IBM

MQSeries® Integrator

User's Guide

Version 1.1

GC34-5504-01

Note: Before using this information, and the product it supports, be sure to read the general information under *Notices* **on page 215.**

Second edition (June 1999)
This edition applies to IBM® MQSeries Integrator, Version 1.1 and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

At the back of this publication is a page titled "Sending your comments to IBM". If you want to make comments, but the methods described are not available to you, please address them to:

IBM United Kingdom Laboratories
Information Development,
Mail Point 095,
Hursley Park,
Winchester,
Hampshire,
England,
SO21 2JN.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Chapter 4: Rules 137

Chapter 5: Visual Tester..............................195

Appendix A: ASCII Extended Character Set..217
Appendix B: EBCDIC Character Set..............223
Appendix C: Notices 237

Chapter 1
Introduction

This user's guide describes MQSeries Integrator Version1.1. It is organized into the following sections:

Chapter 1, *Introduction*, provides Y2K information, a brief overview of MQSeries Integrator, a list of available documentation, and technical support information.

Chapter 2, *MQSeries Integrator Overview* describes the MQSeries Integrator product, its architecture, and its components.

Chapter 3, *Formatter*, describes the NEONFormatter module and provides procedures to create, change, and delete both input and output formats.

Chapter 4, *Rules*, describes the Rules Daemon and provides the procedures to create, change, and delete a rule.

Chapter 5, *Visual Tester*, describes the Visual Tester, which enables users to test queues, formats, and rules.

Appendix A, *ASCII Extended Character Set*, contains an ASCII extended character set table.

Appendix B, *EBCDIC Character Set*, contains an EBCDIC character set table.

Appendix C, *Notices*, contains trademark and service mark information.

A Glossary listing terms used in describing MQSeries Integrator.

Product Documentation Set

The MQSeries Integrator documentation set includes:

- ***MQSeries Integrator Installation and Configuration Guide*** details the installation and initial implementation of MQSeries Integrator and the MQSeries Integrator applications.

- ***MQSeries Integrator User's Guide*** helps users understand and apply the program through its graphical user interfaces (GUIs).

- ***MQSeries Integrator System Management*** Guide is intended for system administrators and database administrators who work with MQSeries Integrator on a day-to-day basis.

- ***MQSeries Integrator Application Development Guide*** assists programmers in writing applications that use MQSeries Integrator APIs.

- ***Programming Reference*s** are intended for users who build and maintain the links between MQSeries Integrator and other applications. This document consists of following volumes:

  - ***MQSeries Integrator Programming Reference for*** NEONFormatter is a reference to NEONFormatter APIs for those who build and maintain the links between MQSeries Integrator and other applications.

  - ***MQSeries Integrator Programming Reference for*** NEONRules is a reference to NEONRules APIs for those who build and maintain the links between MQSeries Integrator and other applications.

> **Note:**
> For information on message queuing, refer to the IBM MQSeries documentation.

Summary of Changes

This document is a major revision in support of the functional changes introduced with Version 1.1. This revision also includes maintenance and editorial changes.

Oracle 8.5 databases are supported for Windows NT, AIX, HP-UX, and Solaris.

Chapter 3: Formatter on page 13 has been updated to include new functionality. The following summarizes the new information:

- Save an Output Control as an Input Control
- Save a compound output format as a compound input format
- Save a compound input format as a compound output format
- Save a flat input format as a flat output format
- Save a flat output format as a flat input format
- Globally apply input and output controls to fields
- Field Map tab
- Math expressions defined to input fields currently mapped to the output fields
- Extended Properties tab
- Output control attribute operations
- Filter from a Components window
- Browse button
- Refresh function added to menu bar, pop-up menu, and toolbar
- Remove function added to menu bar and pop-up menu
- Validation enhanced to limit the length of user-defined names to 32 characters and to ensure single and double quotes are not mismatched

- Filtered formats can be saved as a group (this is reflected in the Browse tab)

- Find option

- Euro support

Chapter 5: Visual Tester on page 195 is new and provides information on testing queues, formats, and rules.

Chapter 5: Visual Tester on page 195 has been updated to include new functionality. The following summarizes the new information:

- DB2 database support added

- Recache button is now on the Test Format, Test Reformat, and Test Rule tabs

- User can view full expressions when testing a rule

Supported Platforms and Compilers

Operating System	DBMS	Compiler
AIX 4.2, 4.3	DB2 5.0 DB2 5.2 Oracle 7.3.4 Oracle 8.0.5 Sybase 11.5 Sybase 11.9	IBM C Set ++ version 3 or later
HP-UX 10.20	DB2 5.0 DB2 5.2 Oracle 7.3.4 Oracle 8.0.5 Sybase 11.5 Sybase 11.9	HP C++ version 10.40 (HP-UX 10.20)
Solaris 2.5.1, 2.6	DB2 5.0 DB2 5.2 Oracle 7.3.4 Oracle 8.0.5 Sybase 11.5 Sybase 11.9	Sparcworks C++ compiler version 4.2
Windows NT 4.0	DB2 5.0 DB2 5.2 Oracle 7.3.4 Oracle 8.0.5 SQL Server 6.5 Sybase 11.5 Sybase 11.9	Microsoft Visual C++ version 6.0

Disk Space and Memory Requirements

Required disk space is dependent on the number of queues, formats, and rules. Recommended memory for satisfactory performance depends on message rates, message sizes, and application-specific factors. For Windows NT/SQLServer, the recommended memory is 128 MB; for other platforms, the recommended memory is 256 MB.

MQSeries Integrator Disk Space Requirements

For Solaris, the /var/tmp file system requires at least 250 MB of free space to unpack the MQSeries and MQSeries Integrator products.

The minimum database allocation requires 20 MB.

MQSeries Integrator binaries require 150 MB.

MQSeries base code and server require a minimum of 25-30 MB of disk space to be available for the product code and data.

MQSeries documentation requires 50 MB of disk space (HTML files — 35 MB, PDF files — 15 MB).

The GUI requires 40 MB.

Year 2000 Readiness Disclosure

MQSeries Integrator, when used in accordance with its associated documentation, is capable of correctly processing, providing, and/or receiving date information within and between the twentieth and twenty-first centuries, provided that all products (for example, hardware, software, and firmware) used with this IBM program properly exchange accurate date information with it.

Customers should contact third party owners or vendors regarding the readiness status of their products.

IBM reserves the right to update the information shown here. For the latest information regarding levels of supported software, refer to: http://www.software.ibm.com/ts/mqseries/platforms/supported.html.

For the latest IBM statement regarding Year 2000 readiness, refer to: http://www.ibm.com/ibm/year2000/.

Chapter 2

MQSeries Integrator Overview

MQSeries Integrator provides the flexibility and scalability that allows true application integration. MQSeries Integrator has four components:

- MQSeries

- NEONFormatter

- NEONRules

- MQSeries Integrator Rules daemon

MQSeries

MQSeries is message-oriented middleware that is ideal for high-value message handling and high-volume applications because it guarantees that each message is delivered only once, and it supports transactional messaging. Messages are grouped into units of work and either all or none of the messages in a unit of work are processed. MQSeries coordinates message work with other transaction work, like database updates, so data integrity is always maintained.

Formatter

NEONFormatter translates messages from one format to another. It can handle multiple message format types from multiple data value sources and has the ability to convert and parse messages. Messages can be converted from any described format to any other described format. If input data formats are missing, you can set up defaults for those fields on output. When a message is provided as input to Formatter, the message is parsed and data values are returned. Formatter can handle virtually any message format, including the following:

- Fixed, for example, COBOL records

- Delimited, for example, C null delimited strings

- Variable, tagged, delimited, repetitive and recursive formats, for example, S.W.I.F.T. messages

Defining message formats in the Formatter database is done through the MQIntegrator Graphical User Interface (GUI). The GUI leads you through the definitions of format components, such as tags, delimiters, and patterns, to the building of complete message definitions.

For a more in-depth description of NEONFormatter, refer to *Formatter* on page 13.

Rules

Using NEONRules, you can create rules for managing message destination IDs, receiver locations, expected message formats, and any processes initiated upon message delivery. The creation and dispatch of multiple messages to multiple destinations from a single input message is supported, and different formats and transport methods for each are allowed. The dynamic nature of the Rules daemon means that rules can be effective immediately, staged over time, or delayed, depending on how the reload messages are timed. This allows flexibility in rapidly changing environments.

Rules can examine the value of any field or group of fields in a message to make its determinations. It can aggregate conditions with the Boolean AND and OR operators without architectural limits as to the number or complexity of the expressions.

For a more in-depth description of Rules, refer to *Rules* on page 137.

MQSeries Integrator Rules Daemon

The MQSeries Integrator Rules daemon combines MQSeries, NEONFormatter, and NEONRules in a generic server process. The MQSeries Integrator Rules daemon processes messages from an MQSeries input queue, uses NEONFormatter to parse messages, uses NEONRules to determine what transformations to perform and where to route the messages, and then puts the output messages on MQSeries queues for delivery to applications.

Chapter 3
Formatter

NEON Formatter translates messages from one format to another, simplifying message exchange between different systems and applications. NEON Formatter standardizes the description of message formats, eliminating the need to custom code between applications to translate each other's messages.

NEON Formatter breaks complex messages into simple messages for processing, handling anything from formats containing fixed, tagged, or delimited fields to multi-part nested formats.

Through NEON Formatter's use of tables, you only have to describe a message format once. The components of the format can be reused in multiple formats. After that, conversion between formats is dynamic. The NEON Formatter GUI simplifies format definition and configuration.

NEON Formatter can reformat messages at different points in the communication process. For example, a message can be reformatted upon receipt or when it is sent to its next destination. The exact points vary based on your configuration.

Control Tables

NEON Formatter uses control tables within the MQSeries Integrator database to recognize and parse input and output messages. These tables describe message formats and format components. NEON Formatter uses this information to interpret values from incoming messages and dynamically construct outgoing messages.

NEON Formatter can:

- Interpret incoming messages
- Add, remove, or rearrange message data
- Insert literals into output as headers or trailers

- Perform arithmetic operations to format numeric data

- Execute user-written functions to format fields

Operating Modes

NEONFormatter operates in two modes --parsing and converting.

- When parsing a message, NEONFormatter breaks the message into its component parts for easy access to message data.

- When converting messages, NEONFormatter parses the input message and formats the output message data based on a different output format.

Euro Support

NEONFormatter supports the euro currency symbol. To display the symbol in the tree and on a tab, download either Service Pack 4 for Microsoft Windows NT 4.0, or the euro support update from http://www.microsoft.com. You must also change the system font to MS Sans Serif.

The euro symbol replaces the ASCII 0128 character or HEX 80 character.

Starting Formatter

To access NEONFormatter:

1. Double-click the Formatter icon. The Formatter Logon dialog box appears.
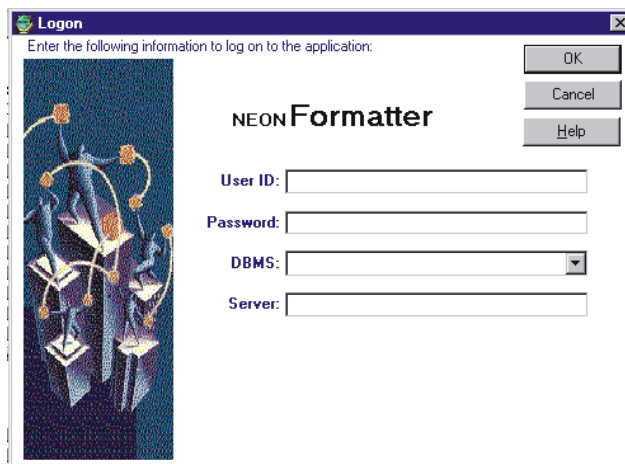
NEONFormatter Logon Dialog Box

2. If all defaults are correct, click the **Password** field, type your password, and click OK.

   If the defaults are not correct, type your user ID in the **User ID** field and press TAB.

3. Type the password associated with the **User ID** and press TAB.

4. Select a database type from the DBMS drop-down list. The server, database, and qualifier fields change based on your selection. Press TAB.

Note:
The DBMS client appropriate for your DBMS must be installed before you can run the GUI. If you do not know the DBMS to connect to, ask your System Administrator.

5. For ODBC DBMS, type the driver name; otherwise, type the server name if the field is blank or you want to overwrite the default. Press TAB.

6. Based on the database type entered in the DBMS field, enter the following in the Database or Qualifier field:

   – ODBC DB2: leave the Qualifier field blank.

   – Sybase and Microsoft SQL: type the database name.

   – ORACLE does not require a database name.

Note:
The Qualifier field is displayed for DB2 on UNIX and AIX. Leave this field blank because the system does not recognize entries in the Qualifier field for these two platforms.

7. Click OK. The NEONFormatter main window appears.

Main Formatter Window

The main NEONFormatter window works similar to Microsoft Explorer in Microsoft Windows NT and Windows 95.
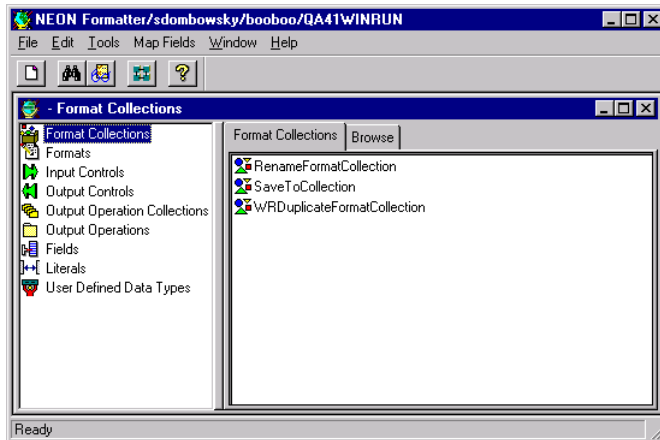
*NEON**Formatter Window***

Format components are displayed in a tree-structured or hierarchical organization in the left pane. The right pane shows the contents of the currently selected root object.

When you select a component in the left pane, the right pane of the window then displays detailed information about the selected component. The detailed information is kept in property sheets labeled with tabs. You can see the properties of a component and change components using property sheets.

To expand a root object, for example Formats, double-click it. The small boxes with a plus sign (+) next to the objects in the left pane indicate that there might be other items subordinate to the object, but the subordinate items are not displayed. Click the plus sign to expand the tree to display the

subordinate items. When the tree is expanded, the plus sign changes to a minus sign (-). To collapse the tree, click the minus sign.
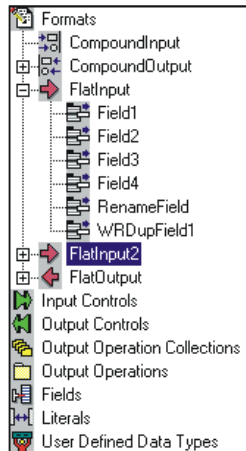
Expanded Tree

When you select a component in the left pane, the right pane of the window then displays detailed information about the selected component. The detailed information is kept in property sheets labeled with tabs. You can see the properties of a component and change components using property sheets.

Opening a New Formatter Window

You can open more than one NEONFormatter window at a time. To open a new window, select File→New from the menu bar.

When you open a new window, you do not create a new instance of the database. You are always connected to the same database.

Tip:

To Simplify Creating Formatter Components:

- When building Input Controls, have two windows available. One window should be open to the new Input Control and the other should be open to Fields.

- When building Output Controls, have four windows available. One for Fields, one for Output Operations, one for Output Operation Collections, and one for the new Output Control.

- When building Flat Input Formats, have two windows available. One window for the new Format and one for Input Controls to be associated with input Fields.

- When building Flat Output Formats, have two windows available. One window for the new Format and one for Output Controls to be associated with output Fields.

Tree Functionality

Depending on the component or element selected, when you click the right mouse button, a pop-up menu appears containing commands such as New, Delete, Remove, Duplicate, and Add Field Components. For more information on pop-up menus, refer to the section See *Formatter Pop-Up Menus* on page 29.

Drag and Drop

Use the drag and drop feature to:

- Associate subordinate components from one window into the related Collection tab of another window.

- Associate input fields with output fields from the Field Map tab of a flat output format.

- Reorder fields (in flat input and output formats).

- Reorder formats (in compound input and output formats).

- Map fields in the Field Mapping window.

- Assign either a single or multiple output operation to an output control from the Output Operation Extended Properties tab.

- Reorder the sequence of operations from the Output Operation Collections tab.

Drag and drop works across both windows and panes. To copy an object using drag and drop, drag the object from the first window's tree to the second window's tabs. You cannot drag and drop in the same pane unless you are reordering fields or formats or using field mapping.

You can only drag and drop at the root level. When you drop an object, it is always added to the end of the list. You cannot drag and drop literals.

When a flat input or output Format or output operation collection is changed (by either adding or deleting a field), it is a global change. Any compound format the flat format is contained in also reflects the addition or deletion of the field.

Note that drag and drop cannot be used with literals, output operation collections, and user-defined types.

Notes:

For more information on reordering fields, refer to the *Using Drag and Drop to Reorder Fields* section on page 21.

For more information on field mapping, refer to the *Field Mapping* section on page 26.

Assigning a Field to an Input or Output Control that was Dragged and Dropped

You can drag and drop to add either an input or output control to a flat input or output format. This procedure describes dragging and dropping an input control. You would use the same procedure to drag and drop an output control. The field for the input or output control component that you drop into the format is named NONE.

To name the component:

1. Close and reopen the format and select NONE. The property sheet for the field appears.

Input Field's Property Sheet

2. Click in the **Field Name** field to open the drop-down list and select the field.

3. Either drag the input control and drop it on the format or select it from the **Input Control Name** drop-down list.

Using Drag and Drop to Field Map

When you are creating flat output formats using the field mapping tab, you can drag and drop an input field into an output field. For more information on field mapping, refer to on *Map Fields* on page 26.

Using Drag and Drop to Reorder Fields

From the Fields tab within a flat input or flat output format, you can use drag and drop to change the order of the fields (in the right pane). When you drop an object from another pane, it is always added to the end of the list. If field order is ordinal (you want the fields to appear in the specified order), you can change the order in which they appear in the Field list to map to an existing format from either a sending or receiving application.

To change the order of the field names, drag and drop the field that you want on top of the field above which you want it to appear above. The field is now repositioned in the list.

To reposition a field in a list longer than the pane, in the right pane, right-click and select View→Details from the pop-up menu to rearrange the fields into columns. You can then drag and drop the field you want to reposition and drag to the top of the field you want it to precede.

Case Sensitivity

WARNING!

If you are using a case-insensitive database, you cannot differentiate between component names with only a change in case. For example, you cannot name one format f1 and another format F1. In a case-insensitive environment, make each item unique using something other than case differences.

If importing components into a case-insensitive database that were exported from a case-sensitive database, these differences cause NNFie to fail during import if a conflict arises when two components are named the same with only case differences. See ***MQSeries Integration System Management Guide*** for information on using NNFie.

See ***MQSeries Integration System Management Guide*** for information on changing a case-insensitive installation to case sensitive.

Formatter Menus

NEONFormatter features both a menu bar and pop-up menus. Use the menu to open, close, and arrange windows, refresh tree items, display and move the toolbar, exit NEONFormatter, and view NEONFormatter version information.

Use the pop-up menu to create, delete, remove, and duplicate format components. See *Formatter Pop-Up Menus* on page 29 for more information.

NEONFormatter menus can also be accessed by pressing ALT plus the letter underlined in the menu name. For example, you can open the File menu by pressing ALT+F. After the menu is opened, you can access its functions by pressing the underlined letter of the function. For example, from the File menu, to open a new Formats window, type N (the underlined letter in the function, New).

The Formatter Menu Bar

NEONFormatter menu bar options are discussed in the following sections:

Note:

If only the NEONFormatter window is open, only the File and Help menus appear on the menu bar.

File

File menu options are described in the following sections.

New

Select File→New to open a new NEONFormatter window, which connects to the same database as the previous window.

Close

Select File→Close to close the active window.

Print Setup

Select File→Print to open the Printer Setup dialog box where you can view or change your printer setup, choose File→Print Setup.

Exit

To exit NEONFormatter, select File→Exit.

Edit

Edit menu options are described in detail in the following sections.

Cut, Copy, Paste

Select Cut, Copy, or Paste to cut text in fields or copy text from one field and paste it into another. For example, you could copy the Tag Value field data from the Input Control property sheet and paste it in the Input Tag Value field of the Output Control property sheet.

Clear

Select Clear to delete the selected text from a user-entry field.

Find

Select Find to search for format, input control, output control, field, literal names, output operations, and output operation collections.

To perform a search

1. Press CTRL+F or select Edit→Find. The Find dialog box appears.
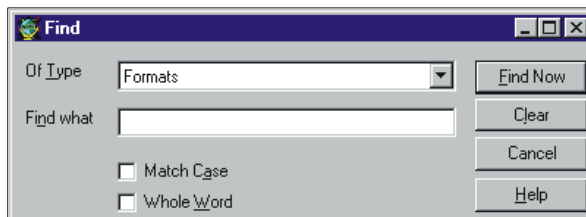
Find Dialog Box

2. Select the item type from the **Of Type** drop-down list.

3. Click the **Find What** field and type either the name or partial name on which you want to search.

Note:
An asterisk can be used as a wildcard in your search.

4. To match the case of the name in the **Find What** field, select the Match Case checkbox.

Note:
If you are using a case-insensitive database, this option might not work, depending on how your data is named.

5. To search for all names that exactly match the name in the **Find What** field, select the **Whole Word** checkbox.

   If you select both **Match Case** and **Whole Name**, ᴺᴱᴼᴺFormatter searches on the literal value of the Find What field.

6. To begin the search, click **Find Now**. The results are displayed at the bottom of the dialog box.
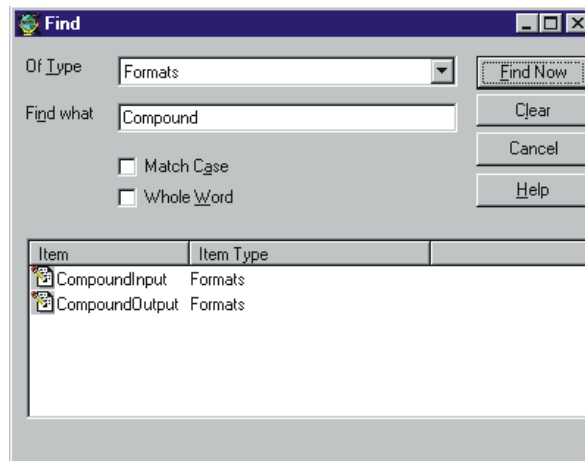
Find Dialog Box with the Search Results Displayed

7. To go to one of the items (in the example window above, a format) found by the search, double-click the name in the Item column. A new window displays the item's placement in the tree. The item is highlighted.

8. To clear the fields to conduct another search, click Clear.

9. To close the dialog box, click Cancel.

Tools

Tools menu options are described in the following sections.

Refresh Item

Select **Refresh Item** to update the active tree item in all open windows. Only the highlighted item and its children are updated.

As you work with the NEONFormatter GUI, items previously retrieved into the data stores can be changed or deleted by other users. Assigning an item that has been changed or deleted causes unpredictable results within associated components or within the GUI itself. Refresh Item gives you current information.

Customize Toolbars

To turn the toolbar either off or on, position it in your window and select the button size.

Map Fields

Map Fields submenus are described in detail in the following sections.

Open Mapped Fields Window

Open Mapped Fields Window opens the Field Mapping window. If you have nothing selected in the Formatter tree, the Field Mapping window opens displaying no formats. If you select either an input or output format in the Formatter tree, the Field Mapping window opens with the same format selected. You can then map input and output formats. For more information, see *Using the Field Mapping Window* on page 120.

View Mapped Fields

The View Mapped Fields functions enables displays the fields that are mapped to other fields.

View Format Types/Access Modes

Use view Format Types/Access Modes function to view the format type and access mode for each output field. This information is displayed next to the format and field names in the Field Mapping window tree views.

View Optional/Repeating

Use the View Optional/Repeating function to see if a format or field is optional and/or repeating. This information is displayed next to the formats and fields in the Field Mapping window tree views.

View Input/Output Control Types

Use View Input/Output Control Types to view input and output control types for selected formats. This information is displayed next to the fields in the Field Mapping window tree views.

Print

Use the Print option to print only the Input Tree, Output Tree, Mapped Fields, or Unmapped Fields. To print the contents of the entire window, select All Details.

Window

Use the Window menu option to cascade, tile either vertically or horizontally, or layer windows. You can also use this menu to view a list of the open windows of the application.

Help

The Help menu contains Help Topics and About submenus.

Select Help Topics menu to access NEONFormatter online help. For online information on how to build formats, select the Help Topics icon and then click the How Do I.... button. To use context-sensitive help, select the item you want to view help and press F1.

Select the About option to display copyright and version information, in addition to database, DBMS, and server information.

Formatter Toolbar

Use the NEONFormatter toolbar icons to execute the functions shown in the following table.

Toolbar Icon	Description
	Opens a new NEONFormatter window. The new window connects to the database as used by the previous window.
	Searches for format, input control, output control, field, literal names, output operations, and output operation collections. For more information, See *Find* on page 24
	Updates the display in an active window or all open windows. For more information, See *Refresh Item* on page 26
	Opens the Field Mapping window. For more information, see *Using the Field Mapping Window* on page 120.
	Opens the online Help Topics. For more information, See *Help* on page 27

Formatter Pop-Up Menus

Both the right and left window panes contain pop-up menus. These menus are accessed by right-clicking the mouse.

Left Pane Formatter Pop-Up Menus

Depending on what is selected in the left window pane, a pop-up menu appears when you press the right mouse button. The pop-up might contain any of the functions described in the following table.

Menu Item	Description	Associated Components
New	Creates a new component of the component category type selected. The New function operates on the root level of the associated components.	Formats*, Format Collections, Input Controls, Output Controls, Output Operation Collections, Output Operations, Fields, Literals, or User-Defined Data Types. An output operation is one of the following: Default, Length, Math Expressions, Prefix/Suffix, Substitute, Substring, Trim, User Exit. *Selecting New→Formats displays two submenus. From the first option, select either Compound or Flat, depending on which type of format you want to construct. From the second submenu, choose either Input or Output.
Delete	Deletes the selected component at the parent level.	Formats, Input Controls, Output Controls, Output Operation Collections, Output Operations, Fields, Literals, User-Defined Data Types

Menu Item	Description	Associated Components
Remove	Removes the selected item's association with its parent. The item (child) itself is not deleted.	Formats, Fields, Output Operation Collections
Duplicate	Duplicates the selected component at the parent level and enables you to change the name of the duplicate.	Formats, Input Controls, Output Controls, Output Operation Collections, Output Operations, Fields, Literals
Save as Output	Saves an input control as an output control or an input format as an output format. See *Saving an Input Control as an Output Control* on page 64, *Saving a Flat Input Format as a Flat Output Format* on page 102, and *Saving a Compound Input Format as a Compound Output Format* on page 108 for details.	Input Controls, Input Formats
Save as Input	Saves an output control as an input control or an output format as an input format. See *Saving a Flat Input Format as a Flat Output Format* on page 102, *Saving a Compound Output Format as a Compound Input Format* on page 116, and *Saving an Output Control as an Input Control* on page 87 for details.	Output Controls, Output Formats

Menu Item	Description	Associated Components
Apply Control Apply Input Control Apply Output Control	Opens the Apply Control window in which you can globally change the input/output control name for multiple fields within a format (or formats). These controls are also available from the Components tab for a format and the Formats tab for all formats by right-clicking one or more components in the tree view.	Input Controls, Output Controls, Fields, Formats
Map Fields	Opens the Field Mapping window in which you can add, delete, and modify field mapping for output fields. See *Using the Field Mapping Window* on page 120 for details.	Input Format, Output Format, Format (root), Format Collection
Add Field Components	Opens the Field window for an easy way to add fields to a new or existing flat input or output format. Select the fields you want to add. Choose Accept Selection. **Note**: Fields are always added in alphabetical order to the end of any existing list of fields for the format.	Flat Input or Output Formats

Menu Item	Description	Associated Components
Add Output Operations	Opens the Output Operations window in which you add output operations or other collections to a new or existing collection. Select the operations you want to add. Click Accept Selection to apply your own changes. **Note**: Operations and other collections are always added in alphabetical order to the end of any existing list of operations for the collection.	Output Operation Collections
Add Substitute Items	Opens the Literal window in which you can add literals as values to be substituted. Select the literals you want to add and click Accept Selection to apply your changes. See *Substitute* on page 95 **Note**: Literals are added in alphabetical order to the end of any existing list of literals for the substitute.	Substitute Operations
Expand Item	Displays all items subordinate to the associated component. Expand Item is associated with components on both the root and parent levels. **Note**: If you are expanding all formats of a large database, this might take a few minutes.	Formats, Format Collections, Input Controls, Output Controls, Output Operation Collections, Output Operations, Fields, Literals, and User-Defined Data Types. Includes the following items from Output Operations: Default, Length, Math Expressions, Prefix/Suffix, Substitute, Substring, Trim, and User Exit.

Menu Item	Description	Associated Components
Collapse Item	Hides all items that are subordinate to the associated component.	Formats, Format Collections, Input Controls, Output Controls, Output Operation Collections, Output Operations, Fields, Literals, and User-Defined Data Types. Includes the following items from Output Operations: Default, Length, Math Expressions, Prefix/Suffix, Substitute, Substring, Trim, and User Exit.
Open Item	Opens a new Formatter window with the item selected that was also selected in the original Formatter window.	Formats, Fields within a Format, Operations, Literals, Output Operation Collections (Substitute Operations from tree), Input Controls (Literal, Tag Value), Output Controls (Output Operations (Default, Default Value, Length, Pad Character), Input Tag Value, Field), Substitute Item, Includes the following items from Prefix/Suffix- Substitutes (New Substitute Name, Substitute Items, Input Value, Output Value, Substring (Pad Characters), Trim (Trim Value)

Right Pane Pop-Up Menu

The right window pane contains a pop-up menu when a component list is displayed. Depending on what is selected, the pop-up menu contains the options discussed in the following sections.

View

Select View options to view either large or small (the default) icons. You can also arrange the icons in a list (the default).

Large icons are displayed from left to right, rather than in a list, and the title of the property sheet is no longer displayed. To see the icons and display the property sheet's name, select the Details option.

Arrange Icons

Select the Arrange Icons option to reorder the icons. By default, the icons are alphabetically displayed from a-z. When you select Arrange Icons→by (root-level name), the icons are arranged alphabetically, from z-a. You can switch between these two alphabetic displays by reselecting Arrange Icons→by (root-level name).

Open Item

Use Open Item to open a new window from within a Properties tab or tree. The new window displays the information for the selected component in the tab. You can change the information in this window.

The Open Item pop-up option is available when you right-click on a user-defined type, literal, operation, operation collection, field, input control, output control, substitute item, or field property in a format on the Properties tab.

To find information about a component:

1. In a NEONFormatter properties tab, right-click the field of the item you want to see.

2. Select Open Item. A new NEONFormatter window displays information for the selected item.

3. Either view or change the information in the window.

Apply Control

After you select either one or multiple fields or formats, you can select the Apply Control option to open the Apply Control window. From this window, you can select an input or output control and apply that control to the selected field(s) or format(s).

Formatter Features

Used In

The Used In tab enables you to displays a list of places the selected literal, input control, output control, field, format, operation, operation collection, or user-defined data type is used.

To view a component's Used In properties:

1. Open the property sheet for a component.

2. Select the Used In tab.

Used In Tab

Note:

A literal can be used in many different ways, for example, as a pad character, suffix, or prefix. Therefore, the Used In list for literals can be extensive.

Tip:

Double-click an item in the Used In list to open a new NEONFormatter window with that item selected. You can make changes in this window.

Browse

The Browse feature can be used only when a either a Format or Format Collection or one of its components is selected. When the root-level Formats or Format Collections are selected, all formats appear collapsed in the Browse tab. When a format or one of its components is selected in the left pane, the Browse tab displays the format completely expanded. You can expand and collapse in the Browse tab.

To save the format(s) displayed in the Browse tab to a file, click Print. The data is saved to a text file that you name.

To print the format(s) displayed in the Browse tab, click Print. The data is printed to your default printer.

Filter

Use Filter function to view specific formats. When you select the Select Filter tab, a list of all formats displays. From this list, you can select the formats that you want to view. If a compound format is selected, all its sub-formats are displayed. When the list of formats is filtered, the root object is displayed as Formats (Filtered). Note that the Collection and Export tabs only display filtered formats.

To filter formats:

1. Select Formats from the tree.

2. Select the Set Filter tab. A list of all formats displays. All available formats are highlighted in the treeview.

   If you are adding fields to a flat format, the Format Filter drop-down displays a list of both input and output formats. You can limit the fields displayed to only those in the selected format.

   If you are adding components to a compound format, the Format Filter drop-down lists input formats if you are adding to an input

compound format and output formats if you are adding to an output compound format.

The list of formats is limited to those appearing in the selected format. To display all fields in the database, select Not Applicable.

3. Select the formats that you want to see (filter) and click Set Filter.

4. To see only the filtered formats, collapse and then expand Formats in the left pane.

To add to the list of filtered formats:

1. Select any tab other than the Set Filter tab, then select the Set Filter tab. The complete list of formats is displayed. The filtered formats currently available are highlighted in the tree view.

2. Press *CTRL* and select the formats you want to add to the filter.

3. Click Set Filter.

4. To see the filtered formats, collapse and then expand Formats in the left pane.

To reset the filter:

1. Select any tab other than the Set Filter tab, then select the Set Filter tab. The complete list of formats is displayed. The filtered formats currently available are highlighted in the tree view.

2. Click Remove Filter.

3. To see the complete list of formats, collapse, then expand Formats in the left pane.

WARNING!
Active filters affect all open windows and panes.

Format Collections

You can group formats together and save them as a unique collection name. You can add formats to a collection from either the Add Member Formats window or Set Filter tab.

To create a collection (from the tree view):

1. Select the Format Collection in the tree view.

2. In the left pane, right-click on Format Collections.

3. From the pop-up menu, select New and type a unique collection name.

Note:
The Format Collection name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

---

4. In the left pane, right-click the collection and select Add Member Formats.

5. Select the formats that you want in the collection.

To either add a new collection or add to an existing collection (using the Set Filter tab):

1. Select Formats.

2. Select the Set Filter tab. A list of all formats is displayed. Each format currently available in the tree view is highlighted.

3. Select the formats that you want to see (filter).

4. Select the collection that you want to add the formats to. If the collection is not listed, the item entered can be the collection name.

5. Click Save to Collection.

To remove a format from a collection (from the tree view):

1. Select the format that you want to remove.

2. Right-click and select Remove. A confirmation box appears.

3. Click Yes. The selected formats are removed from the collection.

To delete a collection (from the treeview):

1. Select the collection that you want to delete.

2. Right-click and select Delete. A confirmation box appears.

3. Click Yes. The collection is deleted.

Export

Use the Export function to export one or more format objects. You can only export one type at a time. For example, you could export input controls and then separately export the delimiters of a format. Export is available from each root object, excluding Output Operation Collections.

The components of a format can be exported in any order. To export a complete format, be sure to export any contained Literals, Output Operations, Output Operation Collections, User-Defined Data Types, Fields, Input Controls, Output Controls, Flat Input or Output Formats, and Input or Output Compound Formats.

To export components:

1. Open with the property sheet for the component you want to export and select the Export tab. (The example below displays literals; all components have a similar display and are exported in the same way.)
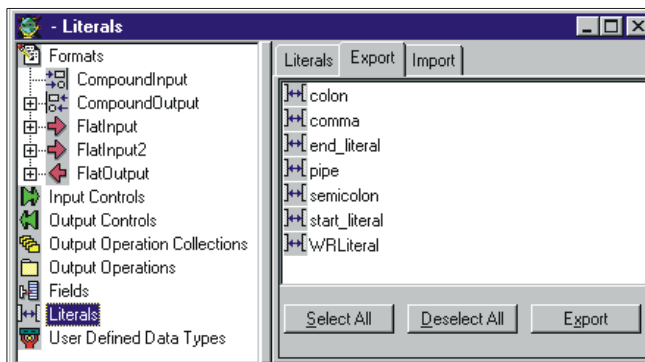
Export Dialog Box

2. Select one of the following:

    ■   Click each component that you want to export.

    ■   To select all available components, click Select All.

    ■   To deselect previously selected components, click Deselect All.

3. Click Export. The Export File dialog box appears.
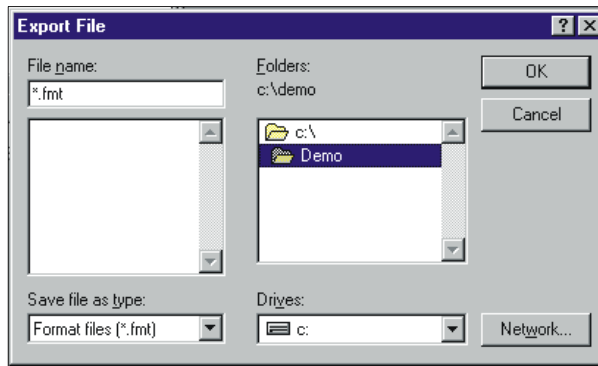
Export File Dialog Box

4. Select the drive and directory that you want to export the components to and click OK. The selected components are exported and the dialog box closes.

WARNING!

The export files created from this GUI cannot be used interchangeably with NNFie export files, which are files exported from the command line and used to migrate complete formats from a database.

Import

***

WARNING!
If you are importing components into a case-insensitive database that were exported from a case-sensitive database, the differences cause the import to fail if a conflict arises when two components are named the same with only case differences.

***

Use the Import function to import components that were previously exported from another database to the current database. When you use the GUI export function to import a complete format, you must import each component separately. The Import function is available from each root object.

You must import the compound components within a compound before importing the compound itself. The compound component A12 within the compound B1 must be defined before B1 can be imported. Also, import flat formats before you import compound formats.

To import a complete format, import the components in the following order:

1. Literals
2. Output Operations
3. Output Operation Collections
4. User-Defined Data Types
5. Fields
6. Input (parse) Controls
7. Output Format Controls
8. Flat Input Formats
9. Flat Output Formats
10. Compound Input Formats
11. Compound Output Formats

***

WARNING!
The import files created from this GUI cannot be used interchangeably with NNFie export files, which are files exported from the command line and used to migrate complete formats from a database.

***

To import formats:

1. Open the property sheet open for the component or format group that you want to import and select the Import tab.

Note:

Import file contents must match the selected area in the tree view. For example, you can only import literals if Literals is selected in the tree view.
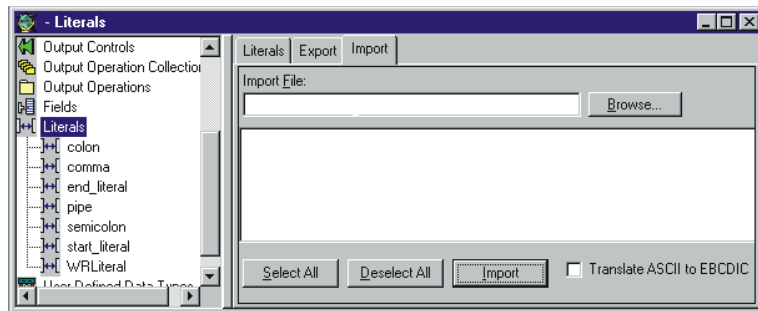
Import Property Sheet Window

2. Click Browse. The Import File dialog box appears.
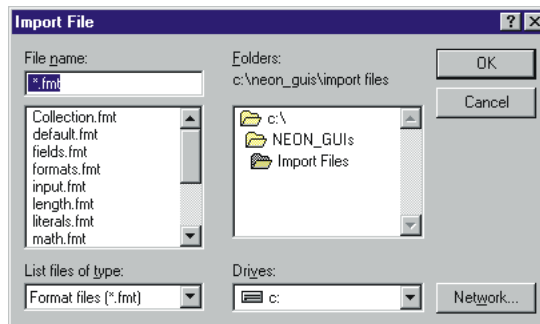
Import File Dialog Box

3. Select the drive and directory. Select the file that to import and click OK. The Import File window closes and the Literal window reappears.

4. Choose one of the following:

   - Click each component you want to import.

   - To select all components, click Select All.

   - To deselect previously selected components, choose the Deselect All button.

5. For DB2, select the Translate ASCII to EBCDIC checkbox if you are importing literals into a platform that uses EBCDIC and the import file was exported in ASCII. The ASCII data will then be automatically converted to EBCDIC.

6. To import the components, click Import.

Building Formats

The easiest way to build formats is to first create the root component, starting with the lowest level literals, input controls, output controls, fields, and formats. The root elements are created same way.

To create a flat input format, first define:

   - Literals (for use as delimiters, tags)

   - Fields

   - Input Controls

   - Flat Input Format

To create a flat output format, first define:

   - Literals

   - Fields

   - Output Operations and/or Output Operation Collections

Note:

Output Operations and Output Operation Collections can also be created from the Output Control window and the Extended Properties tab.

- Output Controls

- Flat Output Format

The following sections explain each root component and describe the procedures to create them.

Literals

A literal is a string of characters that NEONFormatter processes as:

- delimiters

- format terminators

- repeating sequence terminators

- default values

- trim characters

- prefixes and suffixes

- pad characters

- substitute values

- comparison values

- tag values

A literal cannot be restricted for use. A single literal can be used by all of the above.

Notes:

- Literal names must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the literal or cancel the operation.
- When literals are used as pad characters, only the first character is used.

NEONFormatter uses literals in input controls, output controls, and operations, as format terminators, and repeating sequence terminators.

Creating a Literal

Valid literal characters are A-Z, a-z, and 0-9 and any keyboard character available by pressing shift with another printable character. Do not use single quotes in literal names.

1. In the left pane of the Formatter window, select the Literals component category and right-click.

2. Select New from the pop up menu. A new literal component is added to the left pane and the Literal property sheet is displayed in the right pane. Type a new, unique name in the text box.

   Notes:

   The literal name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

   Single quotes, double quotes, and spaces should not be used in names. However, if using quotes, make sure they are not mismatched (for example, a single quote with a double quote), or a message box appears and you must edit the name to reflect either matching quotes or no quotes.

3. Click any mouse button outside the name text box or press ENTER to finish defining the literal. The name is highlighted and moves to its alphabetical location in the list.

4. In the Property tab, enter the value for the literal. Both the ASCII and Hex values are automatically populated in when you type either the ASCII or Hex value. Note that the ASCII and EBCDIC fields have a 127-character maximum (which equals a 254-character Hex value).

> **Note:**
> Refer to Appendix A for ASCII extended character set information.
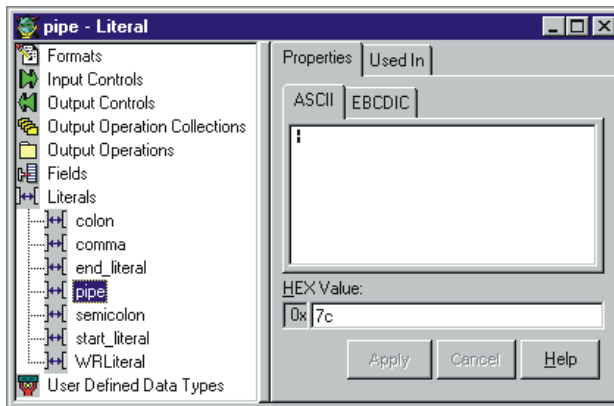
Literal Property Sheet

> **Note:**
> You cannot drag and drop literals.

Fields

Fields are named items of message data and are the smallest possible container for information. In NEONFormatter, the field name is the link between the input and output data. Each field can be used in multiple formats and associated with different input and output controls. To identify a field in a message, a combination of the field name and a control is used.

Note:

Enter a unique name for each format component. Component names can be up to 32 characters.

Creating a Field

1. In the left pane of the Formatter window, select the Fields component category and right-click.

2. Right click and select New. A new field component is added to the left pane and the fields property sheet is displayed in the right pane. The cursor is positioned in the text box where you must type a new, unique field name.

Notes:

The field name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

Names with quotes may only have either single or double quotes, not both, or a message box appears and you must edit the name.

Field names with spaces or underscores must be surrounded by single or double quotes. A field with the name Field_1 could be used as "Field_1" or 'Field_1' (but not "Field_1' or 'Field_").
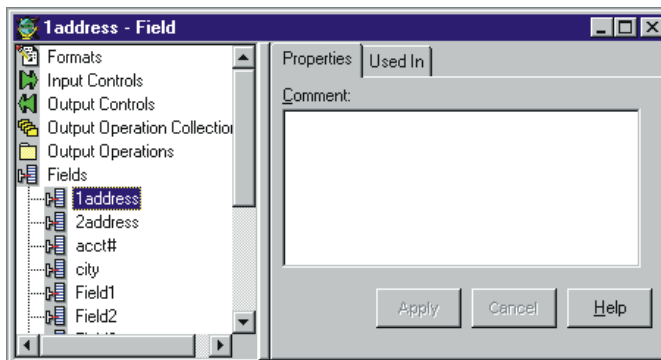
Fields Property Sheet

3. Click any mouse button outside the name text box or press ENTER to finish defining the field. The field name moves to its alphabetical location in the list and is highlighted.

4. To add a descriptive comment for the field, click the Comment field of the property sheet and type the comment.

Note:

The Comment field does not retain formatted text when either imported or exported. Using carriage returns in the Comment field is not recommended.

Tip:

You can change the name of a field by double-clicking on the field name. Type the new name and press ENTER to enter the name and position it in the list.

Input Controls

To build input formats, you must build input controls. Input controls are used to parse input data. NEONFormatter uses input controls to determine how to find the beginning and end of the data in a field. Input controls have various ways of looking for data including: tag search, fixed length positional, delimiters, and embedded length fields.

An input control also determines whether the data for a field is mandatory or optional. Mandatory means that NEONFormatter considers the parse successful if the parse start and end strings are found, even if the associated data is empty. If the parse is successful, the field passes the mandatory test and the parse continues. If the parse parameters are not found, the field fails the mandatory test and the parse fails for the rest of the format. Optional means that the parse continues even if parse parameters for the field are not found.

Input controls can use tags to separate data. Tags are sets of bits or characters explicitly defining a string of data. For example, <NAME> and </NAME> could mark the beginning and end of a name field in a message.

After an input control is created, it can be saved as an output control. For more information, refer to *Saving an Input Control as an Output Control* page 64.

Creating an Input Control

To create an input control:

1. In the left pane of the Formatter window, select the Input Controls component category and right-click.

2. Select New. A new input control component is added to the left pane and the input control property sheet is displayed in the right pane.

3. Type the new, unique input control name in the text box. Input control names should be descriptive.

Notes:

- The input control name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

- Single quotes, double quotes, and spaces should not be used in names. However, if using quotes, make sure they are not mismatched, i.e. a single quote with a double quote, or a message box appears and you must edit the name to either matching quotes or no quotes.

4. Click any mouse button outside the name text box or press ENTER to finish defining the input control. The new input control moves to its alphabetical location in the list and becomes highlighted.
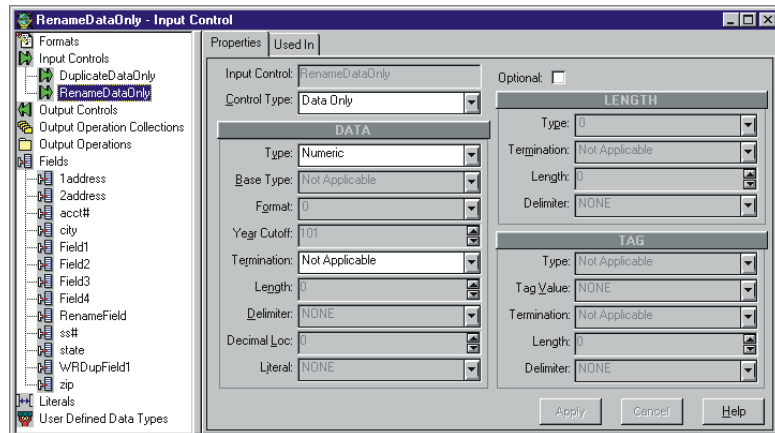
Input Control Dialog Box

5. On the property sheet, click the Input Control Type drop-down list and select an input control type. Input Control Types describe the type of field parsed by Formatter.

Input Control types are described in the following table:

Input Control Types

Value	Description
Data Only	Field has a data component only. Data is the value of the field.
Tag & Data	Field has a tag and data component (in this order). A tag label or identifier for the data.
Tag, Length & Data	Field has a tag, length, and data component (in this order) that indicates the length in bytes of the data portion of the field.
Length & Data	Field has a length and data component (in this order).
Repetition Count	Field contains the count of a repeating component.
Literal	Field value is a literal.
Length, Tag & Data	Field has a length, tag and data component (in this order).
Regular Expression	Compares input data to the value of the literal parse control using pattern matching. A Regular Expression requires an associated string data type and a literal value.

Note:

The Input Control Type determines which sections and fields require information. A section not requiring data is disabled. For example, if the Control Type = Data Only, then only the Data section of the tab is enabled. Also, you do not have to enter a value in every field of an enabled section. A field containing Not Applicable does not require a value.

Data Section

The Data section is almost always enabled; however, the data type selected in this section determines which fields are enabled. All fields are discussed below.

1. In the Data section, click the Type field and select the type of parse control from the drop-down list.

If you select Regular Expression as the Control Type, the Type field contains the String data type.

The data type defines the type of the field in the input format. Data Types are described in the following table.

Data Type Field Values

Data Type Field Value	Description
Not Applicable	No data type is assumed.
String	A string of standard ASCII characters. Note that non-printable characters are valid as long as they are in the ASCII character set. (EBCDIC characters outside the valid ASCII String range are not valid ASCII String characters. During a reformat from ASCII to EBCDIC if a character being converted is not in the EBCDIC character set the conversion results in a EBCDIC space (hexadecimal 40)).
Numeric	A string of standard ASCII numeric characters.

Data Type Field Value	Description
Binary Data	The Binary data type is used to parse any value and transform that value to an ASCII representation of the value internally in the Formatter. The internal representation takes each byte of the input value and converts it to a readable form. An example of this is parsing a byte whose value is (hexadecimal) 0x9C and transforming that to the internal ASCII representation of 9C, which is the hexadecimal value 0x3943. If this value is used in an output format with the output control's data type set to String, the value placed in the message is ASCII 0x9C. If this value is again placed in an output message with the data type Binary, the ASCII value is not printable and occupies one byte with the value of (hexadecimal) 0x9C. Conversely, an input value of ASCII 3B7A parsed with the String data type can be output using the Binary data type. The output value is (hexadecimal) 0x37BA and occupies 2 bytes in the output message. Valid characters that can be converted to Binary from the String data type are 0 through 9 and A through F. All other characters are invalid.
EBCDIC Data	A string of characters encoded using the EBCDIC (Extended Binary Coded Decimal Interchange Code) encoding used on larger IBM computers. During a reformat from EBCDIC to ASCII, if a character being converted is not in the EBCDIC character set, the conversion results in a space (hexadecimal 20).
IBM Packed Integer	Data type on larger IBM computers used to represent integers in compact form. Each byte represents two decimal digits, one in each nibble of the byte. The final nibble is always a hexadecimal F. For example, the number 1234 is stored as a 3-byte value: 01 23 4F (the number pairs show the hexadecimal values of the nibbles of each byte). The number 12345 is stored as a 3-byte value: 12 34 5F. There is no accounting for the sign of a number, so all numbers are assumed to be positive.

Data Type Field Value	Description
IBM Signed Packed Integer	Data type on larger IBM computers used to represent integers in compact form. This data type takes into account the sign (positive or negative) of a number. Each byte represents two decimal digits, one in each nibble of the byte. The final nibble is a hexadecimal C if the number is positive, and a hexadecimal D if the number is negative. An example of how to generate a default value for an IBM Packed Integer is: Data Type: IBM Signed Packed Decimal Default Value: -12345 (default value in ASCII) Data Length: (Null - use the numbers in this field.) The control is optional and there is no corresponding field in the input message, so Formatter uses the default value, converts it to IBM Signed Packed Decimal, and generates the following output: 12 34 5D. Each pair of numbers represents the two nibbles of a byte. The result is three bytes long.
IBM Zoned Integer	Data type on larger IBM computers used to represent integers. Each decimal digit is represented by a byte. The left nibble of the byte is a hexadecimal F. The right nibble is the hexadecimal value of the digit. For example, 1234 is represented as F1 F2 F3 F4 (the number pairs show the hexadecimal values of the nibbles of each byte).
IBM Signed Zoned Integer	Data type on larger IBM computers used to represent integers. Each decimal digit is represented by a byte. The left nibble of each byte, *except* the last byte, is a hexadecimal F. The left nibble of the last byte is a hexadecimal C if the number is positive, and a hexadecimal D if the number is negative. The right nibble of each byte is the hexadecimal value of the digit. For example, 1234 is represented as F1 F2 F3 C4 (the number pairs show the hexadecimal values of the nibbles of each byte). -1234 is represented as F1 F2 F3 D4.

Data Type Field Value	Description
Little Endian 2	Two-byte integer where the bytes are ordered with the rightmost byte being the high order or most significant byte. For example, the hexadecimal number 0x0102 is stored as 02 01 (where the number pairs show the hexadecimal values of the nibbles of a byte).
Little Swap Endian 2	Two-byte integer where the two bytes are swapped with respect to a Little Endian 2 value. For example, the hexadecimal number 0x0102 is stored as 01 02.
Little Endian 4	Four-byte integer where the bytes are ordered with the rightmost byte being the high order or most significant byte. For example, the hexadecimal number 0x01020304 is stored as 04 03 02 01 (where the number pairs show the hexadecimal values of the nibbles of a byte).
Little Swap Endian 4	Four-byte integer where the two bytes of each word are swapped with respect to a Little Endian 4 value. For example, the hexadecimal number 0x01020304 is stored as 03 04 01 02.
Big Endian 2	Two-byte integer where the bytes are ordered with the leftmost byte being the high order or most significant byte. For example, the hexadecimal number 0x0102 is stored as 01 02 (where the number pairs show the hexadecimal values of the nibbles of a byte).
Big Swap Endian 2	Two-byte integer where the two bytes are swapped with respect to a Big Endian 2 value. For example, the hexadecimal number 0x0102 is stored as 02 01.
Big Endian 4	Four-byte integer where the bytes are ordered with the leftmost byte being the high order or most significant byte. For example, the hexadecimal number 0x01020304 is stored as 01 02 03 04 (where the number pairs show the hexadecimal values of the nibbles of a byte).

Data Type Field Value	Description
Big Swap Endian 4	Four-byte integer where the two bytes of each word are swapped with respect to a Big Endian 4 value. For example, the hexadecimal number 0x01020304 is stored as 02 01 04 03.
Decimal, International	Data type where every third number left of the decimal point is preceded by a period. The decimal point is represented by a comma. Numbers right of the decimal point represent a fraction of one unit. For example, the number 12345.678 is represented as 12.345,678. Decimal international data types can contain negative values.
Decimal, U.S.	Data type where every third number left of the decimal point is preceded by a comma. The decimal point is represented by a period. Numbers right of the decimal point represent a fraction of one unit. For example, the number 12345.678 is represented as 12,345.678. Decimal US data types can contain negative values.
Unsigned Little Endian 2	Like Little Endian 2, except that the value is interpreted as an unsigned value.
Unsigned Little Swap Endian 2	Like Little Swap Endian 2, except that the value is interpreted as an unsigned value.
Unsigned Little Endian 4	Like Little Endian 4, except that the value is interpreted as an unsigned value.
Unsigned Little Swap Endian 4	Like Little Swap Endian 4, except that the value is interpreted as an unsigned value.
Unsigned Big Endian 2	Like Big Endian 2, except that the value is interpreted as an unsigned value.
Unsigned Big Swap Endian 2	Like Big Swap Endian 2, except that the value is interpreted as an unsigned value.
Unsigned Big Endian 4	Like Big Endian 4, except that the value is interpreted as an unsigned value.

Data Type Field Value	Description
Unsigned Big Swap Endian 4	Like Big Swap Endian 4, except that the value is interpreted as an unsigned value.
Date and Time*	Based on the international ISO-8601:1988 standard datetime notation: YYYYMMDDhhmmss. See the first paragraph of each of the Date and Time type descriptions for details on representing Date and Time components. Combined dates and times may be represented in any of the following list of base data types. For some data types, a minimum of 8 bytes is required. The list includes: Numeric, String, and EBCDIC.
Time*	Based on the international ISO-8601:1988 standard time notation: hhmmss where hh represents the number of complete hours that have passed since midnight (between 00 and 23), mm is the number of minutes passed since the start of the hour (between 00 and 59), and ss is the number of seconds since the start of the minute (between 00 and 59). Times are represented in 24-hour format. Times may be represented in any of the following list of base data types. For some data types, a minimum of 4 bytes is required. The list includes: Numeric, String, and EBCDIC.
Date*	Based on the international ISO-8601:1988 standard date notation: YYYYMMDD where YYYY represents the year in the usual Gregorian calendar, MM is the month between 01 (January) and 12 (December), and DD is the day of the month with a value between 01 and 31. Dates may be represented in any of the following list of base data types. For some data types, a minimum of 4 bytes is required. The list includes: Numeric, String and EBCDIC.

Data Type Field Value	Description
Custom Date and Time*	Custom Date and Time enables users to specify different formats of dates, times, and combined dates and times. Date/Time formats may include: 1) Variations in year (2- or 4-digit year representation: YY or YYYY). 2) Variations in month –use of a month number (01-12) or three-letter abbreviation (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC). 3) Variations in the day of the month – use of a day of the month number (01-31). 4) Variations in hour – 12-hour or 24-hour representation, with or without a meridian indicator (AM or PM.) 5) Custom date/time formats are available in the Format drop-down list. Custom date/time formats must have a base data type of Numeric, String, or EBCDIC. For information on how to set the Year Cutoff value, refer to the section *Year 2000 Compliance* on page 64.

* If you use Date and Time, Date, Time, or Custom Date Time as the Data Type, select a Base Data Type of ASCII String, Numeric, or EBCDIC. If you select Custom Date Time, also select a format string from the Format drop-down list and a Year Cutoff. For more information, refer to the section *Year 2000 Compliance* on page 64.

Note:
A 16-byte length limit applies to IBM platforms using packed data types (IBM Packed Integer and IBM Signed Packed Integer). However, the 16-byte limit has been removed for the IBM zoned data types (IBM Zoned Integer and IBM Signed Zoned Integer) to reflect the way the data is actually handled by IBM platform assemblers.

Note:

The data types listed above may be followed by user-defined data types.

---

2. If the data in a field is constant, select a control type of Literal. You must define a literal containing the literal value that will be in the message.

    If the Input Control Type is Regular Expression and the data type field contains an ASCII data type, you must enter a regular expression in the Regular Expression field (maximum of 122 bytes). You must define a literal containing the regular expression and assign it.

    Regular expressions are strings that express rules for string pattern matching. Regular expression parse controls function like literal parse controls, but instead of a direct character-by-character match, the parse control value is interpreted as a regular expression to match to the input. The string matching capabilities of this feature comply with the POSIX 1003.2 standard for regular expressions. Examples of regular expressions are shown in the following table.

Regular Expressions

Value	Description
a	match a
\ \	match \
^a	match a, only if it begins the string
a$	match a, only if it ends the string
^ a$	match a, only if it consists entirely of the one character a
. -	match any one character
[abc]	match one of the characters a, b, or c
[^abc]	match any one character that is not a, b, or c

Value	Description
a +	match a string of one or more of the character a
ab*	match the character a optionally followed by any number of the character b
a{3}	match 3 or more concatenated characters a. Equivalent to aaa+ or aaaa*
(a|b|\\\|)	match either the character a or b or |

Note:

When a regular expression is evaluated, a ^ is prepended to begin the match at the first character of the current position of the parser within the message.

3. To specify a year cutoff number for year 2000 compliance, in the Year Cutoff field, type a year cutoff number from 0 to 99. The default is 101, an invalid number. You must enter a valid number to continue.

   To specify a year cutoff, the (Data) Type field must be Custom Date and Time. The format must include a two-digit year notation (for example, MM/DD/YY). For more information, refer to *Year 2000 Compliance* on page 64.

4. Click the Termination field and select the termination type for the data from the drop-down list.

   The termination type tells NEONFormatter when to stop parsing for this field. Data terminators are described in the following table.

Data Terminators

Data Terminator Value	Description
Not Applicable	No data termination. Read to end of message.
Delimiter	The field is terminated by a delimiter.
White Space Delimited	The field is terminated by a white space.

Data Terminator Value	Description
Exact Length	The field has a fixed length.
Minimum Length and Delimiter	Parse a minimum number of characters and then look for a delimiter.
Minimum Length and White Space	Parse a minimum number of characters and then look for a white space.

5. If you specified a data termination type of either exact length or minimum length, choose one of the following:

   ■ If you specified exact length, type the exact length of the data, used for fixed-format messages in the Length field.

   ■ If you specified minimum length, type the number of bytes NEONFormatter skips before it starts looking for the termination control in the Length field.

6. If the termination type is delimited select a literal from the delimiter drop-down list.

7. If the data type requires a decimal, click the Decimal Loc field and either type a decimal location or click the arrows until you reach the correct decimal location. Only a positive value is allowed, and it must not be more than the number of digits the field can hold.

   Note that decimal location pertains only to IBM packed and zoned decimal data. The value determines how many digits are to the right of the decimal location when Formatter interprets the message data. For example, an input value of 12345 with a decimal location of 2 is interpreted as 123.45.

Optional/Mandatory

Check the Optional checkbox to make the input field parse optional. Optional means the parse continues even if parse parameters for the field are not found. By default, input controls are mandatory.

If the parse is successful, the field passes the mandatory test and the parse continues. If the parse parameters are not found, the field fails the mandatory test and the parse fails for the rest of the format.

Length Section

The Length section is enabled if you select one of the following as your Input Control Type:

- Tag, Length & Data
- Length & Data
- Length, Tag & Data

If a length is required and you do not specify one, the default length is zero.

Numeric fields of all types are right justified when the default length is less than the field length.

1. In the Length section, select the parse control length data type from the Type drop-down list.

   Note:
   For a description of the data types, refer to the table on page 52.

2. Click the Termination field and select the parse control length termination type from the drop-down list.

   The parse control length termination type tells Formatter when to stop parsing for the length field value. When the length delimiter is reached, the value that was parsed is the length value.

   For a description of termination types, refer to the table *Data Terminators* on page 60.

3. If the termination type is exact length, click the Length field and type the exact length of the data.

4. If the length termination type is delimited, select a literal from the Delimited drop-down list.

Tag Section

The Tag section is enabled if you select one of the following as your Input Control Type:

- Tag & Data

- Tag, Length & Data

- Length, Tag & Data

1. In the Tag section, click the Type field and select the data type of the tag data from the drop-down list.

   For a description of the data types, refer to the table on page 52.

2. Click in the Tag Value field and select the value from the list.

3. Click in the Termination field and select the termination type for the tag from the drop-down list.

   This tag tells Formatter when to stop parsing the tag value. When the tag delimiter is reached, the value that was parsed is compared to the value in the tag value field to determine if this is the data being searched for.

   For a description of termination types, refer to the table *Data Terminators* on page 60.

4. If the tag termination is fixed length, click the Length field and type the exact tag length.

5. If the tag termination type is delimited, click the Delimiter field and select a literal from the drop-down list.

   Note:
   For information on the Used In tab, refer to the *Used In* section on page 35.

Tip:

You can change the name of an input control by double-clicking the input control name (without moving the mouse). A box appears around the selected input control name and the name is highlighted. Type the new, unique name and press ENTER to enter the name and position it in the list.

Saving an Input Control as an Output Control

Input controls can be saved as output controls. The output controls mirror (as closely as possible) the contents of the input control. The output control may have to be modified, since certain properties of input controls do not have exact matches on the output control side.

T**o save an input control as an output control**:

1. In the left pane of the Formatter window, select the input control you want to save as an output control and right-click. A pop-up menu appears.

2. Select Save as Output. The input control is saved as an output control with the prefix "OFC_#_" and appears in the Output Controls list.

    For example, the first time the input control "Input_Control" is saved as an output control, it is saved as "OFC_1_Input_Control," the second time it is saved as "OFC_2_Input_Control," and so on.

3. To either change or add information for the output control, select it and modify the details on the Output Control property sheet.

For more information on the Output Control property sheet, refer to the section *Output Controls* on page 65.

For information on saving an output control as an input control, see *Saving an Output Control as an Input Control* on page 87.

Year 2000 Compliance

To address Year 2000 compliance, indicate a Year Cutoff number for Input Controls using a (Data) Type of Custom Date and Time, and a Date Format that includes a two-digit year notation (such as MM/DD/YY). The year cutoff number must be from 0 to 99 (inclusive).

Once you specify a cutoff number, NEONFormatter converts a two-digit year (YY) to a four-digit year (YYYY).

The NEONFormatter GUI defaults the year cutoff to 101, an invalid number. For a two-digit year input format, you **MUST** enter a valid number to continue.

Examples of how NEONFormatter interprets the year cutoff number are:

- If the year cutoff is 50, then all two-digit input dates from 50 to 99 are designated as 1950 to 1999 output dates; all two-digit input dates from 00 to 49 are designated as 2000 to 2049 output dates.

- If the year cutoff is 75, all two-digit input dates from 75 to 99 are designated as 1975 to 1999 output dates; all two-digit input dates from 00 to 74 are designated as 2000 to 2074 output dates.

Output Controls

Output controls are used to format output data. NEONFormatter uses output format controls to determine how to justify and trim data, whether to add prefixes or suffixes, whether to perform an arithmetic expression, to use a user exit, conditional field, existence check, or transformed field. Using the output control types of either Input Field Value = or Input Field Exists allows alternative output formatting. For more information on alternative output formatting, refer to the section *Alternative Input and Output Formats* on page 125.

The Output Control tab allows you to add a new output control, create and define default and length output operations, and assign case and justification operations to an output control.

The Extended Properties tab allows you to define and modify output operations and output operation collections and assign them to an output control. For more information, refer to *Using the Extended Properties Tab* on page 82.

Creating an Output Control

To add an output control:

1. From the Formatter window tree, select Output Controls and click the right mouse button. A pop-up menu appears with the New menu item. The right pane displays the current list of output controls.

2. Select New to create a new output control entry in the left pane and open the Output Control property sheet in the right pane. The cursor is positioned in the output control text box where you can either type the new, unique output control name or accept the default name.

Notes:

-   The output control name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

-   Single quotes, double quotes, and spaces should not be used in names. However, if using quotes, make sure the quotes are not mismatched, that is, a single quote with a double quote, or a message box appears and you must edit the name to either matching quotes or no quotes.

3. Either click any mouse button outside the output controls text box or press ENTER to add the output control to the list of output controls.

The output control is highlighted and its name now appears in the Output Control name text box in the Output Control tab.
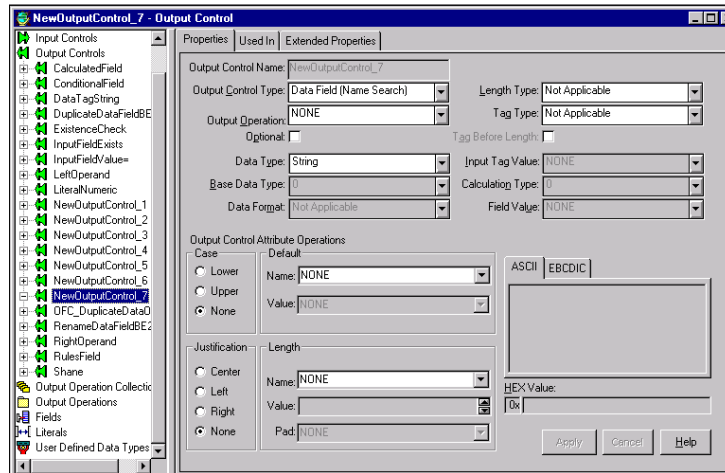
Output Control Property Sheet

4. In the Output Control property sheet, click the Output Control Type field and select a control type from the drop-down list.

An output control defines how you want Formatter to output a message field. Output Control types identify how to map to the input data and the type of reformatting to use. The Data Field (Name Search) and Data Field (Tag Search) values map the output field to the input field by either name or tag. Other output control types define details concerning data mapping.

Output Control types are described in the following table. The Procedure column lists the page number of the procedure you would use when defining each output control type.

Output Control Type	Description	Procedure
Calculated Field	Perform a calculation using the left and right operand fields.	page 72
Conditional Field	Mark field as output only if existence check field exists.	page 73
Data Field (Name Search)	Map the output field to the input field by field name.	page 69
Data Field (Tag Search)	Map the output field to the input field by tag value.	page 70
Existence Check Field	Mark field as an existence check field.	page 75
Input Field Exists	An input field exists.	page 75
Input Field Value =	The input field's value equals a particular value.	page 76
Left Operand Field	Mark field as a left operand.	page 77
Literal	Field value is a literal.	page 77
Right Operand Field	Mark field as a right operand.	page 78
Rules Field	Output format control is chosen based on Boolean logic.	page 79

Optional/Mandatory

Use the Optional check box to specify whether the output control is either optional or mandatory.

Optional means Formatter should continue with the output message if the field was not found in the input message or the input field was NULL.

By default, the output control is mandatory, Mandatory means Formatter should fail with an error message if the field was not found in the input

message or the input field was NULL, and there was no data-producing operation associated with the output control.

A data-producing operation is capable of generating data in the absence of input data. The following operations are data producing:

- Default

- Prefix/Suffix (if Null Action is checked)

- User Exit

- Math Expression

Note:
User Exit or Math Expression operations may fail if they are written to depend on the existence or value of an input field and the input field is missing or NULL.

Data Field (Name Search)

Data Field (Name Search) output controls map the value of an input field directly to an output field. Formatter searches the parsed input message for the input field value by field name specified.

For example, if the contents of an output field map directly to the contents of an input field, you could define a Data Field (Name Search) output control. The output control could look like:

- Output Control Name: Field Name With Comma

- Output Control Type drop-down list: Data Field (Name Search)

- Data Type drop-down list: String

- Tag Before Length checkbox: (not checked)

- Optional checkbox: Checked

Note:
To add a comma suffix to the end of the field data, create a Suffix operation, then select it in the Output Operation drop-down list.

To define a Data Field (Name Search) output control:

1. Select the Data Field (Name Search) in the Output Control Type drop-down list.

2. Select a data type for the output data from the Data Type drop-down list.

3. If the selected data type is Date and Time, Date, Time, or Custom Date and Time, select the Base Data Type (String, EBCDIC, or Numeric) to output the data.

4. If the selected data type is Custom Date and Time, select a date/time format from the Data Format drop-down list.

5. To perform an operation on the data, select one from the Output Operation drop-down list.

6. If the output control is optional, check the Optional checkbox.

7. If the output data is to have an embedded length, specify the Length data type in the Length Type drop-down list.

8. If the output data is to have an associated tag, specify the data type for the tag in the Tag Type drop-down list.

9. If the tag should be output before the embedded length, check the Tag Before Length box.

Data Field (Tag Search)

Data Field (Tag Search) output controls map the value of an input field directly to an output field. Formatter searches for the input field value by tag, looking for the specified literal tag value in the parsed input message.

For example, in a random-ordered flat format, you are searching for a specific tagged field in the parsed input data. If you are looking for an input Tag and Data field with <FirstName> as the tag and want to output the data minus the tag, the output control might look like:

- Output Control Name: InTagDataOutDataComma

- Output Control Type drop-down list: Data Field (Tag Search)

- Data Type drop-down list: Numeric

- Input Tag Value field: FirstName
- Tag Before Length checkbox: (not checked)
- Optional checkbox: (not checked)

Note:

To add a comma suffix to the end of the field data, create a Suffix operation, then select it in the Output Operation drop-down list. For more information, see *Output Operations* on page 87.

To define a Data Field (Tag Search) output control:

1. Select the Data Field (Tag Search) in the Output Control Type drop-down list.

2. Select a data type for the output data from the Data Type drop-down list.

3. If the selected data type is Date and Time, Date, Time, or Custom Date and Time, select the Base Data Type (String, EBCDIC, or Numeric) to output the data.

4. If the selected data type is Custom Date and Time, select a date/time format from the Data Format drop-down list.

5. To perform an operation on the data, select one from the Output Operation drop-down list.

6. If the output control is optional, check the Optional checkbox.

7. If the output data is to have an embedded length, specify the Length data type in the Length Type drop-down list.

8. If the output data is to have an associated tag, specify the data type for the tag in the Tag Type drop-down list.

9. After selecting a Tag Type, choose a Literal from the Input Tag Value drop-down list as the tag.

10. If the tag should be output before the embedded length, check the Tag Before Length box.

Calculated Field

> **Note:**
> The math expression output control offers many more options than Calculated Field. We recommend you use math expression instead of Calculated Field, since Calculated Field will be phased out in a future release.

Calculated Field output controls perform basic arithmetic operations (Add, Subtract, Multiply, and Divide) on the data defined by the Left Operand Field and Right Operand Field output controls.

For example, to use the Add, Subtract, Multiply, or Divide operations with values from two fields in the input format and output the result in an output field, use one Left Operand Field control, one Right Operand Field control, and one or more Calculated Field output controls. The Calculated Field output control performs the specified operation on the fields specified as Left and Right Operands in the flat output format.

A sample output control might look like:

- Output Control Name: AddLeftRightOutDataComma
- Output Control Type drop-down list: Calculated Field
- Data Type (Data Types) drop-down list: Numeric
- Calculation drop-down list: Add
- Tag Before Length checkbox: (Not Checked)
- Optional checkbox: (Not Checked)

> **Note:**
> A Left Operand Field output control and Right Operand Field output control MUST be defined in an output format for a Calculated Field output control to work properly.

To define a Calculated Field output control:

1. Select Calculated Field in the Output Control Type drop-down list.

2. Select a data type for the output data from the Data Type drop-down list.

3. If the selected data type is Date and Time, Date, Time, or Custom Date and Time, select the Base Data Type (String or Numeric) the data will be output as.

4. If the selected data type is Custom Date and Time, select a date/time format from the Data Format drop-down list.

5. To perform an operation on the data, select it from the Output Operation drop-down list.

6. If the output control is optional, check the Optional checkbox.

7. If the output data is to have an embedded length, specify the Length data type in the Length Type drop-down list.

8. To perform a calculation on the data, select a type of calculation from the Calculation drop-down list. Add, Subtract, Multiply, and Divide are available for Calculated Fields.

9. If the output data is to have an associated tag, specify the data type for the tag in the Tag Type drop-down list.

10. If the tag should be output before the embedded length, check the Tag Before Length box.

11. Select a literal from the Field Value drop-down list.

Conditional Field

Conditional Field output controls mark a field as output only if a specific field exists. One other field in the output format must be associated with an Existence Check Field output control.

For example, to output an account number only if a person's name is in the input message. The output control might look like:

- Output Control Name: IfPersonNameOutAccountComma

- Output Control Type drop-down list: Conditional Field

- Data Type (Data Types) drop-down list: String
- Tag Before Length checkbox: (Not Checked)
- Optional checkbox: Checked

Note:

Another field (in this example, the PersonName field) in the output format MUST be associated with an Existence Check Field output control for the Conditional Field output control to work correctly.

To define a Conditional Field output control:

1. Select Conditional Field in the Output Control Type drop-down list.

2. Select a data type for the output data from the Data Type drop-down list.

3. If the selected data type is Date and Time, Date, Time, or Custom Date and Time, select the Base Data Type (String, EBCDIC, or Numeric) the data will be output as.

4. If the selected data type is Custom Date and Time, select a date/time format from the Data Format drop-down list.

5. To perform an operation on the data, select one from the Output Operation drop-down list.

6. If the output data is to have an embedded length, specify the Length data type in the Length Type drop-down list.

7. If the output data is to have an associated tag, specify the data type for the tag in the Tag Type drop-down list.

8. If the tag should be output before the embedded length, check the Tag Before Length box.

9. Select a literal from the Field Value drop-down list.

Existence Check Field

Existence Check output controls are used to define the field that a Conditional Field output control depends on. See *Conditional Field* on page 73 for details.

To define a Existence Check Field output control:

1. Select Existence Check Field in the Output Control Type drop-down list.

2. If the output control is optional, check the Optional checkbox.

   Optional means that Formatter should continue with the output message if the input field was null and there was no data-producing operation associated with the output control.

   By default, the output control is mandatory. Mandatory means Formatter should fail with an error message if the field was not found in the input message or the input field was NULL, and there was no data-producing operation associated with the output control.

Input Field Exists

Input Field Exists output controls indicate that the output format is part of the output message if it exists in the input message.

For example, if a last name input field is in the input message, you may want to output the format that includes first, middle, and last names. You would have four fields in this format:

Field	Output Control Type
LastName	Input Field Exists
FirstName	Data Field (Name Search)
MiddleName	Data Field (Name Search)
LastName	Data Field (Name Search)

The output control for Last Name might look like:

- Output Control Name: IfLastNameOutData

- Output Control Type drop-down list: Input Field Exists

- Tag Before Length checkbox: (Not Checked)
- Optional checkbox: Checked

To define an Input Field Exists output control:

1. Select Input Field Exists as the output control.

2. If the output control is optional, check the Optional checkbox.

Input Field Value =

Input Field Value = output controls indicate that the output format the control is part of should be output if the field has a specific value.

For example, if an input field contains security information, you may want to limit the data in the output message based on that information. The output control might look like:

- Output Control Name: IfSecurity=HighThenOut
- Output Control Type drop-down list: Input Field Value =
- Input Field Value drop-down list: High
- Tag Before Length checkbox: (Not Checked)
- Optional checkbox: Checked

Note:

The Output Control type determines which fields require information. Fields containing Not Applicable or NONE do not require a value.

To define an Input Field Value = output control:

1. Select Input Field Value = as the output control type

2. Select a Literal from the Field Value drop-down list to compare the field contents to.

3. If the output control is optional, check the Optional checkbox.

Left Operand Field

> **Note:**
> The math expression output operation offers many more options than Calculated Field output controls. We recommend you use math expression instead of Calculated Field, since Calculated Field will be phased out in a future release.

Left Operand Field output controls mark an input field as left operand. Once a Right Operand Field has also been defined, the two fields can be used in a Calculated Field output control. See *Calculated Field* on page 72 for details.

For example, to use the Add, Subtract, Multiply, or Divide calculations with values from two fields in the input format and output the result in an output field, use one Left Operand Field, one Right Operand Field, and one Calculated Field output control. The Calculated Field output control performs the specified calculation on the fields specified as Left and Right Operands in the flat output format.

> **Note:**
> Both a Left Operand Field output control and Right Operand Field output control *must* be defined in an output format for a Calculated Field output control to work properly.

To define a Left Operand Field output control:

1. Select the Left Operand Field in the Output Control Type drop-down list.

2. If the output control is optional, check the Optional checkbox.

Literal

Literal output controls insert a literal (static) value into the output message.

For example, there may be a static value separating the header for a message from the body. An output control might look like:

- Tag Before Length checkbox: (Not Checked)

- Optional checkbox: (Not Checked)

- Data Type (Data Types) drop-down list: String

Chapter 3

- Output Control Type drop-down list: Literal

- Output Control Name: Out<FirstName>Literal

To define a Literal output control:

1. Select Literal in the Output Control Type drop-down list.

2. To perform an operation on the data, select one from the Output Operation drop-down list.

3. If the output control is optional, click the Optional checkbox.

4. If the output data is to have an embedded length, specify the Length data type in the Length Type drop-down list.

5. If the output data is to have an associated tag, specify the data type for the tag in the Tag Type drop-down list.

6. If the tag should be output before the embedded length, check the Tag Before Length box.

7. Select a literal from the Field Value drop-down list.

Right Operand Field

Note:

The math expression output operation offers many more options than Calculated Field output controls. We recommend you use math expression instead of Calculated Field, because Calculated Field will be phased out in a future release.

Right Operand Field output controls mark an input field as right operand. Once a Left Operand Field has also been defined, the two fields can be used in a Calculated Field output control. See *Calculated Field* on page 72 for details.

For example, to use the Add, Subtract, Multiply, or Divide calculations with values from two fields in the input format and output the result in an output field, use one Left Operand Field, one Right Operand Field, and one Calculated Field output control. The Calculated Field output control performs

78   MQSeries Integrator User's Guide

the specified calculation on the fields specified as Left and Right Operands in the flat output format.

> **Note:**
> A Left Operand Field output control and Right Operand Field output control MUST be defined in an output format for a Calculated Field output control to work properly.

To define a Right Operand Field output control:

1. Select the Right Operand Field in the Output Control Type drop-down list.

2. If the output control is optional, click the Optional checkbox.

Rules Field

Use the Rules Field (formerly called conditional branching) to create several different output controls for a single output field by integrating NEONRules with NEONFormatter. NEONFormatter uses the boolean logic capabilities of NEONRules to express and evaluate the conditions for formatting a field.

Based on the fields defined for the input format, you build different output controls for the same output field. This eliminates the need to create several output formats for a single input format.

If no rule hits, output data is formatted to the other settings in the Rules Field output control. There will be one subscription or output control for each boolean field in the output format, unless the rule did not evaluate a *true* for that rule. The data is formatted according to the output control specified by the rule.

> **Note:**
> When you define rules for the output control, you cannot delete the fields used in the rules without first deleting the rules that use the fields. If you do not remove fields from the rules, the rules are invalidated when the Consistency Checker is run. The NEONFormatter GUI deactivates any rules that are invalidated. You must then repair the affected rules.
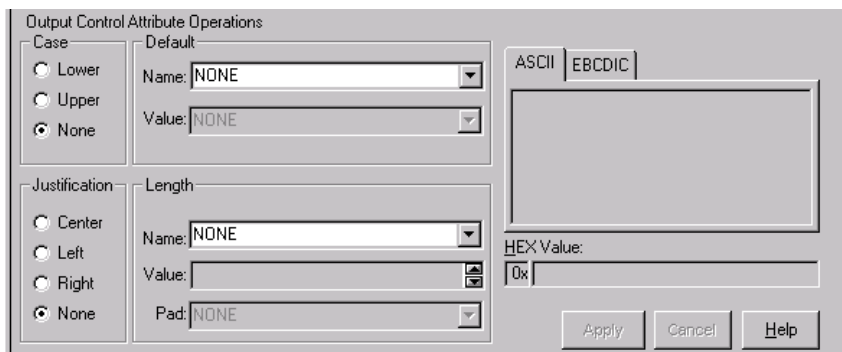
To define a Rules Field output control:

1. Select Rules Field in the Output Control Type drop-down list.

2. To perform an operation on the data, select one from the Output Operation drop-down list.

3. If the output control is optional, check the Optional checkbox.

4. Select a data type for the output data from the Data Type drop-down list.

5. If the selected data type is Date and Time, Date, Time, or Custom Date and Time, select the Base Data Type (String, EBCDIC, or Numeric) the data will be output as.

6. If the selected data type is Custom Date and Time, select a date/time format from the Data Format drop-down list.

7. If the output data is to have an embedded length, specify the Length data type in the Length Type drop-down list.

8. If the output data is to have an associated tag, specify the data type for the tag in the Tag Type drop-down list.

9. If the tag should be output before the embedded length, click the Tag Before Length box.

To add the rules for the field:

1. Choose the Jump to Rules button. The Rules Login dialog box appears.

2. Log in to Rules. The Rules GUI appears.

3. Enter the rules for the field using the instructions in Chapter 3, *Rules* on page 137.

4. Exit the Rules GUI. You are returned to the NEONFormatter GUI.

Output Control Attribute Operations

The Output Control Attribute Operations section of the Output Control window allows you to either assign existing or define new default and length output operations. Case and justification operations also can be assigned from this section of the window. You cannot change existing operations from this section; however, you can change operations using the Extended Properties tab described in *Using the Extended Properties Tab* on page 82.

*Output Control Attribute Operations Section
of the Output Control Window*

To define a new output operation:

1. To define a new default, in the Default Name field, type the new name. If there are matches for the characters you are typing, the matches scroll as you type to allow you to select an existing item. A non-matching name is added as a new name.

2. Click the Value field and either type a new value or select an existing value.

3. To define a new Length, in the Length Name field, type the new name. If there are matches for the characters you are typing, the matches scroll as you type to allow you to select an existing item. A non-matching name is added as a new name.

4. Click the Value field and either type a new value or select an existing value.

5. To assign a pad character, click the Pad field and type a new value or select a value from the drop-down list of literals.

   For more information on literals, refer to *Literals* on page 44.

To assign an existing output operation:

1. To assign a case other than None (the default), click either Lower or Upper.

2. To assign a default, in the Default Name field, either click to open the drop-down list or type the first several characters of the default you want to select to display existing items that match the characters. The list will only show items with matching characters.

3. To assign Length, in the Length Name field, either click to open the drop-down list or type the first several characters of the length you want to select to display existing items that match the characters. The list will only show items with matching characters.

4. To assign a pad character, click the Pad field and type a new value or select a value from the drop-down list of literals.

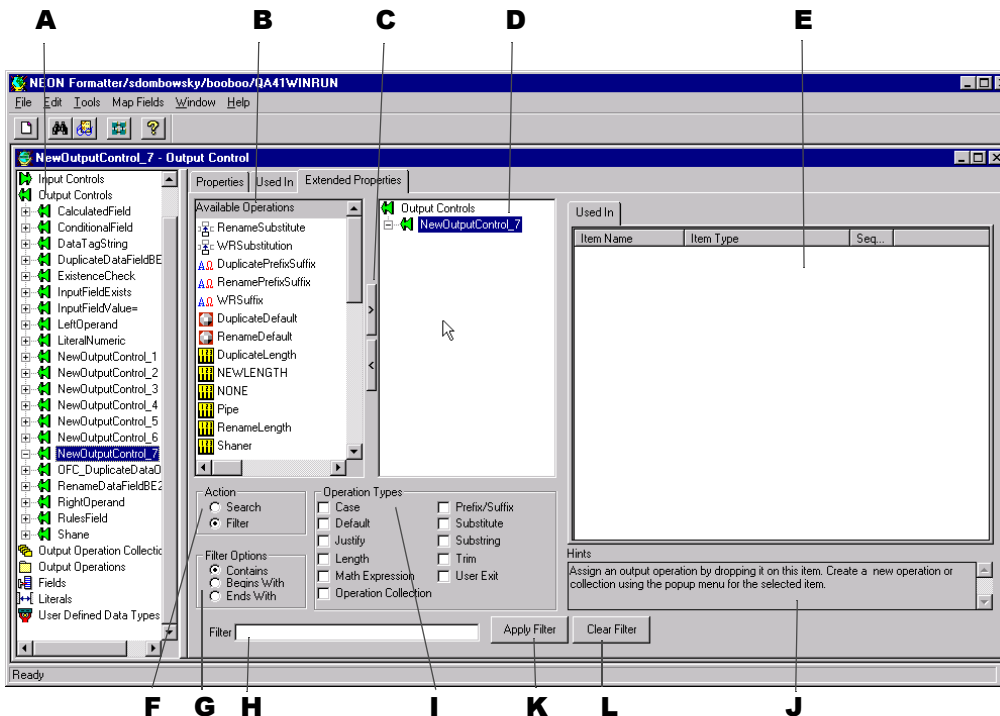   For more information on literals, refer to *Literals* on page 44.

Using the Extended Properties Tab

From the Extended Properties tab you can define and modify output operations and collections and assign them to an associated output control. If you select more than one output operation, a new output operation collection is created that begins with the name of the output control. If more than one output operation exists for a control, a number is appended to the name. For example, SS_1 for an output operation collection assigned to SS.

The value for the literal assigned to the default value is shown in the ASCII/ EBCDIC section if a length field is selected.

Selecting a case, justification, default, and length operation assigns them to a collection in the following order: default, case, length, and justify. This order is enforced by the initial assignment only.

The sections of the window are labeled in the following figure and described in the Extended Properties table.

Extended Properties Window

Extended Properties

Label	Section	Description
A	Formatter tree	The main Formatter tree.
B	Available Operations	Available output operations and output operation collections in read-only mode. This list can be filtered and if a filter is applied the title of the list reflects this. You can select any or all of the items in this list and apply them to the output control by either dragging and dropping the items onto the control itself or by using the Add (>) button.
C	Add (>) and Remove (<) Buttons	The Add (<) and Remove (>) buttons provide a fast way to perform specific tasks. The buttons behave differently, depending on what is selected. ■ To assign operations to an output control, select the output control in the Formatter tree view, select the operations in the Available Operations list, and click > . ■ To remove operations from an output control, select the output control in the Formatter tree view, select the operations in the Available Operations list, and click < . ■ To remove an output operation or output operation collection, select it and click <. A confirmation box appears asking you to confirm the removal.

Label	Section	Description
D	Selected Operations	The Selected operations list is a tree view containing a duplication of the output control from the main tree view, however, functionality not available in the main tree view is available in this list. ■ With a collection selected, you can reorder the sequence of output operations in the output operations collections tab. ■ Pop-up menus are available allowing you to create, remove, open, expand, and collapse items. ■ Creating a new item with an output control selected in the tree view assigns the new item to the control. ■ If an output operation or output operation collection is selected, you can access the pop-up menu and select Duplicate. For more information on creating output operations, refer to *Creating an Output Control* on page 66. WARNING! A change made to any assigned component of an output control affects all controls that use that component.
E	Output Operation Collections	A list of output operation collections.
F	Action	Select the Search button to find items in the list of available output operations. Type the associated text in the Search text box. Select the Filter button to narrow the available output operations. Type the associated text in the Search text box.

Label	Section	Description
G	Filter Options	Use these options to either search or filter the available output operations. Only the Available Operations section is affected, the Formatter tree does not change.
H	Filter Text Box	Use the Filter text box to either search or filter criteria.
I	Operation Types	Select one or more items to filter. If you select Justify, Length is automatically selected also because justify and length are associated. For more information, refer to *Output Controls* on page 65.
J	Hints	Hints guide you through the functionality of the tab. The hint changes to associate with the section you are positioned in.
K	Apply Filter Button	Click this button to apply the filter. The list of operations is filtered to show only the selected items.
L	Clear Filter Button	Click this button to clear all parameters for searching and filtering. All available output operations are then displayed.

Saving an Output Control as an Input Control

Output controls can be saved as input controls. The input controls mirror (as closely as possible) the contents of the output control. The input control may have to be modified, since certain properties of output controls do not have exact matches on the input control side.

T**o save an output control as an input control**:

1. In the left pane of the Formatter window, select the output control you want to save as an input control and right-click. A pop-up menu appears.

2. Select Save as Input. The output control is saved as an input control with the prefix IFC_ExistingName_# and appears in the Input Controls list.

   For example, the first time the output control **Output_Control** is saved as an input control, it is saved as IFC_1_Input_Control, the second time it is saved as IFC_2_Input_Control.

3. To either change or add information for the input control, select it and modify the details on the Input Control property sheet.

For more information on the Input Control property sheet, refer to the section *Input Controls* on page 48.

Output Operations

Output operations provide the different actions that can be performed on an output field. Using operations, you can, for example, change the case of output data, perform mathematical expressions based on input field contents, and extract substrings.

Through the use of output operation collections, you can collect operations to perform them sequentially. For example, you could left justify and right trim a substring of the contents of an input field. The order in which these operations are defined in the collection is the order in which they are performed.

Available operations are described in the following table:

Note:

Operation names have a 32-character maximum length.

Output Operation Type	Description
Case	Case operations affect the case of the field data. The two defined case operations are LOWER_CASE and UPPER_CASE.
Default	Default operations provide a default value for a field if an input field either does not exist in the input message or has a length of zero.
Prefix/Suffix	Prefix/Suffix operations enable you to attach literals to the beginning or end of the field data. **Note**: The NULL value option forces the prefix/suffix value to be generated.
Justify	Justify operations justify field data to be left, center, or right within the length of the field. For pad characters, Justify operations must proceed a Length operation in a collection. To specify a pad character other than the default (a space), select the desired pad character in the associated Length operation. If Center Justify is specified, the data can be padded on both the left and right. The Justify and Length operations are related.
Length	Length operations ensure that an output string is given a length. If the data length is longer than the specified length, the data is truncated. If the data length is shorter than the specified length, pad characters are used. Non-numeric data types are padded on the right, numeric data types are padded on the left. The Length and Justify operations are related.

Output Operation Type	Description
Math Expression	Math Expression operations output a value resulting from an arithmetic expression. The expression can be built using arithmetic operators, constants, and input field values. For more information, see *Math Expression* on page 91.
Substitute	Substitute operations enable you to define a list of input strings to substitute and the output strings to replace them. For each substitute item within a substitute operation, define a literal to look for as the input value, a literal to replace it with, and the data type in which to output the new data. For more information, see *Substitute* on page 95.
Substring	Substring operations enable you to extract a portion of an input string, defined by start byte position and length, and place it in the output field. If the End of Field checkbox is selected, the resulting substitute value is from the start location to the end of field. If the checkbox is not selected, the resulting substring is from the start location for a given length.
Trim	Trim operations remove a defined trim character to the right and/or left of the output data.
User Exit	User Exit operations enable you to run predefined user exit functions. User Exits are external, user-created routines used to compute the value of an output field. **Note**: User Exit ignores results and/or data from any previous operations.

Output Operation Collections

Use Output Operations Collections to group and sequence a series of output operations and/or other output operation collections. Operations are executed in the order in which they appear. For example, to have the contents of an input field take a substring of the left-justified, right-trimmed contents of an input field, LEFT_JUSTIFY, RIGHT_TRIM, and then SUBSTRINGXXXX.

Note:
You cannot create collections that contain themselves. This is recursive. If you see the recursion icon, it is probable that something has corrupted the data in your NEONFormatter database.

To define an output operation collection:

1. In the left pane of the Formatter window, select Output Operation Collection and right-click. A pop-up menu appears with New as a menu item.

2. Select New. A new collection is added to the left pane.

3. The cursor is positioned in the text box where you can type the new, unique Output Operation Collection name. Names should be descriptive.

   ### Note:
   The Output Operation Collection name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

4. Click any mouse button outside the name text box or press ENTER to finish defining the Output Operation Collection. The new entry moves to its alphabetical location in the list and is highlighted.

5. Right-click the collection and choose Add Output Operations. The Output Operations window opens.

6. Select the operations you want to add, then choose the Accept Selection button.

In the collection's pane, to reorder component output operations or output operation collections, click and drag the component to be moved to the highest level of the collection. It is inserted as the first component in the collection.

To move one operation to a partition preceding another operation, drag the operation that you want to move and place it on top of the operation that it should precede.

You can rearrange the other components to fit the new order using this method.

Math Expression

Using math expression operations, you can output a value resulting from an arithmetic expression. The expression can be built using arithmetic operators, constants, and input field values.

When input field names are used in math expressions, the math expression parser extracts the current value for the field(s) specified in the math expression. Alternatively, you can define a single math expression to input fields that are currently mapped to the output for which the output math expression is applied. The string "$MAPPED_INFIELD" in a math expression is used in place of or in addition to field names. The value of the Input field that is currently mapped to the output field replaces "$MAPPED_INFIELD".

For example, suppose the following relationships exist:

- inField1 maps to OutField1
- inField2 maps to OutField2
- inField3 maps to OutField3

Where...

- inField1 holds a value of 10.
- inField2 holds a value of 200.
- inField3 holds a value of 300.

If you apply the math expression "inField1 + $MAPPED_INFIELD" to OutField2 the result is 210. Likewise, if you apply the same math expression to outField3 the result is 310.

Note:
Surround $MAPPED_INFIELD with single or double quotes. For example, "$MAPPED_INFIELD" + Field1 is a valid math expression; whereas, $MAPPED_INFIELD + Field1 is NOT a valid math expression.

The ability to substitute the value of the currently mapped input field is useful when the same calculation is to be applied to multiple input fields. Instead of defining a unique math expression with the same calculation for each field and explicitly naming fields in each math expression, define a single math expression containing "$MAPPED_INFIELD". Use this math expression in all of the output controls associated with the output fields that are mapped to the input fields modifying the calculation.

Operators	Expression Components
Available Operators	+ - * / ( ) and Unary -
Order of Operator Precedence	( ) * / + -
Available Operands	Numeric Constants and Input Field Names

Note:
The math expression function ignores results and/or data from any previous operations.

For example, if an input message is defined with fields **InF1**, **InF2**, and **InF3** and an output message is defined with field **OutF1**. You could define a math expression as part of an output control associated with output message field

OutF1 as: **InF1 + InF2 * -InF3**

This expression is evaluated as **InF1 + (InF2 *(- InF3))** based on the precedence rules.

Other expression examples include:

InF1 + -InF2

InF1 * **8**

InF1 * 9.3

InF1 * -8

InF1 * -9.3

(InF1 + InF2) * 3/InF3

(InF1 * (InF2 + InF3) * 4)

To define a math expression operation:

1. In the left pane of the Formatter window, select the math expression output operation and right-click. A pop-up menu appears with New as a menu item.

2. Select New. A new operation is added to the left pane.

3. The cursor is positioned in the text box where you can type the new, unique math expression operation name. Use descriptive names.

4. Either click any mouse button outside the name text box or press ENTER. The new entry is highlighted and moves to its alphabetical location in the list.
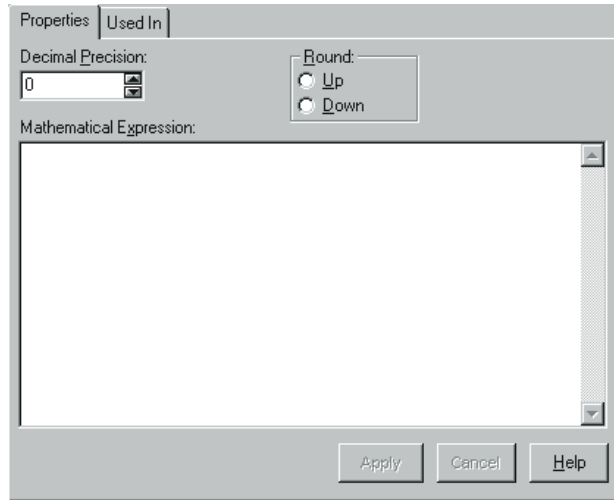
Math Expression Dialog Box

5. Select the Properties tab in the right pane.

6. Click the Decimal Precision field and either use the up/down arrows or type the number of decimal points the expression result should be rounded to.

7. Select whether the expression result should be rounded up or down using the appropriate option button.

8. Type the expression in the Math Expression field. Fields defined in mathematical expressions must be in quotes (for example, "field1" = "field2"). A field name cannot contain a dash (for example, if the field name is abc-def, rename the name to abc_def).

9. Click Apply.

Notes:

- When multiplying fields in math expressions, multiply by 1.0 to avoid erroneous characters after the decimal.

- To set up an output control to handle a 6-digit ASCII number and insert a decimal, multiply the number by 1.000 where the number of zeroes after the decimal equals the amount of decimal precision required in the final number.

- Do not use terminating characters such as carriage returns or line feeds.

Substitute

Substitute operations enable you to define a list of input strings to substitute and the output strings to replace them.

For example, each time Formatter receives the value x as input, you want to output the value as xx. You can change the value from the input of x to the output of xx.

You can add a literal with the value NONE and assign it an output value. This acts as a default substitution if there is no match for the input value.

To define a Substitute operation:

1. In the left pane of the Formatter window, select Substitute operation and right-click. A pop-up menu appears with New as a menu item.

2. Select New. A new operation is added to the left pane.

3. The cursor is positioned in the text box, where you can type the new, unique Substitute operation name. Names should be descriptive.

4. Either click any mouse button outside the name text box or press ENTER to finish defining the Substitute operation. The new entry is highlighted and moves to its alphabetical location in the list.
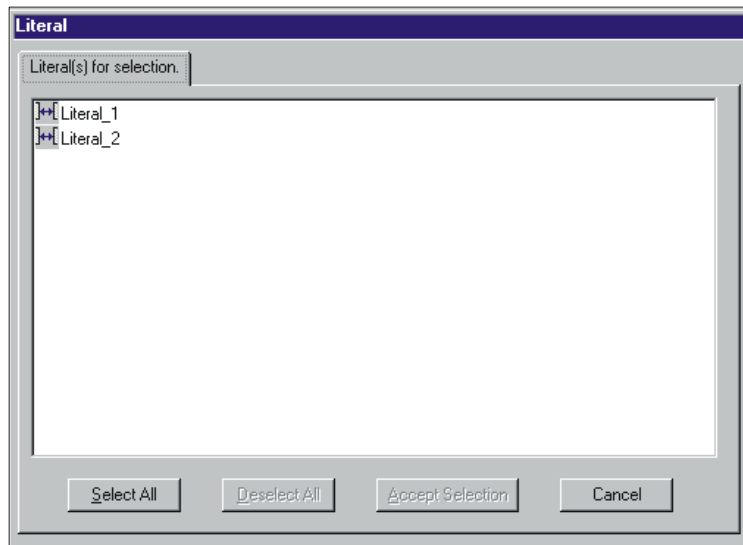
Note:

The Substitute operation itself has no properties. You must add Substitute Items.

To assign Substitute Items:

1. In the left pane of the Formatter window, select the substitute operation you want to add items to and right-click. A pop-up menu appears with Add Substitute Items as a menu item.

2. Select Add Substitute Items. The Literal window appears.

Literal Window

3. Select the literals you want to use as input strings and click Accept Selection.

4. Click the new substitute operation in the left pane.

5. For each substitute item, select:

- A data type from the Output Data Type drop-down list.

- A literal from the Output Value drop-down list. This value will be substituted for the value specified in the Input Value drop-down list.

6. Click Apply.

7. Repeat steps 5 and 6 for each Substitute Item.

Formats

Use the Formats function to build both input and output formats. The formats can be either flat or compound.

The component formats of a compound format are either optional or mandatory. An optional component can have missing components and the parse or reformat will still succeed. A mandatory component must exist, and all its mandatory components must also exist, or the parse or reformat will fail. The component formats are mandatory by default.

Before creating flat or compound formats, define their component parts.

Format	Components
flat input format	literals, fields, input (parse) controls, and user-defined types (if used)
flat output format	literals, fields, and output controls, output operations, and output operation collections
compound input format	flat input formats or other compound input formats
compound output format	flat output formats or other compound output formats

Tip:
To simplify creating NEONFormatter components:

- When building Input Controls, have two windows open. One window should be open to the new Input Control and the other should be open to Fields.

- When building Output Controls, have four windows open. One for Fields, one for Output Operations, one for Output Operation Collections, and one for the new Output Control.

- When building Flat Input Formats, have two windows open. One for the new Format and one for Input Controls to be associated with input Fields.

- When building Flat Output Formats, have two windows open. One for the new Format and one for Output Controls to be associated with output Fields.

- When building Compound Input or Output Formats, have two windows open. One for the new Compound Format and one for other Formats to be placed within it.

Creating a Format

To create a new format:

1. From the tree in the Formatter window, select Formats and click the right mouse button. The New pop-up menu appears.

2. Hold down the right mouse button and move to the right to open the submenu. Choose one of the following:

   - To create a flat format, highlight Flat.

   - To create a compound format, highlight Compound.

3. Continue to hold down the right mouse button and move to the right to open the next submenu. Choose one of the following:

   - To create an input format, highlight Input.

   - To create an output format, highlight Output.

   Release the right mouse button. A new icon appears followed by a blank text box.

4. Type a unique format name and press ENTER. The name is alphabetically inserted into the list of formats and the associated tabs are displayed in the right pane.

The tabs vary, depending on the type of format you are creating. The different formats and their associated tabs are described in the following sections.

Tip:

You can change the name of a format by selecting the format name, then click again on the name. A box appears around the selected format name and the name is highlighted. Type the new name (maximum of 32 characters) and press ENTER to enter the name and position it in the list. Names cannot contain single quotes.
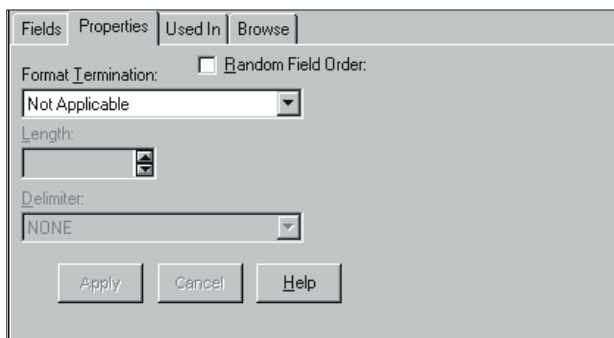

Creating a Flat Input Format

Use the Input Flat Format property sheet to build a list of fields and associated parse controls for an input message. This information matches the input message string sent to Formatter.

To add a flat input format:

1. Follow the steps in the *Creating a Format* section on page 98, selecting flat and input from the pop-up menu. The flat input format dialog box appears.

Flat Input Format Dialog Box

2. To specify the component format order as Random (fields can appear in any order), select the Random checkbox. The default order is ordinal (the fields appear in the specified order in the input message).

3. Click the Format Termination field to open the drop-down list and select the termination type. Format Termination types are described in the following table.

Format Termination Value	Description
Not Applicable	No data termination. Read to end of message.
Delimiter	The format is terminated by a delimiter.
Exact Length	The format has a fixed length.
White Space Delimited	The format is terminated by a white space.
Minimum Length + Delimiter	Parse a minimum number of characters and then look for delimiter.
Minimum Length + White Space	Parse a minimum number of characters and then look for a white space.

4. If the termination type is Delimiter or Minimum Length + Delimiter, select a literal from the Delimiter drop-down list.

   If the termination type is Exact Length, Minimum Length + Delimiter, or Minimum Length + White Space, enter the fixed length of the field or the minimum number of characters to parse before looking for a delimiter or white space in the Length field.

5. To add fields, select the format in the tree and right-click. The pop-up menu appears.

6. From the pop-up menu, select Add Field Components. The Field window appears, displaying a list of the fields.

7. Select the fields that you want to add, then choose the Accept Selection button. The fields are added to the format tree.

Tip:

- To select more than one field, you can select either a group of fields or a range of fields.

    – To select a group of fields, click the first field you want to select, then hold the Ctrl key down while clicking the other fields you want to select.

    – To select a range of fields, click the first field you want to select and hold the Shift key down and select the last field in the range. All fields between the first and last field are also selected.

- To select fields used in another format, click the Format Filter drop down and select a format. The field list changes to list only fields that appear in the selected format. To return to the complete list of fields, click the Format Filter drop down and select Not Applicable.

Note:
When a flat output format is changed by either adding or deleting a field, it is a global change. Any compound format the flat format is contained in also reflects the addition of deletion of the field.

8. When selecting each component for the format, make sure each has associated input controls and fields on their Property sheets. Select fields and input controls from the Field Name and Input Control Name drop-down lists.

Note:
When a flat input format is changed by either adding or deleting a field, it is a global change. Any compound format the flat format is contained in also reflects the addition or deletion of the field.

Saving a Flat Input Format as a Flat Output Format

Flat input formats can be saved as flat output formats. The flat output format mirrors, as much as possible, the contents of the flat input format, however, the termination type, length, and delimiter are defaulted for the output flat format.

T**o save a flat input format as a flat output format**:

1. In the left pane of the Formatter window, select the flat input format you want to save as a flat output format and right-click. A pop-up menu appears.

2. Select Save as Output. A confirmation box appears asking if you are sure you want to save the flat input format as a flat output format. Click Yes. All input controls associated with the flat input format are saved as output controls named OFC_ExistingName_#. A new flat output format OFF_ExistingName_# appears in the Formats section of the tree.

    For example, the first time the flat input format is saved as a flat output format, it is saved as OFF_NewInputFormat_1, the second time it is saved as OFF_NewInputFormat_1.

    The Objects Created Summary window appears containing a list with the new format name and any controls created.

3. To change information for the output controls created, select a control and modify its details.

Creating a Compound Input Format

Compound Input Formats are composed of other flat and compound input formats. You can break down complex formats into many individual formats to simplify parsing message data.

You create compound input formats by adding flat or other compound formats to the format. For example, you might have three flat input formats: Header, Detail, Trailer as in the following table.

Format Name	Component Format	Characteristics
CompleteFormat	Header	N/A
N/A	Detail	Repeating, Ordinal
N/A	Trailer	N/A

The component formats of a compound format are either optional or mandatory. An optional component can have missing mandatory components (for example, fields) and the parse or reformat will still succeed. A mandatory component must exist, and all its mandatory components must also exist, or the parse or reformat will fail.

The order of component formats in an input compound format are ordinal, tagged ordinal, or alternative. Ordinal specifies that component formats must appear in the specified order in the input message. Tagged ordinal format is a flat ordinal format where the first field is a literal. Alternative means that only one of the component formats can appear in the message.

Notes:

- Single quotes, double quotes, and spaces should not be used in names.

- You cannot create compound input formats that contain themselves. These are known as recursive. If you see a recursion icon (two arrows creating a circle), it is most likely that something has corrupted the data in your Formatter database.

To add compound input format:

1. Follow the steps in the *Creating a Format* section on page 98, selecting compound and input from the pop-up menu. The compound input format property sheet appears.
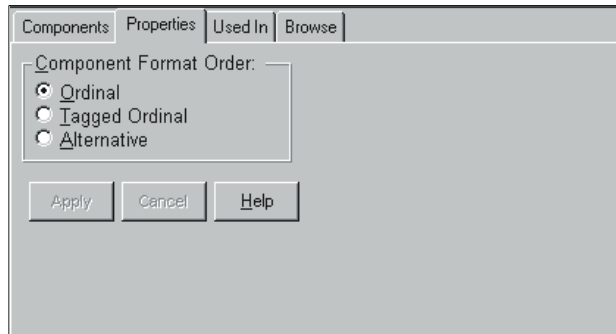
*Compound Input Format Property Sheet*

2. In the Component Format Order section, choose one of the following:

   ■ To have compound formats appear in the specified order in the input message, select ordinal.

   ■ To have the first field of the format be defined as a literal, select Tagged Ordinal.

   ■ To have only one component format appear in the message, select Alternative.

Note:

For each parse or reformat of an alternative format, a different component of the alternative format can apply. For more information, see *Alternative Input and Output Formats* on page 125.

3. To add components, select the format in the tree and right-click. The pop-up menu appears.

4. From the pop-up menu, select Add Component Formats. The Components window appears displaying a list of the available components.
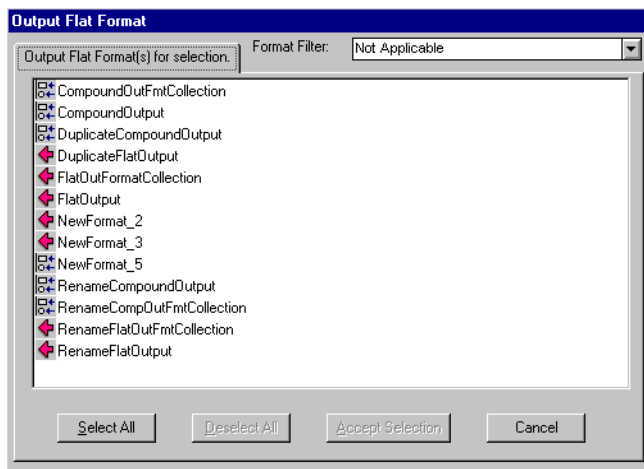
Components Window

5. Select the format that you want to add.

6. To filter the formats, click the Format Filter arrow to open the drop-down list displaying the available formats and fields.

    You can filter the list to display only the formats contained in a specific compound format.

    After you select a filter, the Components window displays only the filtered fields and/or formats.

    To display the entire list of available formats and fields, select Not Applicable from the Format Filter drop-down.

7. Click Accept Selection.

8. To change the properties of the component you just added, select the component in the left pane. The associated property sheet appears in the right pane.
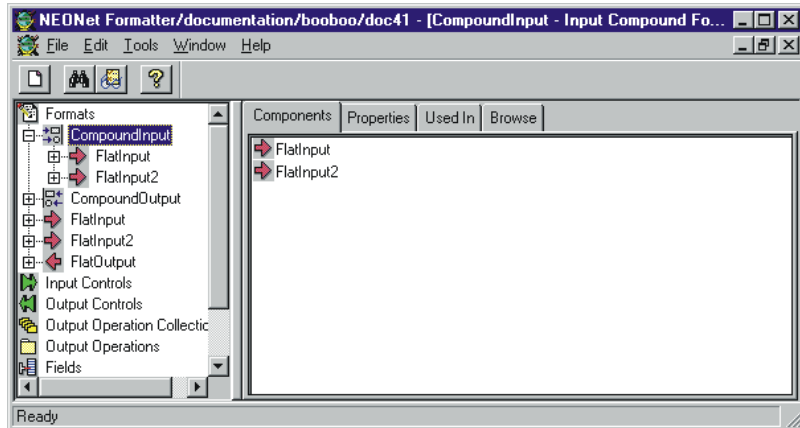
Component's Property Sheet

9. Choose one of the following from the properties tag:

   ■ To define the format as optional, select the Optional checkbox. If the format is optional, skip the rest of this procedure.

   ■ To define the format as repeating, select the Repeating checkbox. The Repeat Terminators field is enabled.

10. Click in the Repeat Termination field to open the drop-down list and select a repeat terminator. Repeat terminators are described in the following table.

Repeat Terminators

Repeat Termination Value	Description
Not Applicable	No format termination. Read to end of message.

Repeat Termination Value	Description
Delimiter	Repeating group is terminated by a delimiter.
White Space	Repeating group is terminated by a white space.
Exact Count	There is an exact count of repeating formats. When you choose Exact Count, you must then indicate the number in the Repeat Count field.
Field contains repeat count	The field has a repeat count. The field containing the repeat count is defined in the Repeat Field. The defined repeat field should also be dragged and dropped into the format. The location of the field in the format's field hierarchy does not matter (the field must exist in the message someplace prior to the repeating component).

11. Choose one of the following:

- If you select Delimiter as a termination type, select one from the Delimiter drop-down list.

- If you select Exact Count, indicate the repeat count in the Repeat Count field.

- If you select Field Contains Repeat Count, specify the field in the Repeat Field drop-down list. After specifying the field, drag the field into the compound format Components tab.

Note:
For information on the Used In tab, refer to the *Used In* section on page 35.

Saving a Compound Input Format as a Compound Output Format

Compound input formats can be saved as compound output formats. The compound output format mirrors, as much as possible.

T**o save a compound input format as a compound output format**:

1. In the left pane of the Formatter window, select the compound input format you want to save as a compound output format and right-click. A pop-up menu appears.

2. Select Save as Output. A confirmation box appears asking if you are sure you want to save the compound input format as a compound output format. Click Yes.

    All input controls associated with the compound input format are saved as output controls named OFC_ExistingName_#.

    Each flat input format that is a component of the compound format is saved as a flat output format with the prefix OFF, for example, OFF_ExistingName_1.

    Each compound input format that is a component of the compound format is saved as a compound output format with the prefix OCF, for example, OCF_ExistingName_1.

    A new root compound format with the prefix OCF, for example, OCF_ExistingName_1, appears in the Formats list in the tree, along with all component formats created.

    The Objects Created Summary window appears containing a list with the new format name and any controls created.

3. To change information for the output control, or flat and compound formats created, select a control or format and modify its details.

Creating a Flat Output Format

Use the Flat Output Format property sheet to build a list of fields and associated parse controls for an input message. The information in this dialog box matches the input message string sent to Formatter.

To create a flat output format:

1. Follow the steps in the *Creating a Format* section on page 98, selecting flat and output from the pop-up menu. The flat output format Fields property sheet appears.

2. To add fields, select the format in the tree and right-click. The pop-up menu appears.

3. From the pop-up window, select Add Field Components. The Fields window appears.
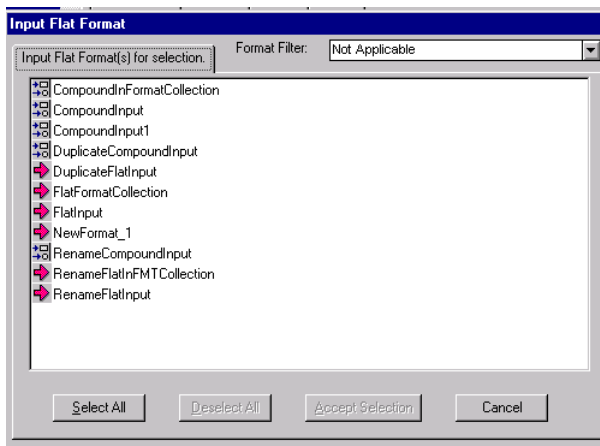
Fields Window

4. Select the field that you want to add.

Tip:

- To select more than one field, you can select either a group of fields or a range of fields.

    - To select a group of fields, click the first field you want to select, then hold the Ctrl key down while clicking the other fields you want to select.

    - To select a range of fields, click the first field you want to select and hold the Shift key down and select the last field in the range. All fields between the first and last field are also selected.

- To select fields used in another format, click the Format Filter drop down and select a format.

    You can filter the list to display:

    - only the fields contained in a specific flat format.

    - only the fields contained in all of the flat formats contained in a compound format.

    After you select a filter, the Components window displays only the filtered fields.

    To display the entire list of available formats and fields, select Not Applicable from the Format Filter drop-down.

---

5. Click Accept Selection. The fields are added to the format tree.

Note:
When a flat output format is changed by either adding or deleting a field, it is a global change. Any compound format the flat format is contained in also reflects the addition or deletion of the field.

To change the properties for fields:

1. To be sure each field has an associated output control and input field name (if appropriate), select fields and output controls from the Input Field Name and Output Control Name drop-down lists.

2. If the access mode you want is other than Not Applicable, select the appropriate access mode from the Access Mode drop-down list.

Saving a Flat Output Format as a Flat Input Format

Flat output formats can be saved as flat input formats. The flat input format mirrors, as much as possible, the contents of the flat output format, however, the termination type, length, and delimiter are defaulted to zero for the output flat format.

T**o save a flat output format as a flat input format**:

1. In the left pane of the Formatter window, select the flat output format you want to save as an flat input format and right-click. A pop-up menu appears.

2. Select Save as Input. A confirmation box appears asking if you are sure you want to save the flat output format as a flat input format. Click Yes.

   All output controls associated with the flat output format are saved as input controls named IFC_ExistingName_#.

   A new flat input format IFF_ExistingName_# appears in the Formats section of the tree.

   The first time the flat input format is saved as a flat output format, it is saved as IFF_NewInputFormat_1, the second time it is saved as IFF_NewInputFormat_2.

   The Objects Created Summary window appears containing a list displaying the new format name and any controls created.

3. To change information for the output controls created, select a control and modify its details.

Access Modes

Each output field has an associated Access Mode. Access Modes define how NEONFormatter accesses fields in the input message to generate fields in the output message. You select output field access modes and associated input field names to tell NEONFormatter how to map fields from the input message to fields in the output message.

The following table provides a description of each access mode supported in NEONFormatter.

Access Mode	Description
Not Applicable	Do not access any field instance. Use for the output of Literals.
Normal Access	Access the instance in the same repeating component as the current controlling field instance. If there is no controlling field, access the first instance. This behaves just like Access sibling instance.
Access with Increment	A field with this access mode is the controlling field for the repeating component (see *Controlling field*). This accesses the current increment and increments it.
Access Using Relative Index	The first field in a repeating component that Formatter encounters with this access mode is the controlling field for the repeating component (see *Controlling field*). Any other field in the repeating component with this access mode behaves as if it has access mode Access sibling instance or Normal access (access the sibling of the controlling field).
Access nth instance of field	Access the nth instance (n = 0 means get the first instance) of the field in the input message.
Controlling field	This field is the controlling field for the repeating component. On each repetition, access the next field instance that is still a child of the current controlling field instance of the parent format. If there is no parent controlling field, the repetitions end with the last field instance from the input message.

Access Mode	Description
Access current instance	Access the same field instance as on the previous access (the first access will get the first instance of the field).
Access next instance	Access the next field instance relative to the previous access.
Access parent instance	Access the instance that is the first ancestor of the current controlling field instance.
Access sibling instance	Access the instance in the same repeating component as the current controlling field instance. If there is no controlling field, access the first instance.

Creating a Compound Output Format

Use compound output formats to combine several flat and compound output formats into a single output message format. Compound output formats are composed of lists of the component flat and compound formats describing the output message.

You create compound output formats by adding flat or other compound formats to the format. For example, you might have three flat output formats: Header, Detail, Trailer. A compound output format could look like:
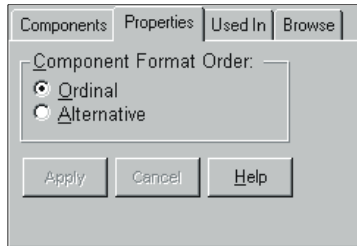
Format Name	Component Format
CompleteFormat	Header
	Detail
	Trailer

To add a compound output format:

1. Follow the steps in *Creating a Format* on page 98, selecting compound and output.

2. A new compound output format component is added to the left pane and the Components and Properties sheets are displayed in the right

pane. The cursor is positioned in the text box where you type the new, unique format name. Format names should be descriptive.

Compound Output Dialog Box

3. Click any mouse button outside the name text box or press ENTER to finish defining the format. The new format is highlighted and moves to its alphabetical location in the list.

4. To select the order for the component formats in the compound, select one of the following:

    ■ To have component formats appear in the specified order in the input message, select Ordinal.

    ■ To have only one component format appear in a message, select Alternative. See *Alternative Input and Output Formats* on page 125 for more information.

Note:
For each parse or reformat of an alternative format, a different component of the alternative format can apply.

5. To begin adding flat and compound format components, select the Components tab. The Components sheet appears.

6. Select the format in the tree and right-click. The pop-up menu appears.

7. From the pop-up menu, select Add Component Formats. The Components window appears.
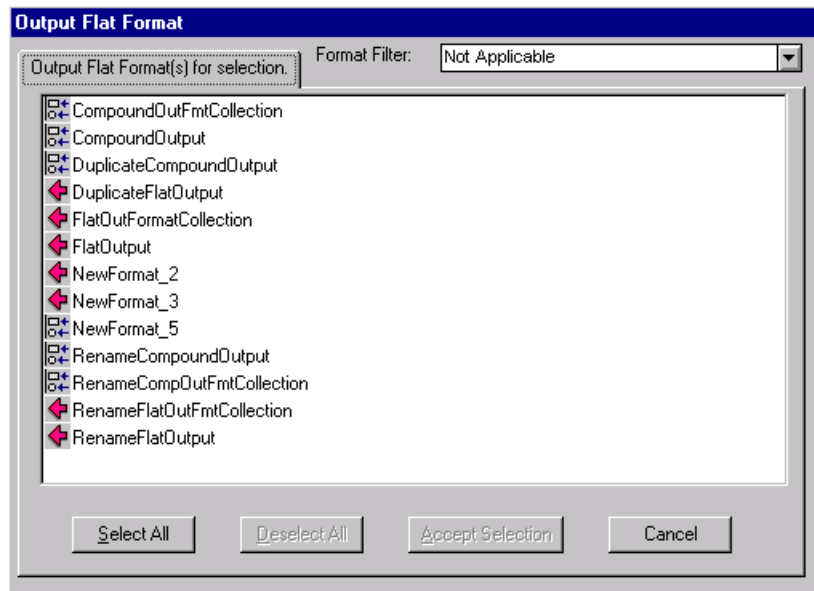
Components Window

8. Select the format that you want to add.

9. To filter the formats, click the Format Filter arrow to open the drop-down list displaying available formats.

   You can filter the list to display only the formats contained in a specific compound format.

   After you select a filter, the Components window displays only the filtered formats.

   To display the entire list of available formats, select Not Applicable from the Format Filter drop-down.

10. Click Accept Selection.

11. Select the components that you want to add, then choose the Accept Selection button. the components are added to the format tree.

To change the properties for component formats:

1. Select that component in the left pane. The associated property sheets appear in the right pane.

2. Select the Properties tab.

3. Check the boxes for one or both of the following:

    ■   To define the format as optional, select the Optional checkbox. (If the format is optional, skip the rest of this procedure.)

    ■   To define the format as repeating, select the Repeating checkbox. The Repeat Termination list becomes active.

4. Select a termination type from the Repeat Termination drop-down list.

5. Choose one of the following:

    ■   If you select Delimiter as a termination type, select one from the Delimiter drop-down list.

    ■   If you select Exact Count, indicate the repeat count in the Repeat Count field.

    ■   If you select Field Contains Repeat Count (refer to the table, *Repeat Terminators* on page 106), specify the field in the Repeat Field drop-down list. After specifying the field, drag the field into the compound format Components tab.

Saving a Compound Output Format as a Compound Input Format

Compound output formats can be saved as compound input formats. The compound input format mirrors, as much as possible, the compound output format.

T**o save a compound output format as a compound input format**:

1. In the left pane of the Formatter window, select the compound output format you want to save as a compound input format and right-click. A pop-up menu appears.

2. Select Save as Input. A confirmation box appears asking if you are sure you want to save the compound output format as a compound input format. Click Yes.

   All output controls associated with the flat output format are saved as input controls named IFC_ExistingName_#.

   Each flat output format that is a component of the compound format is saved as a flat input format with the prefix IFF, for example, IFF_ExistingName_1.

   Each compound output format that is a component of the compound format is saved as a compound input format with the prefix ICF, for example, OCF_ExistingName_1.

   A new root compound format with the prefix ICF, for example, ICF_ExistingName_1, appears in the Formats list in the tree, along with all component formats created.

   The Objects Created Summary window appears containing a list with the new format name and any controls created.

3. To change information for the output control, or flat and compound formats created, select a control or format and modify its details.

Globally Applying Input and Output Controls to Fields

The Apply Controls function lets you apply either an input or output control to:

- Selected field or fields in a format
- Every field in the selected format(s)

To globally apply either an input or output control to selected fields in a format:

1. Select the format that contains the fields you want to apply controls to from the Formatter tree.

2. In either the left pane Formatter tree or the right pane fields list, select the field(s) that you want to apply either an input or output control to. If you are selecting from the Formatter tree, you can only select one field.

3. Right click to access the pop-up menu and choose Apply (*Input* or *Output*) Control. The Apply Control window appears.

4. (Optional). To apply a filter to the list of controls, select the method to filter from the Filter Type drop-down, type the corresponding text in the Filter Text entry box, and click Apply Filter.

5. From the Apply Control list box, select the control that you want to apply. The details for the control appear in the right pane.

6. Choose one of the following:

   – To apply the control to the selected fields and leave the window open, click Apply.

   – To apply the control to the selected fields and close the window, click OK.

To globally apply either an input or output control from the format level:

1. Select either the root-level Formats or a compound format in the left pane.

2. If root-level Formats is selected, from the Formats tab in the right pane, select the format(s) that you want to apply an input or output control to.

3. Right click to access the pop-up menu and choose Apply (*Input* or *Output*) Control. The Apply Control window appears.

4. (Optional). To apply a filter to the list of controls, select the method to filter from the Filter Type drop-down, type the corresponding text in the Filter Text entry box, and click Apply Filter.

5. From the Apply Control list box, select the control that you want to apply. The details for the control appear in the right pane.

6. Choose one of the following:

   – To apply the control to the selected format(s) and fields and leave the window open, click Apply.

   – To apply the control to the selected format(s) and fields and close the window, click OK.

Field Mapping

You can map fields using either:

- The Field Map tab, which is available when you have an output format selected.

- The Field Mapping window, which can be opened from the Formatter menu bar, toolbar, or with a format selected, from the pop-up menu.

Using the Field Map Tab

When building a flat output format, you have the option of mapping a field in the input format to a field with a different name in the output format by selecting the Field Map tab. If you do not want to specify a mapped name, the input fields are mapped to output fields by matching names.

To map an input format field to an output format field:

1. With the output format selected, choose the Field Map tab. The Field Map property sheet appears.

   ### **Note:**
   The output format must have defined and associated output fields before you can map to input fields.
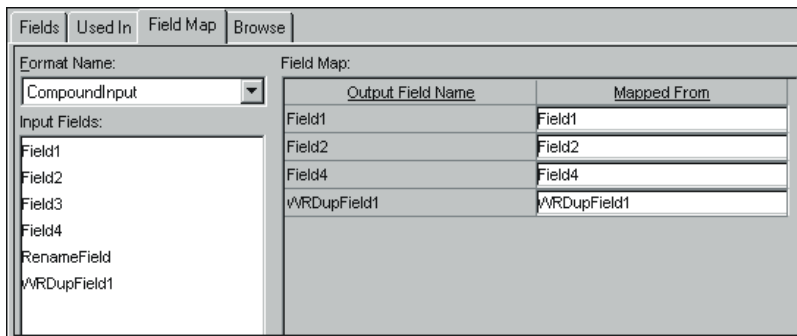
Field Map Property Sheet

2. Click the Format Name field to open the drop-down list and select the input format. The fields for the selected format are displayed in the Input Fields list box.

3. Select the field from the list box and drag the field to the appropriate Mapped From cell.

   The input field is now mapped to an output field. In the example Field Map property sheet above, output field 2 maps to input field 1. The other fields map to identical fields.

   You can map as many fields as you want by repeating the drag-and-drop process.

Using the Field Mapping Window

You can add, delete and modify field mapping for output fields.

To map fields:

1. Select the format containing the field(s) that you want to map. If you do not select a format now, you can select if after you open the Field Mapping window.

   Note that field mappings are attached to flat output formats.

2. To open the Field Mapping window, choose one of the following:

   ■ From the Formatter menu bar, choose Map Fields→Open Mapped Fields Window.

   ■ From the toolbar, choose the Map Fields icon.

   ■ From the main Formatter tree, press the right mouse button to open the pop-up menu and select Map Fields. You must have a format selected to use this method.

The Field Mapping window opens displaying the selected format and its fields.
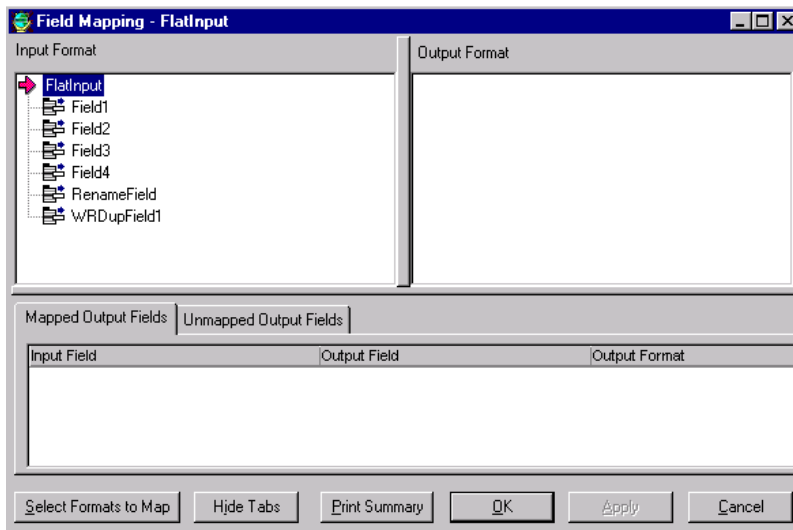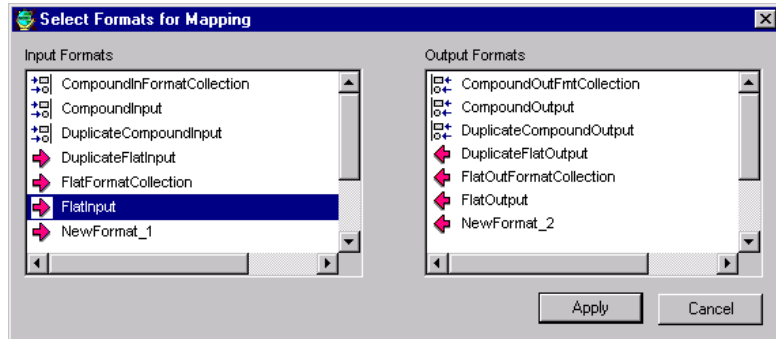

Field Mapping Window

If a single format was selected, some display data is available.

If you did not select a format (or formats) in the Formatter tree, the Mapping Fields Reminder message box appears stating that you must open both an input and an output format. Click OK. The Field Mapping window opens with no data displayed.

3. Click Select Formats to Map. The Select Formats for Mapping window appears displaying input formats in a tree in the left pane and output formats in a tree in the right pane. The format you previously chose is selected. To deselect a format, click it.

Select Formats for Mapping Window

You can double-click to open a format and view its details, but you cannot make changes to the format.

4. Select the input format and output format you want to map. You can only select one input and one output format.

5. Click Apply to begin processing formats. The Select Formats for Mapping window closes.

Depending on the size of the selected formats, processing may take several minutes. The status bar displays and updates the processing taking place. During this process, output fields are treated as follows:

- If the output field was unmapped, Formatter searches for a matching field in the input format. If no match is found, it remains unmapped.

- If the output field was previously mapped, Formatter searches for the mapped input field in the input format. If no match is found, Formatter searches for a matching field for the output field. If no match is then found, the output field becomes unmapped.

When processing completes, format information is displayed in the tree(s).

If both an input and output format were selected, Mapped Fields is the default view.

The Mapped Output Fields tab displays a list of fields from the output format and the input fields to which they are mapped. To have the two trees scroll and select the mapped fields, select a line in the list.

The Unmapped Output Fields tab displays a list of fields from the output format that could not be mapped to the selected input format. If there are unmapped fields, the tab is labeled Unmapped Output Fields. To have the output format tree scroll and select the unmapped field, select the line in the list.

6. To view only the input and output format trees, click Hide Tabs. The bottom pane of the window is hidden. To view this pane again, select Show Tabs.

7. The bottom pane of the window displays the mapped output fields. The information shown includes the input field and the output field it is mapped to, the output format the outfield is associated with, and the field sequence (where the field is in the format). To view the list of unmapped output fields, choose the Unmapped Output Fields tab.

8. To view mapped fields for the output format, choose Map Fields→View Mapped Fields. To the right of each mapped output field is the name of the input field.

9. To view specific format and field information:

   ■ Choose Map Fields→Format Types/Access Modes. The format type of Ordinal, Tagged Ordinal, or Alternative and access mode details are displayed.

   ■ Choose Map Fields→Optional/Repeating. If a field is mandatory, nothing is displayed. If it is optional or repeating, it is marked accordingly.

   ■ Choose Map Fields→Input/Output Control Types. The name and type of associated input and output controls for fields are displayed next to the field name.

The Field Mapping window displays the type of information you requested in the tree view list.

10. Choose one of the following:

- To close the Field Mapping window and apply changes to the database., click OK.

- To apply changes to the database and keep the Field Mapping window open, click Apply.

Printing From the Field Mapping Window

From the Field Mapping window, you can print the contents of the entire window or selected information.

To print the contents of the entire window, either click Print Summary or choose Map Fields→All Details.

To print specific sections of the window, choose:

- Map Fields→Print→Input Tree
- Map Fields→Print→Output Tree
- Map Fields→Print→Mapped Fields
- Map Fields→Print→Unmapped Fields

Alternative Input and Output Formats

Alternative formats are a special form of compound format in which one format in a set of alternatives will apply to a message. For example, if an alternative format is named A, it may contain component formats B, C, and D. A message of format A may actually be of variation B, C, or D.

Exactly one of the alternatives must apply or the entire alternative compound format does not apply.

An alternative format can be used anywhere a format can be used, and each component format can be any kind of its respective parent input or output format.

Optional Components and Fields

Optional components and fields are each characterized as optional within the context of a compound object.

For example, a compound format composed of component formats may have a mix of mandatory and optional component formats. And a format may have a mix of mandatory and optional fields.

If a mandatory component (of a format or field) is not present in a message, the compound or flat format does not apply. However, if an optional component is not present, Formatter continues to the next component. All component formats in an Alternative Compound should be mandatory.

Alternative Input Formats

If an alternative input format is applied to an input message, the first component of the alternative format is compared to the message. If it parses with the first component format, parsing is finished. If parsing fails, Formatter tries the second component format. If that fails, it tries the next component format. If all components fail to parse, then the parse for the entire alternative input format fails.

Alternative Input Formats and Parsing

If a format has optional fields and the fields are delimited and not tagged, it may be impossible to determine which of the optional fields occur in an input message. For example, if a simple space delimited format is:

[F1] F2 [F3] [F4]

The first, third, and fourth fields are optional. If an input message value1 value2 value3 is received, there is no way, without some rules to remove ambiguity, to determine if the message is actually F1 F2 F3, F2 F3 F4, or F1 F2 F4.

Alternative input formats enable you to specify all possible configurations of mandatory and optional fields in a format. You must explicitly define all possible combinations to avoid possible parsing errors because Formatter does not recursively try all combinations by itself.

As an example, if you have the following input format:

Field 1: optional, comma delimited

Field 2: mandatory, colon delimited

Field 3: optional, comma delimited

Field 4: optional, forward-slash delimited

The input message "field1,field2:field3,field4/" parses into four fields correctly.

However, "field1 field2:field3,field4/" fails. The first field of the format is comma delimited and parsed as "field1 field2:field3". The second field is colon delimited, so Formatter looks for a colon in "field4/", detects the end of the message, and fails to parse.

It may seem simple to figure out that since the first field is optional, ignore it and try to parse the second field, but Formatter does not see that. You have to explicitly define all possible combinations (F1 F2 F3, F2 F3 F4, F1 F2 F4). In this case, the second alternative format would have three fields, starting with the mandatory colon-delimited field.

For alternative formats, you determine the order alternative components are parsed in. Remember that components are taken on a first-parsed, only-parsed basis. With that in mind, component order is critical.

For example, if you have a message "abcde," you could specify two alternatives (or more). It could be parsed into a 5-byte field or two separate fields, one 2-bytes long and one 3-bytes long. If the 5-byte field format is the first alternative, you will never parse using the second format. If two parses are valid for the same input, only the first occurs.

Alternative Output Formats

To format an alternative output format, Formatter attempts to create the first component of the alternative format. If creating the first component is successful, formatting is finished. If it fails, Formatter tries to create the second component, and so on. If all components fail to be created, formatting the alternative output format fails.

For information on using alternative formats in Formatter, refer to the section *Output Controls* on page 65.

Note:
A component is not created if a mandatory component or field for the output format is not present in the incoming message.

Tagged Input Formats

A compound format can have a property of Tagged Ordinal. This means that the first field in each component format is a literal. The component format can be flat or compound.

Tagged Input with Alternative Component Example

Tagged input formats can be very useful when used in conjunction with alternative formats. The following is an example of what might come in the data segment of a SWIFT message.

":10:f1 f2 f3<CRLF>:20:C/1234/<CRLF>first description<CRLF>second description<CRLF>:30:f4,f5<CRLF>"

The following is a loose definition of the format:

:10: field1 field2 field3 <CRLF>

:20: [C/acctnum/< CRLF>] (optional)

desc1 <CRLF>

[desc2 <CRLF>] (optional)

:30:first, second <CRLF>

Parse segment :20: using the following rules:

- If there are two slash-delimited fields before the <CRLF>, the credit code and account number exist.

- If not, continue with the mandatory desc1 field, followed by an optional desc2 field.

- Do not run into the :30: segment by parsing ":30:first, second" into the <CRLF>-delimited desc2 field when the desc2 field is actually not present, since it is optional.

With tagged formats, we now have a way to ensure that we do not overrun the boundaries of the :20: segment. Any trailing optional fields in a tagged flat format can be parsed or determined to be absent by parsing only up to the component boundary, instead of looking for a field delimiter (or other termination) beyond the component boundary.

Define a compound tagged format with three components.

Segment10 :

>First Field Literal ":10:"

>space-delim field

>space-delim field

><CRLF> delim field

Segment20 : Alternative format with two components

Segment20_1 (this is the first alternative component)

>First Field Literal ":20:"

>Credit Code : slash delim, mandatory

>AcctNum : /<CRLF> delim, mandatory

>desc1 : <CRLF> delim, mandatory

desc2 : <CRLF> delim, optional

Segment20_2 (this is the second alternative component)

First Field Literal ":20:"

desc1 : <CRLF> delim, mandatory

desc2 : <CRLF> delim, optional

Segment30 :

First Field Literal ":30:"

comma delim field

<CRLF> delim field

When parsing a Segment20, the parser first attempts to parse Segment20_1. If it fails, (the credit code and account number are not part of this particular :20: segment), it parses the second alternative.

You could have different first field literals for 20_1 and 20_2, e.g. :20A: and:20B:, like SWIFT sometimes does.

A tagged input format can be included as a component of a compound, where the other components can be any other kind of input format. For SWIFT, you might define a SWIFT 570 message as a compound ordinal of three components.

Basic Header : Ordinal Flat Format

570 Data Segment : Tagged Compound Format

Trailer : Random Tagged Flat

Tagged formats do not apply to output formats. An output format can certainly have a first field literal, but calling it a tagged format does not gain us anything. It is only when we require an additional way to determine message boundaries on parsing and input message where first field literals become more explicitly necessary.

Alternative Output Format Example

Creating an output format is conceptually clearer. You define a set of alternatives, then define an alternative compound having all of the alternatives as components, sequenced in order of desirability. As with input formats, it may be appropriate to order alternatives from specific to general. (This is not always the case. Many times order is irrelevant on output because of mutually exclusive mandatory fields, i.e., there is only one format that applies.)

Using the previous description of a :20: segment, assume we have to create a :20: segment instead of parse it. We do not know what kind of input message we had, whether it was a Segment20_1, Segment20_2, or some other input format that allowed a parse of some or all of the fields required for a segment 20.

You cannot just say you have some optional leading fields CreditCode and AcctNum, since you have to put in a <CRLF> if the fields exist, but not if they don't. While not the only way, alternative output formats make this easy.

Create a set of (two) alternatives.

Seg20_1 1literal :20:

> 2CreditCode Mandatory with / delim
>
> 3AcctNum Mandatory with /<CRLF> delim
>
> 4desc1 Mandatory with <CRLF> delim
>
> 5desc2 Optional with <CRLF> delim

Seg20_2 1literal :20:

> 2desc1 Mandatory with <CRLF> delim
>
> 3desc2 Optional with <CRLF> delim

Notice the similarity between the two. It is easy to go from most specific to more general by using the GUI Save As and Field Delete features, given this kind of situation. By having several mandatory fields in a format, you are ANDing their existence together. All must be present to create the given format. If you sequenced Seg20_2 before Seg20_1 in the alternative output (compound) format, Seg20_1 would never be applied.

User–Defined Data Types

Use a user-defined data type to define your own data types (based on native data types) and associate your own validation logic with these types. The validation logic is executed for input fields defined as user-defined types. User validation logic is done in addition to native type validation. User-defined types can be assigned to both input and output fields, but user validation is done on input fields only.

For a list of valid native data types, refer to the table on page 52.

Creating a User-Defined Data Type

To create a user-defined data type:

1. From the Formatter tree, select User Defined Data Types and press the right mouse button. A pop-up menu appears displaying the New menu item.

2. Select New. A new user-defined type component is added to the left pane and the user-defined data types property sheet is displayed in the right pane. The cursor is positioned in the text box where you must type a new, unique name.

Notes:

The user-defined data type name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

Single quotes, double quotes, and spaces should not be used in names. However, if using quotes, make sure they are not mismatched, i.e. a single quote with a double quote, or a message box appears and you must edit the name to either matching quotes or no quotes.

3. Click any mouse button outside the name text box or press ENTER to finish defining the user-defined data type. The user-defined data type name is highlighted and moves to its alphabetical location in the list.

4. In the property sheet, click the Native Data Type field to open the drop-down list and select the data type. Data types are described in *Input Controls* on page 48.

User-Defined Property Sheet

5. Type the name of the validation routine in the Validation Routine field. See *User-defined Data Type Input Field Validation* in the **Programming Reference for NEONFormatter APIs** for details about validation routines.

To change the name of a user-defined data type:

1. Select the user-defined data type name.

2. Click on the name again. A text box appears around the selected name and the name is highlighted.

3. Type the new name.

Notes:

The user-defined data type name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

Single quotes, double quotes, and spaces should not be used in names. However, if using quotes, make sure they are not mismatched, i.e. a single quote with a double quote, or a message box appears and you must edit the name to either matching quotes or no quotes.

4. Either press ENTER or click the mouse outside the name text box. The user-defined type name is highlighted and moves to its alphabetical location in the list.

Using a User-Defined Data Type in an Input Control

When creating an input control, you can select a user-defined data type. An input control with a user-defined data type can have both an associated length and tag, but only the data itself can be user defined.The length and tag data cannot be user defined.

The Validation Parameters Property Sheet

When you select a user-defined data type in an input control, the Validation Parameters property sheet appears. You can define the name of the argument for the runtime data lookup function. This function is one of the methods of the validation callback object. For more information on user callback, refer to the *User Callback API Function* section in the **Programming Reference for NEONFormatter APIs**.

The Validation Parameters property sheet also enables you to define name-value pairs. The name identifies the meaning of the value. For example, if the name is phone number, the value would be the actual number.
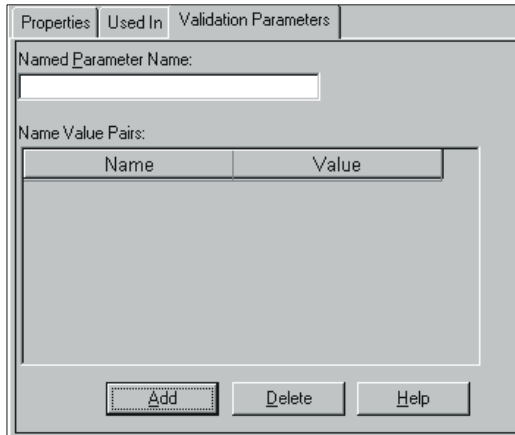
Validation Parameters Property Sheet

Creating a Validation Parameter

To add a validation parameter:

1. From the Validation Parameter property sheet, click the Named Parameter Name field and type the name of the lookup function argument.

2. To add name value pairs, select the Add button. A text entry field appears.

3. Click the Name field and type the name of the function.

4. Either press *Tab* or click the Value field and type the value of the function.

5. Repeat steps 2-4 until you enter all the name value pairs.

Deleting a Name Value Pair

To delete a name value pair, click the Name Value Pair field that you want to delete and select the Delete button. Both the name and value are deleted.

Chapter 4
Rules

NEONRules evaluates the contents of a message (which is a string of data) and uses the evaluation results to perform actions on the message. The following describes what constitutes a rule and a method to define rules through the GUI. Management APIs can also be used to define rules. For more information on APIs, refer to the ***MQSeries Integrator Programming Reference for NEONRules***.

Each rule has a list of evaluation criteria (called an expression) that consists of fields from the message and associated Rules operators linked together with Boolean operators. Field names are defined by the user through the Formatter definition mechanism. Fields can be compared either against constant data or other fields within a message.

Rules are defined within an application group/message type pair. Rules are uniquely identified by the application group/message type/rule name triplet. Rule names are defined by the user.

Rules application groups allow you to easily maintain rules associated with business needs. An application group is a logical grouping used to organize rules. For example, a company can split rules into groups by projects or split projects into logical sub-groups.

Each application group can contain several message types and a message type can be in more than one application group. Message types are defined by the user through the Rules definition mechanism. A message type defines the layout of a string of data. When using Formatter, the message type is the same as the input format name.

When a rule evaluates true, the Rules daemon retrieves specific actions that should be processed by the application. These actions can be thought of as computer commands and the associated parameters required to execute them.

A rule can have multiple subscriptions and each subscription can have multiple actions. A subscription is created in the Subscription list, then assigned to one or more rules within an application group/message type. A subscription is defined by an application group/message type in the same way as a rule. Subscriptions are groupings of actions processed when a rule evaluates true. An action cannot be executed if it is not assigned to a rule. If a subscription is shared by more than one rule, Rules only retrieves the subscription once during evaluation, even if more than one of its rules evaluates to true.

The user defines actions and option name/option value pairs, although two predefined actions are provided with the associated options: *putqueue* and *reformat*.

Starting Rules

To access Rules:

1. Double-click the Rules icon. The Rules Logon dialog box appears.
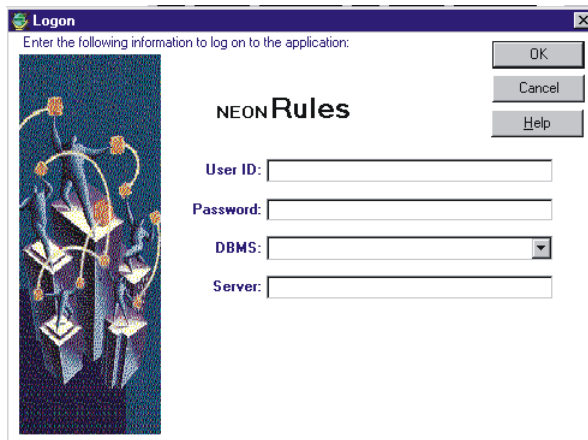
Rules Logon Dialog Box

2. If all defaults are correct, click the Password field, type your password, and click OK.

    If the defaults are not correct, type your user ID in the User ID field and press TAB.

3. Type your password and press TAB.

4. Select a database type from the DBMS drop-down list. The server, database, and qualifier fields change based on your selection. Press TAB.

    Note:
    The DBMS client appropriate for your DBMS must be installed before you can run the GUI. If you do not know the DBMS to connect to, ask your System Administrator.

5. For ODBC DBMS, type the driver name; otherwise, type the server name if the field is blank or you want to overwrite the default. Press TAB.

6. Based on the database type selected in the DBMS field, enter the following in the Database or Qualifier field:

    –   ODBC DB2: leave the Qualifier field blank.

    –   Sybase and Microsoft SQL: type the database name.

    –   ORACLE does not require a database name.

    Note:
    The Qualifier field is displayed for DB2 on UNIX and AIX. You should leave the field blank because the system does not recognize entries in the Qualifier field for these two platforms.

7. Click OK. The Rules main screen appears with the NEONRules tree window open:
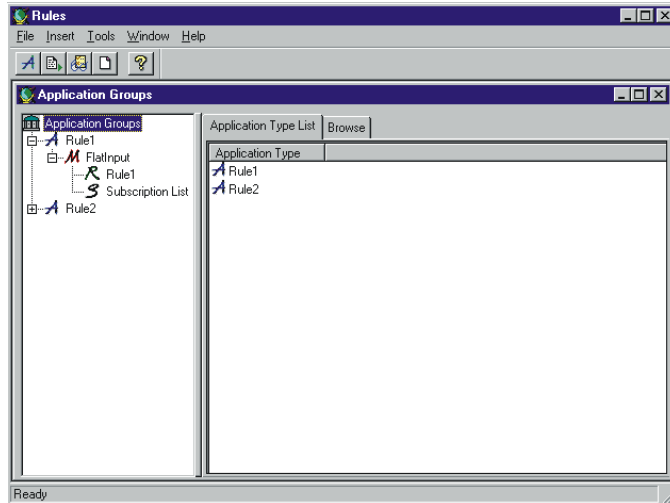
NEONRules Window

Rules Window

The Rules window is designed to work similar to Microsoft Explorer. Rules components are displayed in an hierarchical or tree-structured organization in the left pane, with Application Groups at the top level of the hierarchy. The right pane contains tabs that are associated with the selected object.

The Rules hierarchy demonstrated in the left pane is as follows:
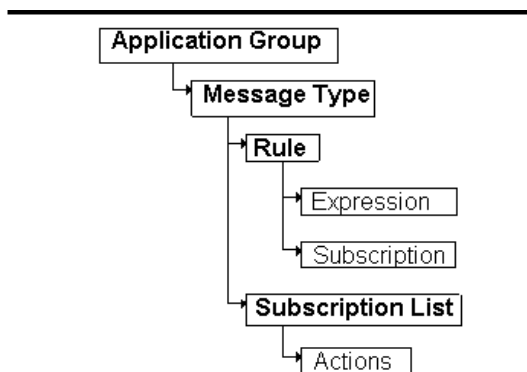
To expand the root object Application Groups, double-click it. The small boxes with a plus sign (+) next to the objects in the left pane indicate that there may be other items subordinate to the object, but the subordinate items are not displayed. Click the plus sign to expand the tree to display the subordinate items. When the tree is expanded, the plus sign changes to a minus sign (-). To collapse the tree, click the minus sign.
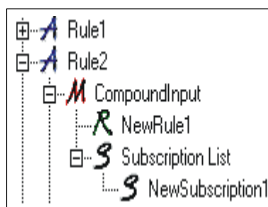
Expanded Tree

Tabs in the right pane contain detailed information about a selected component. If you have Update permission, you can add, change, and delete the information in a tab.

Opening a New/Additional Rules Window

You can open more than one Rules window at a time. To open a new window, select File→New from the menu bar.

Note:

When you open a new window, you do not create a new connection to the database. You are always connected to the same database.

Tree Functionality

Depending on the item selected in the tree, right-click to display a pop-up menu containing commands such as New, Delete, or Duplicate. For more information on pop-up menus, refer to the section *Rules Pop-up Menus* on page 148. Using the pop-up menu, you can add, delete, or duplicate a new rule or subscription.

Security

When you add either a rule or subscription, you are assigned ownership and update permissions. You must set permission for PUBLIC users to update the rule or subscription (Read permission is the default). You can also change the ownership of a rule or subscription that you own to another user. For more information, see the *Security* section for rules on page 162 and for subscriptions on page 184.

Browse

The Browse feature can be used only when an Application or one of its components is selected. When an application is selected, the Browse tab displays the expanded application; showing the application's components and properties. These components and properties consist of message types, fields in the message, rules arguments, subscriptions, and actions.

To view the Browse tab, click the Browse tab or click the Browse button on the application tool bar.

To save the data displayed in the Browse tab to a file, press FILE on the Browse tab. A window appears allowing you to save the data to a file you name.

To print the data displayed in the Browse tab, press PRINT on the Browse tab or select File→Print from the application menu.

Case Sensitivity

WARNING!

If you are using a case-insensitive database, you cannot name components the same with only a change in case to identify them. For example, you cannot name one rule "r1" and another rule "R1". In a case-insensitive environment, make each item unique using something other than case differences.

If importing components exported from a context-sensitive database into a context-insensitive database, these differences will cause NNRie to fail during import if a conflict arises between two components named the same with only case differences. See *System Management* for information on NNRie.

Also, case-sensitive operators (see *Operators* on page 166) may not work correctly on case-insensitive databases.

See *System Management* for information on how to change a current case-insensitive installation to be case sensitive.

Rules Menus

The Rules GUI features a menu bar, toolbar, and pop-up menus. Use the menu bar to open, close, and arrange windows, set up a printer, insert a new application, print the security summary report, refresh a tree item, display and move the toolbar, exit Rules, and view Rules version information.

Use toolbar buttons to:

- Create a new application
- Print the Security Summary Report
- Refresh a tree item
- Open a new window
- Access Help topics.

The tree pop-up menu differs based on which tree item is highlighted.

- When a Message is highlighted, the pop-up menu contains New, to create a new rule.
- When a Rule is highlighted, the pop-up menu contains Delete and Duplicate.
- When a Subscription List is highlighted, the pop-up menu contains New, to create a new subscription.
- When a Subscription is highlighted, the pop-up menu contains Delete and Duplicate.

The tab pop-up menus contain a variety of options such as View and Arrange from the Rules tab and Insert, Update, Delete from various other tabs.

Rules menus can also be accessed by pressing ALT plus the letter underlined in the menu name. For example, you can open the File menu by pressing *ALT+F*. After the menu is opened, you can access its functions by pressing the

underlined letter of the function. For example, to open a new Rules window from the File menu, type N (the underlined letter in the function New).

Note:

Windows has a Cut, Copy, and Paste menu that appears when editing text. This is not a Rules menu. Refer to your Windows documentation for the procedures to use this menu.

Rules Menu Bar

NEONRules menu bar options are discussed in the following sections.

Note:

If only the Rules window is open (the tree window is closed), only the File and Help menus appear.

File Menu

File menu options are described in the following sections.

New

Select File→New to open a new Rules window. The new window connects to the same database as the previous window.

Close

Select File→Close to close the active window.

Print Setup

Use the Print Setup function to select the printer for output. You can also designate the print orientation and the number of copies.

Exit

To exit Rules, select File→Exit.

Insert Menu

The Insert menu contains the New Application function. This function inserts an application group text box at the top of the application group tree where you can enter a new application group name.

Tools Menu

Tools menu options are described in the following sections.

Security Summary Report

Use the Security Summary Report function to print a report showing who has what permission for all of the rules in use. The permissions are grouped by application group and message type and then each rule is listed in the group. Each rule lists the owner's name, user names, and their permission(s).

The report is sent to your designated printer as soon as you activate this function. The report is not displayed on your screen.

Refresh Item

Refresh Item updates the active item in all open windows.

Customize Toolbars

Use the Customize Toolbars to turn the toolbar either off or on, position it in your window and, select the button size.

The Window Menu

Use the Window menu to cascade, tile either vertically or horizontally, or layer windows. For more information, refer to your Windows documentation.

The Help Menu

The Help menu contains Help Topics and About submenus. Use the Help Topics submenu to access Rules online help. The About submenu displays copyright and version information plus database, DBMS, and server information.

Rules Toolbar

Use the Rules toolbar icons to execute the functions shown in the following table.

Toolbar Icon	Description
	Inserts a text box at the top of the tree for you to enter a new Application Group name.
	Prints the Security Summary Report. For more information, see the *Security Summary Report* on page 146.
	Updates the information in an active window or all open windows. For more information, See *Refresh Item* on page 146.
	Opens a new Rules window. For more information, see *Opening a New/Additional Rules Window* on page 142.
	Opens the online Help Topics. For more information, see *The Help Menu* on page 146.

Rules Pop-up Menus

Both the right and left window panes, under certain conditions, contain pop-up menus. These menus are accessed by right-clicking the mouse.

Left Pane

Depending on what is selected in the left window pane, a pop-up menu appears when you right-click the mouse. The pop-up might contain any of the following functions:

- New
- Delete
- Remove
- Duplicate

New

Use the New function to create a new rule or subscription.

The pop-up menu allowing you to create a new rule appears when you select a message type and hold down the right mouse button.

The pop-up menu allowing you to create a new subscription appears when you select a rule and hold down the right mouse button.

Delete

Use the Delete function to delete a selected rule or subscription. You can only delete a subscription from the database if the subscription is not a component in another rule.

If you have a rule or subscription selected and press the right mouse button, the pop-up menu containing the Delete and Duplicate functions appears.

If you are not the owner or do not have Update permission for a rule or subscription, the Delete function is inactive, preventing you from deleting the rule or subscription.

Remove

Use Remove to remove a selected subscription from a rule. The subscription will still be in the database.

If you have a subscription selected under the rule and right-click the mouse button, the pop-up menu containing the Remove function appears.

Duplicate

Use the Duplicate function to duplicate rules and subscriptions. When you select Duplicate, a text entry box appears at the top of the component list. You must enter a unique name. The component's property sheet displays the properties of the rule or subscription that you duplicated. You can change any of these properties.

Right Pane

The right window pane contains a pop-up menu when a component list is displayed. This pop-up menu contains the following functions:

- View
- Arrange Icons
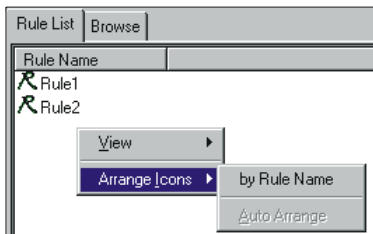
Right Window Pane Pop-up Menu

View

The View function lets you view icons either large or small (the default). You can also select to arrange the icons in a list (the default).

If you select to view large icons, they are displayed from left to right, rather than in a list. To see the icons in a list, select the List function.

Arrange Icons

The Arrange Icons function lets you change the order of the icons. By default, the icons are alphabetically displayed from a-z and numerically displayed in ascending order. When you select Arrange Icons by (for example, Rule Name), the icons are then arranged alphabetically from z-a and numerically in descending order. The sort order is case sensitive.

Building Rules

Application Groups

An application group is a method that lets you logically organize rules associated with a particular subject. For example, an application group could be the Accounting Department of a company.

Adding an Application Group

To add an application group:

1. Choose one of the following:

   ■ Select the New Application icon from the toolbar.

   ■ Select Insert→New Application from the menu bar.

   A text box appears at the top of the application group list. The cursor is positioned in the application group text box where you can type the new, unique application group name.

   Note:
   The application group name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

   WARNING!
   If you are using a case-insensitive database, you cannot use the same name with a change in case to identify components. For example, you cannot name one application group D1 and another d1. In a case-insensitive environment, make each item unique using something other than case differences.

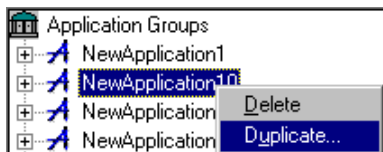2. Press ENTER to add the application group.

Note:

If you click any mouse button outside the application group text box (and you have not pressed ENTER), the application group is added.

The application group is highlighted and alphabetically positioned in the list in the left pane and the New Message property sheet appears in the right pane.

An application group should contain at least one message type, which corresponds to your input formats. For the procedure to add a message type, refer to the section *Adding a Message Type* on page 155.

Renaming an Application Group

To change an application group name, select the name that you want to change and right-click on the name. The name highlights and a cursor appears. Type the new, unique name and press ENTER.
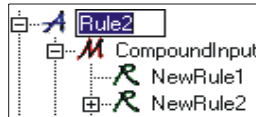
An Application Group Selected for Renaming

> **Note:**
>
> The application group name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

Duplicating an Application Group

To duplicate an application group:

1. Select the application group you want to duplicate.

2. Hold down the right mouse button to activate the pop-up menu and select Duplicate.

3. The cursor is positioned in the application group text box. Type the new, unique application group name. Press ENTER to copy the application group and its components (messages, rules, and subscriptions) to the new application group name.

   > **Notes:**
   >
   > The application group name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.
   >
   > If you click any mouse button outside the application group text box (and you have not pressed ENTER), the new name is saved.

Deleting an Application Group

To delete an application group:

1. Select the application group you want to delete.

2. Hold down the right mouse button to activate the pop-up menu and select Delete. The delete confirmation box appears.

3. To delete the application group, click OK. The Delete box closes and the application group and its components (messages, rules, and subscriptions) are deleted.

   Before any item is deleted, security is checked. After security is checked, one of the following occurs:

   ■ The rule or subscription is deleted if the item's update and owner box are checked in the security tab for the current user.

   ■ The rule or subscription is deactivated if the item's update box is checked and the owner box is NOT checked in the security tab for the current user. This causes the deactivate box to be checked.

   ■ The rule or subscription remains if the item's update and owner box are NOT checked in the security tab for the current user.

   If the item cannot be deleted in the application group, an error messages displays: *Unable to delete entire application group.* The application group remains with the items that could not be deleted.

Message Types

The message type is the input format name from Formatter. A rule is evaluated based on the message type.

An application group should contain at least one message type. For the procedure to add an Application Group, see *Adding an Application Group* on page 151.

Adding a Message Type

Use the New Message property tab to add a message type to the selected application group.

To add a message type:

1. Select the application group that you want to add a message type to. The list of current message types and the New Message tab is displayed in the right pane.

2. Select the New Message property sheet.
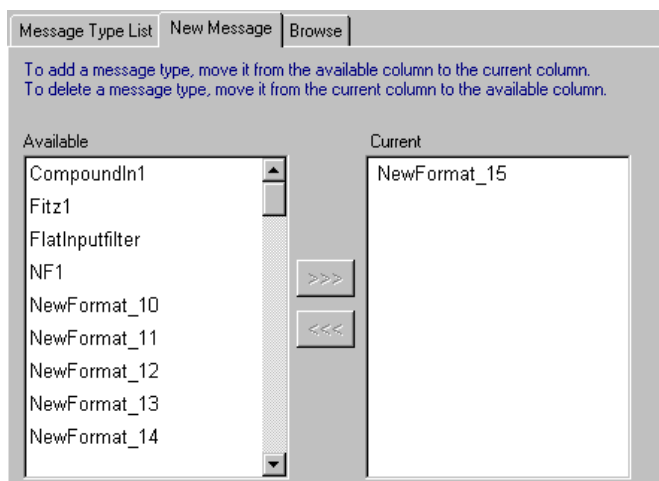
New Message Property Sheet

The Available box on the left displays available message types. This list contains the input format names defined in Formatter.

The Current box on the right displays the message types currently associated with the application group.

3. To add a message type to the selected application group, do one of the following:

   ■ Select the message type from the Available list box, hold down the left mouse button and drag the message type to the Current list box.

- Select the message type from the Available list box, double-click the message type.

- Select the message type from the Available list box, and click

  >>> .

The message type appears both in the Current list box and in the left pane below the Application Group.

Note:

To delete a message type from the selected application group, do one of the following:

- Select the message type from the Current list box, hold down the left mouse button and drag the message type to the Available list box.

- Select the message type from the Current list box, double-click the message type.

- Select the message type from the Current list box and click

  <<< .

Duplicating a Message Type

To duplicate a message type to a different application group:

1. Select the message type to duplicate.

2. Do one of the following:

   - Drag the message type to the application group you want to add it to.

   - Hold down the right mouse button to activate the pop-up menu and select Copy.
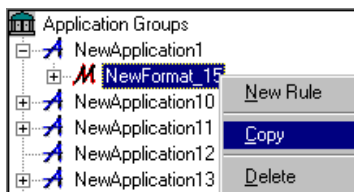
The message is stored in memory for pasting to an application group. To paste the message, select the application group you want to add it to and hold down the right mouse button to activate the pop-up menu and select Paste Message.
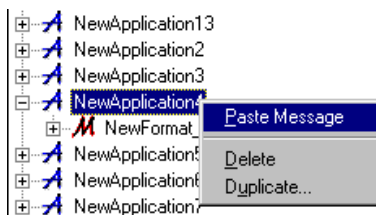
Notes:

The message type name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

> If you click any mouse button outside the message type text box (and you have not pressed ENTER), the new name is saved.

Deleting a Message Type

To delete a message type:

1. Select the message type you want to delete.

2. Hold down the right mouse button to activate the pop-up menu and select Delete. The Delete confirmation box appears.

3. To delete the message type, click OK. The Delete box closes and the message type is deleted.

Rules

A rule contains subscriptions which allow you to define message destination IDs, receiver locations and message formats, and any processes initiated upon message delivery.

Adding a Rule

To add a rule:

1. From the Rules window tree, select the message type that you want to define a rule for and click the right mouse button. A pop-up menu appears with the New menu item.
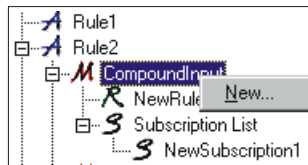
Add New Rule

2. Select New. The cursor is positioned in the rule text box where you can type the new, unique rule name.

Note:

The rule name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

WARNING!

If you are using a case-insensitive database, you cannot use the same name with a change in case to identify components. For example, you cannot name one rule R1 and another r1. In a case-insensitive environment, make each item unique using something other than case differences.

---

3. Press ENTER to add the rule.

Note:

If you click any mouse button outside the rule text box (and you have not pressed ENTER), the rule is added.

The rule is highlighted and alphabetically positioned in the list to create a new rule in the left pane. The Expressions property sheet appears in the right pane. In addition to the Expression property sheet, you can access the Subscription List and the Security property sheets via their tabs.

For the Rules engine to correctly process a rule, you must define the rule's expression, subscription, and user permissions. The procedure to add an expression to a rule is described in the *Expressions* section on page 165. The procedure to add a subscription to a rule is described in *Subscriptions* on page 179. The procedure to add user permissions to a rule is described in the *Security* section on page 142.

Deleting a Rule

When you delete a rule, the expression and links to subscriptions belonging to the rule are also deleted. The subscriptions are not deleted.

You must have Update permission or be the Rule owner to delete a rule. If you do not own the rule, the rule is disabled. For more information on enabling and disabling rules, refer to See *Security* on page 142.

To delete a rule:

1. Select the rule that you want to delete.

2. Hold down the right mouse button to activate the pop-up menu and select Delete. The Delete confirmation box appears.

3. To delete the rule, click OK. The Delete box closes and the rule is deleted.

Duplicating a Rule

The Duplicate function lets you duplicate the selected rule, which includes the rule's associated expression and its links to subscriptions. The new rule is owned by the current user who has Update permission.

To duplicate a rule:

1. Select the rule that you want to duplicate.

2. Hold down the right mouse button to activate the pop-up menu and select Duplicate.
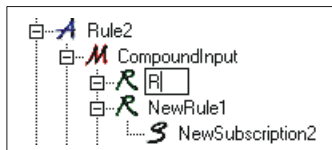
Duplicate Rule Text Entry Box

3. The cursor is positioned in the rule box where you can type the new, unique rule name. Press ENTER to save the name.

Note:

The rule name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

WARNING!

If you are using a case-insensitive database, you cannot use the same name with a change in case to identify components. For example, you cannot name one rule R1 and another r1. In a case-insensitive environment, make each item unique using something other than case differences.

Note:

If you click any mouse button outside the rule text box (and you have not pressed ENTER), the new name is saved.

---

4. To change the user permissions for the new rule, select the Security tab. For more information, refer to the *Security* section on page 142.

5. To change the rule's expressions, select the Expression tab and make the changes. For more information, refer to the *Expressions* section on page 165.

6. To change a rule's subscriptions, select the subscription and make the changes. For more informations, refer to the *Subscriptions* section on page 179.

Renaming a Rule

Use the Rename Rule function to change the name of a rule. You must have Update permission to change a rule's name.
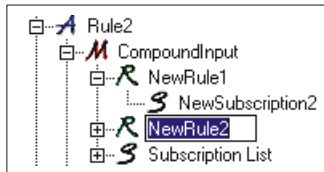
Rule Selected for Renaming

To change a rule name, select the name that you want to change and left-click on the name. The cursor is positioned in the rule box where you can type the new, unique rule name. Press ENTER to save the name.

Note:

The rule name must be 32 characters or less. If you exceed 32 characters, a message box appears and you can either edit the name or cancel the operation.

WARNING!

If you are using a case-insensitive database, you cannot use the same name with a change in case to identify components. For example, you cannot name one rule R1 and another r1. In a case-insensitive environment, make each item unique using something other than case differences.


Rule Security

Use the Security property sheet to enter permissions for an individual rule or subscription. When you add a rule, you are assigned ownership, and must set the permissions for the rule. By default, Rules designates the creator of a rule as the owner, who is given read and update privileges. To give other users the ability to change your rule, give PUBLIC user Update permission. PUBLIC defaults to read only. Read permission is the minimum security currently allowed. The Read checkbox is marked by default and cannot be changed.

When a user deletes a rule and has Update permission, but does not own the rule, the rule is deactivated instead of deleted. The rule appears in the tree, but cannot be changed because it has been deactivated. Users can see that a rule has been deactivated by opening the Security tab and viewing the Disable checkbox at the bottom of the window. A checked box indicates that the rule has been deactivated.

Adding or Changing Rule Security

To add or change user permissions:

1. Select the rule that you want to add or change the user permission for.

2. Select the Security tab. The Security property sheet appears displaying a User List containing the names of all users in the same user group plus the PUBLIC user name. The User Name column in the Permissions box displays users who are assigned permission to the selected rule.
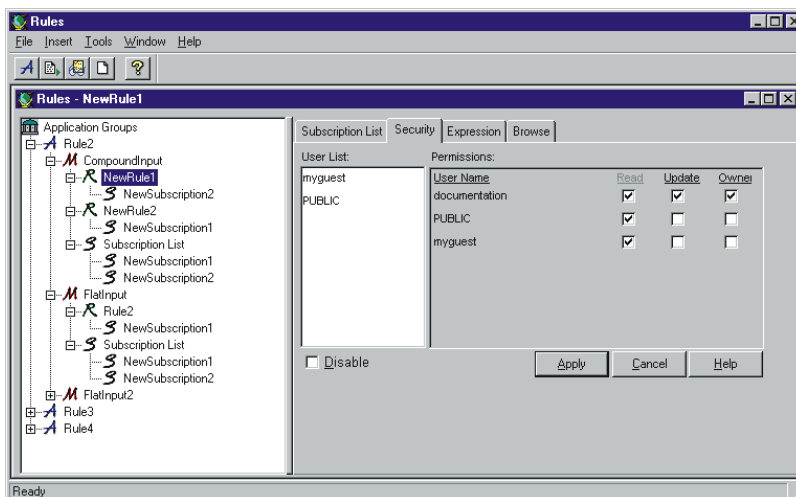
Rules Permissions Dialog Box

3. To add a user to the User Name list in the Permissions Dialog Box, select the user from the User List, hold down the left mouse button and drag the name to the Permissions box.

Note:

Currently, you cannot deny Read permission.

4. To allow the user to update the rule, select the Update checkbox.

5. To have the user own the rule, select the Owner checkbox. Only one user can own a rule, so when you select this checkbox, the Owner checkbox associated with your username is automatically cleared.

6. To disable the rule, the owner can select the Disable checkbox. Only the owner of a rule can enable/disable the rule using this checkbox.

Note:

If a user with Update permission has "deleted" a rule that you own, the rule is really disabled, not deleted. You can uncheck the Disable checkbox to enable the rule.

7. When you are done adding (or changing) users permissions, choose one of the following:

   ■ To add the user permission, click Apply.

   ■ To delete the user permissions you added and return the property sheet to its original configuration, click Cancel.

Note:

If you move the cursor to the left pane and click the mouse button after you add user permissions, the permissions are automatically saved.

Expressions

The Expression tab is used to view, create, modify, or delete an expression for a selected rule.

The following tabs appear within the Expression tab:

- Expression Components: combinations of expression strings used to automate typing for a user.

- Field List: list of fields that appear in the format.

- Operators: comparison operator choices. This list also appears in the pop-up menu.

- Values: masked values for integers, strings, etc.

- Functions: symbols used in the expression such as & (and) and | (or), and parentheses.

The evaluation criteria for a rule consists of a Boolean expression containing the Boolean operators "&" (AND) and "|" (OR), expressions, and parentheses to control the order of evaluation. You can use OR to explicitly direct what Boolean operations do together.

There must be at least one space between the field name and the Rules operator as well as between the Rules operator and the comparison value. The EXIST and NOT_EXIST operators should be followed by at least one space before a parenthesis or a Boolean operator.

Expression Components

The Expression Components tab contains expression templates that allow you to create an expression for a rule. An expression contains arguments (operation expressions, combined with Boolean operators) and parentheses.

Expressions consist of a field name, Rules operator, and either another field name (field-to-field comparisons), a static value (static comparisons), or nothing (existence operators). Boolean operators (Functions) include the & (AND) and | (OR) operators to combine arguments into an expression. Boolean algebra's precedence rules are followed when evaluating an expression, with the innermost parentheses evaluated first, and ANDs evaluated before ORs if no parentheses are present.

Field List

The Field List contains the field names associated with the selected message type. In the Formatter GUI, fields and flat input formats (messages) are created. Fields are assigned to input formats (messages). A rule's message type is (1) a flat input format that contains the fields or (2) a compound input format that contains a flat input format, which in turn contains the fields. Rules field names and message names are identical to the field names and flat input format names in Formatter.

Note:
If the field name or static comparison value contains spaces, quotes, or parentheses, the item must be enclosed in quotes (either single or double-- whatever the value does NOT have). A value must not have both single and double quotes.

Operators

A Rules operator is a data-type comparison with a field. Rules operators can be field existence, field non-existence, or include the following operators:

- <, <=, >, >=, <>, = for INT (whole number),
- FLOAT (decimal number),
- DATE, TIME, DATETIME, and STRING fields,
- field-to-field comparisons (e.g., field1 compares against field2), and
- case-sensitive string comparisons (e.g., where "a" does not equal "A").

Specific operators apply only to a certain data type. For example, there are no operators that compare integer and float data types. Operators are executed against a field name and a value, or between two field names (both field names, as defined by the message type, must be of the same data type). You can use existence operators to determine the existence of a field in a message. Integer, string, and float operators evaluate a message field against a static value using the operator symbol. Field-to-field operators compare two groups of data (fields) within the message.

An operator is defined by type (existence, integer, string, float, date, time, datetime, field-to-field) and associated symbol.

Existence operators enable a user to determine if a field exists and is not empty in a message. Integer, string, float date, time and datetime operators evaluate a message field against a static value using the operator symbol. Field-to-field operators compare two groups of data (fields) within the message.

Operators, except for NOT_EXIST and NOT_EXIST_TRIM, will not hit if a field does not exist or is empty.

For a more detailed explanation of a field, see the MQSeries Integrator Rules Programming Reference for NEONRules.

Existence Operators

Existence operators determine if a field exists or is empty in a message. Existence operators have a TRIM option that trims trailing blanks prior to determining whether a field exists or is empty, thus making a string of blanks a nonexistent field.

Integer Operators

Integer operators compare numeric values. For static value comparisons, the comparison value must be a whole number (which can preceded by '+' or '-'). If the message field is not numeric, its value is assumed to be zero (0), so a rule may hit in this case.

String Operators

String operators compare strings of characters. Case-sensitive operators evaluate the characters 'a' and 'A' differently. Rules may work differently on different platforms. For example, on an EBCDIC machine, the order of characters is: 'a'- 'z' < 'A'- 'Z' < '0' - '9'. In ASCII, the order of characters is: '0' - '9' < 'A' - 'Z' < '0' - '9'. String operators (including field to field, case sensitive, and field to field case sensitive operators) can have a TRIM option that trims trailing blanks prior to comparing fields. For the TRIM operators, trailing blanks are truncated from message fields and comparison values. Therefore, a field containing a string of trailing blanks is considered empty.

Float Operators

Float operators compare decimal (real) numeric values. For static value comparison, the comparison value must be a numeric value (which can be preceded by '+' or '-') and contain a decimal point ('.'). When comparing float values, '1.5' does not always equal '1.5' because of real number precision.

Rules Date, Time, and DateTime Operators

The Rules Date, Time, and DateTime operators are used to create and evaluate the rule arguments that perform Date, Time, and DateTime comparisons. Rules performs comparisons between unmatched Date, Time, and DateTime types based on the operator used in the argument. The Date operators compare the date portion (i.e., YYYYMMDD), the Time operators, the time portion (i.e., hhmmss) and DateTime operators, the entire value (i.e., YYYYMMDDhhmmss). For example, if an argument using a Date operator compares a Date against a DateTime (e.g., F1 F2FDATE=F2, where F1 is a Date and F2 is a DateTime), then the value of the first field (F1) will be compared against only the date portion of the second field (F2).

Note:
The visual representation of dates in the GUI does not adhere to the standard DateTime format (i.e., YYYYMMDD and hhmmss). However, the Management APIs must receive Date, Time, and DateTime values in the standard DateTime format.

Specifying a Year Cutoff Value

The internal application functions of MQIntegrator use date-time information for archiving, time stamping, logging, etc. These functions use 4-digit year notation or Universal Time Coordinated (UTC) for time stamps and are Y2K compliant.

You can use the Date and DateTime comparison functions within the message handling and processing functionality to manipulate the date information (e.g., embed or reformat). Date/DateTime formats and supported Date/DateTime rules facilities are Y2K compliant for accepting input and providing output date information. Default Date and DateTime formats use 4-character years to make them Y2K compliant. MQIntegrator also supports 2-character years as custom formats.

For an input control that specifies a data type of custom Date/Time with a 2-digit year format string, you must specify a "year cutoff" value. This value tells Formatter how to convert the 2-digit year date value to a 4-digit year date value.

To perform this conversion, Formatter compares the year value of the input data to the specified Year Cutoff value and assigns the century designation as required. That is, based on the comparison, Formatter converts the year value "XX" to "20XX" (21st century year) or "19XX" (20th century year) as appropriate.

The year cutoff algorithm is as follows:

- year value >= cutoff value: 19XX

- year value < cutoff value: 20XX

Valid year cutoff values: 0 to 99 inclusive. With this method, any year 00 to 99 can be converted into either 19XX or 20XX.

The following are examples of how Formatter interprets the Year Cutoff number:

- If you specify the Year Cutoff number as 50, all 2-digit input dates from 50 to 99 are designated as 1950 to 1999 output dates. All 2-digit input dates from 00 to 49 are designated as 2000 to 2049 output dates.

- If you specify the cutoff date as 75, all 2-digit input dates from 75 to 99 are designated as 1975 to 1999 output dates. All two-digit input dates from 00 to 74 are designated as 2000 to 2074 output dates.

If the output data type is a custom date and time, you must specify a format (date/time string). The formats that you can specify are predefined by Formatter.

Data Types and Operators

Operator Symbol	Description
Existence Operators	
NOT_EXIST	Required Field Is Not Present
NOT_EXIST_TRIM	Required Field Is Not Present (After Trimming)
EXIST	Required Field Is Present
EXIST_TRIM	Required Field Is Present (After Trimming)
Integer Operators	
INT=	Integer Equals
INT>	Integer Greater Than
INT<	Integer Less Than
INT>=	Integer Greater Than Or Equal To
INT<=	Integer Less Than Or Equal To
INT<>	Integer Not Equal To
String Operators	
STRING=	String Equal To
STRING_TRIM=	String Equal To (After Trimming)
STRING>	String Greater Than
STRING_TRIM>	String Greater Than (After Trimming)
STRING<	String Less Than
STRING_TRIM<	String Less Than (After Trimming)
STRING_TRIM>=	String Greater Than Or Equal To (After Trimming)
STRING>=	String Greater Than Or Equal To

Operator Symbol	Description
STRING<=	String Less Than Or Equal To
STRING_TRIM<=	String Less Than Or Equal To (After Trimming)
STRING<>	String Not Equal To
STRING_TRIM<>	String Not Equal To (After Trimming)
Field To Field Integer Operators	
F2FINT=	Field To Field Integer Equal To
F2FINT>	Field to Field Integer Greater Than
F2FINT<	Field to Field Integer Less Than
F2FINT>=	Field to Field Integer Greater Than Or Equal To
F2FINT<=	Field to Field Integer Less Than Or Equal To
F2FINT<>	Field To Field Integer Not Equal To
Field To Field String Operators	
F2FSTRING=	Field To Field String Equal To
F2FSTRING_TRIM=	Field To Field String Equal To (After Trimming)
F2FSTRING>	Field To Field String Greater Than
F2FSTRING_TRIM>	Field To Field String Greater Than (After Trimming)
F2FSTRING<	Field To Field String Less Than
F2FSTRING_TRIM<	Field To Field String Less Than (After Trimming)
F2FSTRING>=	Field To Field String Greater Than Or Equal To

Operator Symbol	Description
F2FSTRING_TRIM>=	Field To Field String Greater Than Or Equal To (After Trimming)
F2FSTRING<=	Field To Field String Less Than Or Equal To
F2FSTRING_TRIM<=	Field To Field String Less Than Or Equal To (After Trimming)
F2FSTRING<>	Field To Field String Not Equal To
F2FSTRING_TRIM<>	Field To Field String Not Equal To (After Trimming)
Float Operators	
FLOAT=	Float Equals
FLOAT>	Float Greater Than
FLOAT<	Float Less Than
FLOAT>=	Float Greater Than Or Equal To
FLOAT<=	Float Less Than Or Equal To
FLOAT<>	Float Not Equal To
Case Sensitive String Operators	
CSSTRING =	Case Sensitive String Equal To
CSSTRING_TRIM=	Case Sensitive String Equal To (After Trimming)
CSSTRING>	Case Sensitive String Greater Than
CSSTRING_TRIM>	Case Sensitive String Greater Than (After Trimming)
CSSTRING<	Case Sensitive String Less Than
CSSTRING_TRIM<	Case Sensitive String Less Than (After Trimming)

Operator Symbol	Description
CSSTRING>=	Case Sensitive String Greater Than Or Equal To
CSSTRING_TRIM>=	Case Sensitive String Greater Than Or Equal To (After Trimming)
CSSTRING<=	Case Sensitive String Less Than Or Equal To
CSSTRING_TRIM<=	Case Sensitive String Less Than Or Equal To (After Trimming)
CSSTRING<>	Case Sensitive String Not Equal To
CSSTRING_TRIM<>	Case Sensitive String Not Equal To (After Trimming)
Field To Field Case Sensitive Operators	
F2FCSSTRING=	Field To Field Case Sensitive String Equal To
F2FCSSTRING_TRIM=	Field To Field Case Sensitive String Equal To (After Trimming)
F2FCSSTRING>	Field To Field Case Sensitive String Greater Than
F2FCSSTRING_TRIM>	Field To Field Case Sensitive String Greater Than (After Trimming)
F2FCSSTRING<	Field To Field Case Sensitive String Less Than
F2FCSSTRING_TRIM<	Field To Field Case Sensitive String Less Than (After Trimming)
F2FCSSTRING>=	Field To Field Case Sensitive String Greater Than Or Equal To
F2FCSSTRING_TRIM>=	Field To Field Case Sensitive String Greater Than Or Equal To (After Trimming)

Operator Symbol	Description
F2FCSSTRING<=	Field To Field Case Sensitive String Less Than Or Equal To
F2FCSSTRING_TRIM<=	Field To Field Case Sensitive String Less Than Or Equal To (After Trimming)
F2FCSSTRING<>	Field To Field Case Sensitive String Not Equal To
F2FCSSTRING_TRIM<>	Field To Field Case Sensitive String Not Equal To (After Trimming)
Date Operators	
DATE=	Date Equal To
DATE>	Date Greater Than
DATE<	Date Less Than
DATE>=	Date Greater Than Or Equal To
DATE<=	Date Less Than Or Equal To
DATE<>	Date Not Equal To
Field To Field Date Operators	
F2FDATE=	Field To Field Date Equal To
F2FDATE>	Field To Field Date Greater Than
F2FDATE<	Field To Field Date Less Than
F2FDATE>=	Field To Field Date Greater Than Or Equal To
F2FDATE<=	Field To Field Date Less Than Or Equal To
F2FDATE<>	Field To Field Date Not Equal To
Time Operators	
TIME=	Time Equal To

Operator Symbol	Description
TIME>	Time Greater Than
TIME<	Time Less Than
TIME>=	Time Greater Than Or Equal To
TIME<=	Time Less Than Or Equal To
TIME<>	Time Not Equal To
Field to Field Time Operators	
F2FTIME=	Field To Field Time Equal To
F2FTIME>	Field To Field Time Greater Than
F2FTIME<	Field To Field Time Less Than
F2FTIME>=	Field To Field Time Greater Than Or Equal To
F2FTIME<=	Field To Field Time Less Than Or Equal To
F2FTIME<>	Field To Field Time Not Equal To
DateTime Operators	
DATETIME=	DateTime Equal To
DATETIME>	DateTime Greater Than
DATETIME<	DateTime Less Than
DATETIME>=	DateTime Greater Than Or Equal To
DATETIME<=	DateTime Less Than Or Equal To
DATETIME<>	DateTime Not Equal To
Field to Field DateTime Operators	
F2FDATETIME=	Field To Field DateTime Equal To
F2FDATETIME>	Field To Field DateTime Greater Than
F2FDATETIME<	Field To Field DateTime Less Than

Operator Symbol	Description
F2FDATETIME>=	Field To Field DateTime Greater Than Or Equal To
F2FDATETIME<=	Field To Field DateTime Less Than Or Equal To
F2FDATETIME<>	Field To Field DateTime Not Equal To

Values

The Values tab contains the matching data types of the MQIntegrator operator in the Operation Expression field.

Note:
If the field name or static comparison value contains spaces, quotes, or parentheses, the item must be enclosed in quotes (either single or double-- whatever the value does NOT have). A value must not have both single and double quotes.

The following is a list of operators and their values:

Operator	Value
INT	Whole numbers in the integer range for the platform used, which are usually about -2.1 billion to about 2.1 billion. Non-numeric characters are not allowed except for a + or - sign as the first character. Do not use a decimal point.
FLOAT	Whole number in the integer range for the platform used, which is usually about -2.1 billion to 2.1 billion and a decimal mantissa being a whole number with the maximum of 32 digits. No non-numeric characters are allowed except for a + or - sign as the first character. A decimal point must be used.
STRING	Null-terminated strings with at most 64 characters.

Functions

The Functions tab contains the Boolean operators AND and OR and parentheses for use either with a single expression or more than one expression connected with Boolean operators with optional parentheses.

Creating or Modifying an Expression

You must have update permission for the associated rule to modify an expression.

To create or modify an expression:

1. Select the appropriate rule.

2. Select the Expression tab.
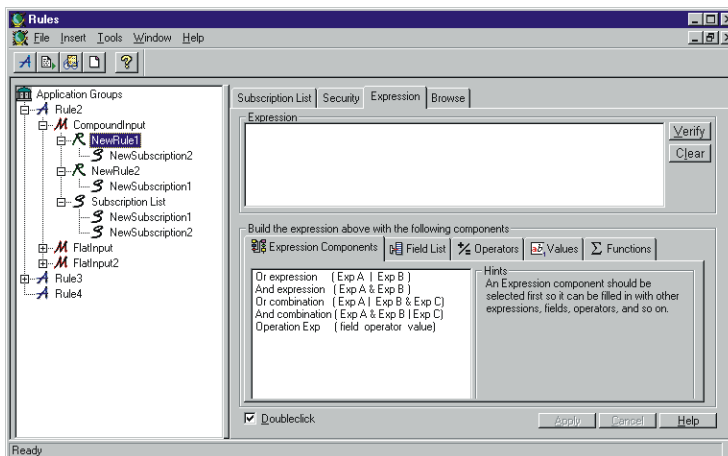
Expression Tab

3. Select the Expression Components tab. Select (by clicking or double-clicking) an Expression Component in the Rules tree to display it in the Expression tab.

Note:

Use the Doubleclick checkbox to modify mouse action to a click or a double-click when selecting expression components.

---

4. Use the Field List, Operators, Values, and Functions tabs to add components to your expression.

Note:

You must enter a valid field name. Otherwise, a message box appears informing you the information is incorrect.

The Literals AND and OR do not evaluate correctly in a Rules expression. If you enter these values, a message box appears stating the expression is okay, but AND will be changed to &, and OR will be changed to |.

You can type the expression directly into the Expression box, without using the component tabs. You can modify the expression by right clicking within the Expression window to access the text edit menu. Use the following text edit functions to modify your expression: Undo your last change, Cut, Copy, Paste, Delete, and Select All of the text within the Expression window.

---

5. Click Verify to validate the final expression. This tests the expression, but does not save it to the database.

6. To clear the window and restart, click Clear.

7. To either create or update an expression, click Apply.

Note:

If you move out of the Expression tab either by using the mouse or the keyboard, the expression is automatically saved.

---

8. To cancel and reload the existing expression from the database, click Cancel.

Expression Examples

Description	Layout	Expression
Single Argument	A	F1 STRING= NEONet
Arguments Anded Together	A & B & C	F1 STRING= NEONet & F2 INT= 100 & F3 INT= 150
Arguments Ored Together	A | B | C	F1 STRING= NEONet | F2 INT= 100 | F3 INT= 150
Precedence	A | B & C	F1 STRING= NEONet | F2 INT= 100 & F3 INT= 150
Nested Parens	(A | ((B & (C)) | D))	(F1 STRING= NEONet | ((F2 INT= 100 & (F3 INT= 150)) | F4 INT= 200))

Deleting an Expression

You can delete an expression for a selected rule. You must have Update permission for the rule to delete an expression.

To delete an expression:

1. Select the rule that contains the expression you want to delete. The expression is displayed in the Expression tab.

2. Click Clear. The expression is no longer associated with the selected rule.

3. Click Apply to update the database.

Subscriptions

After a rule is created, subscriptions that contain actions should be added to the rule's Subscription List. Each application/message group has its own Subscription List. You can drag and drop one or more subscriptions to any rule from the Subscription List or drag and drop into the Subscription List tab of a rule.

A subscription describes the actions that are performed if the rule's expressions pass evaluation. A subscription allows you to define message

destination IDs, receiver locations, and message formats, and any processes initiated upon message delivery. Subscriptions have permissions similar to Rules permissions.

A subscription can only be assigned to a rule within the same application group/message type. You must have Update permission for the rule to add a subscription to a rule.

Note:

Subscription Sharing: A subscription can be assigned to several rules if the rules are in the same application group and message type.

Adding a Subscription

You must have Update permission for the rule to add a subscription to a rule.

To add a subscription:

1. From the Rules window tree, select the Subscription List for the Application Group/Message Type that you want to add a subscription to and click the right mouse button. A pop-up menu appears.

2. Select New. The cursor is positioned in the Subscription List text box where you can type the new, unique subscription name.

Note:

The subscription name must be 64 characters or less. If you exceed 64 characters, a message box appears and you can either edit the name or cancel the operation.

WARNING!

If you are using a case-insensitive database, you cannot use the same name with a change in case to identify components. For example, you cannot name one subscription S1 and another s1. In a case-insensitive environment, make each item unique using something other than case differences.

3. To add the subscription to the Subscription List, press ENTER.

Note:

If you click either mouse button outside the subscription text box (and you have not pressed ENTER), the subscription is added.

If there is no duplicate name for this subscription, the subscription is added and the Actions tab appears.

Each subscription should have at least one action. For the procedure to add an action, see *Adding an Action* on page 186.

Duplicating a Subscription

Use the Duplicate function to duplicate the selected subscription. The new subscription is owned by the current user who has Update permission.

To duplicate a subscription:

1. Select the subscription that you want to duplicate from the Subscription List.

2. Hold down the right mouse button to activate the pop-up menu and select Duplicate. A text entry box appears at the top of the rules list.

3. Type a unique subscription name in the text box and press ENTER to save the name.

Note:

The subscription name must be 64 characters or less. If you exceed 64 characters, a message box appears and you can either edit the name or cancel the operation.

WARNING!

If you are using a case-insensitive database, you cannot use the same name with a change in case to identify components. For example, you cannot name one rule R1 and another r1. In a case-insensitive environment, make each item unique using something other than case differences.

> **Note:**
> If you click any mouse button outside the subscription text box (and you have not pressed ENTER), the new name is saved.

4. To change the user permissions for the new subscription, select the Security tab.

   For more information on security, refer to *Adding an Action* on page 186.

Renaming a Subscription

Use the Rename Subscription function to change the name of a subscription. You must have Update permission to change a subscription's name.

To change a subscription name, select the name that you want to change and click on the name. A box appears around the subscription name and the name is highlighted (see the figure below). Type the new, unique name and press ENTER to save the name.

> **Note:**
> The subscription name must be 64 characters or less. If you exceed 64 characters, a message box appears and you can either edit the name or cancel the operation.

> **WARNING!**
> If you are using a case-insensitive database, you cannot use the same name with a change in case to identify components. For example, you cannot name one subscription S1 and another s1. In a case-insensitive environment, make each item unique using something other than case differences.

Removing a Subscription

Use the Remove function to remove a subscription from a rule. You cannot remove a subscription from the Subscription List, only from a rule. After you remove the subscription from the rule, the subscription will still be in the Subscription List.

To remove a subscription:

1. Select the subscription you want to remove from the specific rule.

2. Click the right mouse button to activate the pop-up menu and select Remove. The Remove confirmation box appears.

3. To remove the subscription, click OK. The remove box closes and the subscription is removed from the rule.
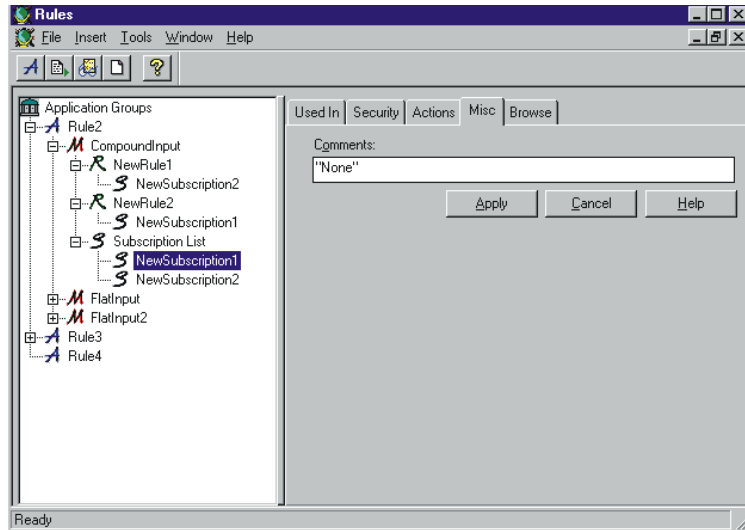
Deleting a Subscription

A subscription can only be deleted if it is not linked to any Rules from the Subscription List. You must be the subscription owner and have Update permission to delete a subscription.

If you own the subscription and the subscription is not linked to any rules, the subscription is deleted. If you do not own the subscription and if the subscription is not linked to any rules, the subscription is disabled. For more information on enabling and disabling subscriptions, see *Subscription Security* on page 184.

To delete a subscription:

1. Select the subscription that you want to delete from the Subscription List.

2. Click the right mouse button to activate the pop-up menu and select Delete. The Delete confirmation box appears.

3. To delete the subscription, click OK. The delete box closes and the subscription is deleted.

Adding a Comment to a Subscription

You can associate a comment with a subscription.

To add a comment:

1. Select the subscription for which you want to add a comment from the Subscription List.

2. Select the Misc tab. A Comments text entry box appears containing the word "None" (if no comments were previously added).

Misc. Tab's Comments Text Box

3. Select and delete the word None and type your comment; then click Apply.

Subscription Security

Use the subscription's Security property to enter permissions for an individual subscription. When you add a subscription, you are assigned ownership and you must set the permissions for the subscription. By default, Rules designates the creator of a subscription as the owner, who is given Read and Update privileges. Owners should give Update permission if they want others to be able to change the subscription. PUBLIC defaults to read only. Read permission is the minimum security currently allowed. The Read checkbox is marked by default and cannot be changed.

Owners should give Update permission if they want others to be able to change the subscription. When a user deletes a subscription and has Update permission, but does not own the subscription, the subscription is deactivated instead of deleted. The subscription appears in the tree, but does not allow

changes because it has been deactivated. Users can see that a subscription has been deactivated by opening the Security tab and viewing the Disable checkbox at the bottom of the window. A checked box indicates that the subscription is deactivated.

Adding or Changing Subscription Security

To add or change user permissions:

1. Select the subscription that you want to add (or change) the user permission for.

2. Select the Security tab. The Permissions tab appears displaying a User List containing the names of all users in the same user group plus the PUBLIC user name. The User Name column in the Permissions box displays users that are assigned permission to the selected subscription.

3. To add a user to the User Name list, select the user from the User List and drag the name to the Permissions box.

    Note:
    Currently, you cannot deny Read permission.

4. To allow the user to update the subscription, select the Update checkbox.

5. To have the user own the subscription, select the Owner checkbox. Only one user can own a subscription, so when this checkbox is selected, the Owner checkbox associated with your username automatically loses its check mark.

6. To disable the subscription, the owner can select the Disable checkbox.

    Note:
    If a user with Update permission has "deleted" a subscription that you own, the subscription is really disabled, not deleted. As the owner, you can uncheck the Disable checkbox to enable the subscription.

7. When you are done adding (or changing) permissions, choose one of the following:

   - To add the user permissions, click Apply.

   - To delete the user permissions you added and return the tab to its original configuration, click Cancel.

Note:

If you move the cursor to the left pane and click the mouse button after you add user permissions, the permissions are automatically saved.

Actions

Actions hold subscription instructions. The three Actions affecting subscriptions are called Generic, Reformat, and Put Message. From the Actions tab, you can add (to the bottom of the list), insert, and delete actions.

You must have Update permission for the subscription to add an action.

Adding an Action

To add an action to a subscription:

1. Select the subscription in which you want to add an action. The Actions tab appears.
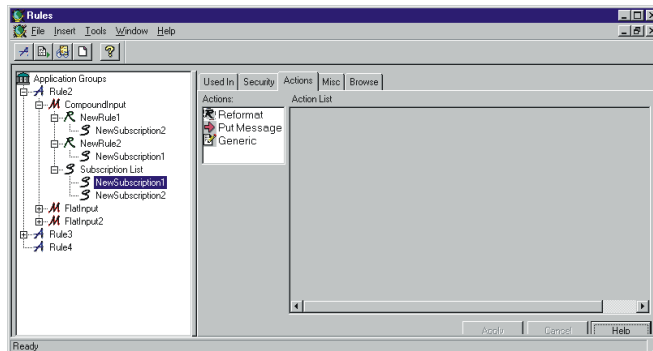
Subscription Selected - Actions Tab Displayed

The Actions box displays the following actions:

Action	Description
Reformat	Reformats a message. Using Formatter, the message format is defined both at input and at output. The Rule Engine daemon uses an input format (what the message looks like coming into Rules) and an output format (what the message looks like after the reformat) to reformat the message contents.
Put Message	Instructs Rules to route (deliver) a message to a specific queue. You define the queue name and any other message options to be set using Rules.
Generic	Allows you to create a user-defined action. Generic actions can have one or more options and the option name and option values can be changed. You can rename a generic action.
ReEvaluate	Adds a re-eval action to any subscription (perhaps after reformat). When the Rules daemon encounters this action, it saves the pertinent information so that once all the other actions are performed for the original evaluation, the Rules instance can be called once again with the new message, application group, and message type. The transaction that began with the get from the input queue does not commit until all the successive re-evaluations (and their subsequent subscriptions) are performed.

2. Select the action that you want to add, hold down the left mouse button, and drag the action to the Action List box. Then do one of the following:

- To add the action to the bottom of the list, drop the action onto the gray space below the last action in the list.

- To insert the action above an existing action, drop the action on top of the position you want the action to be above.

   Note:
   The parameters you enter into in the Action Values box should be based on the type of action selected.

- If you select the **Reformat action**, when it is added to the Action List box, the options INPUT_FORMAT and TARGET_FORMAT appear. These options each have a mandatory associated value. You can either enter the value (format name), or select it from its drop-down list. The INPUT_FORMAT drop-down list displays only input formats and the TARGET_FORMAT drop-down list displays only output formats. Multiple Reformat actions can be added.
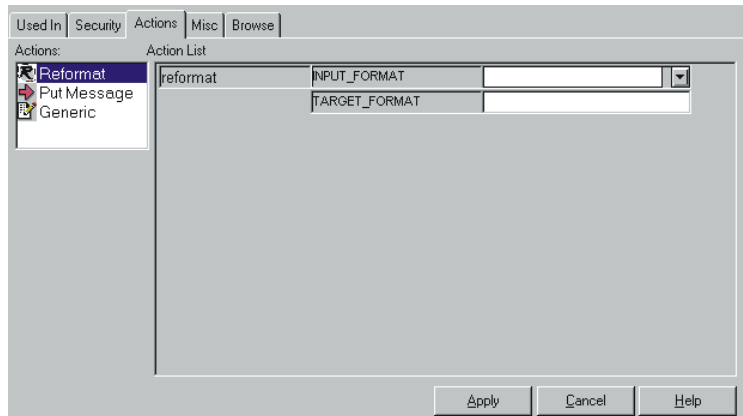
Reformat Action Added

- If you select the **Put Message action**, it is added to the Action List box and labeled putqueue. Multiple Put Message actions can be added.

- If you select the **Generic action**, when it is added to the Action List box, you can change the action name and option name and value. Multiple Generic actions can be added.

3. To save the actions, click Apply.

Notes:

- To delete the new actions and return the tab to its initial configuration, click Cancel.

- The MQIntegrator-supplied Rules daemon does not support generic actions. Users must develop their own Rules daemon to process generic actions.

Adding Multiple putqueue Options

Multiple options can be added. To add more than one option, right click on an Action line to display the pop-up menu, then select Add Option. A blank option field appears under the existing option field. Type the option.

Additional putqueue Option Names and Values

Option Name	Option Value	Default Value	Description
user input	user input	N/A	Data source is user input.
MQS_FORMAT	user input NONE MQADMIN MQCHCOM MQCMD1 MQCMD2 MQDEAD MQHDIST MQEVENT MQIMS MQIMSVS MQHMDE MQPCF MQHREF MQSTR MQTRIG	N/A	Data source is section [putqueue] in the NEONet.ini file. Specifies the MQSeries message format on an outbound message.

Option Name	Option Value	Default Value	Description
MQS_PROPAGATE	PROPAGATE NO_PROPAGATE	NO_ PROPAGATE	Data source is section [MQS_PROPAGATE] in the NEONet.ini file. On a per-message basis, specifies whether to include the MQSeries Integrator header in the user data of the outbound message. For more information, refer to the section, See *MQS_PROPAGATE* on page 191. .
MQS_PERSIST	PERSIST NO_PERSIST	Propagate the expiration setting of the input message to the output message.	Data source is section [MQS_PERSIST] in the NEONet.ini file. Indicates whether the outbound message will be persistent regardless of whether the inbound message was persistent. The MQS_PERSIST option is not set for a given putqueue action. The MQSeries Integrator daemon sets the persistence field of the outbound message's message descriptor to the value of the persistence field of the inbound messages's message descriptor.

Option Name	Option Value	Default Value	Description
MQS_EXPIRY	PROPAGATE NO_PROPAGATE	NO_ PROPAGATE The expiry field of the outbound message descriptor is set to unlimited.	Data source is section [MQS_EXPIRY] in the NEONet.ini file.
OPT_MSG_TYPE		N/A	Formatter application

MQS_PROPAGATE

MQS_PROPAGATE=PROPAGATE

If you set MQS_PROPAGATE=PROPAGATE, if the MQHRF header is present on the inbound message, the format field of the outbound message's MQHRF header will be the same as that of the inbound message.

If the MQHRF header is *not* present on the inbound message, the MQSeries Integrator daemon rejects the message and puts it on the Failure queue.

MQS_PROPAGATE=NO_PROPAGATE

If you set MQS_PROPAGATE=NO_PROPAGATE, if the MQHRF header is present on the inbound message, the format field of the outbound message's message descriptor is set to the value contained in the format field of the inbound message's MQHRF header.

If the MQHRF header was *not* present on the inbound message, the format field of the outbound message's message descriptor is set to the value contained in the inbound messages's message descriptor.

**MQS_PROPAGATE=PROPAGATE and
MQS_FORMAT=SOME_FORMAT**

If you set MQS_PROPAGATE=PROPAGATE and
MQS_FORMAT=SOME_FORMAT, if the MQHRF header is present on the

inbound message, the format field of the outbound message's MQHRF header is set to the value SOME_FORMAT.

If the MQHRF header was *not* present on the inbound message, the MQSeries Integrator daemon rejects the message and puts it on the Failure queue.

MQS_PROPAGATE=NO_PROPAGATE and MQS_FORMAT=SOME_FORMAT

If you set MQS_PROPAGATE=NO_PROPAGATE and MQS_FORMAT=SOME_FORMAT, if the MQHRF header is present or not on the inbound message, the format field of the outbound message's descriptor is set to the value SOME_FORMAT.

MQS_EXPIRY

MQS_EXPIRY=PROPAGATE

If MQS_EXPIRY=PROPAGATE, the Expiry field of the outbound message's message descriptor is set to the value of the Expiry field of the inbound message's message descriptor.

MQS_EXPIRY=NO_PROPAGATE

If MQS_EXPIRY=NO_PROPAGATE, the Expiry field of the outbound message's message descriptor is set to MQEI_UNLIMITED.

Deleting an Action

Use the Delete Action function to delete an action. This function is available from the pop-up menu.

You must have Update permission to delete an action.

To delete an action:

1.   In the Action List, select the action that you want to delete.

2. Position the cursor on the action and press the right mouse button to open the pop-up menu.
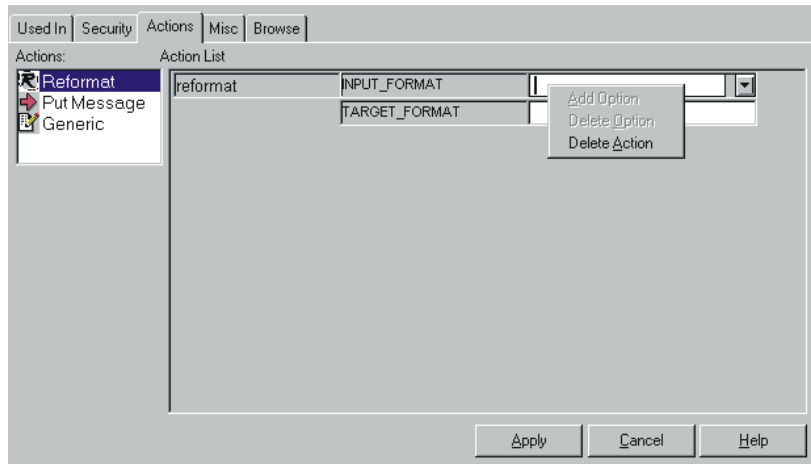
Note:
After you add actions, if you move the cursor to the left pane and click the mouse button, the actions are automatically saved.

3. From the pop-up menu, select Delete Action. The action is deleted from the Actions tab.

4. Choose one of the following:

   ■ To delete the action, click Apply.

   ■ To cancel the delete, click Cancel. The Actions tab is returned to its original configuration.

Note:
Reformat and Put Message must have both options defined to work properly. Only Generic actions are allowed to have user-defined options added and deleted.

Conditional Branching

If the Rules GUI is started from the Formatter GUI, the Rules GUI opens in Conditional Branching mode. The only application group that appears in the Rules tree is NEONET_FORMATTER.

You can add message types, rules, arguments, subscriptions, and actions to the NEONET_FORMATTER application group using the procedures described earlier in this chapter. However, there is a difference in the Actions tab.

The Actions tab contains only the Rules Field action. When adding or inserting Rules Field actions, three options are necessary:

- FIELD_NAME
- OUTPUT_FORMAT
- FORMAT_CONTROL

See *Adding an Action* on page 186 for details on adding and changing actions and options.

Note:

When reformatting from a compound input format to a compound output format, use the compound input format (not a component flat format) as the message type and the compound output format (not a component flat format) as the OUTPUT_FORMAT in the subscription.

Chapter 5
Visual Tester

The Visual Tester lets you test queues, formats, and rules. Each tab in the Visual Tester graphical user interface (GUI) contains a different test.

Visual Tester Tests

Test	Description
Load Q	Loads messages onto a queue.
Test Format	Parses a message based on a format defined in Formatter and graphically displays the results.
Test Reformat	Converts a message from an input format to an output format and displays the results.
Test Rule	Shows how an input message is evaluated by NEONRules.

When you use a Visual Tester test, you specify the data source for the test and using Options and, you can specify settings associated with your input data source. You can also create a message and use this message for more than one test. These concepts are discussed in the following sections.

Visual Tester uses Active X objects to handle interaction with the MQSeries Integrator database.

Starting Visual Tester

To log on to Visual Tester:

1. Select NNVT from the Start menu. The Visual Tester Logon dialog box appears.

    The example log on dialog box will appear if you are using a DB2 database. If you are using an Oracle or MS SQL Server database, the Driver field is labeled Server and the Qualifier field is labeled Database.

2. Choose one of the following:.

    ■   If all defaults are correct, click the Password field, type your password and skip to step 7.

    ■   If there are no defaults or the defaults are not correct, click the User Id field and type your user ID.

3. Click the Password field and type the password associated with the User ID.

4. Choose one of the following:

   - If you are using DB2, click the Driver field and type the driver name.

   - If you are using either Oracle or MS SQL Server, click the Server field and type the server name.

5. Choose one of the following:

   - If you are using DB2, this is labeled the Qualifier field. The field is disabled.

   - If you are using MS SQL Server, click the Database field and type the database name.

   - If you are using Oracle, the Database field information is not necessary for you to sign on. The field is disabled.

6. Click the Queue Manager field and type the name of the queue manager.

7. Click OK. The Load Q window appears. If there is no associated queuing mechanism, the Test Format window appears.

Data Sources

When a Visual Tester test is run, it must have a specified data source.

Visual Tester Data Sources

Data Source	Description
Queue	A second queue is used to provide input messages for the selected test. Refer to *Options* on page 199 for a description of queue-specific options.
File	A file is used as input for the selected test. Since a file contains no message-specific data, you must indicate an Application Group and/or Message Type for a message read from the file if you are testing Formatter and Rules (this is not required if you are only testing Messaging and Queuing). Refer to *Options* on page 199 for a description of file-specific options
Screen	The Message box is used as input for the selected test. You must indicate an Application Group and/or Message Type for your message if you are testing Formatter and Rules (this is not required if you are only testing Messaging and Queuing).

Note:

All data entered into the Message box is in Raw mode. The ASCII, EBCDIC, and Hex filters cannot be used when entering data. For more information on data conversion, refer to the section *Changing Data Representation* on page 215, Appendix A: *ASCII Extended Character Set* on page 217, and Appendix B *: EBCDIC Character Set* on page 223.

Options

Options let you specify various settings, most of which relate to your input data source. Options are divided into three categories:

- General

- Queue

- File
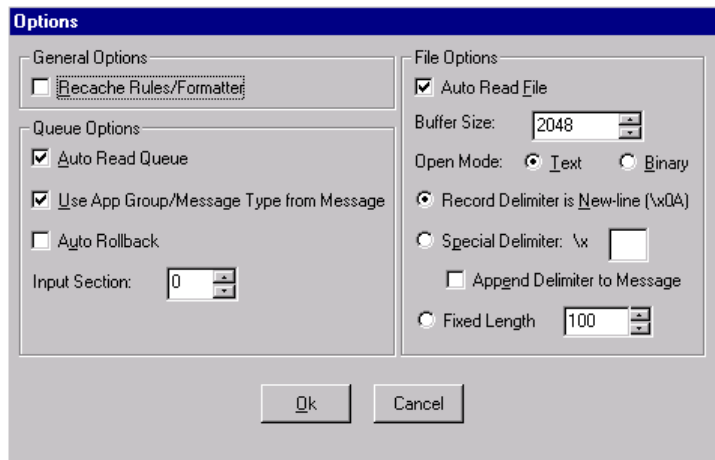
General Options

General options are data source independent.

Recache Rules/Formatter

Recache Rules/Formatter forces the selected test to re-initialize its underlying NEONRules or NEONFormatter object. Note that this option does not apply to Load Q test.

To optimize performance, both NEONRules and NEONFormatter objects cache data they retrieve from the database. However, when you are using Visual

Tester to help debug Rules or Formats, you may occasionally want to force the underlying objects to re-cache data they have read. This would be the case whenever you change a Rule or Format that you have previously tested.

You can also recache Rules/Formatter by selecting the Recache button on the Test Format, Test Reformat, and Test Rules tabs.

Queue Options

Queue options only apply when you have selected a queue as a data source.

Note:

Queue options can be set regardless of the data source, but they only apply when you select Queue as your data source.

Auto Read Queue

When Auto Read Queue is set, the queue is read each time the main action button is selected for the current test (for the Load Q test, this is the Load button, for other tests, it's the Test button). When this option is not set, select the Read button to read the next message from the queue.

Use App Group/Message Type From Message

When Use App Group/Message Type from Message is checked, the application group and message type from the message read from the queue overlays the values for these fields on the screen.

To have values on the screen override the message values, deselect this option.

Auto Rollback

Auto Rollback forces a transaction rollback after a message is read from the queue. The message that was read appears on the screen, but the next message on the queue will be the same message. This option is useful when you are repeatedly testing a particular message and do not want to take the message off the queue until you are done with your testing.

Input Section

The Input Section number specifies the queue section to be used.Queues are usually defined as single-section queues, and the value of the queue section field is normally zero.

File Options

File options only apply when you have selected a file as a data source.

Note:

File options can be set regardless of the data source, but they only apply when you select File as your data source.

Auto Read File

When Auto Read File is set, the file is read each time the main action button is selected for the current test (for the Load Q test, this is the Load button, for other tests, it's the Test button). When this option is not set, select the Read button to read the next message from the queue.

Buffer Size

Buffer Size controls the size of the buffer to be used when reading data from the input file. There are two cases where you might want to change this value from its default (2048 bytes):

- If a file has records larger than 2048 bytes. Set the buffer size to a value that is larger than the largest record length in the file.

- If there are fixed-length records. Set the buffer size to be equal to the length of each fixed record if these are fixed-length records of a specific byte size.

    If the records are fixed length by the number of characters, use Fixed Length instead. For more information, refer to *Fixed Length* on page 202.

Open Mode [Text/Binary]

The Open Mode [Text / Binary] toggle group specifies if the file should be opened in text or binary mode.

When a file is opened in Text mode, it is treated as a file of records, with each record ending in a new-line character. In actuality, the new-line character is a carriage-return/line-feed pair ('\x0D0A').

When a file is opened in Binary mode, it is treated as a stream of data. By default, Visual Tester uses a new-line character ('\x0A') to delimit records within a file. You can specify a different delimiter. For more information, refer to *Special Delimiter* on page 202.

Record Delimiter is New-Line

The new-line character is the default delimiter for records in a file. When Record Delimiter is New-line is checked, data is read from the file until either a new-line character is read or the end of file is reached. The mode that the file is opened determines what is meant by new-line. For more information, refer to *Open Mode [Text/Binary]* on page 202.

Special Delimiter

Special Delimiter lets you specify a delimiter other than NewLine as your record separator. This delimiter must be a single character and entered as hexadecimal. Refer to Appendix A, *ASCII Extended Character Set* for character-to-hex mappings.

Append Delimiter to Message

If Append Delimiter to Message is selected, the delimiter is considered part of the message data. Otherwise, the delimiter is stripped off the record when it is read from the file.

Fixed Length

If Fixed Length is selected, use the arrows to scroll and select the correct length.

Visual Tester Tests

Load Q

Load Q tests the queuing application. You specify a queue and the message to put on the specified queue. The message can be in a separate file, on another queue, or entered into the Message box.
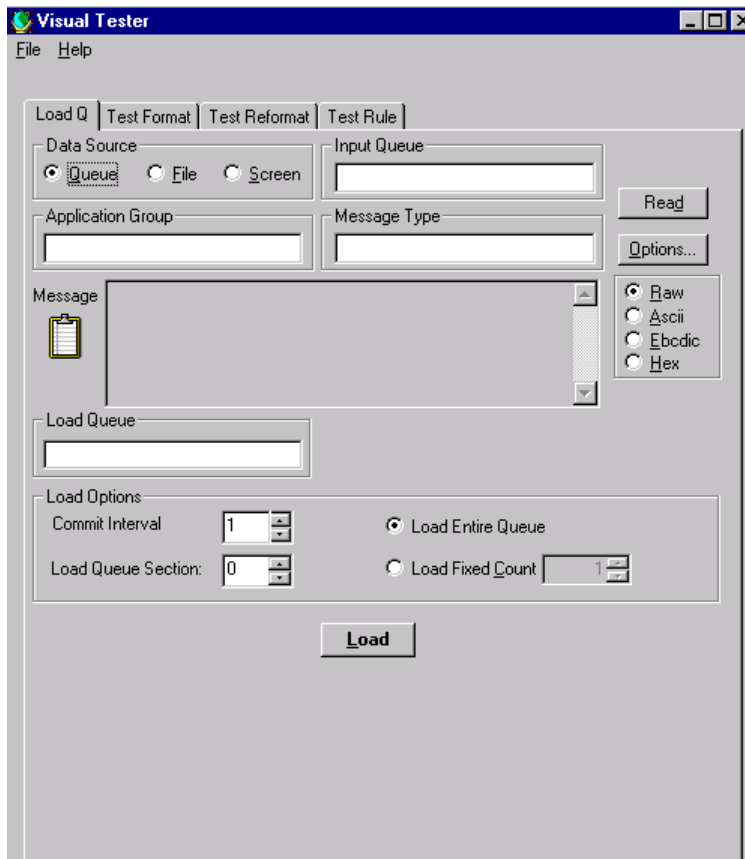
If the message is successfully loaded onto the queue, you receive a notification message. If the message is not successfully loaded onto the queue, you receive a failure message indicating why the put failed.

To add test messages:

1. Select one of the following Data Source option buttons:

   - **Queue**. In the Input Queue field, type the name of the input queue that the message will be taken from.

   - **File**. In the Input File field, either type the full path name of the input file containing the message or click Browse and select the drive, directory, and filename that the message is to be imported from.

     After you indicate the input file, click OK to continue.

   - **Screen**. Type the message in the Message box.

2. To load the message from the selected Queue or File, click Read. The message is displayed in the Message box.

3. To include options necessary for rules evaluation:

   - In the Application Group field, type the Application Group name.

   - In the Message Type field, type the Message Type name.

   These values are set on messages that are loaded onto the destination queue.

4. In the Load Queue field, type the destination queue.

5. Set any other Load Options on the bottom of the screen. Load options are described in the following table:

Load Option	Description
Commit Interval	This option specifies the frequency of database commits while loading the queue. The default value of 1 indicates that a commit will be performed after each message is put to the queue. If a large number of records will be loaded to the queue, the commit interval should be increased for faster loading.
Queue Section does not apply to MQSeries queues	The queue section in which to load data. Most queues only contain one section, so this value is normally zero.
Load Entire Queue/File	This load option indicates that all messages from the input data source will be loaded to the queue. If the input data source is Queue, all messages from the input queue will be moved to the destination queue. If the input data source is File, all data from the file will be loaded to the destination queue. If the input data source is Screen, a single message will be loaded to the destination queue.
Load Fixed Count	This option enables you to limit the number of messages to be loaded to the destination queue. If the input data source is Queue or File, it indicates the number of messages from the input source to be loaded to the destination queue. If the input data source is Screen, it specifies the number of times the message should be loaded to the destination queue.

6. To load the data to the destination queue, click Load.

Test Format

Test Format tests an input format. You provide a formatted message and a message type. You can read the message from a preloaded queue, from a separate file, or enter the message in the Message box. If you are pulling the message from a queue or file, the queue or file name must be specified.
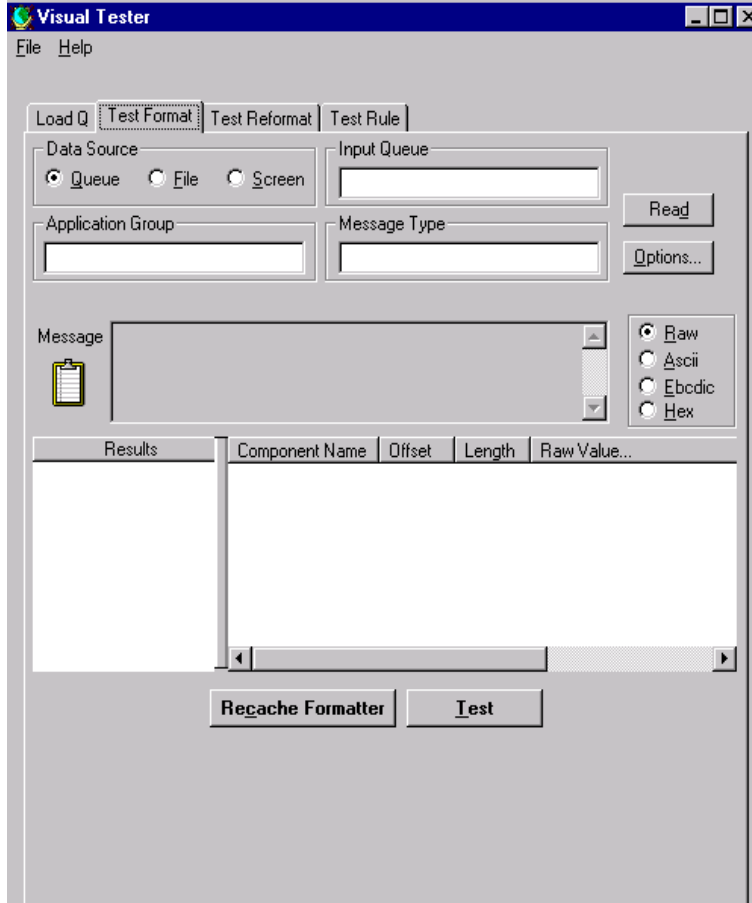
If the message can be successfully parsed, the message components appear in the Results box with their corresponding formatting (for example, type, field

name). If the message cannot be parsed according to the specified message type, an error message appears.

To test a format:

1. Select one of the following Data Source option buttons:

   - **Queue**. In the Input Queue field, type the name of the input queue that the message will be taken from.

   - **File**. In the Input File field, either type the name of the input file containing the message or click Browse and select the drive, directory, and filename that the message is to be imported from.

     After you indicate the input file, click OK to continue.

   - **Screen**. Type the message in the Message box.

2. (Optional) To load the message from the selected Queue or File, click Read. The message is displayed in the Message box.

   If you do not manually read the record, the record is automatically read when the Test button is selected (unless the auto read option has been disabled).

3. To include options necessary for format and rules evaluation:

   - In the Application Group field, type the Application Group name.

   - In the Message Type field, type the Message Type name.

   These values are set on messages that are loaded onto the destination queue.

4. To re-initialize the underlying Formatter object, click Recache Formatter. See *Recache Rules/Formatter* on page 199.

5. Click Test. If the message parses successfully, the results of the parse appear in the lower windows.

   The left window displays a parse tree of the formats and the right window displays the parsed output of the first flat input format

Test Reformat

Test Reformat tests how a message will be parsed and reformatted from an input format to an output format. You provide a formatted input message and a message type. You can read the message from a preloaded queue, from a separate file, or enter the message in the Message box. If you are pulling the message from a queue or file, the queue or file name must be specified. You must also provide an output message format. You can choose if you want the output message to go to the screen, a file, or another queue.
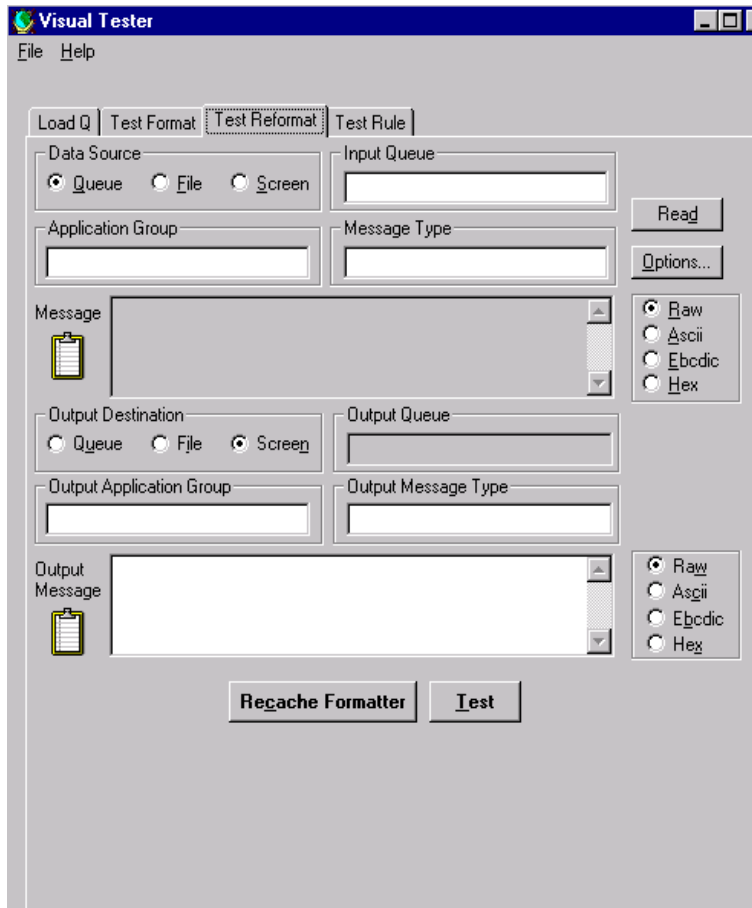
If the message can be successfully reformatted, the message is displayed where you selected, either on your screen, in a file, or on another queue. If the message cannot be reformatted, an error message appears.

To test a format:

1. Select one of the following Data Source option buttons:

   ■ **Queue**. In the Input Queue field, type the name of the input queue that the message will be taken from.

   ■ **File**. In the Input File field, either type the name of the input file containing the message or click Browse and select the drive, directory, and filename that the message is to be imported from.

   After you indicate the input file, click OK to continue.

   ■ **Screen**. Type the message in the Message box.

2. (Optional) To load the message from the selected Queue or File, click Read. This displays the message in the Message box.

   If you do not manually read the record, the record is automatically read when you select the Test button (unless the auto read option has been disabled).

3. To include options necessary for Rules evaluation:

   ■ In the Application Group field, type the Application Group name.

   ■ In the Message Type field, type the Message Type name.

   These values are set on messages that are loaded onto the destination queue.

4. Select an Output Destination -- Queue, File, or Screen. Normally this will be Screen, but you may also choose a queue or a file.

5. In the Output Message Type field, enter an output format name. This will be the destination output format for your message.

6. To re-initialize the underlying Formatter object, click Recache Formatter. For more information on recaching, see *Recache Rules/ Formatter* on page 199.

7. Click Test. If the reformat was successful, the results appear in the lower window (and on the queue or file if you chose an output destination other than Screen).

   If the reformat was not successful, an error message is displayed.

Test Rule

Test Rule evaluates an input message and returns two lists of rules, rules that hit and rules that did not. You provide an input message, message type and application group. You can read the message from a preloaded queue, from a separate file, or enter the message in the Message box. If you are pulling the message from a queue or file, the queue or file name must be specified. Visual Tester uses the NEONRules eval API to parse the message and evaluate it against its associated rules.
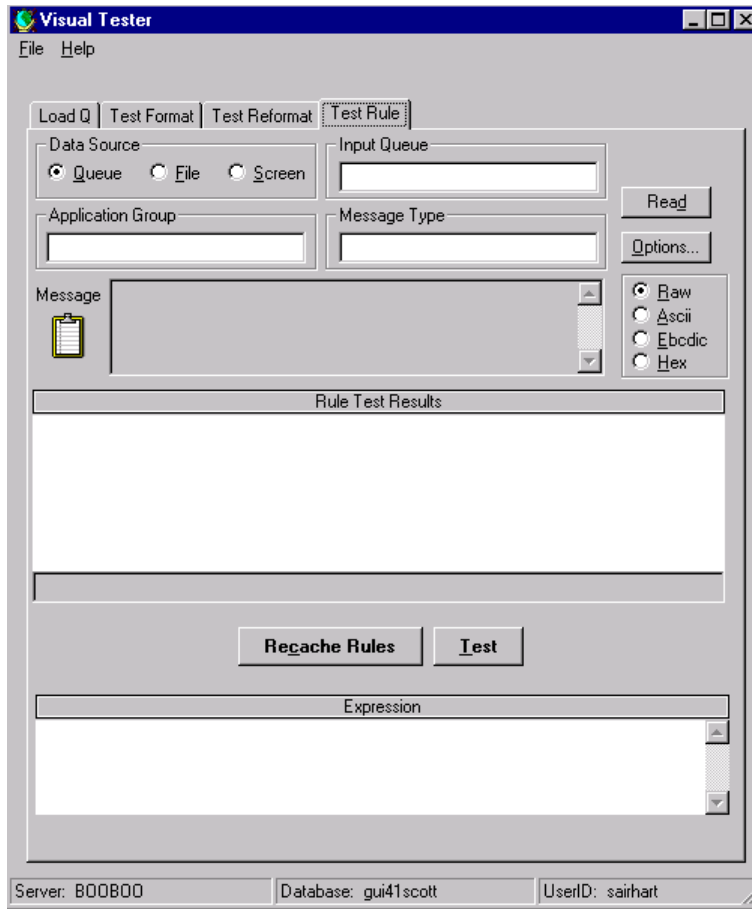
If Rules evaluates the message, a detailed schematic appears in the Rule Test Results box. The schematic is a tree view of both hit and missed rules and their components. The components include the rule name, the expression the rule evaluates, and the rule's subscriptions and actions. You can choose to see the rule's full expression, which will be displayed in the Expression box

To test rules:

1. Select one of the following Data Source option buttons:

    ▪ **Queue**. In the Input Queue field, type the name of the input queue that the message will be taken from.

    ▪ **File**. In the Input File field, either type the name of the input file containing the message or click Browse and select the drive, directory, and filename that the message is to be imported from.

      After you indicate the input file, click OK to continue.

    ▪ **Screen**. Type the message in the Message box.

2. (Optional) To load the message from the selected Queue or File, click Read. This displays the message in the Message box.

    If you do not manually read the record, the record is automatically read when you select the Test button (unless the auto read option has been disabled).

3. To include options necessary for Rules evaluation:

    ▪ In the Application Group field, type the Application Group name.

    ▪ In the Message Type field, type the Message Type name.

    These values are set on messages that are loaded onto the destination queue.

4. To re-initialize the underlying Rules object, click Recache Rules. For more information on recaching, see *Recache Rules/Formatter* on page 199.

5. Click Test. The results of the rule evaluations appears in a tree in the lower window. The tree contains two main branches -- Hit Rules and Missed Rules.

6. In the tree, to see the full expression, select the expression. The full expression is displayed in the Expression box.

Copying Messages Between Tabs

The same message can be used for more than one test. For example, you can first test your message using the Test Format. After you successfully parse the message, you can test the same message using Test Reformat. Finally, you can check the message using Test Rule. Visual Tester allows you to copy message data from one test tab to another using a drag and drop operation.

T**o copy a message from one test to another**:

1. Move the cursor over the message icon.

2. Hold down the left mouse button. The mouse pointer attaches to a copy of the message icon.

3. Drag the pointer to the tab of the test that you want to copy the message to. Your message now appears in the tab of that test.

User Exits

The Visual Tester NEONFormatter User Exit interface is encapsulated in a DLL to allow you to extend the product with your own user exit code.

To add user exits to Visual Tester, a Microsoft Visual C++ compiler (Version 5.0 or greater) is required. Also, a Windows NT version of MQSeries

Integrator must be installed. For more information, refer to the ***MQSeries Integrator Programming Reference for Formatter APIs***.

To add a user exit:

1. Open the Microsoft Developer Project file **NNFExit.dsw** which is in the **NNFExit** directory where Visual Tester was installed.

2. Set the environment variable **NEONET_BUILD** to point to the location of the NEONet for Windows NT installation directory. The NNFExit project uses this variable to locate NEONet header and library files.

3. Choose File→Open. A dialog box displaying the current project files appears.

4. Select **DllUserExit.cpp**.

   This file should be very similar to other user exit source files you have written. One difference is the user exit mapping function is now named **DllNNGetUserExitFuncPtrs** (instead of NNGetUserExitFuncPtr).

5. Follow the TO DO comments and add your user exit prototype, name comparison logic, and function definition. For more information, refer to the ***MQSeries Integrator Programming Reference for Formatter APIs***.

6. Choose Build→Set Active Configuration. Ensure that the default configuration is **Win32** - **Release**.

   WARNING!

   You must build your project in release mode. If you do not, you will get runtime Heap Assertion Failure errors. (This stems from the way Visual C++ uses different heaps for debug and release DLLs. Since the underlying MQSERIES Integrator libraries compiled in release mode, your user exit module must also be compiled in release mode.)

   ---

7. Choose Build→Rebuild All. This compiles and links your project.

8. Exit Visual Tester.

9. Move your new **NNFExit.dll** file (in the **NNFExit/Release** directory where Visual Tester is installed) to the DLL directory where Visual Tester is installed. This links your user exit with Visual Tester.

Note:
If you did not exit Visual Tester before moving the NNFExit.dll file, you will receive a File in Use error.

Changing Data Representation

When viewing a message read from a data source, the default representation is Raw; there is no translation of the data. The default character set used in Raw mode is ASCII. If your data has unprintable characters or is not ASCII, you can view the data using an ASCII, EBCDIC, or Hex filter.

Input Data

The toggle buttons on the right of the Message Text box let you choose an ASCII, EBCDIC, or Hex filter. These filters are not true conversions to any of these character sets, since they ensure that all characters will be viewable. To accomplish this, non-viewable hexadecimal characters are printed as \## where ## represents the hexadecimal value of the character.

Parsed Data

You can change the data representation in the NEONFormatter results list view. The list view uses toggle columns to change the data representation.

To change to a different data representation, press the left mouse button and click the 'Raw Value...' column heading. The column heading changes to 'ASCII Value...' and translates the data in the column to printable ASCII.

Continue to use the left mouse button to click the column heading to cycle through the ASCII, EBCDIC, and Hex filters.

MQSeries Integrator User's Guide

Appendix A

ASCII Extended Character Set

Decimal Value	Hex Value	Extended Character Set	Decimal Value	Hex Value	Extended Character Set
000	00	NUL	016	10	DLE
001	01	SCH	017	11	DC1
002	02	STX	018	12	DC2
003	03	ETX	019	13	DC3
004	04	EOT	020	14	DC4
005	05	ENO	021	15	NAK
006	06	ACK	022	16	SYN
007	07	BEL	023	17	ETB
008	08	BS	024	18	CAN
009	09	HT	025	19	EM
010	0A	LF	026	1A	SUB
011	0B	VT	027	1B	ESCAPE
012	0C	FF	028	1C	FS
013	0D	CR	029	1D	GS
014	0E	SO	030	1E	RS
015	0F	SI	031	1F	US

Decimal Value	Hex Value	Extended Character Set	Decimal Value	Hex Value	Extended Character Set
032	20	SPACE	055	37	7
033	21	!	056	38	8
034	22	"	057	39	9
035	23	#	058	3A	:
036	24	$	059	3B	;
037	25	%	060	3C	<
038	26	&	061	3D	=
039	27	'	062	3E	>
040	28	(	063	3F	?
041	29	)	064	40	@
042	2A	*	065	41	A
043	2B	+	066	42	B
044	2C	,	067	43	C
045	2D	-	068	44	D
046	2E	.	069	45	E
047	2F	/	070	46	F
048	30	0	071	47	G
049	31	1	072	48	H
050	32	2	073	49	I
051	33	3	074	4A	J
052	34	4	075	4B	K
053	35	5	076	4C	L
054	36	6	077	4D	M

Decimal Value	Hex Value	Extended Character Set	Decimal Value	Hex Value	Extended Character Set
078	4E	N	101	65	e
079	4F	O	102	66	f
080	50	P	103	67	g
081	51	Q	104	68	h
082	52	R	105	69	i
083	53	S	106	6A	j
084	54	T	107	6B	k
085	55	U	108	6C	l
086	56	V	109	6D	m
087	57	W	110	6E	n
088	58	X	111	6F	o
089	59	Y	112	70	p
090	5A	Z	113	71	q
091	5B	[	114	72	r
092	5C	\	115	73	s
093	5D	]	116	74	t
094	5E	^	117	75	u
095	5F	_	118	76	v
096	60	`	119	77	w
097	61	a	120	78	x
098	62	b	121	79	y
099	63	c	122	7A	z
100	64	d	123	7B	{

Decimal Value	Hex Value	Extended Character Set	Decimal Value	Hex Value	Extended Character Set
124	7C	|	147	93	"
125	7D	}	148	94	"
126	7E	~	149	95	•
127	7F	DEL	150	96	–
128	80	€	151	97	—
129	81	unused	152	98	˜
130	82	,	153	99	™
131	83	ƒ	154	9A	š
132	84	„	155	9B	›
133	85	…	156	9C	œ
134	86	†	157	9D	unused
135	87	‡	158	9E	unused
136	88	ˆ	159	9F	Ÿ
137	89	‰	160	A0	nonbreaking space
138	8A	Š	161	A1	¡
139	8B	‹	162	A2	¢
140	8C	Œ	163	A3	£
141	8D	unused	164	A4	¤
142	8E	unused	165	A5	¥
143	8F	unused	166	A6	¦
144	90	unused	167	A7	§
145	91	'	168	A8	¨
146	92	'			

Decimal Value	Hex Value	Extended Character Set	Decimal Value	Hex Value	Extended Character Set
169	A9	©	192	C0	À
170	AA	ª	193	C1	Á
171	AB	«	194	C2	Â
172	AC	¬	195	C3	Ã
173	AD	-	196	C4	Ä
174	AE	®	197	C5	Å
175	AF	¯	198	C6	Æ
176	B0	°	199	C7	Ç
177	B1	±	200	C8	È
178	B2	²	201	C9	É
179	B3	³	202	CA	Ê
180	B4	´	203	CB	Ë
181	B5	µ	204	CC	Ì
182	B6	¶	205	CD	Í
183	B7	·	206	CE	Î
184	B8	¸	207	CF	Ï
185	B9	¹	208	D0	Ð
186	BA	º	209	D1	Ñ
187	BB	»	210	D2	Ò
188	BC	¼	211	D3	Ó
189	BD	½	212	D4	Ô
190	BE	¾	213	D5	Õ
191	BF	¿	214	D6	Ö

Decimal Value	Hex Value	Extended Character Set	Decimal Value	Hex Value	Extended Character Set
215	D7	×	238	EE	î
216	D8	Ø	239	EF	ï
217	D9	Ù	240	F0	ð
218	DA	Ú	241	F1	ñ
219	DB	Û	242	F2	ò
220	DC	Ü	243	F3	ó
221	DD	Ý	244	F4	ô
222	DE	Þ	245	F5	õ
223	DF	ß	246	F6	ö
224	E0	à	247	F7	÷
225	E1	á	248	F8	ø
226	E2	â	249	F9	ù
227	E3	ã	250	FA	ú
228	E4	ä	251	FB	û
229	E5	à	252	FC	ü
230	E6	æ	253	FD	ý
231	E7	ç	254	FE	þ
232	E8	è	255	FF	ÿ
233	E9	é			
234	EA	ê			
235	EB	ë			
236	EC	ì			
237	ED	í			

Appendix B
EBCDIC Character Set

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
000	00	NUL	Null	0000 0000
001	01	SOH	Start of Heading	0000 0001
002	02	STX	Start of Text	0000 0010
003	03	ETX	End of Text	0000 0011
004	04	SEL	Select	0000 0100
005	05	HT	Horizontal Tab	0000 0101
006	06	RNL	Required New Line	0000 0110
007	07	DEL	Delete	0000 0111
008	08	GE	Graphic Escape	0000 1000
009	09	SPS	Superscript	0000 1001
010	0A	RPT	Repeat	0000 1010
011	0B	VT	Vertical Tab	0000 1011
012	0C	FF	Form Feed	0000 1100
013	0D	CR	Carriage Return	0000 1101
014	0E	SO	Shift Out	0000 1110
015	0F	SI	Shift In	0000 1111
016	10	DLE	Data Link Escape	0001 0000

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
017	11	DC1	Device Control 1	0001 0001
018	12	DC2	Device Control 2	0001 0010
019	13	DC3	Device Control 3	0001 0011
020	14	RES/ENP	Restore/Enable Presentation	0001 0100
021	15	NL	New Line	0001 0101
022	16	BS	Backspace	0001 0110
023	17	POC	Program-Operator Communication	0001 0111
024	18	CAN	Cancel	0001 1000
025	19	EM	End of Medium	0001 1001
026	1A	UBS	Unit Backspace	0001 1010
027	1B	CU1	Customer Use 1	0001 1011
028	1C	IFS	Interchange File Separator	0001 1100
029	1D	IGS	Interchange Group Separator	0001 1101
030	1E	IRS	Interchange Record Separator	0001 1110
031	1F	IBT/IUS	Intermediate Transmission Block/Interchange Unit Separator	0001 1111
032	20	DS	Digit Select	0010 0000
033	21	SOS	Start of Significance	0010 0001
034	22	FS	Field Separator	0010 0010
035	23	WUS	Word Underscore	0010 0011
036	24	BYP/INP	Bypass/Inhibit Presentation	0010 0100
037	25	LF	Line Feed	0010 0101

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
038	26	ETB	End of Transmission Block	0010 0110
039	27	ESC	Escape	0010 0111
040	28	SA	Set Attribute	0010 1000
041	29	SFE	Start Field Extended	0010 1001
042	2A	SM/SW	Set Mode/Switch	0010 1010
043	2B	CSP	Control Sequence Prefix	0010 1011
044	2C	MFA	Modify Field Attribute	0010 1100
045	2D	ENQ	Enquiry	0010 1101
046	2E	ACK	Acknowledge	0010 1110
047	2F	BEL	Bell	0010 1111
048	30			0011 0000
049	31			0011 0001
050	32	SYN	Synchronous Idle	0011 0010
051	33	IR	Index Return	0011 0011
052	34	PP	Presentation Position	0011 0100
053	35	TRN	Transparent	0011 0101
054	36	NBS	Numeric Backspace	0011 0110
055	37	EOT	End of Transmission	0011 0111
056	38	SBS	Subscript	0011 1000
057	39	IT	Indent Tab	0011 1001
058	3A	RFF	Required Form Feed	0011 1010
059	3B	CU3	Customer Use 3	0011 1011

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
060	3C	DC4	Device Control 4	0011 1100
061	3D	NAK	Negative Acknowledge	0011 1101
062	3E			0011 1110
063	3F	SUB	Substitute	0011 1111
064	40	SP	Space	0100 0000
065	41	RSP		0100 0001
066	42			0100 0010
067	43			0100 0011
068	44			0100 0100
069	45			0100 0101
070	46			0100 0110
071	47			0100 0111
072	48			0100 1000
073	49			0100 1001
074	4A	¢		0100 1010
075	4B	.		0100 1011
076	4C	<		0100 1100
077	4D	(		0100 1101
078	4E	+		0100 1110
079	4F	|		0100 1111
080	50	&		0101 0000
081	51			0101 0001

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
082	52			0101 0010
083	53			0101 0011
084	54			0101 0100
085	55			0101 0101
086	56			0101 0110
087	57			0101 0111
088	58			0101 1000
089	59			0101 1001
090	5A	!		0101 1010
091	5B	$		0101 1011
092	5C	*		0101 1100
093	5D	)		0101 1101
094	5E	;		0101 1110
095	5F	¬		0110 1111
096	60	-		0110 0000
097	61	∕		0110 0001
098	62			0110 0010
099	63			0110 0011
100	64			0110 0100
101	65			0110 0101
102	66			0110 0110
103	67			0110 0111

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
104	68			0110 1000
105	69			0110 1001
106	6A	|		0110 1010
107	6B	,		0110 1011
108	6C	%		0110 1100
109	6D	_		0110 1101
110	6E	>		0110 1110
111	6F	?		0110 1111
112	70			0111 0000
113	71			0111 0001
114	72			0111 0010
115	73			0111 0011
116	74			0111 0100
117	75			0111 0101
118	76			0111 0110
119	77			0111 0111
120	78			0111 1000
121	79			0111 1001
122	7A	:		0111 1010
123	7B	#		0111 1011
124	7C	@		0111 1100
125	7D	'		0111 1101

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
126	7E	=		0111 1110
127	7F	"		0111 1111
128	80	€		1000 0000
129	81	a		1000 0001
130	82	b		1000 0010
131	83	c		1000 0011
132	84	d		1000 0100
133	85	e		1000 0101
134	86	f		1000 0110
135	87	g		1000 0111
136	88	h		1000 1000
137	89	i		1000 1001
138	8A			1000 1010
139	8B			1000 1011
140	8C			1000 1100
141	8D			1000 1101
142	8E			1000 1110
143	8F			1000 1111
144	90			1001 0000
145	91	j		1001 0001
146	92	k		1001 0010
147	93	l		1001 0011

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
148	94	m		1001 0100
149	95	n		1001 0101
150	96	o		1001 0110
151	97	p		1001 0111
152	98	q		1001 1000
153	99	r		1001 1001
154	9A			1001 1010
155	9B			1001 1011
156	9C			1001 1100
157	9D			1001 1101
158	9E			1001 1110
159	9F			1001 1111
160	A0			1010 0000
161	A1	~		1010 0001
162	A2	s		1010 0010
163	A3	t		1010 0011
164	A4	u		1010 0100
165	A5	v		1010 0101
166	A6	w		1010 0110
167	A7	x		1010 0111
168	A8	y		1010 1000
169	A9	z		1010 1001

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
170	AA			1010 1010
171	AB			1010 1011
172	AC			1010 1100
173	AD			1010 1101
174	AE			1010 1110
175	AF			1010 1111
176	B0			1011 0000
177	B1			1011 0001
178	B2			1011 0010
179	B3			1011 0011
180	B4			1011 0100
181	B5			1011 0101
182	B6			1011 0110
183	B7			1011 0111
184	B8			1011 1000
185	B9			1011 1001
186	BA			1011 1010
187	BB			1011 1011
188	BC			1011 1100
189	BD			1011 1101
190	BE			1011 1110
191	BF			1011 1111

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
192	C0	{		1100 0000
193	C1	A		1100 0001
194	C2	B		1100 0010
195	C3	C		1100 0011
196	C4	D		1100 0100
197	C5	E		1100 0101
198	C6	F		1100 0110
199	C7	G		1100 0111
200	C8	H		1100 1000
201	C9	I		1100 1001
202	CA	SHY		1100 1010
203	CB			1100 1011
204	CC			1100 1100
205	CD			1100 1101
206	CE			1100 1110
207	CF			1100 1111
208	D0	}		1101 0000
209	D1	J		1101 0001
210	D2	K		1101 0010
211	D3	L		1101 0011
212	D4	M		1101 0100
213	D5	N		1101 0101

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
214	D6	O		1101 0110
215	D7	P		1101 0111
216	D8	Q		1101 1000
217	D9	R		1101 1001
218	DA			1101 1010
219	DB			1101 1011
220	DC			1101 1100
221	DD			1101 1101
222	DE			1101 1110
223	DF			1101 1111
224	E0	\		1110 0000
225	E1			1110 0001
226	E2	S		1110 0010
227	E3	T		1110 0011
228	E4	U		1110 0100
229	E5	V		1110 0101
230	E6	W		1110 0110
231	E7	X		1110 0111
232	E8	Y		1110 1000
233	E9	Z		1110 1001
234	EA			1110 1010
235	EB			1110 1011

Decimal Value	Hex Value	EBCDIC Value*	Description	Binary
236	EC			1110 1100
237	ED			1110 1101
238	EE			1110 1110
239	EF			1110 1111
240	F0	0		1111 0000
241	F1	1		1111 0001
242	F2	2		1111 0010
243	F3	3		1111 0011
244	F4	4		1111 0100
245	F5	5		1111 0101
246	F6	6		1111 0110
247	F7	7		1111 0111
248	F8	8		1111 1000
249	F9	9		1111 1001
250	FA			1111 1010
251	FB			1111 1011
252	FC			1111 1100
253	FD			1111 1101
254	FE			1111 1110
255	FF	EO	Eight Ones	1111 1111

* In the IBM-DOS Character Set, the nonprinting characters may be displayed as figures, for example, (x03) ETX is shown as a heart, and (x0D) CR is shown as a musical note.

Appendix B

Appendix C
Notices

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

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

```
IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.
```

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

```
IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan
```

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England,
SO21 2JN.
```

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

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

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms.

You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks and Service Marks

The following, which appear in this book or other MQSeries Integrator books, are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

```
MQSeries
AIX
DB2
IBM
```

NEONFormatter and NEONRules are trademarks of New Era of Networks, Inc. in the United States and/or other countries.

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

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

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

Appendix C

Glossary

abend

Acronym for abnormal end. More commonly known as a crash.

accelerator keys

A combination of keystrokes that bypasses the menu system to carry out an action.

action

An action defines a function to perform in a subscription. The Reformat and Put Message actions are predefined and the user can provide generic actions as well. An action must contain at least one option name-value pair.

administration workstation

A form of "client" from which a user carries out NEONet management duties.

alternative format

A special form of compound format where one format in a set of alternatives applies to a message. For example, if the alternative format is named A, it may contain component formats B, C, and D. A message of format A may actually be of variation of only B, C, or D.

application group

A logical grouping of applications used to organize rules.

Application Program Interface (API)

The interface (calling conventions) by which an application program accuses services. An API is defined at source code level and provides a level of abstraction between the application and the kernel (or other privileged utilities) to ensure portability of the code. At NEON, an API refers to a set of object classes, structures, and functions that can be used by an application program to invoke NEONet services. Plural: APIs.

argument

In Rules, an argument is evaluation criteria made up of fields from a message and associated operators.

ASCII

Abbreviation for American Standard Code for Information Interchange.

asynchronous

In electronic messaging, a method of operation in which receiving applications are loosely coupled and independent. The receiver need not respond immediately to a message, and the sender does not have to wait for a response before proceeding with the next operation. Compare to synchronous.

Boolean operators

Logical operators, including AND and OR.

client

A system or process that requests a service from another system or process.

client-server

A paradigm for distributed computing under which the system is split between one or more server tasks, which accept requests according to some

protocol, and client tasks, which request information or actions. Clients and servers can be placed independently on network nodes.

cross-platform

Used to describe programs that can execute in dissimilar computing environments.

Data Type

Describes how to interpret the data.

DBMS

Abbreviation for Database Management System.

delimiter

One or more characters marking either the end or beginning of a piece of data.

daemon

In UNIX, a program that executes in the background. Also known as agents, daemons only activate when defined system conditions become true. For example, a daemon may activate when the system clock reaches 2:00 a.m. every other Saturday. Another daemon may activate when the system senses the arrival of e-mail.

embedded length

A piece of data associated with a field indicating the length of the data in the field. For example, the data may appear in a message as 3.DOG, where the 3 indicates that the field data (DOG) is three bytes in length.

field

The smallest possible container for information. You can use a field in more than none table. If the first_name field exists in one table, for example, you can use the same field in other tables.

flat format

A format only containing fields and associated controls. Flat input formats are composed of fields with associated controls input controls and flat output formats are composed of fields with associated output controls.

format

Formats describe how messages are constructed. Input formats describe how to separate input messages into their component parts. Output formats describe how to build output messages from the parsed components of the input message.

get

A request for the next message in a queue. Compare to put.

GUI

Abbreviation for graphical user interface.

heterogeneous

Composed of different parts of different kinds. Having dissimilar constituents.

hexadecimal (hex)

Hexadecimal means 16. The base 16 numbering system is used to represent binary numbers.

input control

In Formatter, an input control is used to parse input data. The input control is used to determine how to find the beginning and end of the data in a field.

literal

One or more symbols or letters in data that represents itself.

massive mirroring

Massive mirroring is a means of boosting processing power by replicating entire databases on different computers and running them in parallel (keeping the databases synchronized).

message

A packet of data both sent and received with a definite beginning and end.

message type

A message type defines the layout of a string of data. The message type name in Rules is the same as the input format name in Formatter.

middleware

Software intended to work between software applications and the underlying computer operating system.

nesting level

Level within the hierarchy of a specific component. A repeating child format and the fields within it may have a nesting level one greater than that of the parent format and any nonrepeating components of the parent format. The nesting level of the root format is one.

option

An option consists of a name-value pair of data related to an action An option name can be predefined (for Reformat and Put Message) or user-defined.

parent/child

Compound formats contain other flat and/or compound formats. If you have a compound format X that contains a repeating format Y, X is the parent to child Y.

parse

To analyze a message by breaking down into its component fields.

permission

Rules and subscriptions are associated with user permissions that determine whether a user can change the item definition or any part of it. The PUBLIC user includes any user of the system. Read permission is always granted. The user creating the rule or subscription is the Owner. The Owner can change ownership to another user. Update permission is needed to modify component definition.

persistence

The ability of a computerized system to remember the state of data or objects between runs.

pool

The series of slots that NEONet preallocates. The slots can be assigned to any queue.

pop-up menu

A floating list of actions. To open a pop-up menu, click an object and hold down the right mouse button; the available actions depend on the object

protocol

A set of rules that govern the transmission and reception of data.

put

A request to store a message in a queue. Compare to get.

queue

In NEONet, a series of slots in a single segment, stored in the database.

A queue processes its data using First In, First Out (FIFO).

A wrap-around queue as implemented in NEONet's Messaging and Queuing uses space more efficiently than a standard queue. With a standard queue, all messages must be cleared before the queue returns its space to the system. Wrap-around queues release the space as each message is cleared.

regular expression

Strings which express rules for string pattern matching.

repeating component

A component (either a field or format) that may appear multiple times in an input or output message.

rule

A rule is uniquely defined by its application group, message type, and rule name. It contains evaluation criteria (a rules expression) and is associated with subscriptions to perform if the rule evaluates to true. Rules also have permissions that determine user access.

scalable, scalability

Changeable in size and configuration. The ability to expand, implying minimal procedural change to accommodate growth.

segment

An indexing device within a queue section. A series of contiguous slots that store information.

server

A computer that, through its network connections, carries out parts of computing tasks for other networked computers.

shared subscription

A subscription that is associated with more than one rule. This subscription will only be retrieved once by the Rules APIs even if multiple associated rules evaluate true.

slots

The "building blocks" of NEONet's queuing structure. Slots are assigned to queues, where they become components segments. Slot size is fixed at 1,880 bytes. Slots are allocated from the pool to only one queue; once allocated, they become unavailable to other queues.

subscription

A subscription is uniquely identified by its application group, message type, and subscription name. It contains actions with options and can be associated with one or more rules. Subscriptions also have permissions that determine user access to change the subscription definition.

synchronous

In electronic messaging, a method of operation in which sender and receiver applications are tightly coupled and dependent. The receiver must answer the sender's message immediately with a well-defined response; the sender must wait for the receiver's response before proceeding to the next operation Compare to asynchronous.

table

A database remembers relationships between pieces of information by storing the information in tables. The columns and rows in each table define the relationships in a highly structured way. We can classify tables by function

into two types: support tables and data tables. Most tables fit into only one category, but some can serve as both support and data tables.

A support table stores information that changes infrequently and functions as a list from which you make selections.

tag

A set of bits of characters that identifies various conditions about data in a file. In Formatter, a standard value indicating the field name.

transaction

An activity or request. Additions, changes, and deletions are typical transactions stored in a computer.

transaction management

A method of handling electronic messaging such that only committed messages are sent and only messages received and committed are considered delivered.

transaction processing

A method of handling computer operations in which the operations take place immediately upon receipt of the processing request. Also called realtime or online system. Compare to batch processing.

Index

A

access modes 112
actions
    deleting 192
    Generic action 186
    Put Message action 186
    Reformat action 186
Actions property sheet 194
adding actions to subscriptions 186
adding application groups 151, 154, 158
adding comments to subscriptions 183
adding input controls to flat input formats 20
adding parse controls 66
adding Rules 158
adding subscriptions 180
adding user permissions 185
alternative input formats 125
alternative output formats 125
AND operator 177
application groups 151
Arrange Icons function 34, 150
arranging icons 34, 150
ASCII characters 215
ASCII Extended Character Set 217
assigning fields to input controls 20
assigning update permissions 142

B

Boolean operators
    AND operator 177
    OR operator 177
building formats 43

C

Calculated Field output control 72
cascading windows 27
case sensitivity 22, 143

case-sensitive string comparisons 166
changing user permissions 185
Clear function 24
clearing text 24
Close function 23, 145
closing a window 23
closing Formatter 23
closing Rules 145
closing Rules window 145
comments 183
comparison values 44
conditional branching 79, 194
Conditional Branching mode 194
Conditional output control 73
control tables 13
converting messages 14
Copy function 23
copying messages between tabs 213
copying objects 19
copying text 23
creating 99
    compound input formats 102
    compound output formats 113
    expressions 165
    flat input formats 99
    flat output formats 21, 109
    formats 98
    input controls 49
    literals 45
    new rules 148
    new subscriptions 148
    output controls 66
    output formats 19
    user-defined data types 131
    validation parameters 134
Customize toolbars function 146
Cut function 23
cutting text 23

Sending your comments to IBM
MQSeries Integrator
User's Guide
GC34-5504-01

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book only and the way in which the information is presented.

To request additional publications or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, use the Readers' Comment Form
- By fax:
    - From outside the U.K., use your international access code followed by 44 1962 870229
    - From within the U.K., use 01962 870229

Electronically, use the appropriate network ID:

- IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
- IBMLink: HURSLEY(IDRCF)
- Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The page number or topic number to which your comment applies
- Your name/address/telephone number/fax number/network ID

Readers' Comments
MQSeries Integrator
User's Guide
GC34-5504-01

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name   Address

Company or organization

Telephone   Email

You can send your comments POST FREE on this form from any one of these countries:

|					
Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

2 Fold along this line

By air mail
Par avion

IBRS/CCRI NUMBER: PHQ - D/1348/SO

NE PAS AFFRANCHIR

NO STAMP REQUIRED

IBM
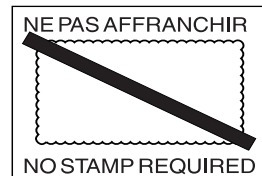
REPONSE PAYEE
GRANDE-BRETAGNE

IBM United Kingdom Laboratories
Information Development Department (MP 095)
Hursley Park
WINCHESTER, Hants
SO21 2ZZ   United Kingdom

3 Fold along this line

From: Name _____

Company or Organization _____

Address _____

_____

EMAIL _____

Telephone _____

4 Fasten here with adhesive tape

Cut along this line

IBM

Printed in U.S.A