Area Fields

Field	Description
Name	Name of the threshold that the set contains
Display Name	Actual display representation of the threshold. This field has a 255-character constraint
Description	Description of the threshold
Units	Unit of measure for the variable defined in a threshold as it applies to the defined data type
Data Type	Data representation (for example, integer or string) of the threshold
Sample	Base value that is defined for a threshold
Current	Current value of the threshold in the base threshold set that the scenario/dataset uses
Min	Minimum threshold value
Max	Maximum threshold value

Using the Modifiable Threshold Fields

Modifiable threshold fields include the following:

- **Name:** Name of each threshold in a threshold set.

   Note: When you create and save a new threshold, the threshold name is no longer modifiable.

- **Display Name:** Name of a threshold as it appears in the Threshold Editor in the Admin Tools.

- **Current Value:** Current value of the threshold.

- **Data Type**: Data types for a threshold from available data types. After making a selection, other fields in this dialog box change to accommodate the required entries. Table 29 describes valid field values.

Table 29. Data Type Field Values or Entries

Data Type Field Value	Field Entries
INTEGER	● **Active fields:** Sample Value, Current Value, Max Value, Min Value, and Units ● **Valid values for *Units* field:** SECONDS, MINUTES, HOURS, DAYS, MINUTES, and MONEY NO FRAC ● **Max Value in Active fields (except units)**: 999999999
REAL	● **Active fields:** Sample Value, Current Value, Max Value, Min Value, and Units ● **Valid values for *Units* field:** MONEY and PERCENTAGE ● **Max Value in Active fields (except units):** 99999999999999999999 (OR 20 times 9)
BOOLEAN	● **Active fields:** Sample Value, Current Value ● **Valid values for *Sample Value* and *Current Value* fields:** TRUE or FALSE
STRING	● **Active fields:** Sample Value, Current Value
LIST	● **Active fields:** Sample Value, Current Value

- **Sample Value:** Base value that is defined for a threshold.

- **Comment:** Additional comments that you want to add about the threshold (up to 4,000 characters).

- **Max Value:** Maximum limit to which you can change a threshold value.

- **Min Value:** Minimum limit to which you can change a threshold value.

- **Units**: Unit of measure for the variable defined in a threshold as it applies to the defined data type. Valid values are MONEY and PERCENTAGE (which apply to the REAL data type) and time increments (which applies to INTEGER data type).

- **Description:** Description of a threshold.

- **Add:** Adds a threshold to a scenario that displays in the Threshold List.

- **Update:** Confirms any changes you made in the threshold fields.

Using the Threshold Manager

The Threshold Manager enables you to create, modify, and delete thresholds to use in a rule or sequence scenario. You can also remove thresholds from a pattern or dataset. The following sections provide instructions for performing the following procedures:

- Defining Thresholds
- Modifying Thresholds
- Removing Thresholds from a Pattern and Dataset

Defining Thresholds

Use the following guidelines when defining thresholds:

- The threshold name must be unique to any other threshold that exists in the current scenario.
- The sample value and current value must be within the minimum and maximum value limits.
- The sample current, minimum, and maximum values must correspond to the data type specified within the current threshold.

To define a threshold, follow these steps:

1. Click the Threshold Definer tab in the Threshold Manager.
2. Click the cursor on a blank line in the Threshold List area. If other thresholds are listed, click after the last threshold or from the Threshold menu, select **New**.
3. Enter the appropriate information for the new threshold.
4. Click **Add** to add the threshold to the current scenario.
5. The threshold displays in the Thresholds List area.

   Optional: Add any comments about the threshold in the Comment field.
6. Click **Update** to confirm your changes.
7. Select **Save** in the **File** menu to save your threshold.

Modifying Thresholds

You can change the parameters of an existing threshold within the Threshold Manager.

To modify a threshold, follow these steps:

1. Click the Threshold Definer tab in the Threshold Manager.
2. Select the threshold that you want to modify in the Threshold List area.

   The details of the threshold display in the fields in the tab.
3. Make the desired changes.

   Optional: Add any comments about the threshold in the Comment field.
4. Click **Update** to confirm your changes.
5. Select **Save** in the **File** menu to save your threshold.

To Delete a Threshold

To delete a threshold from a rule or sequence scenario, follow these steps:

1. Click the Threshold Uses tab in the Threshold Manager.

2. Select the threshold you want to delete from the Threshold List area.

3. From the **Threshold** menu, click **Delete**.

    The Warning dialog box displays (Figure 61).

4. Click **OK**.

5. Select Save in the File menu to confirm the deletion of the threshold from the current scenario.

    Note: You can remove a threshold only if the scenario is in EXP mode.

Removing Thresholds from a Pattern and Dataset

When you remove a threshold from a pattern or dataset, the Threshold Manager replaces each occurrence of the threshold in the pattern with the current value of the threshold in the base threshold set.

A record is added to the threshold history table for the base version of the threshold. This sets the deactivation timestamp to the current date, the deactivation identifier to the current user and adds the deactivation comment.

You can remove a threshold from a pattern or dataset, which disassociates a threshold from a pattern or dataset. You can remove a threshold from a dataset or pattern whenever needed.

To remove a threshold from a pattern or dataset, follow these steps:

Note: You can remove a threshold only if the scenario is in EXP mode.

1. Click the Threshold Uses tab in the Threshold Manager.

2. Select the threshold you want to remove.

3. In the Current Uses of Thresholds area, do one of the following:

    ● Select the pattern that contains the threshold you want to remove and click **Remove from Pattern**.

    ● Select the dataset that contains the threshold you want to remove and click **Remove from DataSet**.

    The Warning dialog box displays (Figure 61).
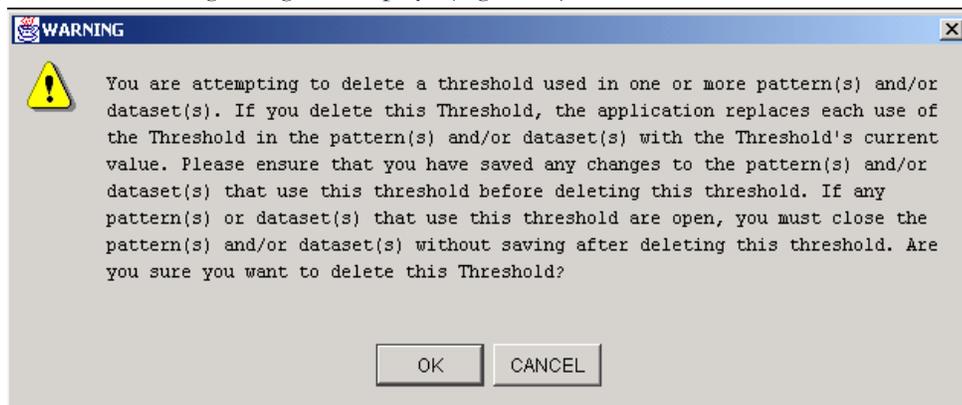
Figure 61. Warning Dialog Box

4. Click **OK** to confirm that you want to remove the threshold from the selected pattern or dataset.

    The pattern or dataset is removed from the list.

5. Click **Save** to confirm the removal of the pattern or dataset.

The Threshold Manager dialog box closes.

If you do not want to remove the selected threshold from the pattern or dataset you selected, click **CANCEL**.

If the pattern or dataset from which you are removing the threshold is currently open in another editor, you must exit the scenario or dataset editor without saving your changes when you return to the editor.

Defining Threshold Sets

Use the following guidelines when defining threshold sets:

- The threshold set name must be unique to any other threshold set that exists in the current scenario.

   Note: Before using the Threshold Manager to define threshold sets for a scenario, you must first create a scenario in a scenario editor. Then you can use the Threshold Manager to define your threshold sets and then use those threshold sets in the Sequence Scenario, Rule Scenario, or DataSet Editors.

To define a threshold set, follow these steps:

1. From the Threshold Set menu of the Thresholds Editor, select **New**.
2. Enter the appropriate name for the new threshold set and click **OK** to confirm.
3. The threshold set displays in the **Threshold Sets** area.

   Optional : Add any comments in the Comment field.
4. Make the desired changes to the threshold in the Threshold Definer tab.
5. Click **Update** to confirm your changes.
6. From the File menu, select **Save** to save your threshold set.

Modifying Threshold Sets

You can change the parameters of an existing threshold set within the Threshold Manager.

To modify a threshold set, follow these steps:

1. Click the Threshold Set**s** tab in the Threshold Manager.
2. Select the threshold set that you want to modify from the **Threshold Set** drop-down menu.

   The details of the threshold set display in the fields in the tab.
3. Make the desired changes.

   Optional: Click the Display Threshold History button to view history of the selected threshold in the Threshold History dialog.

   Optional: Add any comments in the Comment field.
4. Click **Update** to confirm your changes.
5. From the **File** menu, select **Save** to update changes.

Deleting Threshold Sets

You can delete a threshold set from a scenario using the Threshold Manager if the threshold set was not moved to production.

To delete a threshold set from a rule or sequence scenario, follow these steps:

1. Click the **Threshold Sets** tab in the Threshold Manager.

2. Select the threshold set that you want to delete from the Threshold Set drop-down list.

3. From the **Threshold Set** menu, select **Delete**.

4. If the Warning dialog box displays, then delete the Job using this Threshold Set. For deleting Job, refer *About Deleting a Job*, on page 187.

5. Click **OK**.

6. From the File menu, select Save to confirm deletion of the threshold set from the current scenario.

CHAPTER 8 *Augmentation*

This chapter defines the augmentation tool features and functionality that are in the Scenario Manager. This chapter focuses on the following topics:

- About Augmentation
- About Binding Augmentation
- Using Binding Augmentation
- About Match Augmentation
- Using Match Augmentation

About Augmentation

Augmentation is a process that is executed after the completion of a detection job to supplement the information that is stored and displayed with each match.

You can pull data to add to matches in two ways:

- **Binding Augmentation:** Retrieves information from a table not used in the pattern and stores the information with a match. This function is most commonly used to capture information to display in the highlights, such as retrieving a ticker symbol using a security ID number.

- **Match Augmentation:** Adds matched records (in other words, pointers to additional records of interest) that were not written out by the pattern. Doing this creates building blocks on the Alerts Details page of the Web Application based on the dataset key IDs that are stored with a match. Some key IDs are written directly by the pattern as it creates matches. Match Augmentation is used for the following reasons:

  - Since pattern datasets have only one key ID attribute but may join multiple data sources they can provide pointers to the sources that were not specified as the dataset key.

  - Information not used in the pattern logic may be pertinent to show on the Alert Details page within the Web Application and can be provided through the use of Match Augmentation.

Augmentation Editor Terminology

Table 30 lists the terms that define the components of the Augmentation Editors.

Table 30. Augmentation Editor List of Terms

Term	Definition
Augmentation	Process that is executed after the completion of a detection job to supplement the information stored and displayed with each match. Two augmentation mechanisms are available: ● **Binding Augmentation:** Retrieves information from a table not used in the pattern and stores the information with a match ● **Match Augmentation**: Adds to additional matched records that display in the Web Application
Binding	Variable captured in a pattern
Constraint	Condition used to compare an attribute from an augmentation dataset to a variable
Target Binding Name	Variable that stores the result of a binding augmentation instruction

About Binding Augmentation

Binding augmentation enables you to look up a value from the database and store it with a match. This is most often done to retrieve information to display in an alert's highlights.

For example, many detection patterns use security IDs in the financial services industry that are not recognizable to business users. You can use binding augmentation to look up the ticker symbol ID so that it appears in the highlights.

Components of the Binding Augmentation Rule Editor

The Binding Augmentation Editor contains the Binding Augmentation Rule and Associated Rule Properties areas (Figure 62).
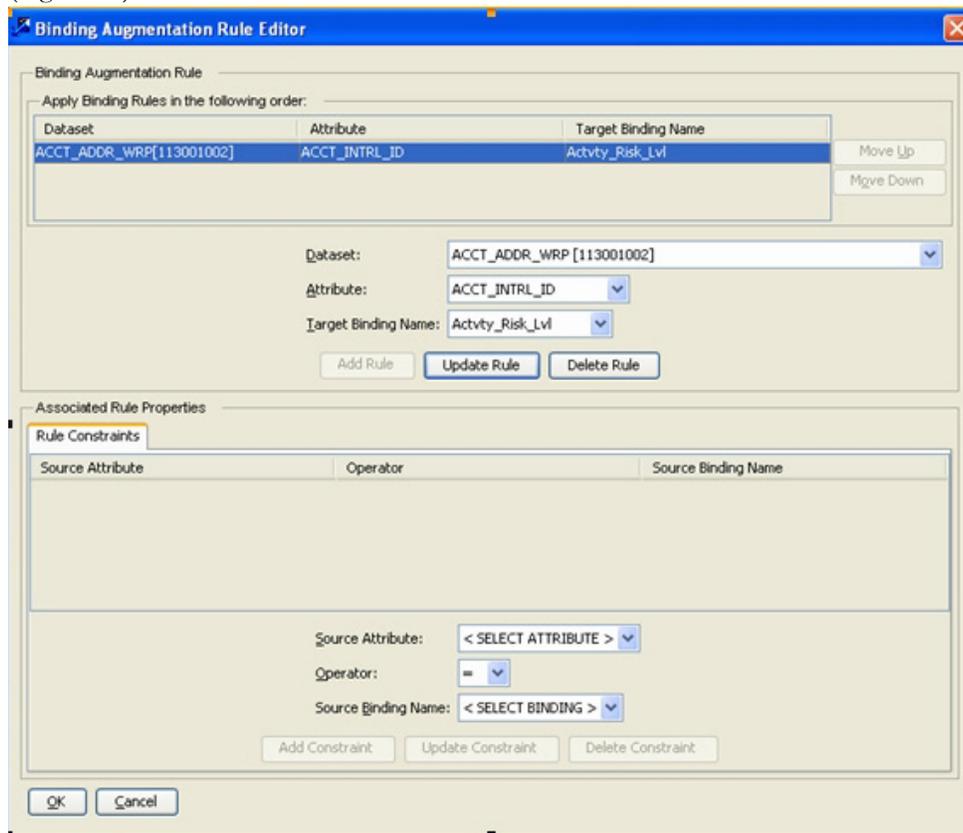
Figure 62. Binding Augmentation Rule Editor

About the Binding Augmentation Rule Area

The Binding Augmentation Rule area displays the properties of the rules you create. Within this area, you can add, update, and delete rules. The Apply Binding Rules in the following order area is blank until you add a rule. This area contains the following lists and buttons:

- **DataSet:** Enables you to select the dataset from which you want to retrieve information in the binding rule.

- **Attribute:** Enables you to identify the attribute you want to retrieve from the dataset in the previous menu.

- **Target Binding Name:** Lists the bound variables saved with each match and enables you to identify the variable that stores the result of this augmentation instruction.

- **Move Up or Move Down:** Moves the selected rule up or down one level.

- **Add Rule:** Adds a new rule with the specified dataset, attribute, and target binding name.

- **Update Rule:** Updates the dataset, attribute, and target binding name of the selected rule based on your changes.

- **Delete Rule:** Deletes a binding rule from the current scenario.

About the Associated Rule Properties Area

The Associated Rule Properties area enables you to define a constraint within a binding rule and to add, update, and delete rule constraints. This area remains inactive until you add a rule. Once you add a rule, you must specify rule constraints for the new rule before you can close the Binding Augmentation Rule Editor.

The Associated Rule Properties contains the following drop-down lists under the Rule Constraints tab:

- **Source Attribute:** Lists the attributes in the dataset from which you are retrieving information and enables you to select the one that is compared to the source binding for the rule constraint.

- **Operator:** Provides the allowable operators that you can use to compare the Source Attribute and Source Binding Name.

- **Source Binding Name:** Lists existing bindings and enables you to select a source binding for the constraint. The source binding is used to build a *where* clause by comparing it to the Source Attribute to identify the dataset record for information retrieval.

The Associated Rule Properties area contains the following buttons:

- **Add Constraint:** Adds a new constraint with the dataset, attribute, and target binding name you specify.

- **Update Constraint:** Updates the dataset, attribute, and target binding name of the selected constraint based on your changes.

- **Delete Constraint:** Deletes a binding constraint from the current scenario.

- **OK:** Saves all changes and closes the Binding Augmentation Rule Editor.

- **Cancel:** Disregards any changes and resets the fields on the Binding Augmentation Rule Editor to reflect previous values.

Using Binding Augmentation

To use binding augmentation, you first need to define your source and target variables with the pattern, and designate them to be saved with the pattern results by adding pattern roles for them.

Caution: You should not create augmentation instructions that involve selecting records from a dataset that has thresholds. You should only do this in cases where the parameter values remain the same across threshold sets.

Applying a Binding Augmentation Rule to a Scenario

To apply a binding augmentation rule to a scenario with a sequence or rule scenario open, follow these steps:

1. From the **Scenario** menu, select **Edit Augmentation Rules** and then again select **Binding Augmentation Rules**.

   The Binding Augmentation Rule Editor dialog box displays (refer to Figure 62).

2. Click the **DataSet** arrow to select the dataset that contains the attribute that you want to retrieve in this augmentation instruction.

3. Click the **Attribute** arrow to select the attribute that you want to retrieve.

4. Click the **Target Binding Name** arrow to select the variable to which you want the selected attribute bound.

   If the Target Binding Name field is not active, you need to add bindings to the scenario pattern before you can select the binding.

5. Click **Add Rule**.

   You now need to apply rule constraints to your binding augmentation instruction.

6. Click the **Source** attribute menu to select the attribute from the augmentation dataset. This is compared to the source binding.

7. Click the **Operator** menu to select the appropriate comparison operator between source Attribute and Source Binding name.

8. Click the source binding name to select the variable that is compared to the source attribute in the rule constraint.

Modifying a Binding Augmentation Instruction

Use the following procedure to modify a bound rule.

To modify a bound rule, follow these steps with a sequence or rule scenario open:

1. Select the rule in the **Apply Binding Rules in the following order** area and adjust the parameters of the selected rule, as desired.

2. Click **Update Rule**.

3. Click **OK** to save your changes.

   The Apply Binding Rules in the following order area reflects the new rule parameters.

Modifying a Binding Augmentation Rule Constraint

You can change the parameters of an existing constraint using the Binding Augmentation Rule Editor.

To Modify a Rule Constraint

To modify a rule constraint, follow these steps with a sequence or rule scenario open:

1. Select the constraint you want to change and select the desired parameters.

2. Click the **Update Constraint** button.

3. The new constraint parameters are reflected in the Rule Constraints area.

4. Click **OK** to save your changes.

Deleting a Binding Augmentation Rule

After adding binding rules to a scenario, you may decide that you no longer want a rule to be associated with that specific pattern. Using the Binding Augmentation Rule Editor, you can delete a binding rule.

If you delete a rule, all constraints that are associated with the rule are also deleted. If you delete a constraint, only that constraint is deleted (refer to *Deleting a Binding Augmentation Rule Constraint* for information about how to delete an individual constraint).

With a sequence or rule scenario open, follow these steps:

1. From the **Scenario** menu, select **Edit Augmentation Rules** and then again select **Binding Augmentation Rules**.

    The Binding Augmentation Rule Editor dialog box shows all rules and constraints that are defined in the selected scenario (Figure 62).

2. Select the rule you want to delete in the **Apply Binding Rules in the following order** list box.

3. Click **Delete Rule**.

    The rule is removed from the Apply Binding Rules in the following order list box.

    Note: If the rule is associated with other binding augmentation rules within a scenario, the system prompts you to confirm the deletion. Click **OK** to confirm this deletion, or **Cancel** to retain the selected rule.

4. Click **OK** to save your changes.

Deleting a Binding Augmentation Rule Constraint

You can delete a constraint without deleting the rule to which it applies using the Binding Augmentation Rule Editor.

To Delete a Rule Constraint

With a sequence or rule scenario open, follow these steps:

1. From the **Scenario** menu, select **Edit Augmentation Rules** and then again select **Binding Augmentation Rules**..

   The Binding Augmentation Rule Editor dialog box displays showing all constraints defined in the selected scenario (Figure 62).

2. Select the constraint you want to delete in the **Rule Constraints** tab.

3. Click **Delete Constraint**.

   The constraint is removed from the list.

4. Click **OK** to save your changes.

   The Constraint is removed from Rule constraint list box.

Changing the Order of Binding Augmentation Rules

You can change the order in which you want the rules within a scenario to be processed. You determine the order that the rules are applied by placing the rules listed in the Apply Binding Rules in the following order list box in the desired order using the Move Up and Move Down buttons.

With a sequence or rule scenario open, follow these steps:

1. From the **Scenario** menu, select **Edit Augmentation Rules,** and then again select **Binding Augmentation Rules**.

   The Binding Augmentation Rule Editor dialog box shows all rules that are defined in the selected scenario (Figure 62).

2. Select the rule you want to reorder and click the desired **Move Up** or **Move Down** buttons.

3. Repeat Step 2 for each rule you want to move.

4. Click **OK** to save your changes.

   The Apply Binding Rules in the following order list box displays all rules in the specified order.

About Match Augmentation

As part of each scenario match, patterns create a set of *matched records* that identify the business data involved in detecting the behavior of interest. The Web UI presents the matched records as building blocks on the Alert Details screen. Match augmentation creates references to additional data records that are pertinent to a match, so that the UI shows additional building blocks.

Match augmentation instructions can function in two ways:

- **Match Augmentation from Rule Constraints:** Adds matched records by looking up business data using bound variables solved with the match.

- **Match Augmentation from Matched Data Binding:** Adds matched records by looking up business data using the matched records that the pattern originally captured.

A Match Augmentation instruction can have any of the following combinations:

- One constraint

- One match data binding

- One or more constraints and one match data binding

Parts of the Match Augmentation Rules

The **Match Augmentation Editor** contains the following components (Figure 63):

- Match Augmentation Rule Area

- Associated Rule Properties Area

- Saving or Discarding Changes

Match Augmentation Rule Area

The Match Augmentation Rule area displays the dataset from which you add records, and the key attribute for that dataset. The Match Augmentation Rule area is blank until you add a rule (Figure 63). This area contains the following lists, text boxes, and buttons.

- **DataSet:** Enables you to add a dataset to the current rule.

- **Attribute:** Shows the key attribute for the selected dataset.

- **Add Rule:** Adds a new rule with the specified dataset and attribute.

- **Delete Rule:** Deletes a binding rule from the current scenario.
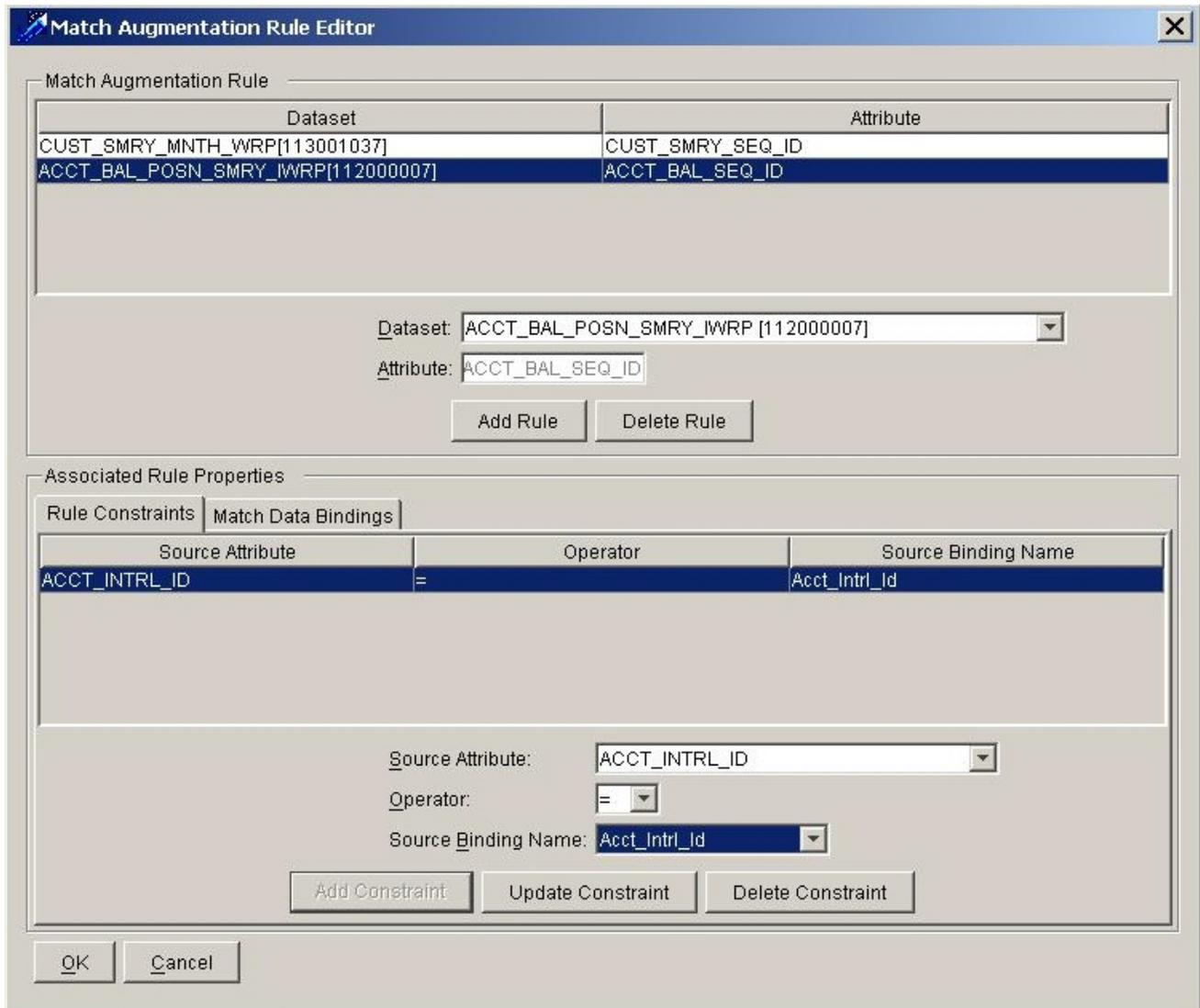
Figure 63. Match Augmentation Rule Editor Dialog Box

Associated Rule Properties Area

The Associated Rule Properties area captures the logic used to retrieve information from the dataset identified in the Match Augmentation Rule area. This area contains the following tabs:

- **Rule Constraints:** Displays a list of all constraints created for the selected augmentation instruction, and provides access to the match augmentation for bindings functions that enable you to add, update, or delete rule constraints. You must set at least one constraint per match data binding for each rule that you create.

- **Match Data Bindings:** Provides access to the match augmentation for binding functions that enable you to add, update, and delete match data bindings to an augmentation instruction.

Rule Constraints Tab

The Rule Constraints tab provides access to the rule constraint functions that enable you to add, update or delete rule constraints for a match augmentation rule (refer to Figure 63 on page 153).

You define parameters in the following drop-down lists for a constraint:

● **Source Attribute:** Enables you to select an attribute from the dataset (selected in the Match Augmentation Rule area) to use in the rule constraint.

● **Operator:** Provides the allowable operators that you can use to compare the source attribute to the source binding name.

● **Source Binding Name:** Enables you to select the binding that is compared to the source attribute in this constraint.

In the Associated Rule Properties area, you can add, update, or delete a rule constraint by using the following buttons:

● **Add Constraint:** Adds a new constraint with the specified source attribute and source binding name.

● **Update Constraint:** Updates the source attribute and source binding name of the selected constraint based on your changes.

● **Delete Constraint:** Deletes a match constraint from the current augmentation instruction.

Match Data Binding Tab

The Match Data Binding tab provides access to the Match Augmentation for Matches rule functions that allow you to add, update or delete match data bindings within a rule (Figure 64).
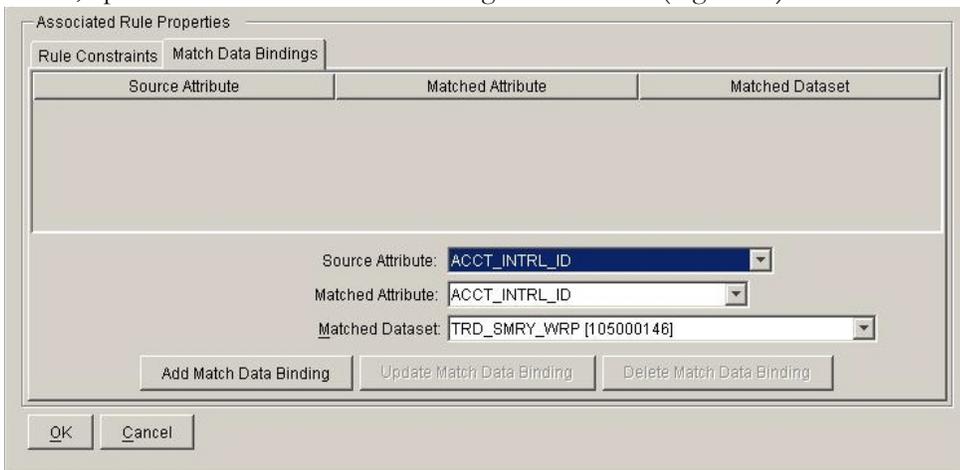
Figure 64. Match Augmentation for Match Data Bindings Tab

Use the parameters in the following drop-down lists to create a match data binding:

- **Source Attribute:** Specifies the attribute from the dataset in the Match Augmentation Rule that links the record to the Matched Dataset.

- **Matched Attribute:** Specifies the attribute from the Matched Dataset that links the record to the dataset in the Match Augmentation Rule.

- **Matched Dataset:** Specifies the wrapper dataset containing the dataset key that the pattern matches.

The Match Data Bindings tab contains the following buttons:

- **Add Match Data Binding:** Adds a new binding with the attributes and dataset you specify.

- **Update Match Data Binding:** Updates a new binding with the attributes and dataset you specify, based on your changes.

- **Delete Match Data Binding:** Deletes the selected binding from the current scenario.

Refer to section *Creating a Match Augmentation with Match Data Binding*, on page 159, for additional information about defining match data bindings within a rule.

Saving or Discarding Changes

In the Match Augmentation Editor, use the following buttons to save or discard your changes:

- **OK:** Saves all changes and closes the Match Augmentation Rule Editor.

- **Cancel:** Discards any changes and resets the fields on the Match Augmentation Rule Editor to reflect previous values.

Using Match Augmentation

To add a match augmentation instruction follow these steps:

1. Specify an augmentation dataset from which you select records. Refer to section *Specifying an Augmentation Dataset*, on page 156, for instructions.

2. Create the associated rule properties, which enables you to define the logic used to retrieve information from the target dataset.

After completing the procedures for these steps, you can modify the defined rules and constraints. Using the Match Augmentation Editor, you can make changes by:

- *Specifying an Augmentation Dataset*. Refer to section *Specifying an Augmentation Dataset*, on page 156, for instructions.

- *Specifying Associated Rule Properties*. Refer to section *Specifying Associated Rule Properties*, on page 157, for instructions.

Specifying an Augmentation Dataset

When specifying an augmentation dataset, you can:

- Create a new dataset. Refer to *Creating an Augmentation Dataset*, on page 156., for more information.

- Modify an existing dataset. Refer to *Modifying an Augmentation Dataset*, on page 157., for more information.

The following sections describe these procedures:

Note: If a dataset used in augmentation contain thresholds, views used in augmentation will use the values of the thresholds from the scenario's base threshold set. The values of the thresholds in the base set may be different from the values used in the detection job.

Creating an Augmentation Dataset

To create an augmentation dataset, follow these steps with a sequence or rule scenario open:

1. Click the **DataSet** arrow to select the dataset from which you want to select records for this augmentation instruction. The Attribute text box displays the key attribute for this dataset.

2. Click **Add Rule**.

    The rule displays in the Match Augmentation Rule area. The Associated Rule Properties area becomes active.

Refer to section *Specifying Associated Rule Properties*, on page 157, for additional information about creating rule constraints.

Modifying an Augmentation Dataset

Follow these steps to modify an augmentation dataset, with a sequence or rule scenario open:

1. Click the **DataSet**, **Attribute**, and **Target Binding Name** arrows to change these parameters, as desired.

2. Click **Update Rule**.

   The modified rule displays in the Match Augmentation Rule list box.

3. Click **OK** to confirm any changes.

Specifying Associated Rule Properties

Associated rule properties provide the logic for selecting records from the dataset specified in the Match Augmentation Rule area.

When specifying associated rule properties, you can:

- Create constraints for a match rule. Refer to *Creating Constraints for a Match Augmentation Rule*, on page 157., for more information.

- Modify the constraints of a Match Rule. Refer to *Modifying an Augmentation Dataset*, on page 157., for more information.

- Delete a rule constraint. Refer to *Deleting a Rule Constraint*, on page 158., for more information.

- Delete a match augmentation instruction. Refer to *Deleting a Match Augmentation Instruction*, on page 158., for more information.

Creating Constraints for a Match Augmentation Rule

To create constraints for a match augmentation rule, follow these steps in the Associated Role Properties area:

1. From the Scenario menu, select Edit Augmentation Rules, and then again select Match Augmentation Rules.

   The Match Augmentation Rule Editor dialog box displays.

2. Select the **Match Augmentation Rule** for which constraint is being created. For new, refer section *Creating an Augmentation Dataset*, on page 156.

3. Click the **Source Attribute** arrow to select the desired name.

4. Click the **Operator** arrow to select the operator that you want to use to compare the source attribute and source binding name.

5. Click the **Source Binding Name** arrow to select the desired name.

6. Click **Add Constraint**.

7. Verify that the selected constraints are accurate by reviewing the contents of the **Rule Constraints** field.

8. Click **OK** to save your changes.

   Note: You must add a minimum of one constraint or match data binding for each rule you create.

Modifying the Constraints of a Match Rule

To make changes to constraint parameters set within the augmentation rule, follow these steps with a sequence or rule scenario open:

1. From the **Scenario** menu, select **Edit Augmentation Rules**, and then again select **Match Augmentation Rules**.

   The Match Augmentation Rule Editor dialog box displays.

2. Click the Rule Constraints tab.

3. Select the constraint you want to change.

4. Click the **Source Attribute** and **Source Binding Name** arrows to change these parameters, as desired.

5. Click **Update Constraint**.

   The updated constraint is listed in the Associated Rule Properties Rule Constraint tab list box showing the updated information.

6. Click **OK** to save your changes.

Deleting a Rule Constraint

To delete a constraint without deleting the rule to which it applies, follow these steps with a sequence or rule scenario open:

1. From the **Scenario** menu, select **Edit Augmentation Rules** and then again select **Match Augmentation Rules**.

   The Match Augmentation Rule Editor dialog box shows all constraints defined in the selected scenario.

2. Select the constraint you want to delete in the **Rule Constraints** tab.

3. Click **Delete Constraint**.

   The constraint disappears from the list.

4. Click **OK** to save your changes.

Deleting a Match Augmentation Instruction

To delete a match augmentation rule from a scenario pattern, follow these steps with a sequence or rule scenario open:

1. From the **Scenario** menu, select **Edit Augmentation Rules** and then again select **Match Augmentation Rules**.

   The Match Augmentation Rule Editor dialog box shows all rules defined in the selected scenario (Figure 63).

2. Select the rule you want to delete and click **Delete Rule**.

   The deleted rule is removed from the Match Augmentation Rule list box.

3. Click **OK** to save your changes.

Creating a Match Augmentation with Match Data Binding

To create a match augmentation with match data binding, follow these steps with a sequence or rule scenario open.

Note: Make sure you define these parameters in the order shown; if this is done out of order, you cannot define the required fields.

1. From the **Scenario** menu, select **Edit Augmentation Rules** and then again select **Match Augmentation Rules**.

    The Match Augmentation Rule Editor dialog box displays.

2. Select the rule to which you want to add match data binding.

3. Click the Match Data Bindings tab.

    The Associated Rule Properties area displays all attributes for defining match data bindings for a rule.

4. Click the **Source Attribute** arrow to select the attribute that links the dataset specified in the **Match Augmentation Rule** area to the existing matched records.

5. Click the **Matched Dataset** arrow to select the wrapper dataset that contains the key attribute that the pattern matches.

6. Click the **Matched Attribute** arrow to select the attribute from the matched dataset that links the records to the dataset specified in the Match Augmentation Rule area.

7. Click **OK** to save your changes.

Modifying Match Data Bindings Within a Rule

To modify a match data binding that you define within a rule, follow these steps with a sequence or rule scenario open:

1. From the **Scenario** menu, select **Edit Augmentation Rules** and then again select **Match Augmentation Rules**.

    The Match Augmentation Rule Editor dialog box displays.

2. Select the rule to which you want to change the attributes of the match data bindings.

3. Click the Match Data Bindings tab.

    The Associated Rule Properties area displays all attributes for defining match data bindings for a rule.

4. Select the match data binding rule you want to change.

5. Click the arrow in the fields to change the current values until all changes are complete.

6. Click **Update Match Data Binding**.

7. Click **OK** to save your changes.

To Delete Match Data Bindings Within a Rule

To delete a match data binding from any rule, follow these steps with a sequence or rule scenario open:

1. From the **Scenario** menu, select **Edit Augmentation Rules** and then again select **Match Augmentation Rules**.

   The Match Augmentation Rule Editor dialog box displays.

2. Select the rule from which you want to delete match data bindings.

3. Click the Match Data Bindings tab.

   The Associated Rule Properties area displays all attributes for defining match data bindings for a rule.

4. Select the match data binding rule you want to delete.

5. Click **Delete Match Data Binding**.

   The match data binding is deleted and removed from the Associated Rule Properties list box.

6. Click **OK** to save your changes.

CHAPTER 9

Job Editors

This chapter defines the Job Editors' features and functionality within the Scenario Manager. This chapter focuses on the following topics:

- About the Job Editors
- Common Job Editor Parts
- About Jobs with Salient Entities
- Using the Editors to Create a Job
- Performing Additional Job Tasks

About the Job Editors

The job editors within the Scenario Manager are tools that enables you to create instruction parameters that are then executed through a scheduling tool.

Each job set up in Sequence Matcher and Rule Matcher uses one or more scenarios and sequence-based or rule-based algorithms to search for patterns. The patterns available to choose for a job were created using the editors of the Scenario Manager.

The Scenario Manager contains the following job editors:

- **Rule Matcher:** Create a job for scenarios created within the Rule Scenario Editor.
- **Sequence Matcher:** Create a job for scenarios created within the Sequence Scenario Editor.
- **Link Analysis:** Create a job for network definitions created using the Network Definition Editor.

Job Editor Terminology

Table 31 provides a list of terms used to define the concepts of the job editors.

Table 31. Job Editors List of Terms

Term	Definition
dispatcher	Daemon process that polls the job metadata for new jobs to be executed.
job	Set of input parameters you identify within a job editor for a Oracle Financial Services algorithm, such as datasets, patterns, and date ranges.
job editors	Tools within the Scenario Manager used to create jobs.

Table 31. Job Editors List of Terms (continued)

Term	Definition
job status code	Three-letter abbreviation for the status of a job: ● **ERR:** The system places a job in this status if the job's aborted during the run due to error. ● **FIN:** The system places the job in this status when the job runs successfully without errors. ● **IGN:** You, the user, places a job in this status to tell the dispatcher to ignore and not process the job. This type of job contains a *clean copy* of the setup parameters for a job and is never intended to be executed. Also used for job templates. ● **NEW:** You, the user, place a job in this status to indicate to the dispatcher that a new job is available for processing. When creating a new job, this is the default status. ● **RES:** You, the user, place a job in this status after you have corrected a failure due to errors. This RES indicates to the dispatcher that the job should be restarted. ● **RUN:** The system places the job in this status while the job is running.
run	Audit trail the system creates when the Oracle Financial Services algorithm processes, such as the start and end times, the status, or the associated job. A *run* is created when the dispatcher processes a job. A job can have multiple runs. Refer *looping*.
salient entity	Dataset attribute that is set up in the job editor to perform a special function during a job run, such as directing the job to run in multiple iterations.

Common Job Editor Parts

Across all the job editors are several common areas, common menu options, and tab pages. Common components of the job editors include:

- Job Editor Menus

- Status Bar

- Job Area

- Job Options Tab

- DataSets Tab

An example of the Rule Matcher Job Editor window is shown in Figure 65.

Figure 65. Rule Matcher Job Editor Window

Job Editor Menus

All job editors have the following common menus: Jobs, Windows, and Help. The menus contain available commands, functions, and options. The following sections describe the job editor menus.

Job Menu

The **Job** menu contains the following options:

- **New:** Entering the Job Editor essentially selects New in the Job menu (in other words, it is unnecessary to click this menu option). However, you can select this option while you are working in the Job Editor and want to create a new job. If you have unsaved changes, the system displays a message that instructs you to do one of the following:

    - **Save:** Saves the job you are creating or modifying and then clears the columns in the Job Options tab to the default values so that you can begin work on a new job.

    - **OK:** Does not save the data and clears the columns in the Job Options tab to default values.

    - **Cancel:** Returns you to the job without saving your data or creating a new job.

- **Save:** Saves a new job or an existing job that you have modified.

- **Copy:** Copies an existing job.

- **Delete:** Deletes a job. You can delete only jobs that you own (created).

- **Load All Jobs:** Displays jobs of all statuses within the domain.

- **Load Unfinished Jobs:** Displays all jobs other than the status of FIN (finished) within the domain when you access a job editor. This is the default.

- **Close:** Closes the job editor you are in and returns you to the DataSet Editor.

- **Exit:** Exits you from the Scenario Manager.

Windows Menu

Refer to Chapter 2, *Common Features*, on page 3., for descriptions of the options in the **Windows** menu.

Help Menu

Refer to Chapter 2, *Common Features*, on page 3., for descriptions of the options in the **Help** menu.

Status Bar

Each job editor provides instructional text on the Status Bar, located at the area of the bottom of a job editor window. The instructional text provides information related to messages, options to select, and what steps to take.

Job Area

The Jobs area lists all jobs within the domain that do not have the status of *FIN* by default. The Jobs area displays in the left side of most job editors (Figure 65). To populate this area, use the Load All Jobs and Load Unfinished Job options in the Job menu.

In the Jobs area, you can select jobs from this list for modifying, copying, or deleting. This area includes the following resizable columns:

- **ID:** Contains the ID of the job

- **Description:** Contains a description of the job

- **Creator:** Contains the user ID of the user that created the job

- **Run Start:** Specifies the date the job was last executed

● **Status:** Displays the status of the job

Job Options Tab

Each job editor contains a Job Options tab (Figure 65). Use the Job Options tab for the following:

● Enter the job name (up to 80 alphanumeric characters) in the Description field.

● Assign a status code to the job by selecting a value from the Status drop-down list:

■ **NEW:** Run the job immediately.

■ **IGN:** Retain the job. Use this status when setting up a job as a template, copying, and modifying it for subsequent runs.

■ **RES:** The job was previously run but failed due to errors. Before setting the job to this status, you should correct the errors.

Note: You cannot change the Status to ERR, RUN, or FIN.

Other read-only fields on the Job Options tab include the following:

● **ID:** Unique number the system assigns to a new job

● **Creator:** Login ID of the user who creates a job

● **Start Time:** Start time and date when the system last executed the job

Completing the Job Options tab is usually the first step in creating any type of job.

DataSets Tab

Each job editor, except for the Link Analysis Job Editor, contains a DataSets tab (Figure 65). Use the DataSets tab to identify the datasets to use in the job:

● The datasets that the job contains are usually the same as the datasets in the pattern; however, the datasets in the job override the pattern datasets.

● You can substitute different datasets from those identified in the pattern, but this only works if the alternate datasets have the same attributes as the datasets used in the pattern.

Note: Substituting alternate datasets does not automatically update the Matched Data Bindings field in match augmentation constraints.

● List the datasets in the same order as in the pattern; you can update the list order by clicking and dragging a specific dataset.

Use the Salient Entities function on the DataSets tab to set up *looping jobs* or *multiple runs* (in other words, set up separate, independent runs based on the values in one of the dataset fields). We recommend that you follow these guidelines before using this function:

● Use the **Looping Entity** option to set up separate runs based on the values in one of the dataset fields.

Note: This method is not recommended as most situation can be set up efficiently by ordering datasets.

● Use the **Multiple Run** option to specify values in the dataset fields to set up independent runs based on user-defined ranges specified in the dataset fields.

● Use the Date Control (KDD_CAL) table in the applicable datasets in place of the Start Date and End Date fields.

DataSets Area on the DataSets Tab

To select a dataset for the job, click the **Select** button. The Select DataSet dialog box displays. Refer to Chapter 2, *Common Features*, on page 3., for instructions on using this dialog box.

Salient Entities Area on the DataSets Tab

Refer to section *About Jobs with Salient Entities*, on page 166, for information about this section of the DataSets tab.

About Jobs with Salient Entities

To create a job with salient entities, you must access the Salient Entities Area on the DataSets tab. The Salient Entities Area contains the following fields:

- **Entity Name:** Enter the name of the salient entity you want to use in the job. When you use a date attribute as the salient entity, you must enter the name of the attribute as `Date`. This format, which is not case sensitive, enables the system to recognize the attribute as a date.

- **Attribute:** Click the **Select** button next to the Attribute field (which is non-editable) after entering the entity name. This displays the Attribute Selector dialog box that list attributes that correspond to the selected dataset.

- **Looping Entity:** Enables you to use this function for your job. When using a date salient entity, do not use the Looping Entity option.

- **Multiple Run:** Enables you to use this function for your job by defining the ranges specified in the dataset fields.

- **Job Thread Count:** Is enabled when you click the **Looping Entity** or **Multiple Run** check box. Enter the maximum number of threads per job. We recommend setting this to twice the number of processors or cores available on the system. This field is available for both Sequence Matcher and Rule Matcher jobs.

- **Add Entity:** Enables when you enter an entity name and select an attribute. Click this button to add the entity to the Salient Entities area.

- **Update Entity:** Modifies or changes an existing salient entity listed in the Salient Entities areas. Example modifications include changing the name of the salient entity or changing the attribute associated to it. Click the salient entity you want to modify, make your changes, and then click **Update Entity**. The system replaces the salient entity with the newly entered information.

- **Delete Entity:** Deletes a salient entity from the Salient Entity list. Click the salient entity you want to delete and then click **Delete Entity**.

- **Start Date** and **End Date**: Enter information within the these text boxes that corresponds to the date salient entity. These dates limit the examination of the data to that which is within the range specified. If you did not identify a salient entity, leave these fields blank.

  Use one of the following formats when entering a date: M/DD/YYYY, MM/DD/YY, DD-MONTH-YY, or DD-MONTH-YYYY. Regardless of your input, the system displays the date in MM/DD/YYYY format after you press **Enter**. If you type only two digits for the year, the system assumes the 21st century and automatically inserts two digits before the two entered digits.

Using Multiple Salient Entities

You can create a job with one dataset that has a date salient entity and an additional salient entity. These limit the target data to a specific date range and examination based on the values of the additional salient entity. For example, a security is the second salient entity. The results of the job are based on the date range and the various securities, in other words, the results for Security A, the results for Security B, the results for Security C, and so on. When executing the job, the system examines the data for matches to the pattern in loops over the second salient entity values.

The salient entities are used per dataset in the order that the Salient Entities area includes. You can adjust that order for run efficiency. If the selected dataset is based on a table, you may need to adjust the order of salient entities (by dragging and dropping until desired order is established) to ensure that the appropriate table index is used. If you make no adjustments, the system uses the salient entities in the order of their selection for the setup.

Using Multiple Runs

The multiple run feature helps to improve performance of the job, and takes advantage of multiple processors. The multiple run feature applies to both Sequence Patterns and Rule Patterns. Each run is executed in a separate thread.
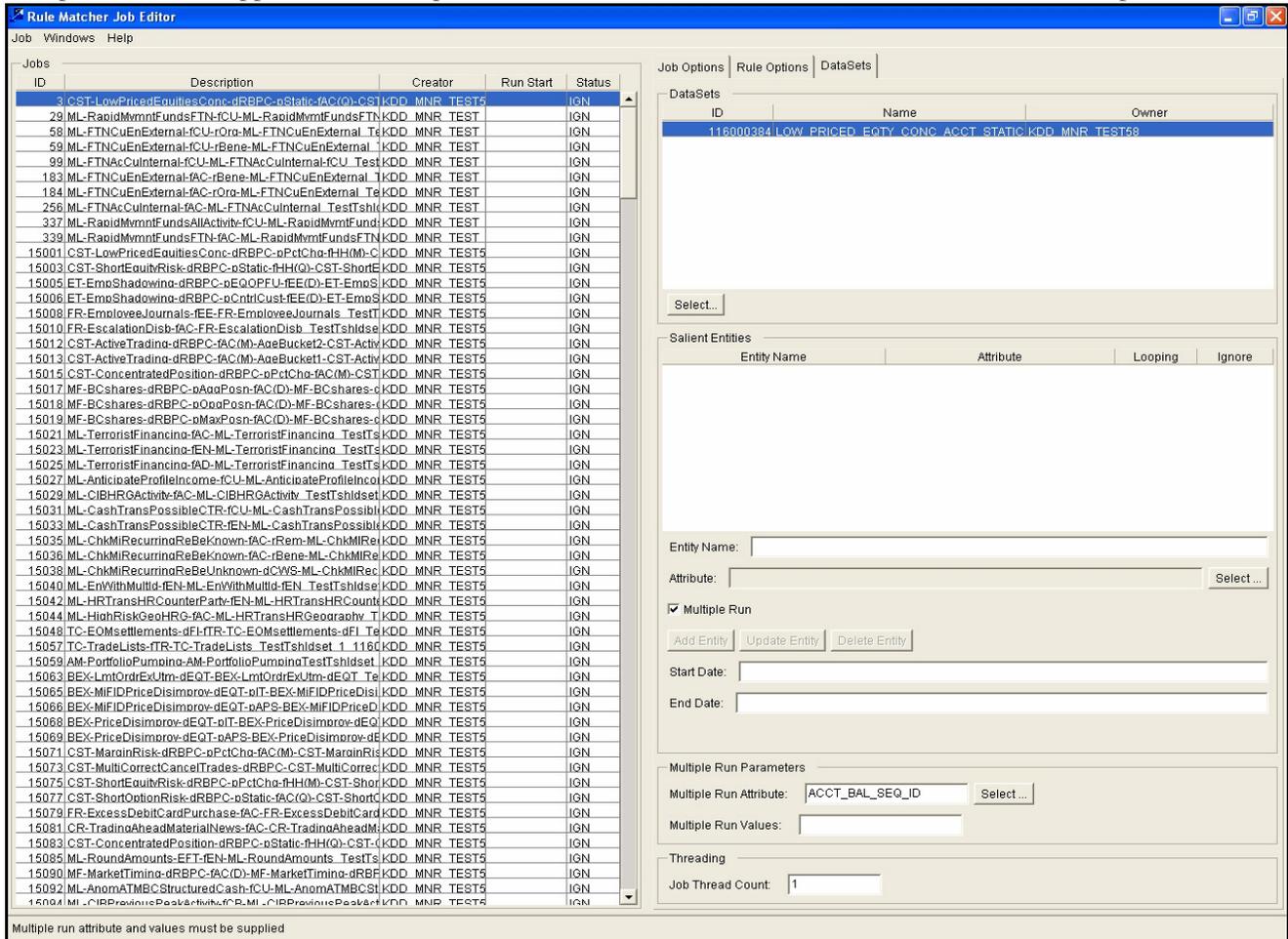
Figure 66. Rule Matcher Job Editor for Multiple Run

To create a multiple run job, implement the following steps:

1. On the Datasets tab, select the first dataset in the Datasets list.

   The **Multiple Run** check box enables.

2. Select the **Multiple Run** check box.

   The Multiple Run Attribute field enables.

3. Select the attribute of the dataset that is used to separate the data.

   Typically, this is the attribute that represents the focus of the pattern. The **Multiple Run Values** and **Job Thread Count** field enables.

4. Enter value separators in a comma delimited list in the **Multiple Run Values**.

   For example, for a Security-focused pattern, the data miner determines that the securities can be divided into four relatively equal buckets by dividing them alphabetically. The different buckets are A-G, H-M, N-S, and T-Z. The **Multiple Run Values** should be populated with G,M,S. The first thread will handle securities where the Security Internal ID is less than or equal to G. The second thread handles securities where the Security Internal ID is greater than G and less than or equal to M. The third thread will handle securities where the Security Internal ID is greater than M and less than or equal to S. The fourth thread will handle securities where the Security Internal ID is greater than S.

5. In the **Job Thread Count** field, enter the desired number of threads for this job.

   This value should be less than or equal to the number of buckets specified in the Multiple Run Values field. In the previous example, four threads would be the maximum value. This value should take into account the number of buckets specified in the Multiple Run Values field, and the number of processors available on the detection machine.

Note: Repeat the steps for each dataset in the job. The same set of values should be used for the **Multiple Run Attribute**, **Multiple Run Values**, and **Job Thread Count** fields for each dataset.

Using the Editors to Create a Job

This sections provides instructions for creating the following types of jobs:

- Creating a Sequence Matcher Job

- Creating a Rule Matcher Job

- Creating a Link Analysis Job

These sections cover the tab pages that are not common among all the job editors, and provide the step-by-step instructions for creating the job.

Creating a Sequence Matcher Job

The Sequence Matcher Job Editor includes the following tabs:

- Job Options Tab

- Seq Matcher Options Tab

- DataSets Tab

Seq Matcher Options Tab

Use the Seq Matcher Options tab to select the sequence pattern for data matching, to indicate the search direction in the data, and to designate inter-dataset sorting of the data (Figure 67).
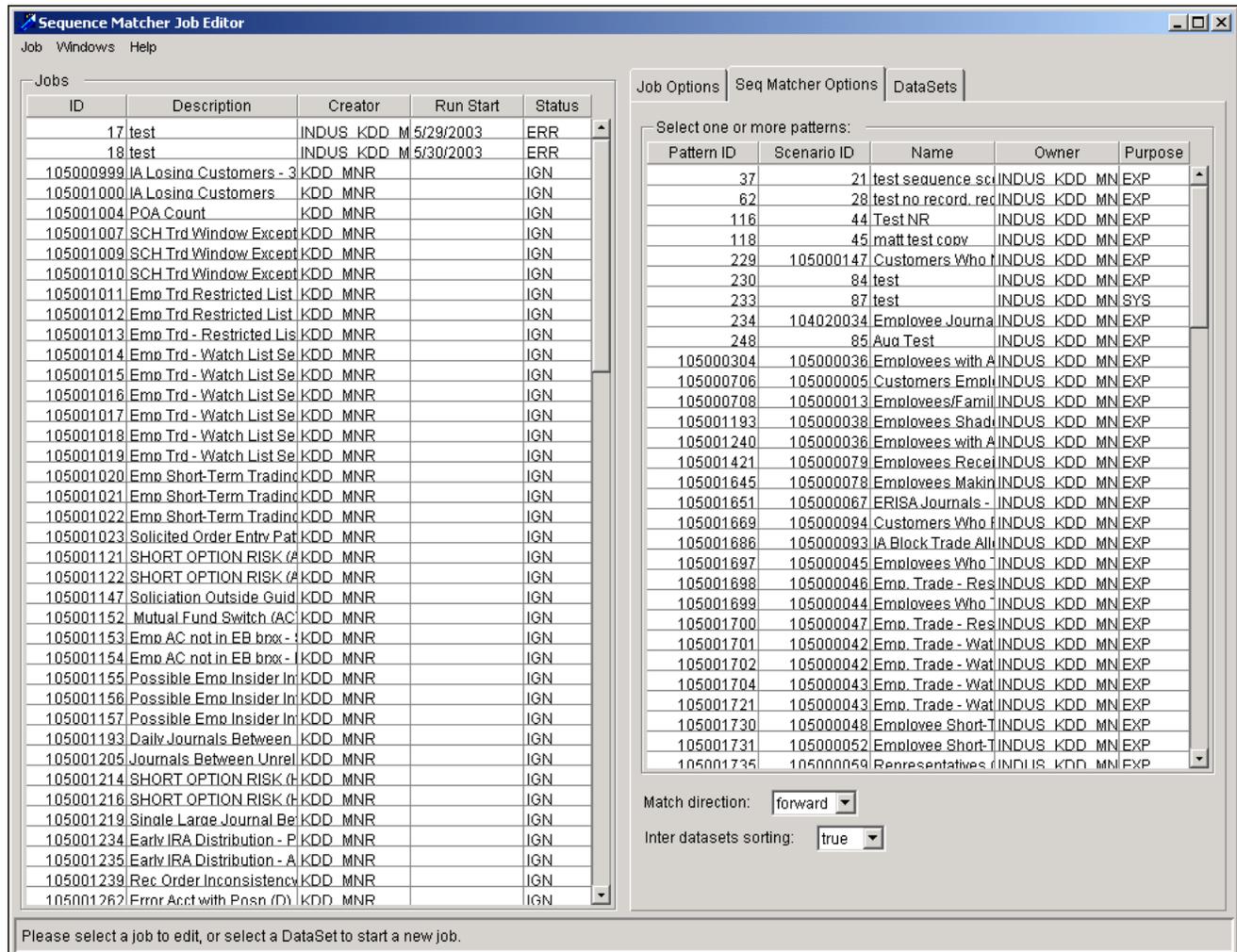
Figure 67. Job Editor Sequence Matcher Options Tab

The Seq Matcher Options tab includes the following fields:

- **Select one or more patterns** area: Lists available patterns within the tab. These patterns are established in the Sequence Matcher Editor; you cannot modify the content of a pattern within the job editor. The selected pattern must have the same patterns as the attributes of the dataset you selected on the DataSets table. If it does not, an error occurs when the job is run. If you are using patterns with thresholds, select only patterns within the same scenario.

  When you select a pattern, the system automatically populates the DataSets tab with the associated datasets of the selected pattern.

- **Match direction**: Enables you to select the sort order of rows for the selected dataset. The system default is a Forward search option; however, sequences for some patterns were written to expect rows to appear sorted in reverse order. If you select such a pattern, you need to select the Reverse option.

- **Inter datasets sorting**: Controls whether datasets are intermingled (that is, threaded together) based on the sorting attributes.

- **Select Threshold Sets:** Contains value appropriate to data to be used.

Creating a Sequence Matcher Job

To create a sequence matcher job, follow these steps:

1. Click **Windows** in the **File** menu, select the **Jobs** option, and then click **Sequence Matcher**.

2. From the **Job** menu, select **New**.

3. Click the Job Options tab and enter information in the following fields:

   - Description

   - Status

4. Click the Seq Matcher Options tab and select one or more patterns and any appropriate option.

5. Click the DataSets tab and follow these steps:

   - Click **Select** to select a dataset. Refer to Chapter 2, *Common Features*, on page 3, for instructions on using the Select a Dataset dialog box.

     Optional: Configure the job for a Salient Entity.

6. From the Job menu, select **Save**.

   The system saves your job. If you selected the New option as the status, the job is made available to the dispatcher to execute immediately.

Creating a Rule Matcher Job

The Rule Matcher Job Editor includes the following tabs:

- Job Options Tab

- Rule Options Tab

- DataSets Tab

Rule Options Tab

Use the Rule Options tab to select the rule pattern for detection matching (Figure 68).

Figure 68. Rule Matcher Tab

The Select a pattern area of the tab contains the list of active rule patterns associated with the current domain established in the Rule Matcher Editor. You cannot modify the content of a pattern within the job editor.

Creating a Rule Matcher Job

To create a rule matcher job, follow these steps:

1. Click **Windows** in the **File** menu, select the **Jobs** option, and then click **Rule Matcher**.

2. From the **Job** menu, select **New**.

3. Click the Job Options tab and complete the **Description** and **Status** fields.

4. Click the Rule Options tab and select a pattern and threshold set.

5. Click the DataSets tab and follow these steps:

   ● Click **Select** to select a dataset. Refer to Chapter 2, *Common Features*, on page 3, for instructions on using the Select a Dataset dialog box.

   Optional: Configure the job for a Salient Entity.

6. From the **Job** menu, select **Save**.

   The system saves your job. If you select a status of New, the job becomes available to the dispatcher to execute immediately.

Creating a Link Analysis Job

Refer to Chapter 5, *Link Analysis*, on page 67, for information about creating a Link Analysis Job.

Performing Additional Job Tasks

The following sections provide instructions on the following tasks:

- Modifying a Job

- Copying a Job

- Deleting a Job

- Creating a Job Template

- Executing a Job

Modifying a Job

You can modify a template or a job that the system has not yet run. You cannot modify a job that has been run, but rather you must make a copy of it and modify the copy. Refer to *About Copying a Job*, on page 186, for more information.

Modifications you can make to a job include, but are not limited to, the following:

- Adding or deleting patterns

- Adding or deleting datasets

- Adding or changing date ranges

When you make modifications to a job and save it, all changes are saved to the job. The information on the Job Options tab is *not* updated. For example, the ID of the job is not updated with a new ID and the Creator does not change to reflect your logon ID.

To modify a job, refer section *Using the Editors to Create a Job*, on page 169.

.

Copying a Job

After configuring a job to your specifications, you may want to use the job to create other jobs. To reuse a template or an existing job that has been run, you must make a copy of the job first and then any modifications, if applicable. This is equivalent to a *Save As* function, in which everything is the same except for the name.

When you make a copy of the job, all information in the tab pages of the copied job are transferred to the new job, with the exception of the Job Options tab:

- The ID is null, in other words, it did not copy the ID from the copied job.

- The Creator now contains your logon ID.

- The Start Time is null, because the job never ran so it does not have a start time.

- The Status drop-down list value is NEW, which can be different than the job from which you copied information.

To copy a job, follow these steps:

1. Click a job from the Jobs area of the job editor to copy.

2. From the **Job** menu, select **Copy**.

   The system copies the job, displays it in the Jobs area, and populates the job tab pages accordingly.

   Optional: Make appropriate changes to the job, such as a new Description (name) or new status code (for example, NEW) on the Job Options tab.

3. From the **Job** menu, select **Save**.

Deleting a Job

To delete a job, verify that:

- You are listed as the creator of the job.

- The job has the status of NEW or IGN.

If not, the system disables the Delete option from the Job menu.

To delete a job, follow these steps:

1. Access the appropriate job editor.

2. Select a job from the Jobs area of the job editor.

3. From the **Job** menu, select **Delete**.

4. Click **OK** to confirm the deletion.

   The system removes the job from the system and the Jobs area.

Creating a Job Template

If you set up a new job with a status code of IGN rather than NEW, the dispatcher ignores the job and the system never runs the job. You should follow this process when a job is not intended to be run but instead is used as a job template (that is, copying the job description many times, changing some aspect of the job slightly in each copy, and then setting each copy to NEW to run automatically). This copy procedure is possible because with each copy of the job with an IGN status code, the system assigns a new, unique ID to it and treats it like a new job.

Using the IGN status code as a way to copy a job eliminates the need to recreate the entire job each time the you want to run a similar job or the same job but for a different date range or entity.

Executing a Job

After creating a job with the status of NEW using one of the job editor tools, the `dispatcher` script starts the job for processing. The `dispatcher` script is a daemon that periodically checks the meta data tables for new jobs. This script resides on the Database Server.

Refer to *Administration Guide* for information about the `dispatcher` script functions.

APPENDIX A
Scenario Design Guidelines

This appendix describes technical guidelines for the design of scenarios and their components. Specifically, this appendix covers the following topics:

- About Technical Guidelines
- About Improving Pattern Performance
- About Constructing Datasets
- Examples of Common Business Problems
- About Scenario Constructions to Avoid
- About Temp Tables

About Technical Guidelines

The detection process consists of three major components, of which a variety of factors impact the performance:

- **Datasets:** Select, join, filter, and transform records from tables in the database or from other datasets. (refer to *Datasets*, on page 175, for more information).

- **Patterns:** Retrieve and sort records from one or more datasets and then search through them to identify a sequence of activity or a set of conditions of interest. The Scenario Manager includes sequence patterns and rule patterns. (refer to *Sequence Patterns, on page 175*, and *Rule Patterns*, on page 176, for more information).

- **Jobs:** Associate patterns to datasets and contain parameters associated with the execution of a detection process; for example, specifies whether the match direction is forward or reverse (refer to *Jobs*, on page 176, for more information).

Datasets

The datasets are stored as database views, which are virtual tables that are based on an underlying SQL definition. Dataset retrieval is often the biggest contributor to the completion time for a detection process. Some datasets use *group by* expressions to perform aggregate analysis on a set of records. Some datasets are also created by performing *union* operations on multiple queries.

Contributors to dataset retrieval time include the following:

- Volume of data being searched in the dataset
- Extent to which conditions can be applied to filter the returned records
- Number of PDM tables that are joined together in the dataset
- Execution plan that the database uses to run the dataset query

Sequence Patterns

Sequence patterns search through one or more datasets to identify instances of a behavior of interest. When a sequence pattern searches through records, it creates a *match state* each time it finds a record that satisfies the specified initial condition in this pattern. The detection engine must hold this event in memory as it searches through additional records to verify whether the remaining conditions in the pattern are satisfied. The following drive pattern performance:

- Total number of records through which the pattern must search based on the size of the datasets it is using.

- Number of match states (that is, partial matches) that it creates in the course of a run.

- Number of match states that the pattern must hold in memory simultaneously.

  This is often a function of the amount of data that the pattern must search through after the initial event is identified to determine whether the remaining conditions are satisfied.

Rule Patterns

The rule matcher detection tool searches large datasets to identify situations of interest, either based on an individual occurrence of a particular event or an aggregate analysis of the activity that a particular entity performs. The rule matcher can search across multiple datasets for the presence or absence of particular events. The tool can access data incrementally, retrieving records for only those entities that satisfied the conditions specified earlier in the pattern. The following drive the performance of rule patterns:

- Amount of time required for retrieving records from the pattern's primary dataset

- Total number of datasets that the pattern uses.

- In order to avoid accessing unnecessary data records, the pattern can eliminate cases from consideration early in the pattern.

Jobs

Jobs associate patterns and datasets and control some aspects about execution of the detection process. Jobs have a limited number of configurable options. Tuning jobs also involves ensuring that the detection work is balanced appropriately between pattern and dataset components.

About Improving Pattern Performance

The following sections describe design tips for sequence and rule patterns.

Improving Performance of Sequence Patterns

The data miners should apply the following tips to improve the performance of sequence patterns:

- Choose an unusual occurrence to serve as the initial event in a pattern, whenever possible. The sequence matcher creates a match state each time an event satisfies the initial condition for the pattern, and holds this condition in memory as it checks for the remaining conditions. Selecting an unusual event to start the sequence reduces the number of match states that are created during a run.

- Limit the rows retrieved in the datasets that a pattern uses. Specifically, if certain records cannot satisfy the pattern criteria, it is best to apply logic to filter them out up-front. If certain detection conditions can be checked in the dataset, it is generally more efficient to allow the database to filter them out rather than applying this logic in the pattern. It is also helpful to determine the appropriate lookback period range that is necessary to identify the behaviors of interest, so that the patterns do not look through a larger volume of historical data than is necessary.

- Use dataset sorting to the maximum extent possible. To optimize performance, sort records that are used together to form an alert into adjacent positions in the dataset. For example, if a pattern is looking for a behavior within one account, sort the datasets it uses so that the records are clustered together by account. Doing this minimizes the amount of data through which the pattern must search to find a match, and reduces the number of partial matches that are held in memory at any one time.

- Apply criteria that terminate a search as soon as it becomes evident that a particular case cannot result in a match. Ending a search as early as possible prevents the pattern from performing unnecessary work or creating extra match states. For example, the activity of interest in a particular scenario must occur in the same account and the same security within a specified time frame, the pattern should have explicit conditions that cause it to terminate when any of these conditions can no longer be satisfied.

Improving Performance of Rule Patterns

Data miners should apply the following tips to improve the performance of rule patterns:

- Limit the rows retrieved in the datasets that a pattern uses. Specifically, if certain records cannot satisfy the pattern criteria, it is best to apply logic to filter them out up-front. If certain detection conditions can be checked in the dataset, it is generally more efficient to allow the database to filter them out rather than applying this logic in the pattern. It is also helpful to determine the appropriate look-period that is necessary to identify the behaviors of interest, so that the patterns do not look through a larger volume of historical data than is really necessary.

- Apply the most restrictive pattern conditions in the primary rule. Rule patterns access data incrementally, retrieving records only for the entities that satisfy the conditions of the primary rule. Applying the most limiting conditions up-front helps to reduce the total number of records that are retrieved and evaluated in the detection process.

- Use periodic checkpoints to terminate the search at the point when the conditions for a match can no longer be satisfied. Rule patterns include explicit checkpoints that must be satisfied in order for the detection process to move forward. If the conditions in a checkpoint are not met for a particular entity, the detection process does not continue to retrieve data for that entity. Developers should apply checkpoints in patterns wherever they can substantially reduce the population of records being considered for a match.

● Use aggregate functions and conditional expressions to avoid performing redundant evaluation of the same data records. The rule matcher has several built-in functions to perform aggregate analysis of the activity that a particular entity performs. These functions can be used in conjunction with conditional expressions to analyze a set of records in two different ways without retrieving the records twice. For example, if a scenario requires computing the total monetary amount for all transactions, and the total amount for the subset of transactions that are considered high-risk, accomplish this through the use of two aggregate functions such as:

TOT_TRANS_AM = Sum(trans_am)

HR_TRANS_AM = Sum(cond(risk_lvl_nb ≥ @high_risk_nb; trans_am; 0))

Note: If you want to create a bound variable that compares these results (for example, to compute a high-risk transaction percentage), you can do so in the same rule as long as the new binding appears further down on the list than the bindings that it is comparing.

● Use datasets that have a leading index on the focal entity attribute. All sub-rule datasets retrieve records for the subset of entities that have satisfied the conditions of the primary rule, as well as the conditions of any checkpoints that appear before the sub-rule. For example, if the scenario is account-focused, the sub-rule datasets retrieve data only for those ACCT_INTRL_IDs that are candidates for a match based on the constraints of the primary rule and any preceding checkpoints. The retrieval of data from the sub-rule datasets is much more efficient if it can access the records for this entity using an index (in the case of this example, an index on ACCT_INTRL_ID is required).

About Constructing Datasets

The data miners should apply the following tips when constructing datasets:

- Apply as much filtering as possible at the dataset level. If certain cases cannot satisfy the criteria for a scenario, it is best to eliminate them from consideration through dataset filtering rather than processing these records in the pattern.

- Create dataset thresholds for any filtering criteria that may require adjustment over time. When creating these thresholds, the data miner should not set the maximum and minimum values in a way that would cause the scenario to perform too slowly. For example, if the lookback period is a configurable parameter, the data miner must ensure that the scenario performs adequately when this threshold is set to its maximum value.

- Make appropriate considerations for null values when creating dataset filters. Remember that null values cannot be joined across tables, and null values will not satisfy the criteria for comparisons to specific values. For example, `the dataset filter` `ACCT_TYPE_CD != 'IRA'` eliminates cases where the `ACCT_TYPE_CD` field is null. We recommend using the NVL expression when checking that a nullable field is not equal to a particular value (for example, `NVL(ACCT_TYPE_CD, 'NONE') != 'IRA'`). Use NVL expressions when performing computations with nullable attributes.

- When creating derived attributes, specify the correct data type (that is String, Float, Integer, or Date). The most common mistake is to specify a numeric field as a string. This causes problems when making comparisons in patterns. For example, *100* is less than *99*.

- Avoid any situation that can cause a divide-by-zero error. If a zero value is not expected in the denominator of a calculation, the data miner should apply specific logic to handle this situation should it occur.

- Remove references to tables that the detection process does not require. In some cases, information that is used for display—in places such as the alert *highlights*—is not directly needed to identify matches. Retrieve this information after a match is generated using the augmentation process, rather than bringing extra tables into the original pattern dataset. For example, if a dataset uses the Security Master table only so that the pattern can display a ticker symbol in the alert highlights, the data miner should instead use binding augmentation to add the ticker symbol to the highlights as part of post processing.

- Rely on summarized data whenever possible, rather than calculating aggregate statistics by searching through a large volume of data. For example if a scenario requires information on aggregate trade or transactional data, it is best to retrieve it from the summary tables (for example, `ACCT_SMRY_MONTH`) rather than computing it in a dataset. If possible, arrange the frequency and lookback periods for scenarios.

- Whenever possible, create the dataset by using the graphical icons. Using Edit SQL mode is required whenever the user needs to apply advanced SQL clauses (for example, exists, not exists, union), aggregate functions, or performance hints. However, datasets that are in Edit SQL mode are somewhat harder to maintain, so data miners should try other options before moving to Edit SQL.

- For sequence scenarios, consider using temporary tables to narrow down the records included in a dataset. In some cases, a relatively simple set of SQL logic can remove a large percentage of situations that have no chance of satisfying the criteria for a particular scenario. Under these conditions, it is often an efficient strategy to perform an initial screening step that identifies candidate entities and stores them in a temporary table. The other datasets can then use the temporary table as a starting point, thereby limiting the volume of data they need to retrieve.

- Use the comment boxes to document your datasets rather than writing in-line comments in the SQL directly. It is not necessary to put comments directly into SQL, and this practice should be avoided. Do not write

dataset names (for example, `dsview12345`) into SQL comments under any circumstances because this causes problems during job runs.

- If it is necessary to paste in the dataset SQL for some reason, do not start from any application that can include unprintable formatting characters with the text, such as Microsoft Word. It is much safer to paste in the information from a text editor such as Notepad. Unprintable characters can cause problems during scenario extraction and migration.

- Review the execution plan for each query either with a Database Engineering (DBE) or via a SQL tuning utility. If the execution plan indicates that the query is inefficient, either modify the query or have the DBE add appropriate indexes. As a last resort, if the query is not using the correct indexes that it should, apply database hints to instruct the query optimizer appropriately.

- Verify that all SQL expressions used are in conformance with the Oracle Financial Services SQL Usage Guide. This ensures that the datasets are compliant with both Oracle and DB2 or UDB.

Examples of Common Business Problems

Many scenarios have similar detection requirements. This section describes common problems (by case number) and the recommended detection approaches for addressing them.

Case #1

In Case #1, each occurrence of a particular event should result in a match. For example, identify each instance in which an employee trades a security on a restricted list.

The recommended approach to this issue is to create a dataset to identify the cases of interest—in this case by joining together tables involving trades, restricted securities, system user/account mappings, and date control information. If additional restrictions apply (for example, minimum order size), create dataset filters to eliminate the unwanted cases. The matches should be generated with a rule pattern that contains only a primary rule. If desired, use the Focus Clustering option in the Rule Matcher to collect all activity for an employee into a single match.

Case #2

In Case #2, a match should be generated if the aggregate activity that an entity performs meets or exceeds specified parameters. For example, identify cases in which an account performs two or more (configurable) cash deposits in a two-day period that are each below \$3,000 (configurable) and collectively sum to at least this amount.

We recommend that you create a dataset that retrieves cash transactions over the lookback period, filtering out those that are above the size limit. Implement the detection pattern as a two-row rule pattern, as shown below:

- **Primary Rule:** Use the dataset described above, and apply focus clustering. No constraints are required—consider all of the records from the dataset. Create aggregate bindings for:

  ```
  TOT_TRANS_AM = sum(TRXN_AM)

  TOT_TRANS_CT = count()
  ```

- **Checkpoint:** Test the aggregate bindings against the scenario thresholds:

  ```
  TOT_TRANS_AM > or < or >= or <= @MIN_TRANS_AM && TOT_TRANS_CT > or < or >= or <=
  @MIN_TRANS_CT
  ```

Case #3

Create a match when one event follows another within a specified time period. For example, identify cases where an account performs a withdrawal that is preceded by a change of registration information within seven days.

The recommended approach to this issue is to identify instances of the behavior of interest by creating a dataset that joins the tables involved. For example:

SELECT a.BO_TRXN_SEQ_ID, a.ACCT_INTRL_ID

From BO_TRXN a, CHG_LOG_SMRY b

Where a.ACCT_INTRL_ID = b.ACCT_INTRL_ID

And a.DBT_CDT_CD = 'C'

```
And b.ACCT_RGSTN_CHG_CT > 0

And b.CHG_DT ≤ a.EXCTN_DT

And b.CHG_DT ≥ (a.EXCTN_DT - 7)
```

Bring in additional tables (such as KDD_DATE_CONTROL) to control the lookback period.

An alternate strategy is to create a dataset that selects from the transaction table, and use an *exists* clause to detect the presence of an account registration change in the change log summary table.

Generate the matches by using a one-row rule pattern that consists only of a primary rule. Apply focus clustering to generate only one match for each account that exhibits this behavior. The pattern matches transaction building blocks because this field is specified as the dataset key. Use augmentation to add the building blocks for the relevant change log summary records.

Generally, it is more efficient to solve a business problem at the dataset-level, rather than through a sequence pattern, if the scenario involves a situation that consists of only two events or the behavior involves a series of events that are not time ordered.

Case #4

In Case #4, the behavior of interest involves three or more events that must take place in a particular order. For example, identify cases in which an employee places orders shortly before and shortly after a large customer order for the same security.

The recommended approach to this issue is to create a dataset that includes orders placed during the lookback period. To be of interest in this scenario, an order must either originate from an account associated with an employee or be of sufficient size to meet the definition of a large order. Use dataset filters to eliminate all other orders from consideration, because they cannot be used to satisfy the criteria for an alert. Generate the matches by using a sequence pattern. The pattern should sort the records by security and then by time so that the events of interest are arranged into adjacent positions in the dataset. The pattern should have the following general structure:

1. ROW – Employee Order (1 occurrence)

2. ROW – Orders from Other Emps (0 or more, not recorded)

3. ROW – Large Customer Order (1)

4. ROW – Orders from Other Emps (0 or more, not recorded)

5. ROW – Employee Order Offsetting the Initial Order (1)

All events in the pattern must involve the same security, and events (1), (3), and (5) must occur within a specified amount of time. The pattern should have explicit conditions causing the sequence to terminate whenever the securities involved are not the same, or if the time parameters are exceeded.

About Scenario Constructions to Avoid

We recommend avoiding several techniques in scenario design, because they may either lead to errors or performance problems. Scenario developers should refrain from using the following constructions:

- **Jobs with looping entities:** Setting up a job with a looping entity causes it to execute in multiple runs—one run for each unique value of the looping entity. The purpose of this is to create small job runs that include only data for single entity, so that evaluation of the records for that entity can occur quickly. However, using looping often results in creating a large number of runs, which can result in an unacceptable completion time for the job. Instead of using looping, scenario developers should use dataset sorting (multi-dataset row ordering). This technique sorts the records for a particular entity into adjacent positions in the recordset, which is similar to the way they are arranged in a looping job. However, the dataset sorting approach is more efficient because it selects all of the records for the job at once, avoiding the overhead of running multiple queries to retrieve the records.

- **Failing to account for all possible terminating conditions in a sequence pattern:** All sequence patterns use sorted datasets, and many patterns do not generate a match until they have evaluated all of the records that may be of interest based on the starting condition. For example, if a pattern is looking for a particular behavior within one account, the datasets that it uses are typically sorted by account, and the pattern may not check the final alert criteria until discovery of a record for a different account. This approach ensures evaluation of all of the records for one account before generation of a match.

  If multiple datasets are involved, note the pattern of all possible datasets in which the *next account* record may be found. Failure to do so causes the pattern to terminate when it does not find the next account record in the expected dataset, which results in missed alerts.

- **Looping-within-looping:** The most common form of this error is to have:

  SEQ loop 1-

  or

  ROW A loop 1-

  ROW B loop 1-

  ROW C loop 1-

  The desired effect of this pattern is to find one or more As, Bs, or Cs. The pattern structure above is inefficient because if it sees an A followed by another A, the sequence matcher does not know whether to match the same A ROW again or to loop back through the SEQ to match the A. So the sequence matcher branches and does both. This can cause exponential growth in the number of states maintained in memory and slow down the algorithm significantly. A better structure is as follows:

  ```
  SEQ loop 1-
  ```

  or

  ROW A loop

  ROW B loop

  ROW C loop

  If an A is followed by an A, only one path can be followed.

- **Misuse of bind or rebind for pattern variables**: If a variable has already been bound once in a pattern, if it is bound again later in the pattern, use *rebind*. If the variable is bound again using *bind* causes, the sequence

matcher does not produce an error message and also does not change the variable's value. This can have unintended results if the pattern writer thinks the variable's value is being changed. Some recommended practices are as follows:

- Bind all variables that you intend to use in a pattern in the very first row using *bind* even if they have to be initialized to 0 or a dummy value. Then use *rebind* rebind everywhere else in the pattern.

- Always use *rebind* within a looping construct.

● **Alternation conditions that overlap or fail to cover all possible conditions:** Mutual exclusion and covering all possible cases. When a pattern contains an OR statement, it is usually (but not always) the case that the pattern writer intends that the ROWs in the OR statement are both mutually exclusive and cover all possible cases. This is especially true when the OR statement is within a looping construct. Pattern writers often accidentally write OR statements that have gaps or overlap. For example:

- OR statement with a gap is `OR: ROW x > 5, ROW x < 5`. This has a gap because if x equals 5, it does not match either row and the match dies.

- OR statement with overlap is `OR: ROW x ≥ 5, ROW x ≤5`. This has overlap because if x equals 5, it matches both rows and branches.

If a pattern is missing alerts that it should be catching, an OR statement may have a gap. You may need to add *junk* rows that match everything other than the entities in the other rows. The constraints in junk rows can be complex and are often where the gap is introduced. If a pattern that you are writing has a gap or overlap, be sure that is what you intend, and be sure to document it.

About Temp Tables

In some cases, a relatively simple set of SQL logic can remove a large percentage of situations that have no chance of satisfying the criteria for a particular scenario. Under these conditions, it is often an efficient strategy to perform an initial screening step that identifies candidate entities and stores them in a temporary table. The other datasets can then use the temp table as a starting point, thereby limiting the volume of data that they need to retrieve.

For example, an earlier release of the *Brokerage and Investor Protection* product included a scenario to identify *mutual fund switches*, which are cases in which a customer performs short-term trading in a mutual fund. Specifically, the customer must:

1. Buy a mutual fund.

2. Sell the same mutual fund after less than six months after event (1).

3. Buy a different mutual fund sells within seven days of event (2).

The situation is of interest if a representative solicits event (2) or event (3). The scenario is run daily.

The challenge in developing this scenario is that large brokerage firms may have more than two million mutual fund trades per month. Consequently, a scenario takes a long time to run if it has to process all mutual fund trades for a six-month period. To shorten the run time, the scenario uses a temp table that identifies the distinct list of accounts that met three conditions:

1. The account performed a mutual fund purchase on the most recent day.

2. The account had performed a mutual fund sale in last seven days.

3. A representative solicited at least one of the account's mutual fund trades during this period.

The SQL script for the temp table ran within a few minutes and the output identified a limited number of accounts. The system uses this temp table in the final pattern dataset to retrieve the entire six months of data only for the accounts in the temp table. Using this approach, the system retrieves a manageable amount of data and the detection process runs in an acceptable amount of time.

Creating Temp Tables

To create a temp table, follow these steps:

1. Create a database view that contains the SQL definition for the temp table. Name the view with a prefix that contains *VW* and the name of the scenario group in which the view is used (that is, BC, TC, AML, FR, or INSURANCE). The rest of the name should provide a meaningful description of the table's purpose (for example, VWBSM_RCNT_MFUND_TRDS).

2. Create a table that contains the records from the view. The table name should be the same as the view with the extension _TMP
(for example, VWBSM_RCNT_MFUND_TRDS_TMP).

3. Create a wrapper dataset for the temp table.

4. Document the SQL scripts to create the view, and then refresh the temp table.

Using Temp Tables

Consider these issues when using temp tables:

- Use temp tables only in cases where they significantly improve the performance of a detection process. Test the performance of the job with and without the temp table to ensure that the temp table shortens the completion time substantially.

- Never use temp tables with the Rule Matcher detection tool. The Rule Matcher can access data incrementally, retrieving records only for those entities that satisfied the conditions specified earlier in the pattern. This capability obviates the need for temp tables.

- Note that data records cannot be matched directly from temp tables. All datasets used in detection must have a KEY_ID that is derived from one of the business tables in the data mart. The user interface does not have any means of displaying a record that is matched out of a temp table.

Multi-Dataset Row Ordering

Multi-dataset row ordering uses the least common denominator approach for inter-sorting and any additional attributes for intra-sorting.

An example of row ordering.

Dataset	Sort Attributes
DS1	Employee, Account, Trans Am
DS2	Employee, Account, Logon ID
DS3	Employee, Account

In this example, the INTER sort occurs across datasets DS1, DS2, and DS3 using Employee and Account. Additionally, DS1 is INTRA-sorted on Trans Amt and DS2 is INTRA sorted on Logon ID.

Multi-dataset row ordering uses the minimum *number* of attributes listed for sorting for each dataset. The sort algorithm does not recognize names as are specified in the front end. Therefore, if you specify *Employee* as the first attribute in DS1 and *Account* as the first attribute in DS2 and DS3, the algorithm reads one attribute as the least common denominator and attempt to inter-sort on *Employee* and *Account*, which is incorrect. As data miner, you must specify the order of sort attributes for each dataset correctly so that the system performs the inter-sort on the same logical attribute. It is not necessary for an inter-sort attribute to have the same name across datasets, but they must have the same meaning and data type.

Threshold Sets

In the **Threshold Definer** tab of the Threshold Manager, you can add, modify, and delete thresholds from the base threshold set. Any thresholds that you add through this tab are added to all derived threshold sets. Any threshold that you update through this tab has its values copied into derived threshold sets that are set up to inherit the base threshold value. Any threshold that you delete through this tab is deleted in all derived threshold sets and removed from any patterns or datasets with which it is associated.

You must create the needed scenario in the Sequence Scenario Editor or Rule Scenario Editor before it is available in the Threshold Manager.

APPENDIX B

Naming Conventions

This appendix describes naming conventions to follow when creating datasets and scenario components. Specifically, this appendix covers the following topics:

- About Dataset Naming Conventions
- About Naming Scenario Components

About Dataset Naming Conventions

Use the following conventions when creating datasets:

- For all datasets that you create for a specific scenario, begin the name with an abbreviated version of the scenario name that the scenario or pattern catalog ID represents (for example, BC-ShortOptionRisk-dRBPC for Short Option Risk).

- If the scenario uses multiple datasets, start each dataset with the same prefix and provide an extension that specifies the purpose of each dataset. Make the extensions as meaningful as possible; avoid extensions like _1 and _2 because they may not be recognizable to another scenario developer. Use the _p prefix and the pattern code if the dataset is pattern specific.

- If you no longer need a particular dataset, rename it with the prefix zzz_ (for example, zzz_OPTN_ASGMT_RISK). This causes the dataset to sort to the bottom of the dataset list.

- If using a dataset across multiple scenarios, do not give it a name that is specific to one scenario. Instead, give it a name that describes the purpose of the dataset. These cases should be rare. Remember that when datasets are transferred between environments they are made parts of scenarios, so treat the *multi-scenario* datasets with care.

About Naming Scenario Components

Use the following guidelines when naming bound variables, thresholds, table attributes, and external bindings:

- Write table attribute names in UPPER_CASE. For example:

  `ACCT.ACCT_INTRL_ID`

- Write bound variables that other Oracle Financial Services modules (external bindings such as variables that represent focal entities) require and define in the UPPER_CASE. For example:

  `ACCOUNT, SECURITY, ASSOC_SECURITY`

- Write bound variables that the scenario defines in Title_Case and use the `$` prefix. For example:

  `$Min_Trans_Am`

- Write thresholds that the scenario defines in Title_Case and use the `@` prefix. For example:

  `@Max_Trans_Am`

In general, use the upper case when referring to components defined outside of the scenario, and the title case when referring to bound variables or thresholds that the scenario defines.

Reserved Words

Use lowercase when referring to reserved words.

For example:

`select ACCT.ACCT_INTRL_ID from ACCT where ACCT.ACCT_TYPE_1 = 'A.'`

Bound Variables

Use an extension on bindings that are computed or derived from other variables and attributes that describes the type of information it is capturing. Use the same naming conventions applied in the database:

- o_Ct = Count
- o_Qt = Quantity
- o_Am = Amount
- o_Pt = Percent
- o_Rt = Ratio

Variable names should describe the purpose of the variable as clearly as possible.

In the Sequence Matcher, the focal entity must be bound to a variable with the focus name (for example, ACCOUNT). Do not use the name of any focus type under any other circumstances as a binding name for other purposes.

Thresholds

Use the following guidelines when creating threshold names:

- For thresholds that represent minimum or maximum counts/quantities/amounts, use the prefix `Min_` or `Max_`.
- Use the extension `_Lmt`, for thresholds that represent a time limit.

- List thresholds (which are applicable only in datasets) should contain the extension `_Lst`.

- Use the extension `_Fl`, if using a string threshold as a flag (which has a valid value of Y or N).

- Use the extension `_Cd`, if using a string threshold to specify a code.

Jobs

The general format for a job name is:

```
<scenario//focus/pattern catalog ID> <(frequency)>:
```

Spell each job name exactly as it appears in the scenario configuration table in the appropriate Technical Scenario Description (TSD).

For the frequency period, use the codes D (Daily), W (Weekly), BW (BiWeekly), M (Monthly), and Q (Quarterly). If the frequency period is a tunable parameter, use the scenario defaults when creating the job name.

For example, the job name would look like:

TC-ImprCloseOrder-dEQT-fAC-pCNL (D)

APPENDIX C

Scenario Manager to User Interface Mapping

This appendix provides examples of correspondence of the elements within the Scenario Manager to a particular area of the Web Application User Interface.

- About Scenario Editor to Web Application User Interface Mapping

- About Pattern Role Editor to Web Application User Interface Mapping

- About Threshold Sets Tab to Web Application User Interface Mapping

- About Threshold Analyzer Role to Threshold Analyser Utility User Interface Mapping

About Scenario Editor to Web Application User Interface Mapping

In the Web Application user interface (UI), you can view elements on a particular page that correspond to scenario elements of interest.

For example, use a scenario editor to define a scenario (Figure 69).

Figure 69. Defining a Scenario in a Scenario Editor

Correspondence of these elements within the scenario editors to the Alerts & Case List in the Web Application is shown in Figure 70.
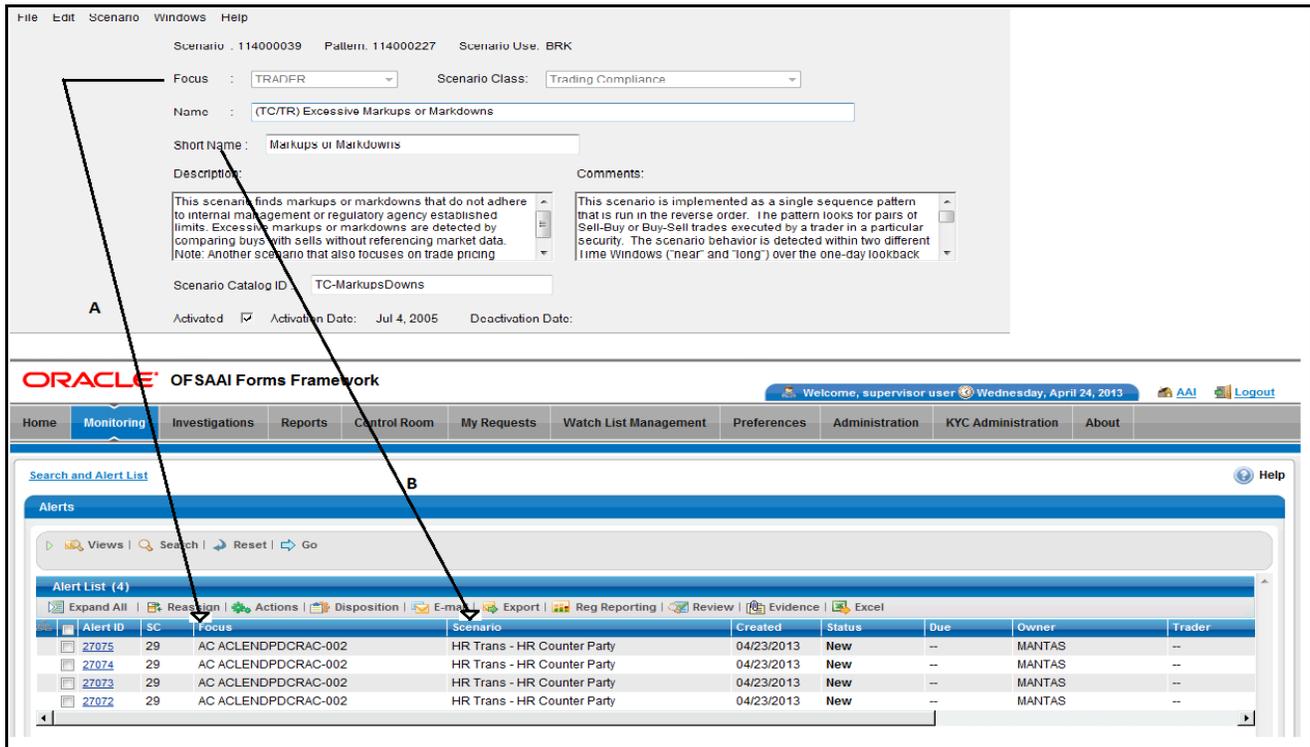
Figure 70. UI Mapping Sample- Scenario Editor to Alert List

In this example, the UI's Alert & Case List page contains scenario elements from the scenario editor dialog box:

- In the scenario editor, the entry in the Focus field maps to the focus in the first row under the UI's Focus column heading (A).

- In the scenario editor, the entry in the Short Name field maps to the Scenario name in the first row under the UI's Scenario column heading (B).

About Pattern Role Editor to Web Application User Interface Mapping

In the Web Application user interface (UI), you can view elements on a particular page that correspond to Pattern elements of interest.

In the Pattern Roles dialog box, you can view related pattern attributes, as in the sample entries in Figure 71.
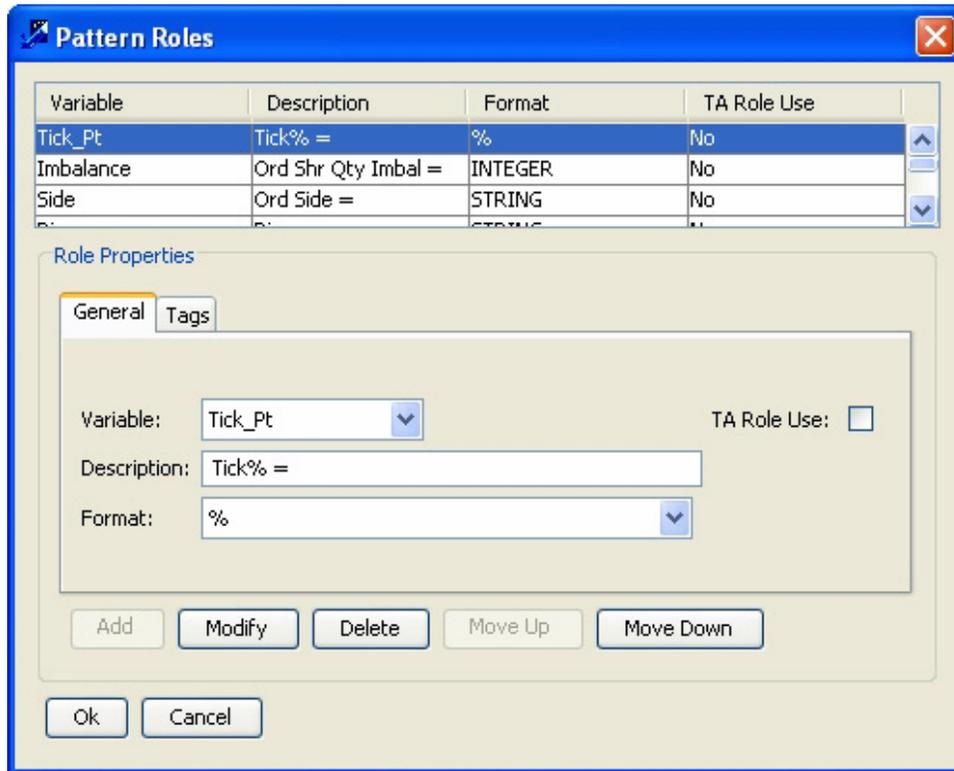
Figure 71. Pattern Roles Dialog Box

Correspondence of these elements within pattern roles to the Alerts & Case List in the Web Application is shown in Figure 72.
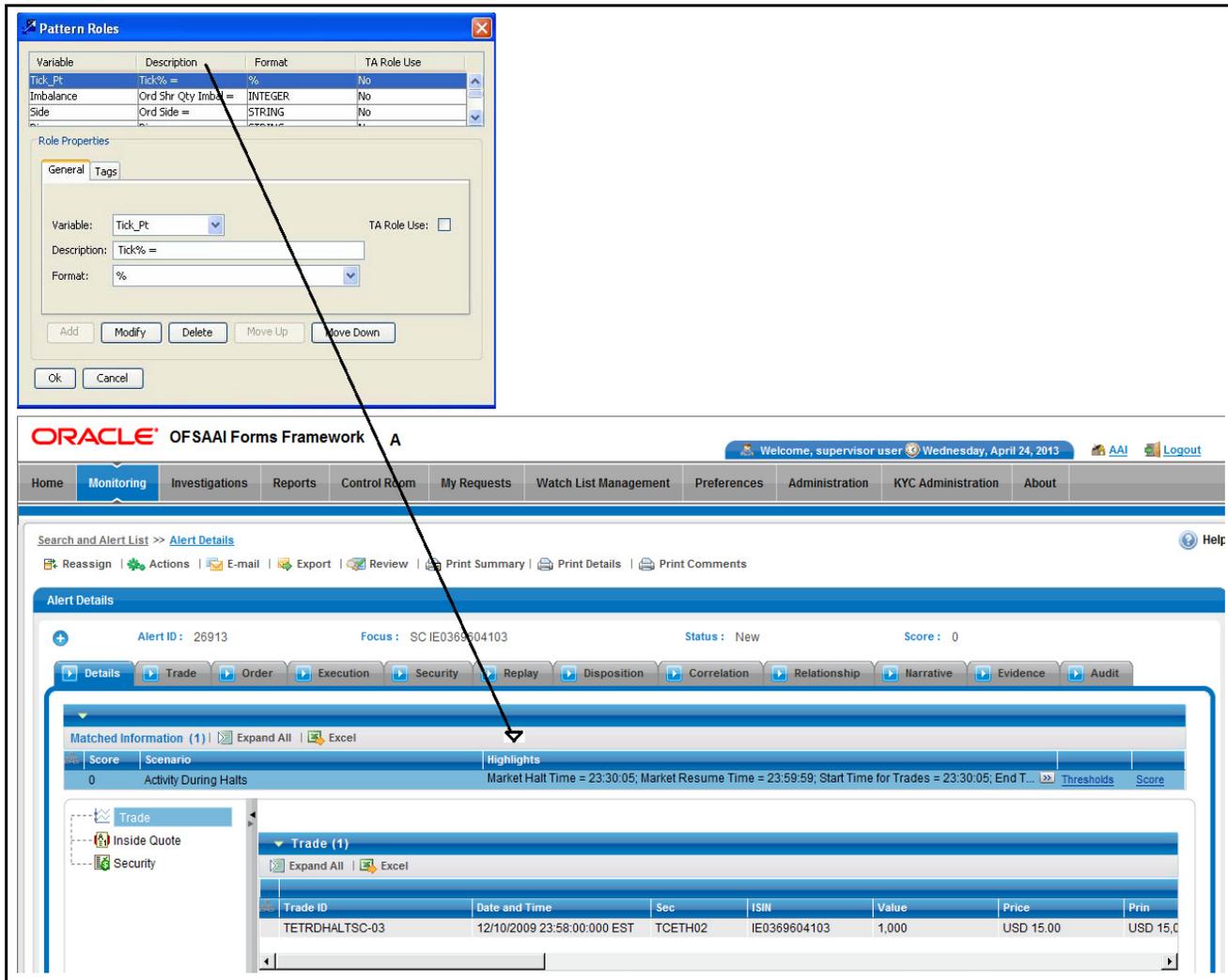
Figure 72. UI Mapping Sample- Pattern Role Editor to Alert List

In this example, the UI's Alert & Case List page contains scenario elements from the Pattern Roles Editor dialog box:

- In the Pattern Roles dialog box, the first entry under the Description column heading maps to the highlight in the first row under the UI's Highlights column heading (A)

About Threshold Sets Tab to Web Application User Interface Mapping

In the Web Application user interface (UI), you can view elements on a particular page that correspond to scenario threshold sets of interest.

For example, use the Threshold Sets tab to view and modify the contents of the Base Threshold Set and any existing sets (which you access in the **Threshold Sets** drop-down menu) (Figure 73).
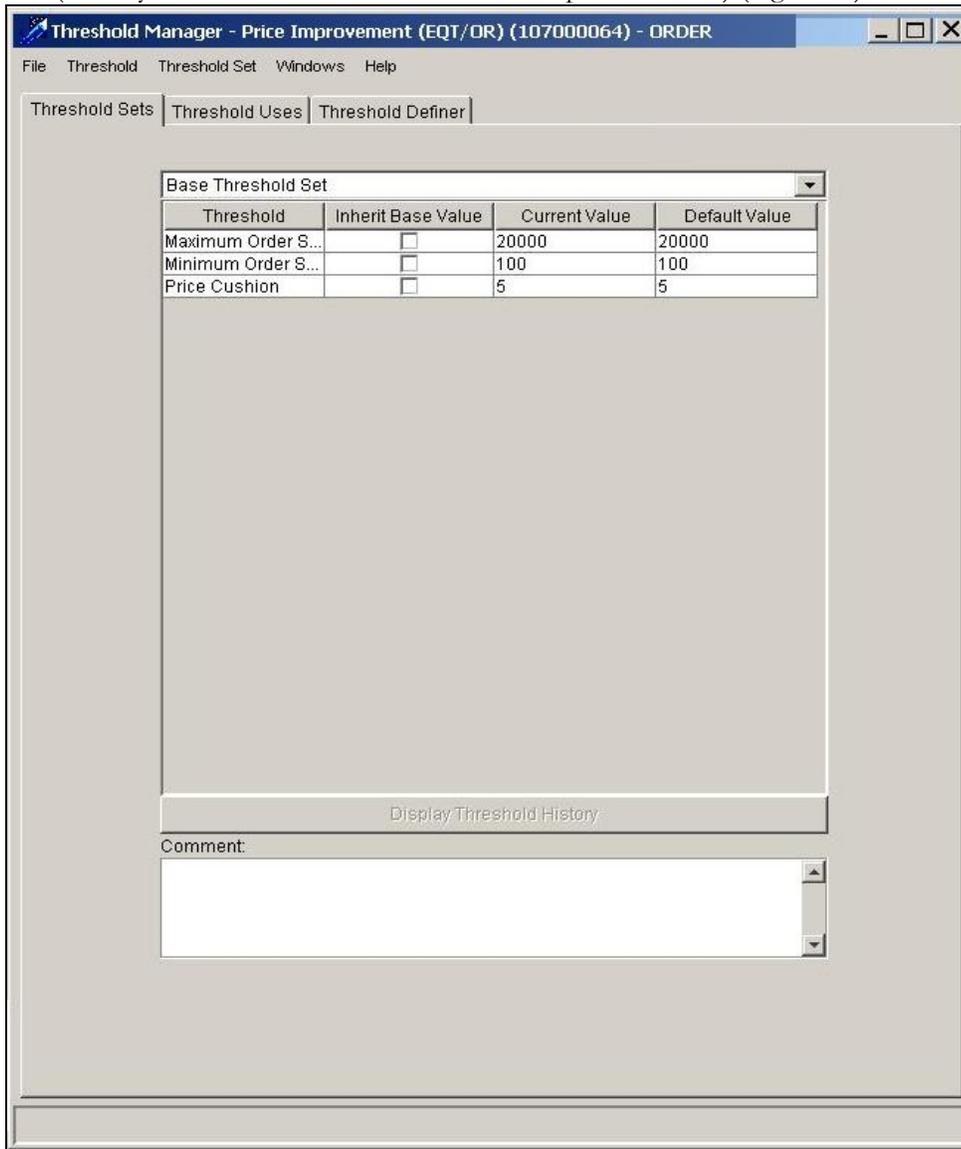
Figure 73. Threshold Sets Tab

Correspondence of these elements within the Threshold Set editor to the Alert & Case List in the Web Application displays Figure 74.
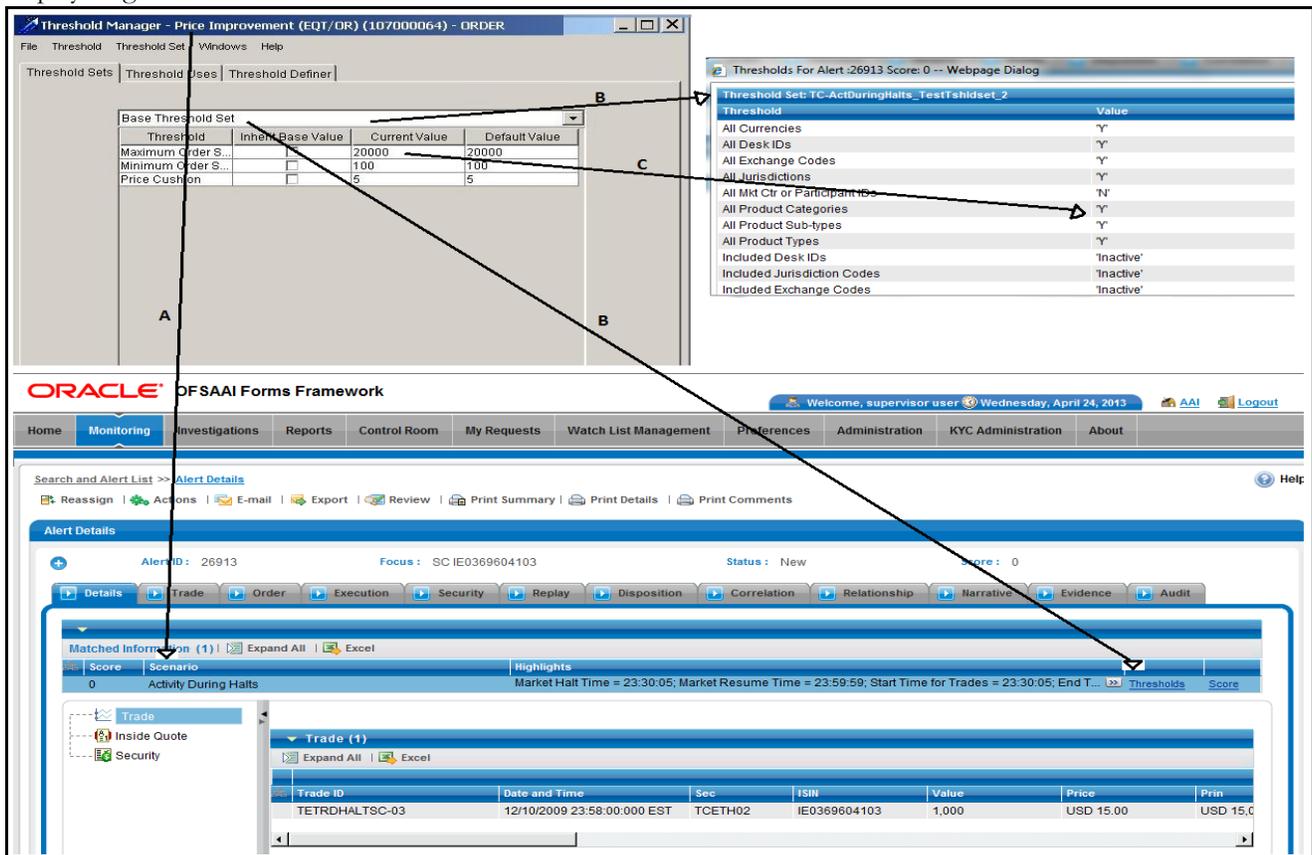
Figure 74. UI Mapping Sample - Threshold Sets Tab to Alert Details

In Figure 74, the UI's Alert & Case Details page contains scenario elements from the Threshold Sets tab: :

- In the Threshold Sets tab, the Scenario name maps to the UI's **Scenario** field in the Alert Display (A)

- In the Threshold Sets tab, the Threshold Sets drop-down list selection maps to the Threshold Set name that displays in the Threshold link window (B)

- In the Threshold Sets tab, the entries under the **Current Value** column heading maps to the UI's **Value** column in the **Threshold** link window (C)

About Threshold Analyzer Role to Threshold Analyser Utility User Interface Mapping

In the Threshold Analyzer user interface (UI), you can view elements on a particular page that correspond to threshold analyzer role and threshold analysis elements of interest.

In the Threshold Analyzer Roles dialog box, you can view related threshold attributes, as in the sample entries (Figure 75).
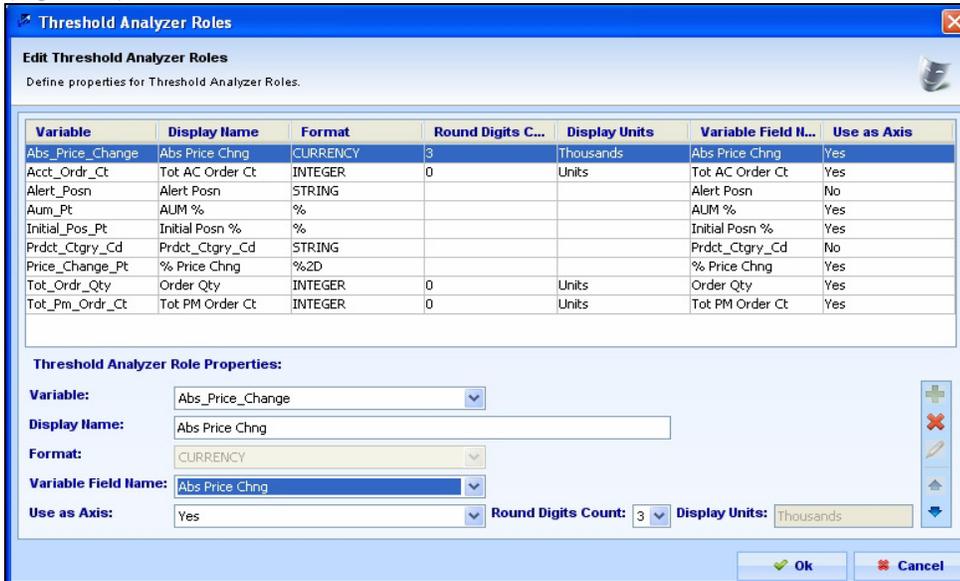
Figure 75. Threshold Analyzer Role Dialog Box

Correspondence of these elements within the threshold analyzer role and threshold analysis in the Threshold Analyzer utility (Figure 76).

.

Figure 76. UI Mapping Sample - Threshold Analyzer Roles to Threshold Analysis

In Figure 76, the UI's Threshold Analysis section contains threshold elements from the Threshold Analyzer Roles dialog box:

- In the Threshold Analyzer Roles dialog box, the Variable name maps to the UI's Variables drop-down (A)

● In the Threshold Analyzer Roles dialog box, the Use as Axis field maps to the display of the X/Y axis variables drop-down list (B)

● In the Threshold Analyzer Roles dialog box, the Display Name field maps to the display name of X and Y axis of the graph (C)

Index