

OpenFox™
Form Configurator Guide
Version 1.3



Computer Projects of Illinois, Inc.
475 Quadrangle Drive
Bolingbrook, IL. 60440
(630) 754-8835

* * * THIS PAGE LEFT INTENTIONALLY BLANK * * *

Table of Contents

CHAPTER 1: INTRODUCTION AND OVERVIEW	5
1.1 ORGANIZATION OF RELATED FORM CONFIGURATOR DOCUMENTATION.....	5
1.2 FORM CONFIGURATOR GENERAL DESCRIPTION	6
1.3 DESKTOP ENVIRONMENT DESCRIPTION	7
1.4 INSTALLATION AND SYSTEM REQUIREMENTS	7
1.5 LICENSE REGISTRATION.....	8
1.6 COMMUNICATION WITH THE OPENFOX™ MESSAGE SWITCH	8
CHAPTER 2: GENERAL OPERATION, MAINTAINING MESSENGER XML RESOURCES: FORMS	9
2.1 DESKTOP GENERAL OPERATION	9
2.2 FORM CONFIGURATOR GENERAL OPERATION	9
2.3 EXAMPLE WALK-THROUGH OF A SIMPLE FORM MAINTENANCE OPERATION.....	10
2.4 XML TREE [SCREEN-AREA].....	12
2.4.1 Action: Add	15
2.4.2 Action: Change To	18
2.4.3 Action: Delete	20
2.4.4 Action: Move Up and Move Down.....	21
2.4.5 Action: Cut, Copy, Paste and Paste from History.....	24
2.4.6 Action: Drag-And-Drop.....	27
2.4.7 Action: Shift-Drag-And-Drop	27
2.5 ATTRIBUTE-VALUE TABLE [SCREEN-AREA].....	28
2.6 FORM DISPLAY [SCREEN-AREA].....	31
2.7 OPEN RESOURCE FUNCTIONALITY	32
2.8 NEW RESOURCE FUNCTIONALITY	34
2.9 SAVE AND SAVE-AS RESOURCE FUNCTIONALITY	35
2.10 DELETE RESOURCE FUNCTIONALITY	38
2.11 UNDO-REDO FUNCTIONALITY.....	40
2.12 FIELD ORDER WINDOW.....	41
2.13 FIND WINDOW	43
2.14 PRINTING	45
2.15 MENU-ITEM: REFRESH DISPLAYED FORM.....	46
CHAPTER 3: MAINTAINING MESSENGER XML RESOURCES: FORMS, CODE LISTS, SCRIPTS AND FORMS MENU	49
3.1 GENERAL DESCRIPTION	49
CHAPTER 4: MAINTAINING MESSENGER XML RESOURCE: NCIC VEHICLE CODES LISTS 	51
4.1 GENERAL DESCRIPTION	51
4.2 EDIT MAKE AND/OR AUTOMOBILE MODELS [TAB].....	51
4.2.1 Example Walk-Through: Adding a Make Record	52
4.2.2 Category [Radio Buttons] and [Combo Box].....	53
4.2.3 Make [Table]	54
4.2.4 Make [Fields] and [Buttons]	55
4.2.5 Vehicle Type('s) [Check Boxes] and V.T. Clear All [Button].....	56
4.2.6 Example Walk-Through: Adding an Automobile Model Record.....	58
4.2.7 Automobile Model [Table].....	60
4.2.8 Automobile Model [Fields] and [Buttons].....	61
4.3 'EDIT MODEL AND/OR STYLE' [TAB]	62

4.3.1	Example Walk-Through: Adding a Model Record.....	63
4.3.2	Vehicle Type [Radio Buttons] and [Combo Box]	64
4.3.3	Model [Table].....	65
4.3.4	Model [Fields] and [Buttons].....	66
4.3.5	Example Walk-Through: Adding a Style Record	67
4.3.6	Style [Table].....	68
4.3.7	Style [Fields] and [Buttons]	69
CHAPTER 5: MAINTAINING ALL GENERIC XML RESOURCES.....		71
5.1	GENERAL DESCRIPTION	71
5.2	CHANGES TO BEHAVIOR OF ‘XML TREE’ [SCREEN-AREA]	72
5.3	ELEMENT FIELD [SCREEN-AREA]	72
5.4	CHANGE TO BEHAVIOR OF ‘XML ATTRIBUTE-VALUE TABLE’ [SCREEN-AREA].....	73
5.5	TEXT FIELD [SCREEN-AREA].....	73
5.6	‘ADD XML CHILD ELEMENT’ WINDOW	74
CHAPTER 6: PREFERENCES		75
6.1	GENERAL DESCRIPTION	75
6.2	MOVE TO NEW ROW ON AD.....	77
6.3	POPUP ‘FIELD ORDER WINDOW’ ON ADD	78
6.4	REFRESH ‘FORM DISPLAY’ AFTER UPDATING A XML ELEMENT.....	78
6.5	CHANGE COLORS IN ‘XML TREE’ [SCREEN-AREA] AND FOR PRINTING	80
CHAPTER 7: EXPORT/IMPORT FUNCTIONALITY		83
7.1	GENERAL DESCRIPTION	83
7.2	EXPORT FUNCTIONALITY	83
7.3	IMPORT FUNCTIONALITY	85
CHAPTER 8: RUNNING HELP SYSTEM COMMANDS.....		89
8.1	GENERAL DESCRIPTION	89
8.2 HELP SYSTEM INTRODUCTION.....		89
8.3 ‘COMPLETE CLEAN & REBUILD’ COMMAND		91
CHAPTER 9: BACKUPS		93
9.1	GENERAL DESCRIPTION	93
9.2	RESTORING A BACKUP	94
CHAPTER 10: MISCELLANEOUS		97
10.1	ERROR HANDLING AND REPORTING.....	97
10.2	OPERATION USING KEYBOARD ONLY.....	98
10.2.1	Using Alt-Keys and Ctrl-Keys.....	98
10.2.2	Accessibility Menu	99
10.2.3	Using Tab and Shift-Tab.....	100
10.2.4	Keyboard Only Behavior of ‘Attribute-Value Table’ [Screen-Area].....	100
10.3	ABOUT WINDOW.....	102
APPENDIX 103		
A.	DEFINITION OF TERMS	103

Chapter 1: Introduction and Overview

1.1 Organization of Related Form Configurator Documentation

The Form Configurator can be viewed as a programming tool. The languages that it programs in are the ‘Messenger Definition Languages’. The object is to use Form Configurator to create a ‘Messenger Definition Language’ file/program that is then used by Messenger to display and provide a look-and-feel for a specific Messenger form. All the ‘Messenger Definition Languages’ are XML compliant languages. Note that there are 6 ‘Messenger Definition Languages’. They are as follows:

- **Messenger Form Language** – This the most commonly used and the most complicated language. It is used to display and provide a look-and-feel for Messenger forms. There is one ‘Messenger Form Language’ file/program for each form that Messenger displays.
- **Messenger Code Language** – This language is used to maintain a list of codes, which can then be shared across forms. For example, all valid entries for ‘Scars Marks and Tattoos’ are maintained in one ‘Messenger Code Language’ file/program. This file is then in turn used by many ‘Messenger Form Language’ files/programs to populate ‘Scars Marks and Tattoos’ drop down lists displayed within the forms.
- **Messenger Script Language** – This language defines logical relationships between different text-fields (or other field types) in a Messenger form. For example, these relationships can tell a form that, if one text-field contains data, then another text-field must also contain data, before the form can be successfully submitted.
- **Messenger Menu Language** – This language is used to define the menu of forms Messenger presents to users in the lower left-hand-side of the Messenger main screen.
- **Messenger NCIC Vehicle Codes Language** – This language is used to maintain all vehicle make, model and style information used across all Messenger forms.
- **Messenger Default Language and Messenger Install Language** – These Messenger languages are treated as plain XML by Form Configurator, hence they are grouped together. These languages are rarely used. The few Messenger resources that use them are most likely set up at the time of system installation and are accessed only by CPI personnel after that time.

There are several documents that describe various subjects concerning Form Configurator, Messenger and the ‘Messenger Definition Languages’ they use. A brief description of the other documentation is provided below:

- OpenFox™ Messenger XML Forms Spec – The purpose of this document is to describe all the XML elements (and their corresponding XML attributes)

that are used in the ‘Messenger Form Language’ to create a Messenger form. This document covers the following subjects:

- The parent – child relationship of XML elements.
- Valid values (or range of values) of all XML attributes.
- The default value of XML attributes.
- A brief description of the effect XML attributes have on their corresponding XML element or the Form.
- A brief Java programmer oriented description for some XML elements. Please refer to the document itself for more information.
- **Messenger Form Developer Guide** – This document is intended as a guide for developers of OpenFox™ Messenger forms and discusses high-level issues related to the usage of the ‘Messenger Definition Languages’ used by Form Configurator and Messenger. Some issues discussed are: naming conventions, form tightness, form grouping, when to capitalize, form menu structure, etc. Please refer to the document itself for more information.
- **OpenFox™ Form Configurator Guide** – This document. Its purpose is to describe, from a users standpoint, how to use the ‘Messenger Definition Language’ programming tool, how to use Form Configurator to create/modify/delete Messenger forms and all other related Messenger resources. This document does not, however, discuss the ‘Messenger Definition Languages’ that are used to create Messenger forms and other Messenger resources. For example, this document describes how to add, delete, cut, copy, paste, move, edit, save, print, undo/redo, etc. ‘Messenger Definition Language’ XML elements in a ‘Messenger Definition Language’ file, but, it will not explain which ‘Messenger Definition Language’ element should be used nor why, nor the relationship between ‘Messenger Definition Language’ elements or which ‘Messenger Definition Language’ resource/file should be used to create/modify in order to obtain a specific look-and-feel when creating or modifying a Messenger form.
- **OpenFox™ Desktop Guide** – The OpenFox™ Form Configurator module runs within the OpenFox™ Desktop Java application. Please refer to the *OpenFox™ Desktop Guide* document for more information on OpenFox™ Desktop.

1.2 Form Configurator General Description

The purpose of the OpenFox™Form Configurator Desktop Java module is to maintain all ‘Messenger Definition Language’ resources utilized by the OpenFox™ Messenger Java modules, and, furthermore, all XML resources used by OpenFox™ Desktop along with all associated Java GUI (Graphical User Interface) modules (such as Messenger, Form Configurator, Archive Retrieval, Interceptor, etc.). The Messenger GUI is a very important and dynamic functionality. For example, the information that Messenger uses to display each and every form (and control the

forms look-and-feel) is stored in ‘Messenger Definition Language’ files. Those ‘Messenger Definition Language’ files are maintained by Form Configurator and hence change the look-and-feel of the Forms that Messenger displays.

Form Configurator can be viewed as a programming tool. The languages that it utilizes are the ‘Messenger Definition Languages’. The object is to use Form Configurator to create a ‘Messenger Definition Language’ file/program that is then used by Messenger to display and provide a look-and-feel for a specific Messenger form. The ‘Messenger Definition Languages’ are XML compliant languages. For more information about the ‘Messenger Definition Language’ used to program Form Configurator, please refer to other documentation (See section 1.1 of this document as a general guide to other documentation).

Note that Form Configurator allows the maintenance of various Messenger resources not just Messenger form resources (written in the ‘Messenger Form Language’). For example, a given Messenger form commonly uses resources such as Messenger codes (i.e. ‘Messenger Code Language’) and Messenger scripts (i.e. ‘Messenger Script Language’). A Messenger code resource contains a specific list that the form displays, such as the list displayed for ‘Scars Marks and Tattoos’ or ‘Guns’. A Messenger script resource defines logical relationships between different text-fields (or other field types) in a Messenger form. For example, these relationships can tell the form that if one text-field contains data, then another text-field must also contain data before the form can be successfully submitted.

1.3 Desktop Environment Description

Please refer to the *OpenFox™ Desktop Guide* documentation for more information on this subject. In general, OpenFox™ Desktop is a client application designed to run on client workstation computers. After logging onto Desktop, the various OpenFox™ Desktop Module applications that run within the Desktop environment can be launched.. The user must have a properly configured, licensed, and registered instance of OpenFox™ Desktop before he or she may run a specific Desktop Module, such as Form Configurator. Once logged into Desktop, an authorized user can run Form Configurator.

1.4 Installation and System Requirements

Form Configurator is an OpenFox™ Desktop module. Therefore all installation and system requirement issues are the same as those for OpenFox™ Desktop, which needs to be installed on the OpenFox™ Message Switch server (usually a UNIX server). This is usually done during system installation. Desktop also needs to be installed on each Windows PC that will be running Interceptor. Please refer to your appropriate support personnel and the *OpenFox™ Desktop Guide* for specific information and help on installing and setting up Desktop.

1.5 License Registration

Form Configurator requires that the user already has an installed, configured, licensed, and registered copy of OpenFox™ Desktop. A user will not be able to access Form Configurator until he or she completes all the steps required to run OpenFox™ Desktop. For more information on OpenFox™ Desktop, please consult the *OpenFox™ Desktop Guide* documentation.

1.6 Communication with the OpenFox™ Message Switch

All Form Configurator communication with the OpenFox™ Message Switch is done through OpenFox™ Desktop. Please refer to the *OpenFox™ Desktop Guide* document for more specific information on this subject. OpenFox™ Desktop uses the FoxTalk™ communications protocol to communicate with the OpenFox™ Message Switch. FoxTalk™ provides the necessary means to negotiate end-of-line sequences, encryption, object encoding, and guaranteed delivery. All FoxTalk™ communication is accomplished using many different internet technologies, such as TCP/IP, XML and data encryption.

Chapter 2: General Operation, Maintaining Messenger XML Resources: Forms

2.1 Desktop General Operation

Please refer to the *OpenFox™ Desktop Guide* document for specific information on this subject. When running Form Configurator within Desktop, the main interaction a user will have with specific Desktop functionality is to logon to OpenFox™ Desktop and then to open Form Configurator from within Desktop. Another possible situation for the user operations is to view any Form Configurator error messages with the Desktop log file viewer. For more information about standard features provided by OpenFox™ Desktop, please refer to the *OpenFox™ Desktop Guide* documentation.

2.2 Form Configurator General Operation

The primary purpose of the OpenFox™ Form Configurator is to maintain all OpenFox™ Messenger forms and all OpenFox™ Messenger resources for forms. More specifically, OpenFox™ Form Configurator maintains ‘Messenger Definition Language’ resources used by the OpenFox™ Messenger, as well as all XML resources used by OpenFox™ Desktop and all associated Java GUI (Graphical User Interface) modules (such as Messenger, Form Configurator, Archive Retrieval, Interceptor, etc.). For a brief introduction to the different ‘Messenger Definition Languages’ that Form Configurator supports, please refer to section 1.1. Additionally, Form Configurator can be viewed as a programming tool. The languages that it programs in are the ‘Messenger Definition Languages’.

A high-level overview of a typical Form Configurator form modification operation consists of the following steps:

1. Open a Messenger resource (typically a Messenger form resource).
2. Edit the resource. This is primarily done using the ‘XML Tree’ screen-area and the ‘XML Attribute-Value’ screen-area in Form Configurator.
3. Use the ‘Form Display’ screen-area in Form Configurator to preview and test the modified form that is being worked on.
4. When done using Form Configurator to edit and test the form, save the modified form to a different temporary name. For example, when changing the XA form (NCIC Cancel Article), change its name to XA-TEST before saving it. In this way the changes made will not affect the XA form that is currently being used by other users.

5. After the form is saved, use Messenger to re-open the form that was just edited and saved, being sure to use the forms temporary test name, which was XA-TEST in this case. Next, perform any final testing of the changes that were made to that form using Messenger. For example, at this point it may be wise to actually fill out and submit the form using Messenger for a complete system-wide test of the form.

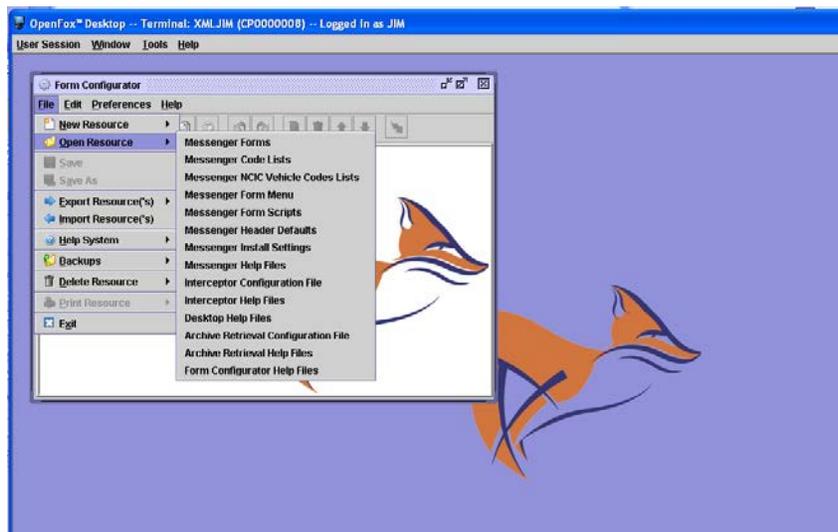
When testing the form with Messenger is finished, go back to Form Configurator and change the form name back to its proper name (change XA-TEST back to XA), and finally save the form again. Lastly, use Form Configurator to delete the temporary XA-TEST form.

Note that once a form is saved (or any resource) with Form Configurator, that form is immediately made available to all Messenger users. However, if a Messenger user had the form open while the form was saved with Form Configurator, the changes made won't be seen until exiting and reopening the form.

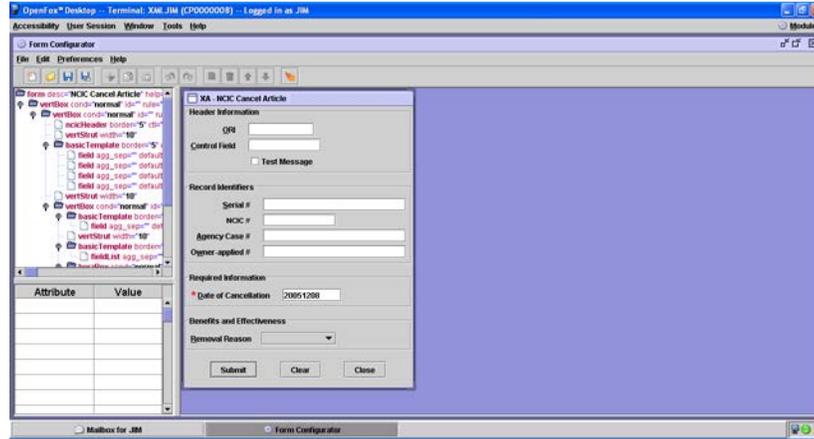
2.3 Example Walk-Through of a Simple Form Maintenance Operation

Following is a detailed walk-through of a Messenger form modification operation. Assume the goal is to modify the XA form (NCIC Cancel Article), specifically modifying the title of the 'NCIC#' field to 'NCIC Number'. Also, assume that the OpenFox™ Desktop has been logged onto and the Form Configurator has been brought up by clicking 'Modules' on the Desktop menu-bar and then clicking on 'Form Configurator' in the drop down list. The steps are listed below:

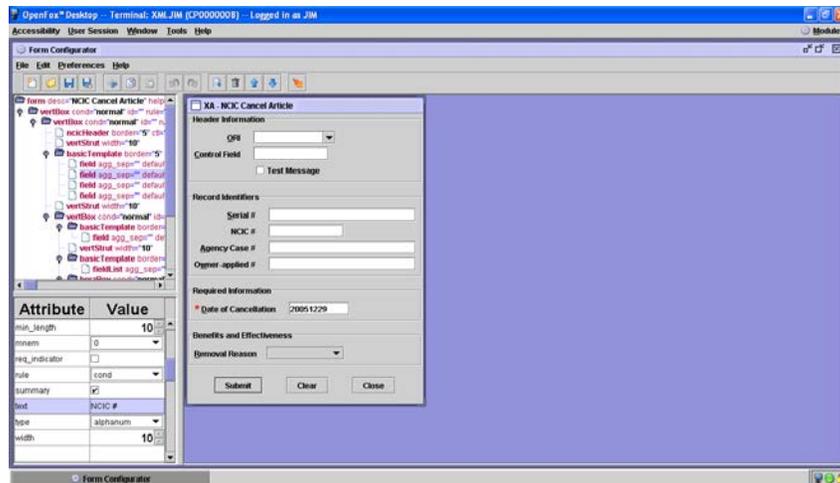
1. Click 'File' on the Form Configurator menu-bar.
2. Highlight 'Open Resource' from the displayed list. At this point the screen should look as shown below.



3. Highlight and click on ‘Messenger Forms’. This causes a dialog box to appear listing all forms available on the OpenFox™ system. The cursor should be in the ‘Open’ field. Type XA. Hit the enter key.
4. The XA (NCIC Cancel Article) form should now be opened and displayed in Form Configurator. The screen should look as shown below:



5. Click on the ‘NCIC#’ field. This will cause the row to be highlighted in the ‘XML Tree’ screen-area that controls the behavior and displaying of the NCIC# text and field in the form. Also once a row is highlighted in the ‘XML Tree’ screen-area, data is populated in the ‘XML Attribute-Value Table’ screen-area. The ‘XML Attribute-Value Table’ is populated with all the XML attributes, and the value of each attribute, for the highlighted row in the ‘XML Tree’ screen-area.
6. There is a vertical scrollbar on the right-side of the ‘XML Attribute-Value Table’. Move the cursor over this scrollbar and hold down the left mouse button. Scroll down your mouse on the scrollbar until the ‘text’ attribute is visible in the ‘Attribute’ column.
7. Now double-click on the table cell that contains the value of the ‘text’ attribute, which is the cell that currently contains the text ‘NCIC#’. The screen should look like this:

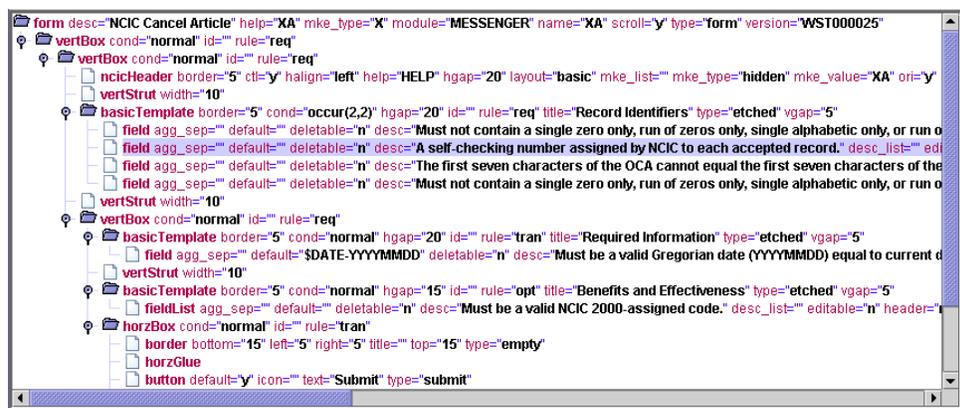


8. Next, type in the new value desired for the ‘text’ attribute ‘NCIC Number’, and hit the enter key. When looking at the displayed form, notice that the text has changed to ‘NCIC Number’.
9. The last thing needed is to save this form, after which the form will be released to the OpenFox™ system, where it can be used by everyone. Normally, when modifying a form, first view and test the form within Form Configurator. Then save the form to a temporary form name, for example XA-TEST, and then use Messenger to open the form and perform final testing of the form. Once all testing is completed, go back to Form Configurator and save the form again, using the ‘Save-As functionality’. This time save the form using its real name, XA in this case. Finally use Form Configurator to delete the temporary XA-TEST form. However, since the change made was so minor, further testing on the form using Messenger did not need to be done, so for this example, just directly save the form. Go to the Form Configurator menu-bar and click on ‘File’, and then click on ‘Save’. That completes this example.

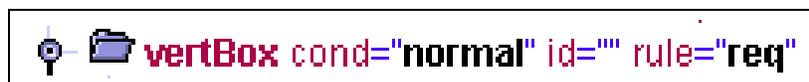
2.4 XML Tree [Screen-Area]

The primary purpose of the ‘XML Tree’ screen-area is to display the underlying XML data that defines the current resource, typically a form that the user is viewing with Form Configurator. It also allows the user to add, change-to, delete, move-up, move-down, cut, copy, paste, paste-history, and drag-and-drop the XML elements of the resource. The ‘XML Tree’ screen-area also shows the order and parent-child relationship of the XML elements that define a resource.

An expanded view of the ‘XML Tree’ screen-area for the XA form is shown below.



Each XML element is represented by an entire row in the ‘XML Tree’. Following in detail is one XML element from the XA (NCIC Cancel Article) form:



The four different parts of this XML element (and for that matter any XML element) are as follows:

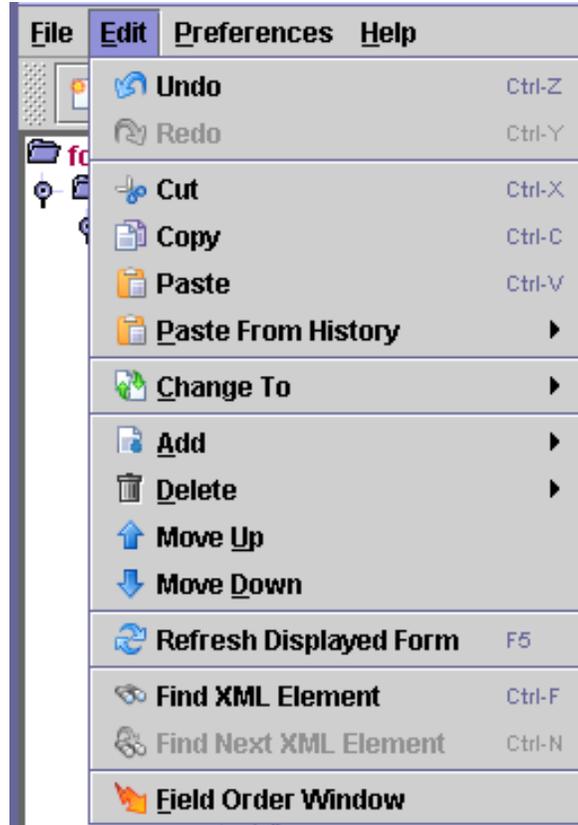
- Name (or Tag) – ‘vertBox’
- Attributes – this XML element has three attributes: ‘cond’, ‘id’ and ‘rule’
- Attribute-Values – normal for ‘cond’, no value for ‘id’ and req for ‘rule’.
- XML markup characters – The equal signs and quotes. These are by default shown in blue and are of no concern since handling of all XML markup characters is automatically handled by Form Configurator.

The color coding of the ‘XML Tree’ screen-area and the colors of the different parts that make up an XML element are completely configurable. This is discussed in Chapter 6: Preferences. The folder icon shown on the left indicates that this is a parent element, hence meaning it contains child elements. If a closed folder icon was displayed instead, it still means this is a parent XML element, but the children of this parent are not currently being shown on screen. Within the ‘XML Tree’ screen-area, clicking on the magnifying glass shown on the left opens and closes the parent XML elements. This is useful when wishing to collapse or expand the ‘XML Tree’. Also, when clicking on and highlighting an XML element, the left and right arrow keys can be used to collapse or expand a parent element. A page icon is shown on the left-hand side for child XML elements. Child XML elements do not contain any other elements. However, note that some child XML elements can be converted into parent XML elements by simply adding a child XML element under them.

The curser can now be moved around the ‘XML Tree’ screen-area using the mouse, the up and down arrow keys, the page-up and page-down keys or by using the scrollbars.

There are scrollbars shown on both the right-hand side and bottom of the ‘XML Tree’ screen-area. These allow different sections of the XML resource in Form Configurator to be viewed. Also, there are fat borders shown with a dotted texture on both the right-hand side and bottom of the ‘XML Tree’ screen-area. After placing the cursor on these borders, hold down the left mouse button to drag these borders in order to expand or contract the size of the ‘XML Tree’ screen-area.

After selecting by highlighting a row (XML element) in the ‘XML Tree’ screen-area, right-click the mouse (or just hit the enter key) to bring up an ‘Action’ popup window which displays various actions the can be performed at this location in the XML resource. The possible actions that can be preformed are: add, change-to, delete, move-up, move-down, cut, copy, paste and paste-history. All these actions can also be performed by selecting a row and then navigating under the ‘Edit’ option located on the Form Configurator menu-bar shown following:



Most of these actions are also available by selecting a row and then clicking on the appropriate button located in the Form Configurator toolbar shown below.



Note that when selecting a row (XML element) and right-clicking the mouse button to bring up the 'Action' popup window, only the actions that are valid at that location in the XML resource are shown. For example if a 'vertBox' XML element is selected and the mouse button is right-clicked, the 'Action' popup window contains 'Add' but not 'Change To' in the list of actions you can perform.



However, if a 'field' XML element is selected and the mouse is right-clicked, the 'Action' popup window contains 'Change To' but not 'Add' in the list of actions you can perform.

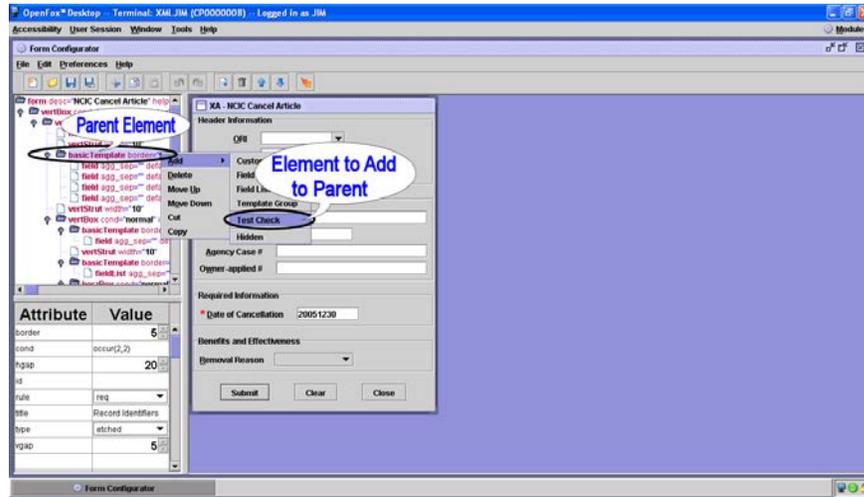


These actions are covered in more detail in later sections of this document.

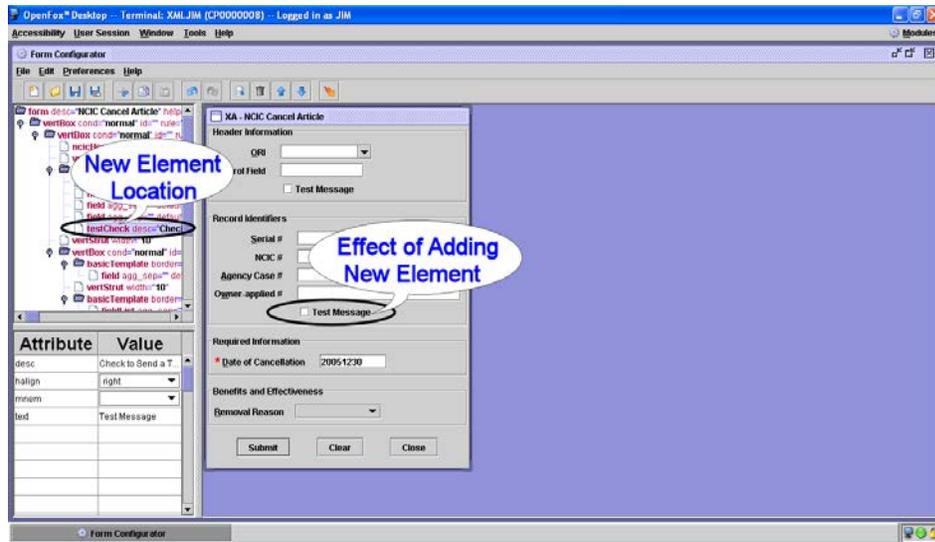
Note: When making a change in the 'XML Tree' screen-area, the effect of that change (or action) on the form currently open is automatically displayed in the 'Form Display' screen-area.

2.4.1 Action: Add

'Add' is a valid action whenever a row is selected that contains an XML element that can contain child elements. For example, if a 'vertBox' is selected, which can always have child elements, and the mouse is right-clicked, the 'Action' popup window will be brought up. Moving the mouse over the 'Add' action will display the list of valid children that can be added under a 'vertBox'.



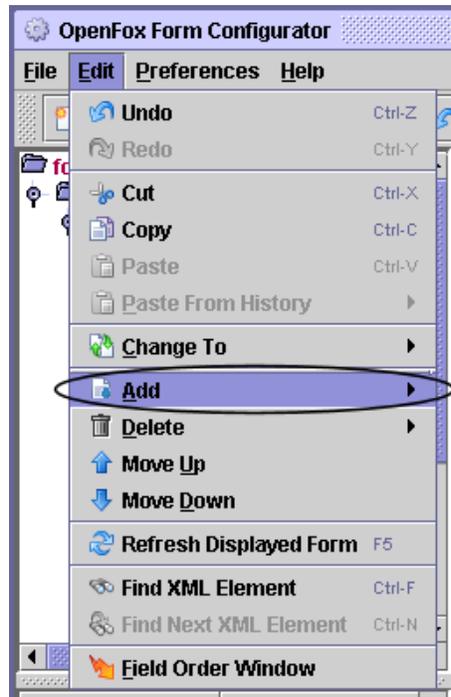
The new element is added as the last child for that parent XML element.



Notice that immediately after completing an 'Add' action in the 'XML Tree' screen-area, the effect of that action on the form currently open is automatically displayed in the 'Form Display' screen-area.

Also, by default, the selected row in the 'XML Tree' screen-area automatically moves to the new XML child element which was just added. If preferred, this can be stopped from happening by un-checking the Preferences option 'Move to new row on add' shown in the 'Preferences' window. Please refer to "Chapter 6: Preference" for more information on this subject.

‘Add’ action can also be performed by selecting an XML element that can have children and then navigating to ‘Add’ under the ‘Edit’ option located on the Form Configurator menu-bar.



An ‘Add’ can also be executed by selecting a row and clicking on the add button on the toolbar.

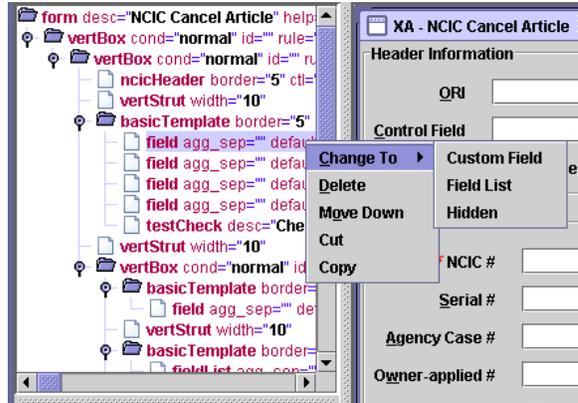


2.4.2 Action: Change To

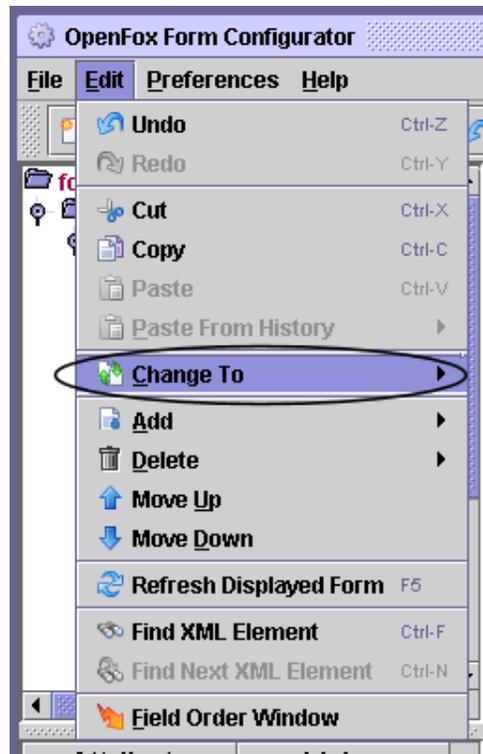
The ‘Change To’ action is only available for elements that are similar to other XML elements, such as the ‘edit’ and ‘edit Area’ elements in the ‘Messenger Forms Language’. Also, it is not available for any elements that can or do have child elements (i.e. parent elements). ‘Change To’ is a time saving action that allows one element to change or convert into another. Ideally when a ‘Change To’ is executed, only the name of the element is changed. All attributes that are contained in both the source and target element remain unchanged when a ‘Change To’ is executed. In this way the attribute values for the attributes that are shared across both the source and target elements do not have to be re-entered..

‘Change To’ action can be executed in one of two ways after selecting an element in the ‘XML Tree’ screen-area that is a valid ‘Change To’ candidate. Use the Action popup window, which can be accessed by

hitting the enter key, or by clicking the right mouse button while the cursor is over the selected element.



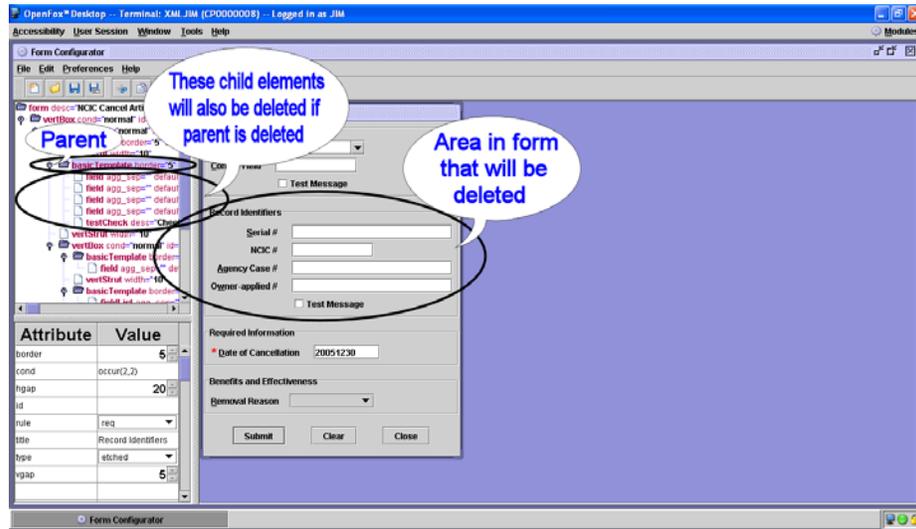
Or, navigate to 'Change To' under the 'Edit' option on the Form Configurator menu-bar.



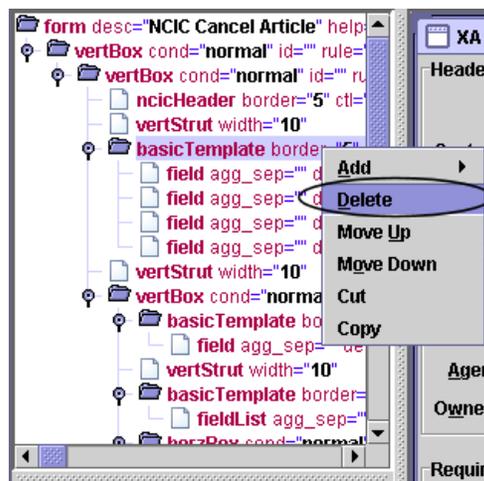
After clicking on 'Change To', simply pick from the list of elements to which the target element can be changed. After completing a 'Change To' action in the 'XML Tree' screen-area, the effect of that action on the form currently open is automatically displayed in the 'Form Display' screen-area.

2.4.3 Action: Delete

The ‘Delete’ action can be performed on all elements except for the root element (i.e. the ‘form’ element for ‘Messenger Forms Language’ resource type). When a parent element is deleted, all its child elements are also deleted.



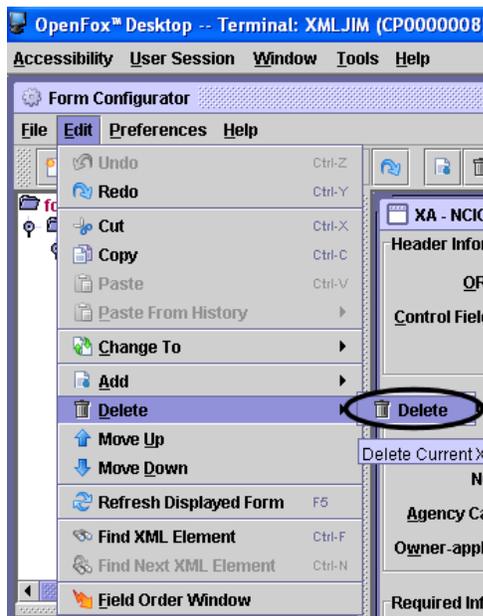
After selecting an element in the ‘XML Tree’ screen-area, the ‘Delete’ action can be executed in several ways. Either hit the delete key or the backspace key. Use the Action popup window which can be brought up by hitting the enter key or by clicking the right mouse button while the cursor is over the selected element.



Another choice is to use the ‘Delete’ button located on the Form Configurator toolbar and shown below:



Or lastly, navigate to ‘Delete’ under the ‘Edit’ option on the Form Configurator menu-bar.

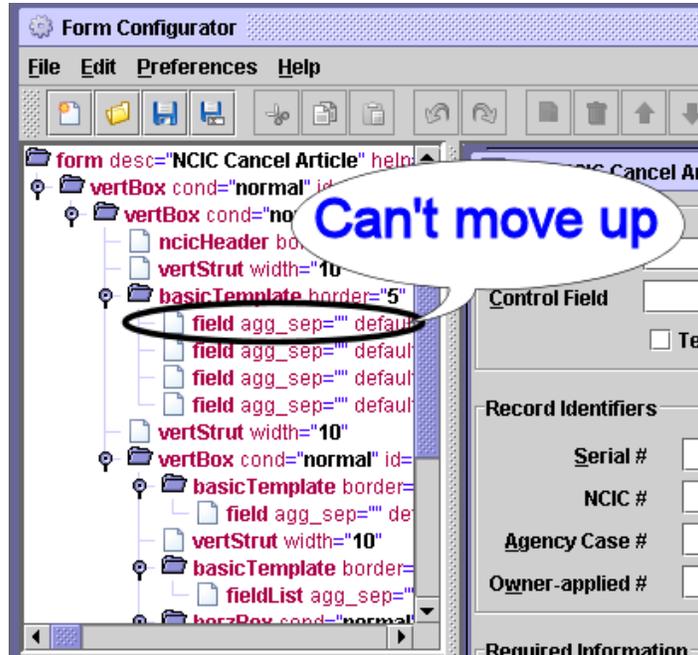


After completing a ‘Delete’ action in the ‘XML Tree’ screen-area, the effect of that action on the form currently open is automatically displayed in the ‘Form Display’ screen-area.

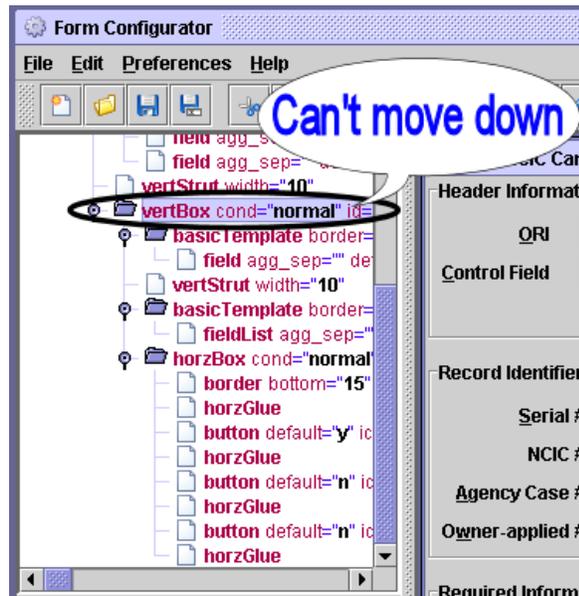
2.4.4 Action: Move Up and Move Down

As long as an element has sibling elements both above and below it in the ‘XML Tree’ screen-area, either the ‘Move Up’ or ‘Move Down’ action can be executed. If the element being moved is a parent element, all children of that element remain children of that element and thus are moved along with the parent element.

Note that ‘Move Up’ is not a valid action if there are no siblings above the selected element attempting to be moved.



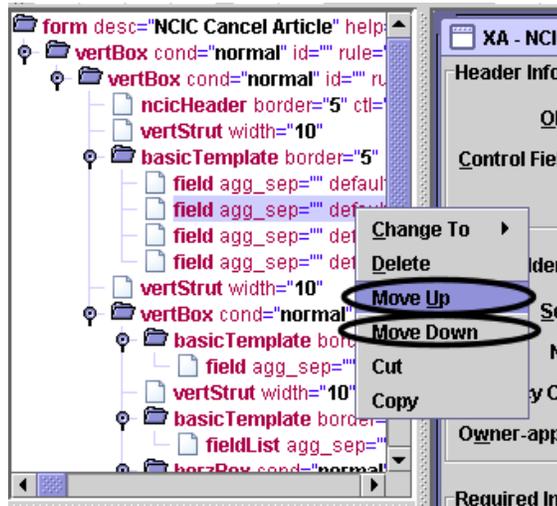
Likewise 'Move Down' is not valid if there are no sibling elements below the selected elements.



Therefore, the 'Move Up' and 'Move Down' actions can not change the parent of the selected element being moved. Please refer to section '2.4.6 Action: Drag-And-Drop' for an operation that allows an element to be moved and its parent element to be changed. Also, refer to section '2.4.5 Action: Cut, Copy, Paste and Paste from History' and to section '2.4.7

Action: Shift-Drag-And-Drop' for an action that allows an element to be moved above/below several sibling elements in one operation.

After selecting an element in the 'XML Tree' screen-area, the 'Move Up' and 'Move Down' actions can be executed in several ways. The Action popup window can be accessed by hitting the enter key, or by clicking the right mouse button while the cursor is over the selected element.

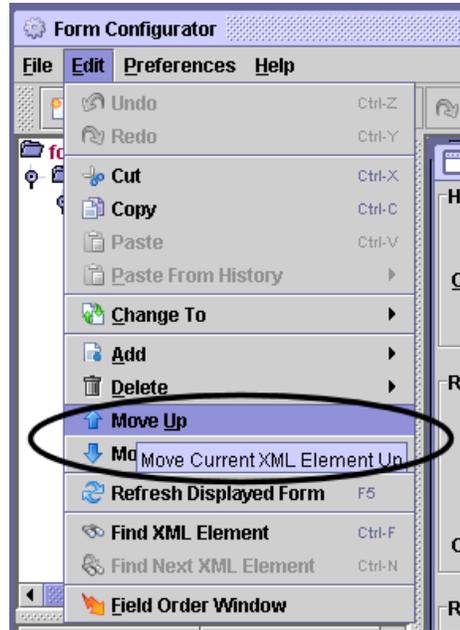


Note that if either 'Move Up' or 'Move Down' are not valid for the element being selected, then they will not appear in the Action popup window.

Secondly, the 'Move Up' or 'Move Down' buttons shown in the Form Configurator toolbar can be clicked.



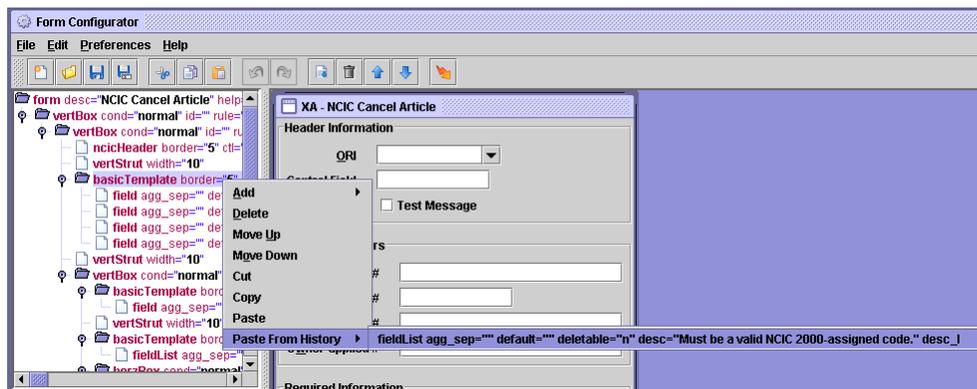
Or lastly, you can navigate to 'Move Up' or 'Move Down' under the 'Edit' option on the Form Configurator menu-bar.



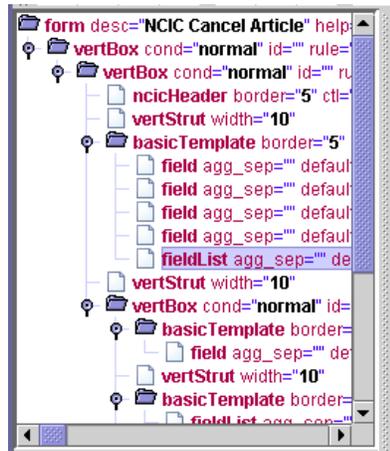
After completing a 'Move Up' or 'Move Down' action in the 'XML Tree' screen-area, the effect of that action on the form currently open is automatically displayed in the 'Form Display' screen-area.

2.4.5 Action: Cut, Copy, Paste and Paste from History

The 'XML Tree' screen-area supports standard cut, copy and paste like functionality. An XML element can only be pasted onto a XML element that can contain children. When pasting an element onto a parent element, the pasted element is automatically added to the end of the children for that parent element. The following screen shot shows how the screen looks just before a paste from history.



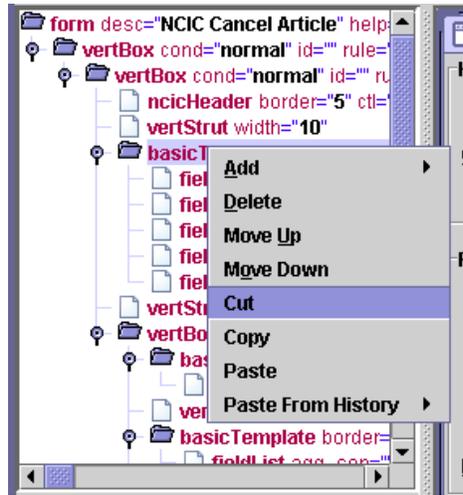
Shown below is the screen immediately after a paste from history.



It should be noted that Form Configurator performs validation when a 'Paste' or 'Paste from History' action is executed. If the paste action is not valid according to the rules of the 'Messenger Definition Language' for the resource currently being edited in Form Configurator, the paste action is aborted and an error message is displayed.

The 'Paste from History' functionality will store up to 10 of the elements that were cut or copied, whereas the 'Paste' option just stores the last element that was cut or copied. When closing one resource and opening another, for example closing the XA form and then opening the EW form, the elements stored in 'Paste' and 'Paste from History' remain unchanged. This is useful when copying elements from one form (or resource) to another. However, if during the process of closing one resource and opening another resource, a switch is made to a different resource type, such as closing the XA form resource and opening the HDR_MKE_EW codes resource, all elements stored in 'Paste' and 'Paste From History' are cleared. This occurs because an XML element cannot be copied from one resource type to another resource type.

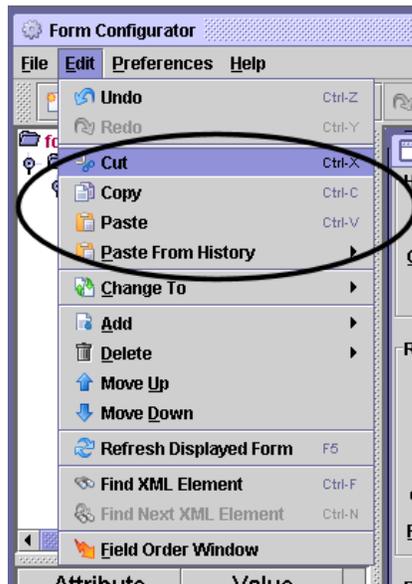
The 'Cut', 'Copy', 'Paste', or 'Paste from History' actions can be executed in several ways after you have selected an element in the 'XML Tree' screen-area. The standard cut, copy and paste hotkeys can be used. These are usually available in most editors, Ctrl-X, Ctrl-C and Ctrl-V. They can also be executed by using the Action popup window which can be accessed by hitting the enter key, or by clicking the right mouse button while the cursor is over the selected element.



The 'Cut', 'Copy' or 'Paste' buttons shown in the Form Configurator toolbar can be clicked.. Note there is no 'Paste from History' button on the toolbar.



Or lastly, navigate to 'Cut', 'Copy', 'Paste, or 'Paste from History' under the 'Edit' option on the Form Configurator menu-bar.



After completing a 'Cut', 'Copy', 'Paste' or 'Paste From History' action in the 'XML Tree' screen-area, the effect of that action (if any) on the

form currently open is automatically displayed in the 'Form Display' screen-area.

2.4.6 Action: Drag-And-Drop

There are two choices to move an element from under one parent element, and place it under another parent element. Either do a cut and paste, or just do a 'Drag-And-Drop'. An element can only be dropped onto an element that can have children. The dropped element is automatically added to the end of the children for that parent element. It should be noted that Form Configurator performs validation when a 'Drag-And-Drop' action is executed. If the 'Drag-And-Drop' action is not valid according to the rules of the 'Messenger Definition Language' for the resource currently being edited in Form Configurator, the 'Drag-And-Drop' action is aborted and an error message is displayed.

'Drag-And-Drop' is executed by clicking the left mouse button on an element in the 'XML Tree' screen-area. To add the element to a new destination, hold down the left mouse button and drag the mouse over the destination parent element being dropped onto. . Releasing the left mouse button will cause the element you were dragging to be dropped on to the parent element the mouse is currently over. Note that while executing a 'Drag-And-Drop' operation, scrolling to the very top or bottom of the 'XML Tree' screen-area while dragging with the mouse, will cause the 'XML Tree' screen-area to automatically scroll up or down. In this way, a 'Drag-And-Drop' operation can be done even if the new location of where the element is being moved is not displayed on screen when you started the 'Drag-And-Drop' operation.

After completing a 'Drag-And-Drop' action in the 'XML Tree' screen-area, the effect of that action (if any) on the form currently open is automatically displayed in the 'Form Display' screen-area.

2.4.7 Action: Shift-Drag-And-Drop

Moving an element up or down several positions, while still keeping the element under the same parent, can be done in two ways. Several 'Move Up's' or 'Move Down's' can be done in a row, or a 'Shift-Drag-And-Drop' can be done instead. The purpose of 'Shift-Drag-And-Drop' is to allow, in one operation, the user to move a child element multiple positions up or down while keeping that child element under the same parent.

‘Shift-Drag-And-Drop’ is executed in a manner very similar to ‘Drag-And-Drop’, except for the addition of the shift key. **The shift key must be held down first before clicking with the left mouse button to start a ‘Shift-Drag-And-Drop’ operation.** The entire sequence of events is as follows. First locate the element that is to be moved. Now hold down the shift key, then click and hold down the left mouse button. While holding down the shift key and left mouse button, scroll the mouse up or down to the new position desired for the target element to move to. Finally release the left mouse button to drop the target element into its new location. The ‘Shift-Drag-And-Drop’ operation will cause the target element to be inserted before or after the sibling element that it was dropped onto. Note that while executing a ‘Shift-Drag-And-Drop’ operation, scrolling to the very top or bottom of the ‘XML Tree’ screen-area while dragging with the mouse, will cause the ‘XML Tree’ screen-area to automatically scroll up or down. In this way, a ‘Shift-Drag-And-Drop’ operation can be done even if the new location of where the element is being moved is not displayed on screen when you started the ‘Shift-Drag-And-Drop’ operation.

After completing a ‘Shift-Drag-And-Drop’ action in the ‘XML Tree’ screen-area, the effect of that action (if any) on the form currently open is automatically displayed in the ‘Form Display’ screen-area.

2.5 Attribute-Value Table [Screen-Area]

The primary purpose of the ‘XML Attribute-Value Table’ screen-area is to allow the user to edit the attribute values of the current selected element in the ‘XML Tree’ screen-area. This table is also useful for just viewing the attributes and attribute values of the current selected element in the ‘XML Tree’ screen-area. Hit enter to commit any changes you made to an attribute-value in the ‘XML Attribute-Value Table’. The effect of those changes (if any) on the form currently open is automatically displayed in the ‘Form Display’ screen-area. The following is a snap shot of the ‘XML Attribute-Value Table’ when basic Template is selected in the ‘XML Tree’ screen-area.

Attribute	Value
border	5
cond	occur(2,2)
hgap	20
id	
rule	req
title	Record Identifiers
type	etched
vgap	5

The mouse, the up and down arrow keys, the page-up and page-down keys and the scrollbars can be used to move around the ‘XML Attribute-Value Table’ screen-area .

There are scrollbars shown on the right-hand side of the ‘XML Attribute-Value Table’ screen-area for scrolling up and down in the table. Also, there are fat borders shown with a dotted texture on both the right-hand side and top of the ‘XML Attribute-Value Table’ screen-area. When the cursor is placed on these borders and the left mouse button is held down, these borders can be dragged to expand or contract the size of the ‘XML Attribute-Value Table’ screen-area.

The Attribute column is not editable, unless the current open resource is a generic XML resource. This topic is covered in ‘Chapter 5: Maintaining All Generic XML Resources’. The Value column contains the editable data. The cells in the Value column can be of different editable types. The editable type of each cell in the Value column depends on the attribute it’s associated with, and furthermore the ‘Messenger Definition Language’ of the current resource opened in Form Configurator. Ultimately it’s the ‘Messenger Definition Language’ that defines what values are valid for a specific attribute. Form Configurator only uses the different cell types to enforce these rules.

Following is a list of the different cell types that can appear in the Value column.

- Text – All standard characters can be entered into these cells. Depending on the associated attribute, some text fields only allow upper cased letters, and some have limited lengths as well.
- Check Box – These are cells that are shown with a small box inside, either with or without a check in the box. They represent a true / false value. If checked, means set to true. False if unchecked. A Check Box cell is shown below.



- Number – Shown with small ‘spinner’ up/down arrow buttons on right-hand side of the cell. These cells only allow number characters to be entered. Depending on the attribute that’s associated with the cell, negative values can be entered as well. When editing cells of this type the number can be edited directly by clicking on the up/down arrow buttons to change the numbers value or by hitting the up/down arrow keys to change the numbers value. A number cell is shown below:



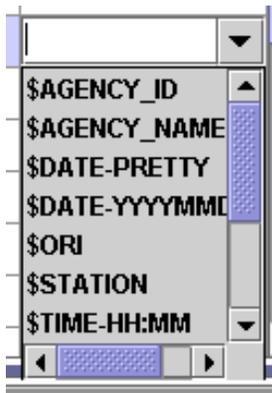
- Drop Down List – An editable Drop-Down List looks as follows when it is not selected.



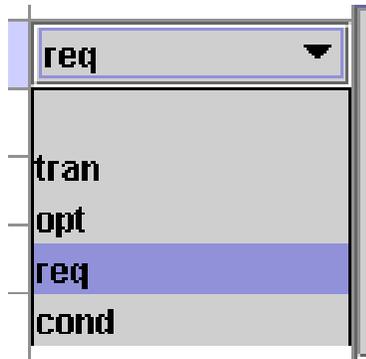
A non-editable Drop-Down List looks as follows when it is not selected.



Clicking on these cells results in a drop down list being displayed that allows the user to choose one value by scrolling through the list. Depending on the attribute associated with the cell, the drop down list may be editable. This allows a value to be typed into the cell just like a text cell instead of picking from the list of values. An expanded editable Drop-Down List is shown below.



An expanded non-editable Drop-Down List is shown below.



- Constant – There are a few attributes that have constant values and hence can not be changed. This is a rarely used cell type.

After editing a cell, hit the enter key to commit any changes or hit the escape key to cancel your changes and restore the cell to its previous value (its value before you started to edit the cell in question).

2.6 Form Display [Screen-Area]

This is the screen-area where the current open form (or other resource) is displayed. **The result of any changes made while editing a form (or other resource) is immediately displayed in the ‘Form Display’ screen-area.** The primary purpose of this screen-area is to allow a Form Configurator user to easily see how the form looks and behaves after making changes to the underlying ‘Messenger Definition Language’ resource that defines the form. Basically the ‘Form Display’ screen-area provides a means to test a form. Note that the ‘Form Display’ screen-area can not completely replace the need to eventually test a form in OpenFox™ Messenger. It’s recommended that you perform final testing on a form with OpenFox™ Messenger.

The ‘Form Display’ screen-area shown below contains an EW or ‘NCIC Enter Wanted Person’ form. Notice the scrollbars on the right-hand side and bottom that automatically appear if the entire form can not be displayed in the ‘Form Display’ screen-area. These allow the user to scroll to view the rest of the form.

The screenshot shows a web-based form titled "EW - NCIC Enter Wanted Person". The form is organized into several sections:

- Header Information:** Includes fields for *Message Key (dropdown), ORI (dropdown), Control Field (text), and a Test Message checkbox.
- Personal Information:** Includes fields for *Name, *Sex, *Race, Place of Birth, Date of Emancipation, *Height, *Weight, Eye Color, *Hair Color, FBI #, Skin Tone, Scars, Marks, Tattoos, Fingerprint Classification, Miscellaneous #, Social Security #, DNA, DNA Location, and Caution/Medical Condition.
- Offense Information:** Includes fields for *Offense, Original Offense, *Date of Warrant, *Agency Case #, Warrant #, Court Identifier, Notify Agency, and Extradition Limitation.
- Optional Linkage Information:** Includes fields for Linkage Agency ID and Linkage Case #.
- Operator Information:** Includes fields for Operator's License #, Operator's License State (dropdown), and Expiration Year.

Note: For the ‘Messenger Script Language’, ‘Messenger Default Language’ and ‘Messenger Install Language’ resource types nothing is displayed in the ‘Form Display’ screen-area. Also, for ‘Messenger NCIC Vehicle Codes Language’ resource type the ‘Form Display’ screen-area is not shown at all.

When a ‘Messenger Form Language’ resource type is open, click on a field in the ‘Form Display’ screen-area. The row in the ‘XML Tree’ screen-area that is responsible for displaying the field you clicked on is automatically selected and highlighted.

2.7 Open Resource Functionality

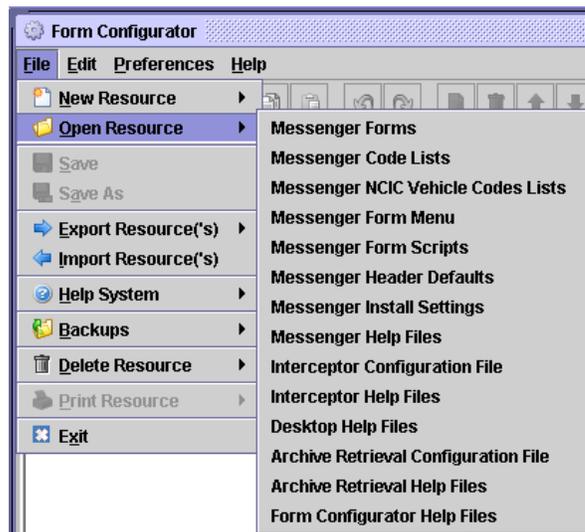
In order to edit or change a resource in Form Configurator, the resource must be opened first. The ‘Open Resource’ functionality can be accessed by clicking on the ‘Open Resource’ button on the Form Configurator toolbar.



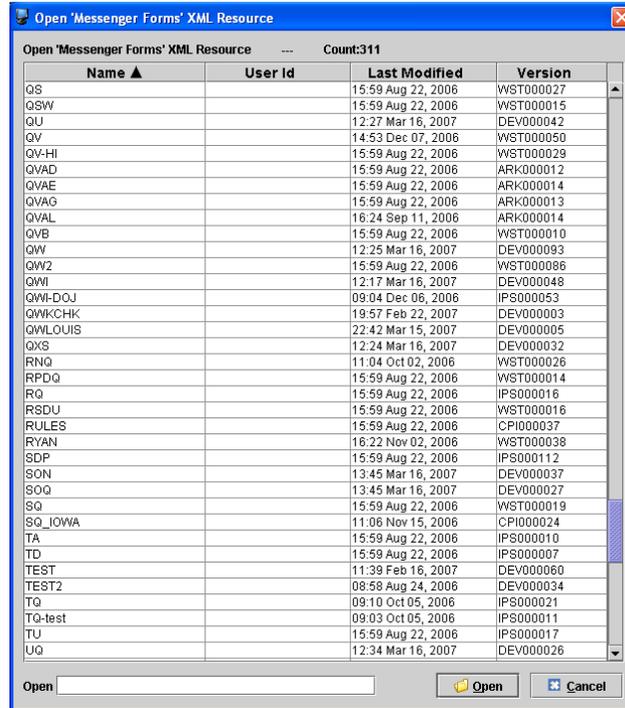
Or it may be accessed by clicking on ‘File’ in the Form Configurator menu-bar and navigating to ‘Open Resource’.



After clicking on ‘Open Resource’ (or the ‘Open Resource’ button on the Form Configurator toolbar) a list of resources appears.



Pick the resource type you want to edit from the list of available resource types. This causes a popup Resource dialog box to appear displaying all available XML resources for the resource type you choose. If 'Messenger Forms' is chosen as the resource type, the popup Resource dialog box should look something like the following screen snapshot:



There are four columns in the displayed table: 'Name', 'User Id', 'Last Modified' and 'Version'. All resources are listed in this table in alphabetical order. Notice there is an 'Open' field in the lower left-hand corner. This is where the name of the resource you want to open is typed. Then click the 'Open' button or instead just type the enter key. A resource can also be opened by using the mouse and double clicking a resource name in the table.

Notice that as characters are typed in the 'Open' field, the table automatically scrolls to display the row that best matches the character string currently entered in the 'Open' field.

Also, notice that there is a scrollbar on the right-hand side that can be used to scroll the table up or down. Look toward the top of the popup Resource dialog box. A count of the total number of resources of this type is always displayed here.



2.8 New Resource Functionality

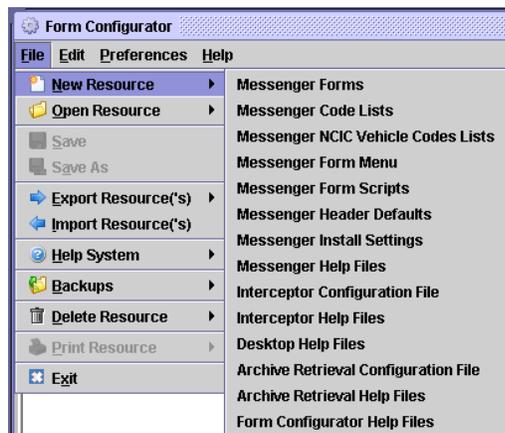
A new resource can be created by clicking on 'File' in the Form Configurator menu-bar and then navigating to 'New Resource'.



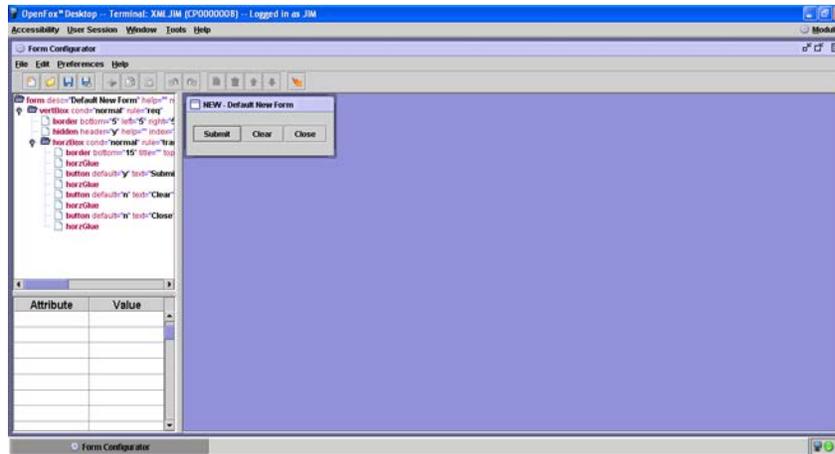
A new resource can also be created by clicking on the 'New Resource' button on the Form Configurator toolbar.



After doing one of the above, a list of the valid resource types appears.



Click on the resource type to be created. At this point, the Form Configurator screen will initialize itself to the new resource state for the resource type that was chosen. Note that the resource will contain the default values for that resource. For resource type ‘Messenger Forms’, the screen should look as follows.



2.9 Save and Save-As Resource Functionality

When making changes to a ‘Messenger Definition Language’ resource (or any resource for that matter) with Form Configurator, the changes are actually being made to a local copy of that resource which is stored on your computer. It is not until a ‘Save’ or ‘Save-As’ action is done that the changes are made available to all OpenFox™ Desktop System and all users (i.e. that includes all OpenFox™ Messenger users).

The following list details the primary actions that take place when a ‘Save’ a XML resource file with Form Configurator is done successfully.

- ➔ Form Configurator sends a copy of the current resource that is open (whether or not it’s a backup file version) to the FM server (Form Manager Server).
- ➔ The FM server calculates the next version number value for that resource, and modifies the version number field in the XML resource with that value.
- ➔ The FM server writes the XML resource to file, which now becomes the current version for this XML resource. Anyone using Messenger, or Form Configurator, etc. who tries to access or open this resource will be sent a copy of this version of the resource.
- ➔ The FM server writes a copy of the XML resource to the backup directory.
- ➔ The FM server sends back a successful acknowledgment message to the Form Configurator client. Imbedded in this message is the new version number for this resource. Form Configurator will use this new version

number to update the version number field of the resource, which it is currently displaying on screen.

The changes made to a resource can be saved by clicking on ‘File’ in the Form Configurator menu-bar and then navigating to ‘Save’.



Or, changes can be made by clicking on the ‘Save Resource’ button on the Form Configurator toolbar.



If a new resource was created and is being saved for the first time, the ‘Save’ will automatically become a ‘Save As’ operation.

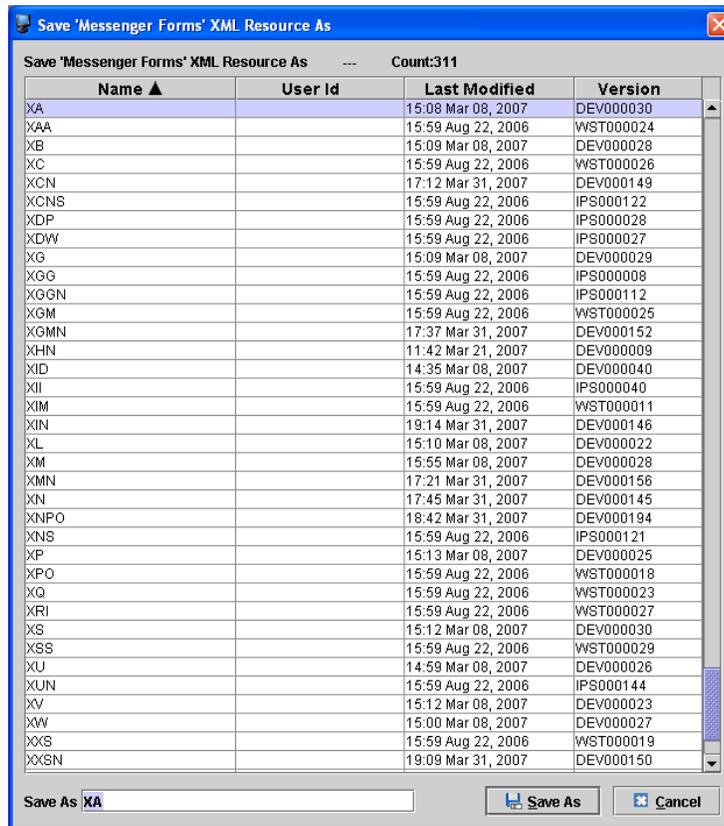
If a resource is to be saved and its name is to be changed as well, do a ‘Save As’ by clicking on ‘File, in the Form Configurator menu-bar and then navigating to ‘Save As’.



Or, the save may be done by clicking on the ‘Save as Resource’ button on the Form Configurator toolbar.



After clicking on the ‘Save as Resource’ button (or choosing ‘Save As’ under the ‘File’ menu option) a popup Resource dialog box appears displaying all available XML resources for the resource type currently open. If, for example, the XA ‘NCIC Cancel Article’ form is open, which is a ‘Messenger Form’ resource type, the popup Resource dialog box should look something like the following screen snapshot:



There are four columns in the displayed table: ‘Name’, ‘User Id’, ‘Last Modified’ and ‘Version’. All resources are listed in this table in alphabetical order. Notice there is a ‘Save As’ field in the lower left-hand corner. This field is initialized to the name of the form currently open, such as XA in this case. This field is where the name of the resource being saved can be changed. Click the ‘Save As’ button or instead just type the enter key. Or, the name of the resource can be saved to a new name by using the mouse and double clicking a resource name in the table.

Notice that when typing characters in the ‘Save As’ field, the table automatically scrolls to display the row that best matches the character string that was currently entered in the ‘Save As’ field.

Notice that there is a scrollbar on the right-hand side that can be used to scroll the table up or down. Also look toward the top of the popup Resource dialog box. A count of the total number of resources of this type is always displayed here.

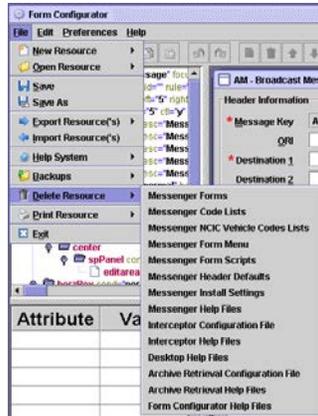
2.10 Delete Resource Functionality

A specific resource **can be permanently deleted** from the **entire OpenFox™ Desktop system** by utilizing the Form Configurator ‘Delete’ functionality. Be careful when using this functionality as a resource can not be recovered or retrieved after this is done..

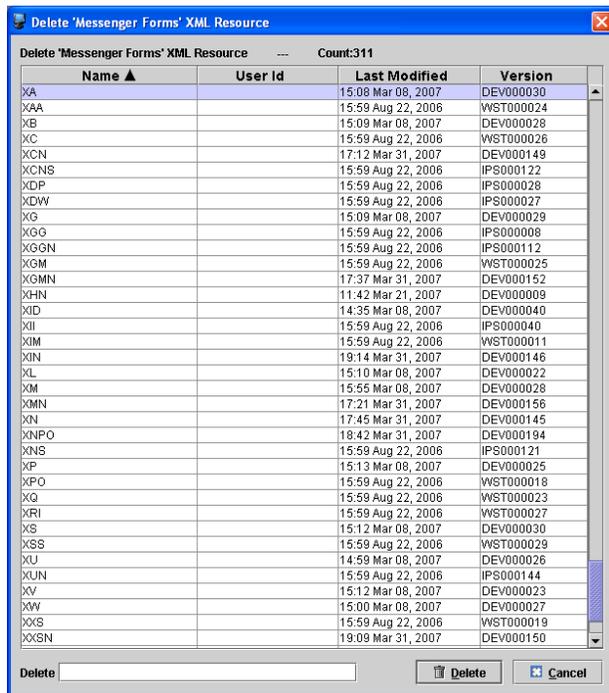
Execute this functionality by clicking on the ‘File’ option on the Form Configurator menu-bar and navigating to ‘Delete Resource’ option.



Pick the resource type to be deleted from the list of available resource types.



A popup Resource dialog box appears displaying all available XML resources for the resource type that was chosen. If it was a ‘Messenger Form’ resource type, the popup Resource dialog box should look something like the following screen snapshot.



There are 4 columns in the displayed table: ‘Name’, ‘User Id’, ‘Last Modified’ and ‘Version’. All resources are listed in this table in alphabetical order. Notice there is a ‘Delete’ field in the lower left-hand corner. This field is where the name of the resource that is to be deleted is typed.. Next, click the ‘Delete’ button or instead just type the enter key. The resource can also be deleted by using the mouse and double clicking a resource name in the table. Before the resource is actually deleted from the OpenFox™ Desktop system, Form Configurator will ask for confirmation to be sure “that you want to permanently delete the resource from the OpenFox™ Desktop System”.

Notice that as characters are typed in the 'Delete' field, the table automatically scrolls to display the row that best matches the character string currently entered in the 'Delete' field.

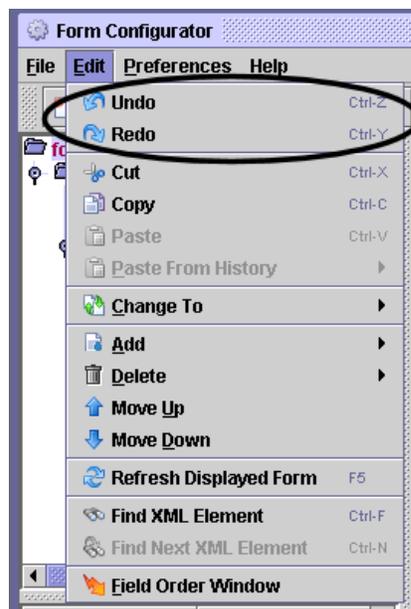
Also notice that there is a scrollbar on the right-hand side that can be used to scroll the table up or down. A count of the total number of resources of this type is always displayed at the top of the popup Resource dialog box.

2.11 Undo-Redo Functionality

A very useful Form Configurator feature is the 'Undo-Redo' functionality. This functionality allows changes to the resource you currently have open to be undone. This functionality works just like in Microsoft Word with the exception of one difference. It is possible to only undo up to and including the last 10 changes made to the current open resource. Likewise it is possible to redo up to and including the last 10 changes that were 'undone'. Every time a change is made to the current open resource, the state of the resource just before that change is saved in the 'Undo-Redo' history. Note that once you 'Save' or 'Save As' a resource, the 'Undo-Redo' history is cleared.

Every time an undo or redo is done, the resource is restored back/forward to a given snapshot of the resource, as stored in the 'Undo-Redo' history. Also the 'Form Display' screen-area is updated to show the new view of the resource.

To execute an undo or redo, click the 'Edit' option on the Form Configurator menu-bar and then navigating to the Undo or Redo options.



This can also be done by clicking on the Undo or Redo buttons on the Form Configurator toolbar.



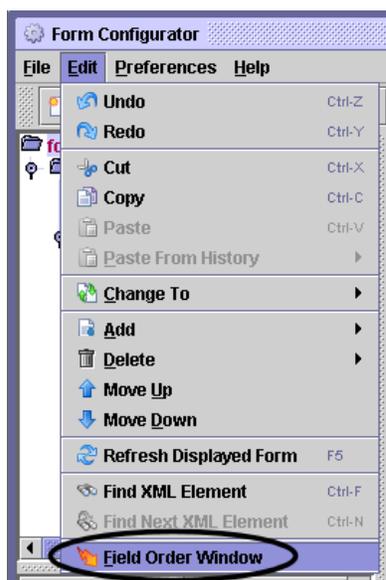
Or, just simply use the ‘Undo-Redo’ hot-keys and hit the Ctrl-Z key combination for Undo, or the Ctrl-Y key combination for Redo. Note that a warning message will popup if there is currently nothing to undo or redo.

2.12 Field Order Window

The ‘Field Order’ functionality is only available for ‘Messenger Forms’ resource types. The purpose of this functionality is to allow the fields to be presented in a certain order when displaying the form in OpenFox™ Messenger, but then to allow transmitting the fields in a different specified ‘Field Order’ when submitting the form to the Message Switch.

Note that it isn’t necessary to use the ‘Field Order Window’ to set the field orders. Instead, just manually edit the ‘index’ attribute in the ‘Attribute-Value Table’ for all appropriate elements. The ‘Field Order Window’ is merely a convenience feature that is designed to help manage a form’s ‘Field Order’ behavior by displaying all a form’s field order ‘index’ and ‘id’ attribute values in one table.

The ‘Field Order Window’ functionality can only be executed by clicking ‘Edit’ on the Form Configurator menu-bar and navigating to ‘Field Order Window’.



Clicking the 'Field Order Window' option will cause the 'Field Order Window' to be displayed. For the EW 'NCIC Enter Wanted Person' form, the 'Field Order Window' should look something like the following.

Index	Id
1	NAM
2	SEX
3	RAC
4	POB
5	DOB
6	DOE
7	HGT
8	WGT
9	EYE
10	HAI
11	FBI
12	SKN
13	SMT
14	FPC
15	MNU
16	SOC
17	DNA
18	DLO
19	CMC
20	OFF
21	OOC
22	DOW
23	OCA
24	WNO
25	CTI
26	NOA
27	EXL

Save
Clear All Cancel

Note that this is a fairly complex form and hence causes many records to be created in the 'Field Order Window'. Also notice there is one record for each individual element in the 'XML Tree' screen-area that has both an 'index' attribute and 'id' attribute that represent field order information.

There is a table with two columns in the 'Field Order Window': the 'Index' column and the 'Id' column. Only the 'Index' column is editable. The 'Index' column is populated by counting from the top of the form, as displayed in the 'XML Tree' screen-area, what position each element with field order information has in the form, and then using this 'position' value to populate the 'Index' cell for that element in the 'Field Order Window'. This logic is repeated for each element that is displayed in the 'Field Order Window'. This logic is why an element can currently have a different value for 'Index' in the 'XML Tree' screen-area, then the one currently displayed in the 'Field Order Window'. This is true only until hitting the 'Save' button in the 'Field Order Window', after which the values in the

'Index' column are applied to the 'index' attribute for all the pertinent elements. The 'Id' column is populated with the 'id' attribute value for that element.

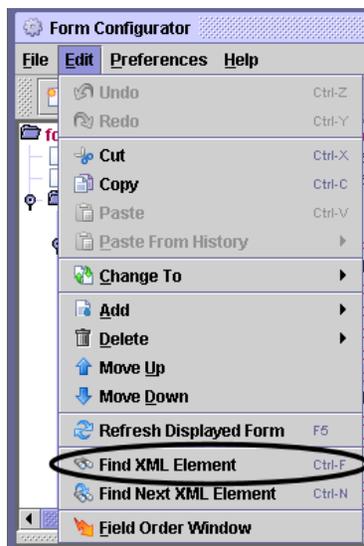
When creating new elements and adding them to a 'Messenger Form' resource, the default value for the 'index' attribute is -1. If the 'Clear All' button in the 'Field Order Window' is hit, it will set all 'index' values to -1 in the 'Field Order Window'. Hitting the 'Cancel' button in the 'Field Order Window' cancels the operation, exits the 'Field Order Window', returns Form Configurator to its previous state and does not change any values. Hitting the 'Save' button, of course, saves all values displayed in the 'Field Order Window'.

Note that when the 'Field Order Window' is displayed most other Form Configurator functionality is temporarily disabled. Also, when needed, scrollbars will appear on the right-hand side of the 'Field Order Window' to aid in scrolling up/down the table. Lastly note that 'Drag and Drop' functionality is available for moving 'Index' values in the 'Field Order Window' table. This functionality is of course controlled by the use of your mouse.

2.13 Find Window

The Form Configurator 'Find' functionality is useful when looking for a specific element, attribute or attribute value when working with a large resource.

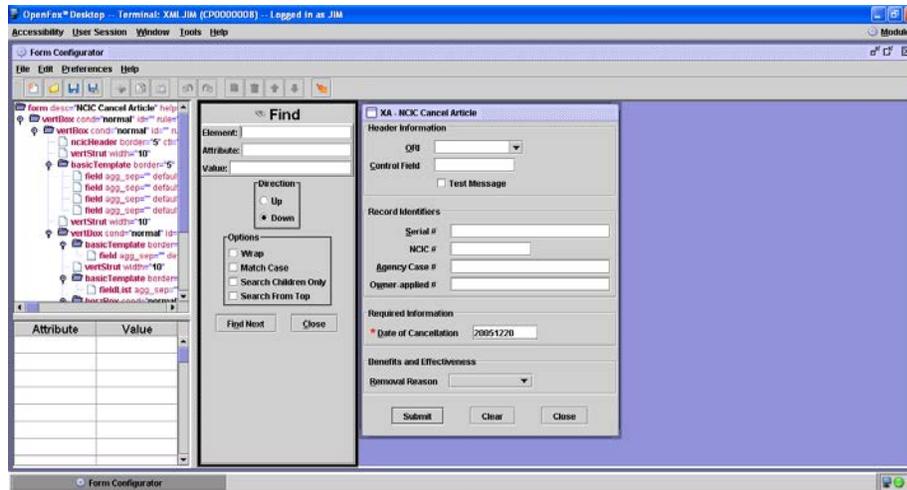
Bring up the 'Find Window' by clicking on the 'Edit' Form Configurator menu-bar option and navigating to the 'Find XML Element' option.



Or, use the hot-key combination of Ctrl-F to bring up the 'Find Window'. Note that after doing a find, the hot-key combination of Ctrl-N can be used to execute an

immediate ‘Find Next’ operation. Or, click on the ‘Edit’ Form Configurator menu-bar option and navigate to the ‘Find Next XML Element’ option.

The ‘Find Window’ pops up just to the right of both the ‘XML Tree’ screen-area and the ‘Attribute-Value Table’ screen-area. Once displayed the ‘Find Window’ stays on screen until clicking the ‘Cancel’ button in the ‘Find Window’, or until opening another resource in Form Configurator. When the ‘Find Window’ is displayed, the Form Configurator screen should look something like the following.



Notice that the ‘Form Display’ screen-area gets pushed off to the right.

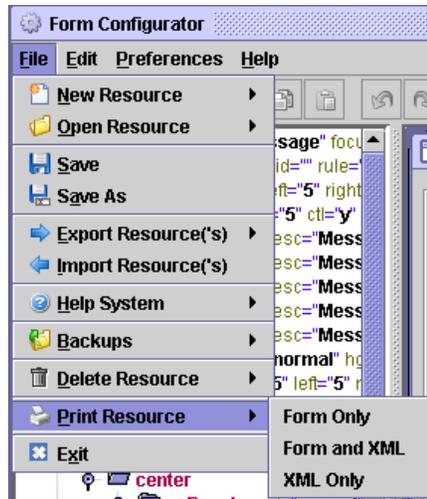
There are 3 primary fields for entering find criteria: Element, Attribute and Value. In most cases only one or more of these fields will be filled in with find criteria. Click the ‘Find Next’ button to execute the find operation. Find by default starts searching at the row just after the current selected row in the ‘XML Tree’ screen-area. If no row is selected, find starts its search from the first element in the ‘XML Tree’ screen-area. Find tries to find the element/row in the ‘XML Tree’ screen-area that matches the find criteria, and then selects that row. Note that this also causes the ‘Attribute-Value Table’ to be populated with the data from the selected element. If Find can not find a match for your search criteria, Find causes a popup message to be displayed indicating Find failed. Also it isn’t necessary to click the ‘Find Next’ button to execute a Find operation. If the cursor is in either the Element, Attribute or Value field, just hit the enter key or use the Alt-N hotkey for the ‘Find Next’ button.

The default direction that Find searches is down from the current selected row. The direction that Find searches can be changed by clicking on the ‘Up’ radio button. There are several Find options to choose from. If ‘Wrap’ is checked, Find will search to the end of the ‘XML Tree’ screen-area and wrap to the top (or bottom if the Find direction is ‘Up’) and continue the Find operation. In this way the current selected row in the ‘XML Tree’ screen-area will be the last row searched.

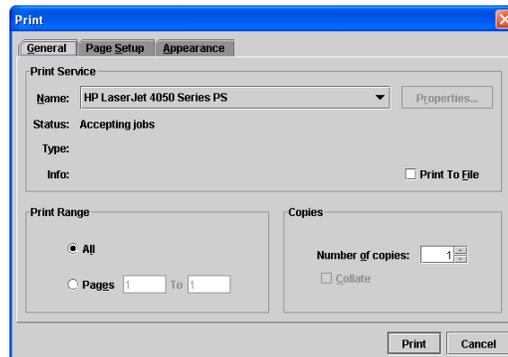
If 'Match Case' is checked, Find takes into account upper/lower case when Find searches for matches. When 'Search Children Only' is checked, Find only searches within all child nodes of the current parent. In this way, Find will search an unlimited number of levels deep of child elements, as long as all searched elements are ultimately under the parent element from where the search was started. When 'Search Children Only' is checked, the current selected row must be a parent element or Find causes a popup error message to be displayed. Lastly if 'Search from Top' is checked, Find always starts its search from the first or top element in the 'XML Tree' screen-area.

2.14 Printing

Form Configurator allows either or both the form that is displayed in the 'Form Display' screen-area and/or the underlying XML code as displayed in the 'XML Tree' screen-area to be printed. Note that when printing to a color printer, the print out will be in the colors as shown on the Form Configurator screen. To print, click on the 'File' option in the Form Configurator menu-bar and highlight the 'Print Resource' option. The screen should appear like the following:



After choosing one of the three options, 'Form Only', 'Form and XML' or 'XML', only the Print dialog box will appear.

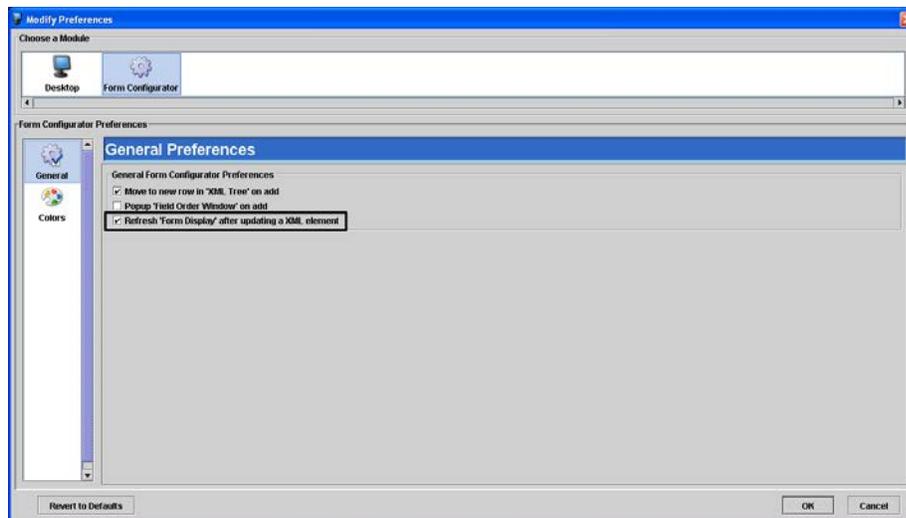


After filling out the Print dialog box, click on the ‘Print’ button to print, or the ‘Cancel’ button to cancel. It is important to pay attention to the primary option in the Print dialog box where you pick the ‘Name’ of the printer to which you want to print.

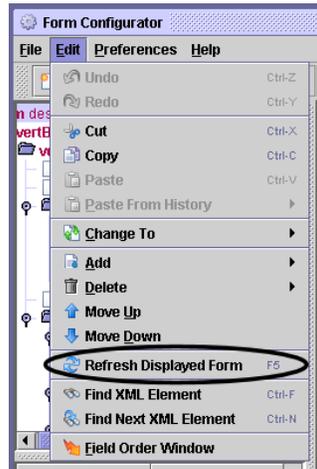
2.15 Menu-Item: Refresh Displayed Form

The “Refresh ‘Form Display’ after updating a XML element” preference option affects the behavior of the ‘Form Display’ screen-area. This topic is also covered in “Chapter 6: Preferences”.

By default, this option is turned on, however, when working on huge resources, it may at times be desirable to turn off the “Refresh ‘Form Display’ after updating a XML element” check box option. Please refer to “Chapter 6: Preferences” for detailed information about how to access this check box option.



This feature is best turned OFF for huge resources because it can cause the performance of the Form Configurator to slow down . You can speed up the performance of the Form Configurator by not updating the ‘Form Display’ screen-area after every change to the underlying XML code. In this way, many changes can be made quickly to the resource without any lag-time. To view the changes, hit the F5 hotkey to “Refresh the Displayed Form”. Note that this functionality can also be executed by clicking on the ‘Edit’ option on the Form Configurator menu-bar and navigating to the ‘Refresh Displayed Form’ option.



The “Refresh ‘Form Display’ after updating an XML element” check box option remains checked or uncheck even when opening another resource. If this box is uncheck when working on a huge resource, don’t forget to re-check-it before you open up another ‘regular sized’ resource.

* * * THIS PAGE LEFT INTENTIONALLY BLANK * * *

Chapter 3: Maintaining Messenger XML Resources: Forms, Code Lists, Scripts and Forms Menu

3.1 General Description

The Form Configurator behavior and the screen layout that is displayed is basically the same if you are editing a resource that is defined by one of the following ‘Messenger Definition Languages’:

- Messenger Form Language – for OpenFox™ Messenger forms resources
- Messenger Code Language – for OpenFox™ Messenger code-list resources
- Messenger Script Language – for OpenFox™ Messenger script resources
- Messenger Menu Language – for the OpenFox™ Messenger menu resource

Notice that the ‘Messenger Definition Language’ of a given resource you have open in Form Configurator dictates items like the following:

- The elements that can exist in this resource.
- The behavior of OpenFox™ Messenger when it processes the elements, attributes and attribute-values of a resource.
- The attributes and default attribute-values for elements.
- What elements can be added under a specific parent.
- How many child elements a specific type of parent element can have.
- The order of elements in the resource.

But in regards to how you use Form Configurator to edit resources of the types listed above, the Form Configurator behavior and displayed screen layout are the same. It is the responsibility of the Form Configurator user to have an understanding of the underlying ‘Messenger Definition Language’ of the resource being edited with Form Configurator. As mentioned in section “1.1 Organization of All Related Form Configurator Documentation“, details of the ‘Messenger Definition Languages’ are not discussed in this manual.

One major item of note is that for ‘Messenger Script Language’ resources nothing is displayed in the Form Configurator ‘Form Display’ screen-area.

*** THIS PAGE LEFT INTENTIONALLY BLANK ***

Chapter 4: Maintaining Messenger XML Resource: NCIC Vehicle Codes Lists

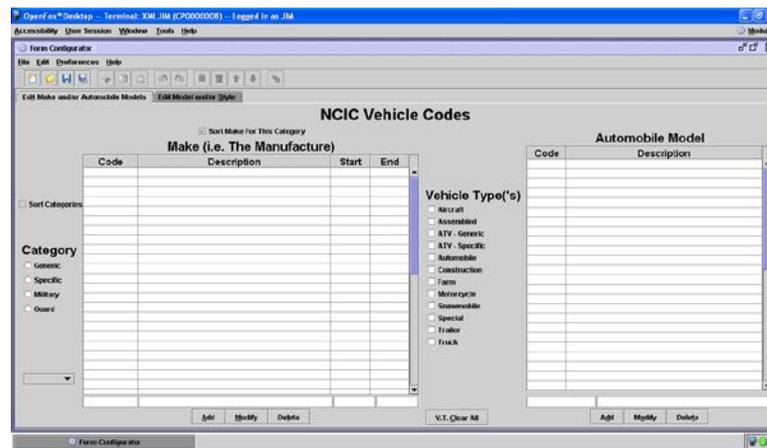
4.1 General Description

The screen layout that Form Configurator presents when editing ‘NCIC Vehicle Codes Lists’ resource type is completely different than the screen layout shared for all other resource types. This chapter discusses these different Form Configurator screen layouts and their behavior.

Note that all the functionality grouped under the ‘Edit’ option on the Form Configurator menu-bar is not available when editing the ‘NCIC Vehicle Codes Lists’, **except for the Undo and Redo functionality.**

4.2 Edit Make and/or Automobile Models [Tab]

When Form Configurator opens the ‘NCIC Vehicle Codes Lists’ resource type, the default screen layout shown is the ‘Edit Make and/or Automobile Models’ tab. This screen layout looks as follows.



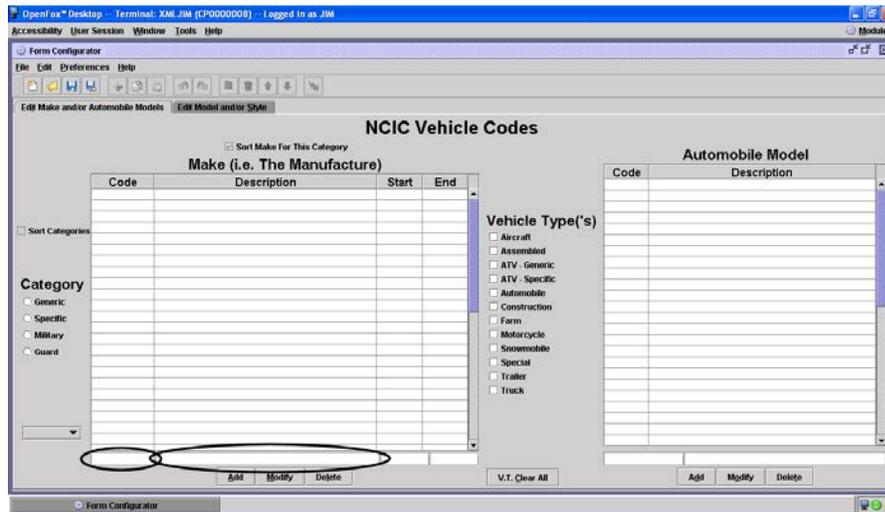
From a high level perspective there are 4 major screen areas in this tab that map to the underlying XML data relationship.

- Category
- Make
- Vehicle Type('s)
- Automobile Model

There is a logical left to right relationship between these different screen areas. What this means is that the screen must be filled out starting at the left-hand side. So the first action, for example, would be to pick a category. That action causes the ‘Make’ table to be populated. Next, pick a make, which causes the ‘Vehicle Type(s)’ check boxes to be populated. And if the ‘Vehicle Type(s)’ Automobile check box is checked, then the ‘Automobile Model’ table will be populated.

4.2.1 Example Walk-Through: Adding a Make Record

The following steps describe how to add a make record in the ‘Edit Make and/or Automobile Models’ tab screen layout. It is necessary to first refer to your NCIC documentation to gather information about the make record that is to be added. From this documentation, determine the Category of the make record and click on that Category. Next, in the ‘Make Code’ field and ‘Make Description’ field, enter the make’s code and description.



Be careful when entering the ‘Make Code’, as this is case sensitive data (although for most cases only upper cased alphanumeric characters can be entered anyway). Enter the make ‘Start’ or ‘End’ values. ‘Start’ and ‘End’ values are only used for Category ‘Generic’ make records. These values tell OpenFox™ Messenger the range of character positions which to highlight when displaying this editable make record in a form. If there is no Start or End value the make record is not highlighted when selected in an OpenFox™ Messenger form.

Now, check and uncheck the appropriate ‘Vehicle Type(s)’ check boxes so that all the Vehicle Type(s) this manufacture makes are properly described.

Vehicle Type('s)

- Aircraft
- Assembled
- ATV - Generic
- ATV - Specific
- Automobile
- Construction
- Farm
- Motorcycle
- Snowmobile
- Special
- Trailer
- Truck

V.T. Clear All

Clicking on the ‘Add’ button located under the ‘Make Description’ field (the ‘Add’ button that’s below the ‘Make’ table) will validate the make record. If an error occurs, a popup detailed error message is displayed. If there are no errors, the make record is added to the above ‘Make’ table and the new row in the table that contains the new make record is selected.

Note that the Undo and Redo functionality is available and can be used to undo all add, modify and delete actions. The undo/redo functionality can be accessed by using the Ctrl-Z or Ctrl-Y hotkeys, clicking on the ‘Undo’ or ‘Redo’ buttons on the Form Configurator tool bar, or by clicking on the ‘Edit’ option on the Form Configurator menu-bar and navigating to ‘Undo’ or ‘Redo’.

4.2.2 Category [Radio Buttons] and [Combo Box]

A category may be chosen by clicking on the appropriate ‘Category’ radio button.

Category

- Generic
- Specific
- Military
- Guard

A category may also be chosen by clicking on the ‘Category’ combo box and choosing from the categories listed.



Notice that there is a blank category value among the choices in the ‘Category’ combo box. Picking this value will clear the ‘Edit Make and/or Automobile Models’ tab screen-area.

4.2.3 Make [Table]

The ‘Make’ table is a non-editable table only used for viewing make data. The table contains four columns: Code, Description, Start and End.

Make (i.e. The Manufacture)			
Code	Description	Start	End
AAA	AAA Mobile Home Mfg. Co.		
AAB	A.A.B. Co., Inc.		
AALI	Aalite Co.		
AAPX	AAPEX Trailers, Inc.		
AARD	Aardvark Co.		
AATM	All-American Trailer Mfg. Co.		
AATS	AAA Trailer Sales		
AAWL	AA Welding Service, Clint R. Woods, Inc.		
AAWS	A-A Welding Service		
ABAR	Abarth		
ABBO	Paul Abbot Co., Inc.		
ABC	ABC (Aluminum Body Corp.)		
ABCC	ABC Coach Co.		
ABCH	ABC Homes		
ABCO	ABCO Trailer Mfg. Co.		
ABER	Aberdeen Boat Trailer		
ABFI	ABF Industries		
ABIA	ABI Leisure, Inc.		
ABTM	Amco Boat Trailer		
ABTR	A & B Trailer Mfg. Co., Inc.		
AC	A C		
ACAD	Acadian		
ACAE	Academy Mobile Home Corp.		
ACAL	Antique & Collectible Autos, Inc.		

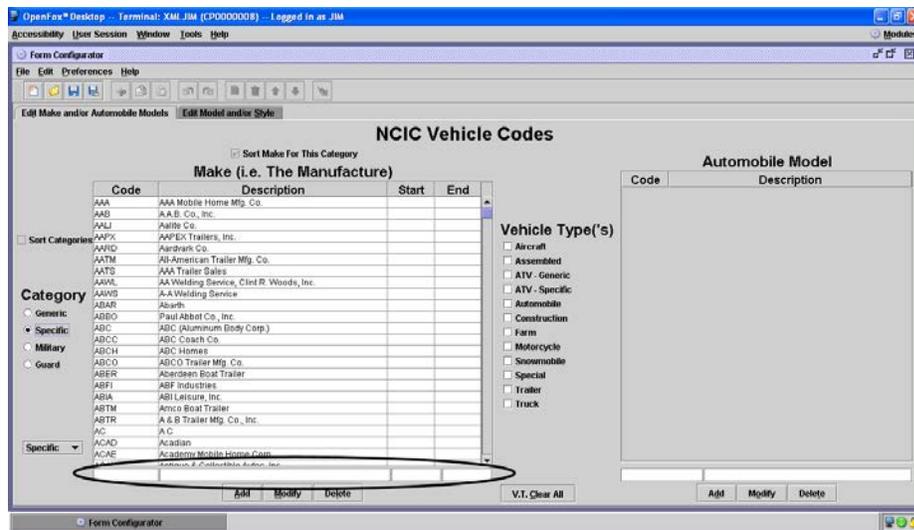
The data in the table is sorted alphabetically by the ‘Code’ column. Each value of code must be unique, thus, there can be no duplicate code values. The ‘Code’ column can only contain alphanumeric characters and depending on the current value for category, either lower and upper case characters, or just upper case characters are valid. Furthermore, depending on the category, the code data has a different maximum size.

Use the ‘Make’ fields and ‘Make’ buttons located below the table to add, modify or delete a make record. When a row from the ‘Make’ table is selected, the ‘Make’ fields are automatically filled in with the selected make records data. Furthermore, upon editing the ‘Make Code’ field, the ‘Make’ table will automatically try to scroll to select the make record that best matches that ‘Make Code’ field’s value.

The ‘Make’ table has scrollbars on the right-hand side to assist in scrolling up and down the table. Also, page-up, page-down, up-arrow and down-arrow are all valid keys used to aid in navigating around the ‘Make’ table.

4.2.4 Make [Fields] and [Buttons]

The ‘Make’ fields and ‘Make’ buttons are used together to add, modify and delete make records. There are 4 unlabeled ‘Make’ fields: ‘Make Code’, ‘Make Description’, ‘Make Start’ and ‘Make End’.



When selecting a row in the ‘Make’ table, the ‘Make’ fields are filled in with the data from that make record. The ‘Make Code’ field is especially tied with the ‘Make’ table. Whenever characters are entered in the ‘Make Code’ field, the ‘Make’ table will try to scroll to select the make record that best matches that ‘Make Code’ field’s value.

The ‘Make Code’ field can only contain alphanumeric characters. Depending on the current value for category, either lower and upper cased characters, or just upper cased characters are valid. Furthermore, depending on the category, the code data has a different maximum size.

The 'Make' fields are tied together with the functionality of the 'Make' 'Add', 'Modify' and 'Delete' buttons which are located just below the 'Make' fields. When clicking on the 'Add' make button (or using the Alt-A hotkey), the data in the four 'Make' fields is used to create the new make record. If successful, the new make record is added alphabetically into the 'Make' table, and the new make record is also selected in the 'Make' table. In order to successfully add a make record, the make code value must be unique and therefore must not already exist in the 'Make' table. Also, a category must be chosen for the add operation to work. Lastly don't forget to make sure the 'Vehicle Type' check boxes are properly checked and unchecked to indicate what vehicle types this manufacture makes. If not the resulting make record will have no associated vehicle type. Note that a make (manufacture) can be associated with one or more vehicle types.

If the 'Modify' make button (or the Alt-M hotkey) is clicked, the data in the four 'Make' fields is used to modify an existing make record. Therefore the value in the 'Make Code' field must match an existing make code in the 'Make' table, or the modify operation will fail.

If the 'Delete' make button (or the Alt-L hotkey) is clicked, the data in the 'Make Code' field is used to identify and delete a make record. Therefore the make code value must exactly match a record in the 'Make' table. When a make record is deleted, the make record is removed from the 'Make' table and the 'Make' fields are cleared.

Note that the Undo and Redo functionality is available and can be used to undo all add, modify and delete actions. The undo/redo functionality can be accessed using the Ctrl-Z or Ctrl-Y hotkeys, clicking on the 'Undo' or 'Redo' buttons on the Form Configurator tool bar, or by clicking on the 'Edit' option on the Form Configurator menu-bar and navigating to 'Undo' or 'Redo'.

4.2.5 Vehicle Type('s) [Check Boxes] and V.T. Clear All [Button]

The 'Vehicle Type('s)' check boxes are located in between the 'Make' table and the 'Automobile Model' table.

Vehicle Type(s)

- Aircraft
- Assembled
- ATV - Generic
- ATV - Specific
- Automobile
- Construction
- Farm
- Motorcycle
- Snowmobile
- Special
- Trailer
- Truck

V.T. Clear All

The 'V.T. Clear All' button is associated with the 'Vehicle Type' check boxes. If the 'V.T. Clear All' button is clicked, all the 'Vehicle Type' check boxes are unchecked or cleared.

The 'Vehicle Type' check boxes are associated with the 'Make' table. Whenever a row in the 'Make' table is selected, the 'Vehicle Type' check boxes are checked and unchecked to reflect the vehicle types associated with that make record. If no row is selected in the 'Make' table, then no 'Vehicle Type' check boxes are checked.

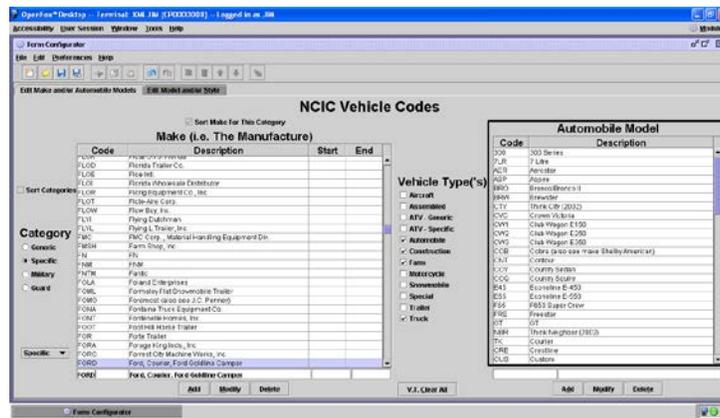
Whenever a make record is added or modified, the current checked 'Vehicle Type' check boxes are captured and stored with that make record. And henceforth used to indicate which vehicle type('s) that make (manufacture) makes.

One interesting point is that in order to display or add any records in the 'Automobile Model' table, the vehicle type 'Automobile' check box must be checked for the current selected make record in the 'Make' table.

4.2.6 Example Walk-Through: Adding an Automobile Model Record

The following steps describe how to add an automobile model record in the ‘Edit Make and/or Automobile Models’ tab screen-area for a specific make record. First, please refer to the specific NCIC documentation to gather information about the automobile models for the make record in question. After having done so, select the make record in the ‘Edit Make and/or Automobile Models’ tab screen-area. The first step is to pick the category for the desired make record. Next select the make record from the ‘Make’ table. At this point the ‘Automobile’ check box in the vehicle type(‘s) screen-area should be checked. If it is not checked, it will not be possible to add an automobile model record. Note that other vehicle type check boxes can also be checked as they will have no effect on this operation.

Now that the make record is selected with the ‘Automobile’ vehicle type check box checked, the ‘Automobile Model’ table may or may not be populated with data. For example, for the ‘FORD’ make record, the ‘Automobile Model’ table should look something like the following.



Now, enter the automobile model record data in the ‘Automobile Model Code’ and ‘Automobile Model Description’ fields.

Automobile Model	
Code	Description
300	300 Series
7LR	7 Litre
AER	Aerostar
ASP	Aspire
BRO	Bronco/Bronco II
BRW	Brewster
CTY	Think City (2002)
CVC	Crown Victoria
CW1	Club Wagon E150
CW2	Club Wagon E250
CW3	Club Wagon E350
COB	Cobra (also see make Shelby American)
CNT	Contour
COY	Country Sedan
COQ	Country Squire
E45	Econoline E-450
E55	Econoline E-550
F65	F650 Super Crew
FRE	Freestar
GT	GT
NBR	Think Neighbor (2002)
TK	Courier
CRE	Crestline
CUS	Custom

<input type="text"/>	<input type="text"/>
----------------------	----------------------

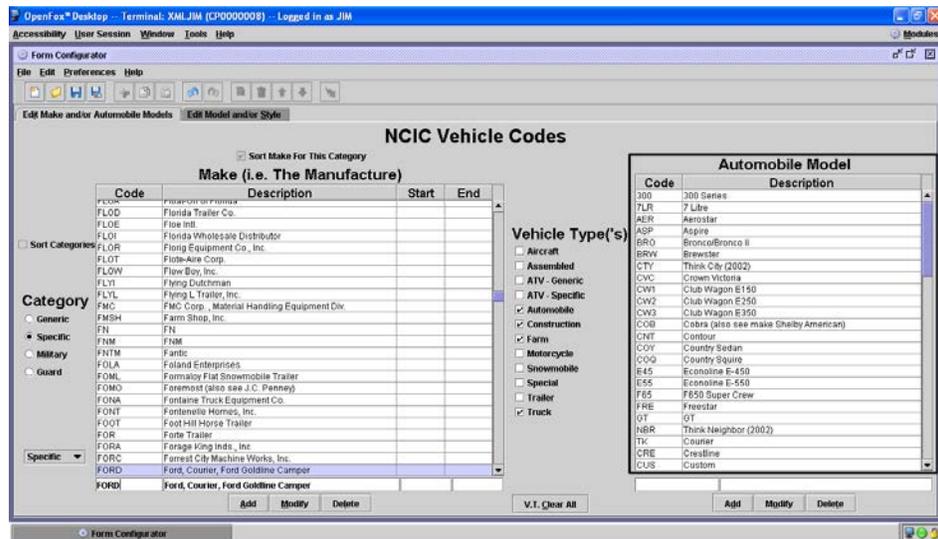
The automobile model code is limited to 3 characters maximum length and must consist of upper cased characters and digits. The automobile model description is a free formatted character field where all printable characters are valid.

After entering the automobile model data in the 'Automobile Model Code' and 'Automobile Model Description' fields, click on the Automobile Model 'Add' button (or use the Alt-D hotkey) to add the automobile model record. At this point Form Configurator validates the automobile model record. If an error occurs, a popup detailed error message is displayed. If there are no errors, the automobile model record is added to the above 'Automobile Model' table, and the new row in the table that contains the new automobile model record is selected.

Note that the Undo and Redo functionality is available, and can be used to undo all add, modify and delete actions. The undo/redo functionality can be accessed using the Ctrl-Z or Ctrl-Y hotkeys, clicking the 'Undo' or 'Redo' Buttons on the Form Configurator tool bar, or by clicking on the 'Edit' option on the Form Configurator menu-bar and navigating to 'Undo' or 'Redo'.

4.2.7 Automobile Model [Table]

The ‘Automobile Model’ table is a non-editable table only used for viewing automobile model data. The table contains two columns: Code and Description.



The data in the table is sorted alphabetically by the ‘Code’ column. Each value of code must be unique. That is there can be no duplicate code values. The ‘Code’ column can contain up to 3 characters maximum length and must consist of upper cased characters and digits.

When wanting to add, modify or delete an automobile model record, the ‘Automobile Model’ fields and ‘Automobile Model’ buttons located below the table must be used. Upon selecting a row from the ‘Automobile Model’ table, the ‘Automobile Model’ fields are automatically filled in with the selected automobile model records data. Furthermore, when editing the ‘Automobile Model Code’ field, the ‘Automobile Model’ table will automatically try to scroll to select the automobile model record that best matches that ‘Automobile Model Code’ field’s value.

The ‘Automobile Model’ table has scrollbars on the right-hand side to assist in scrolling up and down the table. Also, page-up, page-down, up-arrow and down-arrow are all valid keys used to aid in navigating around the ‘Automobile Model’ table.

4.2.8 Automobile Model [Fields] and [Buttons]

The ‘Automobile Model’ fields and ‘Automobile Model’ buttons are used together to add, modify and delete automobile model records. There are 2 unlabeled ‘Automobile Model’ fields: ‘Automobile Model Code’ and ‘Automobile Model Description’.

Automobile Model	
Code	Description
300	300 Series
7LR	7 Litre
AER	Aerostar
ASP	Aspire
BRO	Bronco/Bronco II
BRW	Brewster
CTY	Think City (2002)
CVC	Crown Victoria
CW1	Club Wagon E150
CW2	Club Wagon E250
CW3	Club Wagon E350
COB	Cobra (also see make Shelby American)
CNT	Contour
COY	Country Sedan
COQ	Country Squire
E45	Econoline E-450
E55	Econoline E-550
F65	F650 Super Crew
FRE	Freestar
GT	GT
NBR	Think Neighbor (2002)
TK	Courier
CRE	Crestline
CUS	Custom

When selecting a row in the ‘Automobile Model’ table, the ‘Automobile Model’ fields are filled in with the data from that automobile model record. The ‘Automobile Model Code’ field is especially tied with the ‘Automobile Model’ table. Whenever characters are entered in the ‘Automobile Model Code’ field, the ‘Automobile Model’ table will try to scroll to select the automobile model record that best matches that ‘Automobile Model Code’ field’s value. The ‘Make Code’ field can contain up to 3 characters maximum length and must consist of upper cased characters and digits.

The ‘Automobile Model’ fields are tied together with the functionality of the ‘Automobile Model’ ‘Add’, ‘Modify’ and ‘Delete’ buttons which are located just under the ‘Automobile Model’ fields. When the ‘Add’ automobile model button (or the Alt-D hotkey) is clicked, the data in the two ‘Automobile Model’ fields is used to create the new automobile model record. If successful, the new automobile model record is added alphabetically into the ‘Automobile Model’ table, and the new automobile model record is also selected in the ‘Automobile Model’ table. In order to successfully add an automobile model record, the automobile model code value must be unique and therefore can not already exist in the ‘Automobile Model’ table. Also, a make record from the ‘Make’ table that has the ‘Automobile’ ‘Vehicle Type’ check box checked must already have been selected.

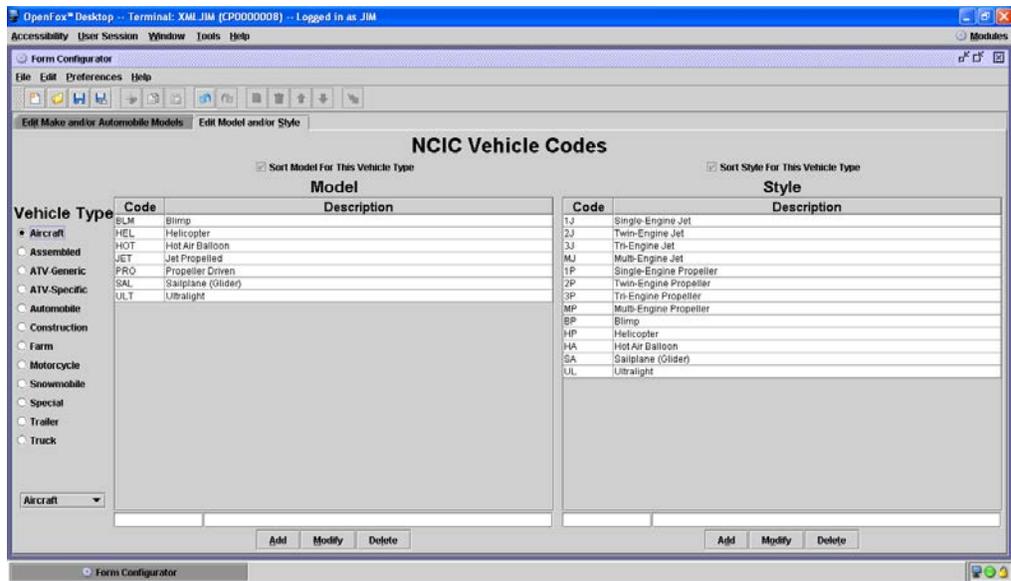
After clicking on the ‘Modify’ automobile model button (or using the Alt-O hotkey), the data in the two ‘Automobile Model’ fields is used to modify an existing automobile model record. Therefore the value in the ‘Automobile Model Code’ field must match an existing automobile model code in the ‘Automobile Model’ table or the modify operation will fail.

After clicking on the ‘Delete’ automobile model button (or using the Alt-T hotkey), the data in the ‘Automobile Model Code’ field is used to identify and delete an automobile model record. Therefore the automobile model code value must match a record in the ‘Automobile Model’ table. When an automobile model record is deleted, the automobile model record is removed from the ‘Automobile Model’ table and the ‘Automobile Model’ fields are cleared.

Note that the Undo and Redo functionality is available and can be used to undo all add, modify and delete actions. The undo/redo functionality can be accessed using the Ctrl-Z or Ctrl-Y hotkeys, clicking on the ‘Undo’ or ‘Redo’ buttons on the Form Configurator tool bar, or by clicking on the ‘Edit’ option on the Form Configurator menu-bar and navigating to ‘Undo’ or ‘Redo’.

4.3 ‘Edit Model and/or Style’ [Tab]

This functionality can be opened by clicking on the ‘Edit Model and/or Style’ tab after opening the ‘NCIC Vehicle Codes Lists’ resource type. When the ‘Aircraft’ vehicle type check box is checked, the screen layout should look something like the following.



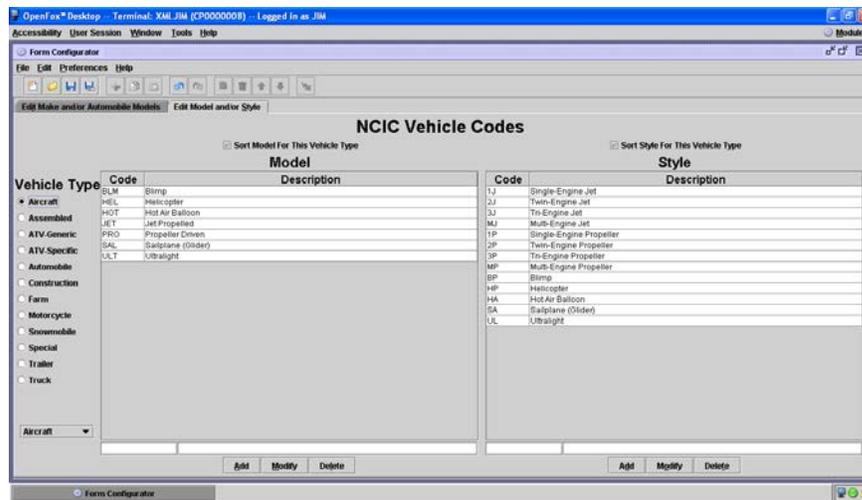
From a high level perspective there are 3 major screen areas in this tab that map to the underlying XML data relationship:

- Vehicle Type
- Model
- Style

Picking a vehicle type value causes both the ‘Model’ table and ‘Style’ table to be populated.

4.3.1 Example Walk-Through: Adding a Model Record

The following steps describe how to add a model record in the ‘Edit Model and/or Style’ tab screen layout. It is necessary to first refer to the specific NCIC documentation to gather information about the model record to be added. From this documentation, determine the vehicle type of the model record and click on that vehicle type. Now enter the model’s code and description in the ‘Model Code’ field and ‘Model Description’ fields.



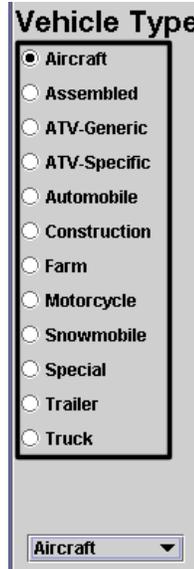
The model code can only contain up to 3 characters and consists of upper cased characters and digits. The ‘Model Description’ field can contain any printable characters.

Next, click on the ‘Add’ button located under the ‘Model Description’ field (the ‘Add’ button that’s below the ‘Model’ table). At this point, Form Configurator validates the model record. Note that the ‘Model Code’ must be unique and therefore must not already exist in the ‘Model’ table. If an error occurs, a popup detailed error message is displayed. If there aren’t any errors, the model record is added to the above ‘Model’ table, and the row in the table that contains the new model record is selected.

Note that the Undo and Redo functionalities are available and can be used to undo all add, modify and delete actions. Access the undo/redo functionality using the Ctrl-Z or Ctrl-Y hotkeys, click on the 'Undo' or 'Redo' buttons on the Form Configurator tool bar, or click on the 'Edit' option on the Form Configurator menu-bar and navigate to 'Undo' or 'Redo'.

4.3.2 Vehicle Type [Radio Buttons] and [Combo Box]

Choose a vehicle type by clicking on the appropriate 'Vehicle Type' radio button:



Or choose a vehicle type by clicking on the 'Vehicle Type' combo box and choosing from the vehicle types listed.



Notice that there is a blank vehicle type value in the ‘Vehicle Type’ combo box. Pick this value to clear the ‘Edit Model and/or Style’ tab screen-area, if desired.

4.3.3 Model [Table]

The ‘Make’ table is a none-editable table only used for viewing model data. The table contains two columns: Code and Description. The following ‘Model’ table is shown populated with vehicle type ‘Aircraft’ data.

Model	
Code	Description
BLM	Blimp
HEL	Helicopter
HOT	Hot Air Balloon
JET	Jet Propelled
PRO	Propeller Driven
SAL	Sailplane (Glider)
ULT	Ultraight

The data in the table is sorted alphabetically by the ‘Code’ column. Each value of code must be unique. That is there can be no duplicate code values. The data in the ‘Code’ column can only be up to 3 characters in length and can only contain upper cased characters and digits.

To add, modify or delete a model record, use the ‘Model’ fields and ‘Model’ buttons located below the table. When selecting a row from the ‘Model’ table, the ‘Model’ fields are automatically filled in with the selected model records data. Furthermore, when editing the ‘Model Code’ field, the ‘Model’ table will automatically try to scroll to select the model record that best matches that ‘Model Code’ field’s value.

The ‘Model’ table has scrollbars on the right-hand side to assist in scrolling up and down the table. Also, page-up, page-down, up-arrow and down-arrow are all valid keys used to aid in navigating around the ‘Model’ table.

4.3.4 Model [Fields] and [Buttons]

The 2 ‘Model’ fields and 3 ‘Model’ buttons are used together to add, modify and delete model records. There are 2 unlabeled ‘Model’ fields: ‘Model Code’ and ‘Model Description’ located below the ‘Model’ table.

Model	
Code	Description
BLM	Blimp
HEL	Helicopter
HOT	Hot Air Balloon
JET	Jet Propelled
PRO	Propeller Driven
SAL	Sailplane (Glider)
ULT	Ultralight

Below the table are two empty input fields for 'Model Code' and 'Model Description', and three buttons: 'Add', 'Modify', and 'Delete'.

When selecting a row in the ‘Model’ table, the ‘Model’ fields are filled in with the data from that model record. The ‘Model Code’ field is especially tied with the ‘Model’ table. Whenever characters are entered in the ‘Model Code’ field, the ‘Model’ table will try to scroll to select the model record that best matches the current ‘Model Code’ field’s value. The ‘Model Code’ field can contain up to 3 characters in length and can only consist of upper cased characters and digits.

The ‘Model’ fields are tied together with the functionality of the ‘Model’ ‘Add’, ‘Modify’ and ‘Delete’ buttons which are located just below the ‘Model’ fields. When clicking on the ‘Add’ model button (or using the Alt-A hotkey), the data in the two ‘Model’ fields is used to create the new model record. If successful, the new model record is added alphabetically into the ‘Model’ table, and the new model record is also selected in the ‘Model’ table. In order to successfully add a model record, the model code value must be unique and therefore must not already exist in the ‘Model’ table.

When you clicking on the ‘Modify’ model button (or using the Alt-M hotkey), the data in the two ‘Model’ fields is used to modify an existing model record. Therefore the value in the ‘Model Code’ field must match an existing model code in the ‘Model’ table, or the modify operation will fail.

When clicking on the ‘Delete’ model button (or using the Alt-L hotkey), the data in the ‘Model Code’ field is used to identify and delete a model record. Therefore the model code value must match a record in the ‘Model’ table. When a model record is deleted, the model record is removed from the ‘Model’ table and the ‘Model’ fields are cleared.

Note that the Undo and Redo functionality is available and can be used to undo all add, modify and delete actions. Access the undo/redo functionality using the Ctrl-Z or Ctrl-Y hotkeys, clicking on the ‘Undo’ or ‘Redo’ buttons on the Form Configurator tool bar, or by clicking on the ‘Edit’ option on the Form Configurator menu-bar and navigating to ‘Undo’ or ‘Redo’.

4.3.5 Example Walk-Through: Adding a Style Record

The following steps describe how to add a style record in the ‘Edit Model and/or Style’ tab screen layout. It is necessary to first refer to the specific NCIC documentation to gather information about the style record to be added. From this documentation determine the vehicle type of the style record and click on that vehicle type. Next, enter the style’s code and description in the ‘Style Code’ field and ‘Style Description’ fields, located just below the ‘Style’ table,

Style	
Code	Description
1J	Single-Engine Jet
2J	Twin-Engine Jet
3J	Tri-Engine Jet
MJ	Multi-Engine Jet
1P	Single-Engine Propeller
2P	Twin-Engine Propeller
3P	Tri-Engine Propeller
MP	Multi-Engine Propeller
BP	Blimp
HP	Helicopter
HA	Hot Air Balloon
SA	Sailplane (Glider)
UL	Ultralight

--	--

The style code can only contain up to 2 characters and consists of upper cased characters and digits. The ‘Style Description’ field can contain any printable characters.

Now click on the ‘Add’ button located under the ‘Style Description’ field (the ‘Add’ button that’s below the ‘Style’ table). At this point, Form Configurator validates the style record. If an error occurs, a popup detailed error message is displayed. If there are no errors, the style record

is added to the above ‘Style’ table, and the row in the table that contains the new style record is selected.

Note that the Undo and Redo functionality is available and can be used to undo all add, modify and delete actions. The undo/redo functionality can be accessed using the Ctrl-Z or Ctrl-Y hotkeys, clicking on the ‘Undo’ or ‘Redo’ buttons on the Form Configurator tool bar, or by clicking on the ‘Edit’ option on the Form Configurator menu-bar and navigating to ‘Undo’ or ‘Redo’.

4.3.6 Style [Table]

The ‘Style’ table is a non-editable table only used for viewing style data. The table contains two columns: Code and Description. The following ‘Style’ table is shown populated with vehicle type ‘Aircraft’ data.

Style	
Code	Description
1J	Single-Engine Jet
2J	Twin-Engine Jet
3J	Tri-Engine Jet
MJ	Multi-Engine Jet
1P	Single-Engine Propeller
2P	Twin-Engine Propeller
3P	Tri-Engine Propeller
MP	Multi-Engine Propeller
BP	Blimp
HP	Helicopter
HA	Hot Air Balloon
SA	Sailplane (Glider)
UL	Ultralight

The data in the table is sorted alphabetically by the ‘Code’ column. Each value of code must be unique. That is there can be no duplicate code values. The data in the ‘Code’ column can only be up to 2 characters in length and can only contain upper cased characters and digits.

To add, modify or delete a style record, use the ‘Style’ fields and ‘Style’ buttons located below the table.

Style	
Code	Description
1J	Single-Engine Jet
2J	Twin-Engine Jet
3J	Tri-Engine Jet
MJ	Multi-Engine Jet
1P	Single-Engine Propeller
2P	Twin-Engine Propeller
3P	Tri-Engine Propeller
MP	Multi-Engine Propeller
BP	Blimp
HP	Helicopter
HA	Hot Air Balloon
SA	Sailplane (Glider)
UL	Ultralight

When selecting a row from the ‘Style’ table, the ‘Style’ fields are automatically filled in with the selected style records data. Furthermore, when editing the ‘Style Code’ field, the ‘Style’ table will automatically try to scroll to select the style record that best matches that ‘Style Code’ field’s value.

The ‘Style’ table has scrollbars on the right-hand side to assist in scrolling up and down the table. Also, page-up, page-down, up-arrow and down-arrow are all valid keys used to aid in navigating around the Style table.

4.3.7 Style [Fields] and [Buttons]

The 2 ‘Style’ fields and 3 ‘Style’ buttons are used together to add, modify and delete style records. There are 2 unlabeled ‘Style’ fields: ‘Style Code’ and ‘Style Description’ located below the ‘Style’ table.

Style	
Code	Description
1J	Single-Engine Jet
2J	Twin-Engine Jet
3J	Tri-Engine Jet
MJ	Multi-Engine Jet
1P	Single-Engine Propeller
2P	Twin-Engine Propeller
3P	Tri-Engine Propeller
MP	Multi-Engine Propeller
BP	Blimp
HP	Helicopter
HA	Hot Air Balloon
SA	Sailplane (Glider)
UL	Ultralight

--	--

When selecting a row in the ‘Style’ table, the ‘Style’ fields are filled in with the data from that style record. The ‘Style Code’ field is especially tied with the ‘Style’ table. Whenever characters are entered in the ‘Style Code’ field, the ‘Style’ table will try to scroll to select the style record that best matches the current ‘Style Code’ field’s value. The ‘Style Code’ field can contain up to 2 characters in length and can only consist of upper cased characters and digits.

The ‘Style’ fields are tied together with the functionality of the ‘Style’ ‘Add’, ‘Modify’ and ‘Delete’ buttons which are located just below the ‘Style’ fields. When clicking on the ‘Add’ style button (or using the Alt-D hotkey), the data in the two ‘Style’ fields is used to create the new style record. If successful the new style record is added alphabetically into the ‘Style’ table, and the new style record is also selected in the ‘Style’ table.

In order to successfully add a style record, the style code value must be unique and therefore must not already exist in the 'Style' table.

When clicking on the 'Modify' model button (or using the Alt-O hotkey), the data in the two 'Style' fields is used to modify an existing style record. Therefore the value in the 'Style Code' field must match an existing style code in the 'Style' table, or the modify operation will fail.

When clicking on the 'Delete' style button (or using the Alt-T hotkey), the data in the 'Style Code' field is used to identify and delete a style record. Therefore the style code value must match a record in the 'Style' table. When a model record is deleted, the style record is removed from the 'Style' table and the 'Style' fields are cleared.

Note that the Undo and Redo functionalities are available and can be used to undo all add, modify and delete actions. The undo/redo functionalities can be accessed using the Ctrl-Z or Ctrl-Y hotkeys, clicking on the 'Undo' or 'Redo' buttons on the Form Configurator tool bar, or by clicking on the 'Edit' option on the Form Configurator menu-bar and navigating to 'Undo' or 'Redo'.

Notice that nothing is displayed in the ‘Form Display’ screen-area for these resource types. Also, there are two additional fields for these resource types, ‘Element’ and ‘Text’, which are displayed on the left-hand side of the screen.

5.2 Changes to Behavior of ‘XML Tree’ [Screen-Area]

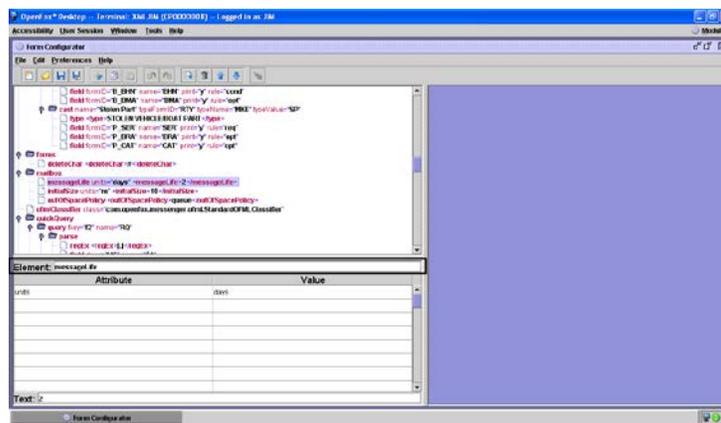
There are two differences to the behavior of the ‘XML Tree’ screen-area for these resource types. First, whenever a new element is added, an ‘Add XML Child Element’ window appears on the right-hand side of the screen. This topic is covered in section “5.6 Add XML Child Element’ Window”. The second difference deals with the display of elements with text data. The following screen snap-shot shows how an element with both attributes and text data is displayed in the ‘XML Tree’ screen-area.

```
messageLife units="days" <messageLife>2</messageLife>
```

The name of the element is ‘messageLife’. The element has one attribute: ‘units’ and also has text data associated with it: ‘2’ in this case. Notice that the text data is wrapped with a start XML tag and an end XML tag. Text data is always shown in the ‘XML Tree’ screen-area wrapped within a start XML tag and end XML tag, and is always displayed after all attributes are displayed. The start XML tag and end XML tag names are always derived from the name of the element. In this case the start tag is ‘<messageLife>’ and the end tag is ‘</messageLife>’. Only the resource types listed in section 5.1 of this chapter can have text data. Note that if an element has text data, it can not be a parent element and hence can not contain any child elements under it.

5.3 Element Field [Screen-area]

The ‘Element’ field is displayed on the left-hand side of the screen in between the ‘XML Tree’ screen-area and the ‘Attribute-Value Table’ screen-area.



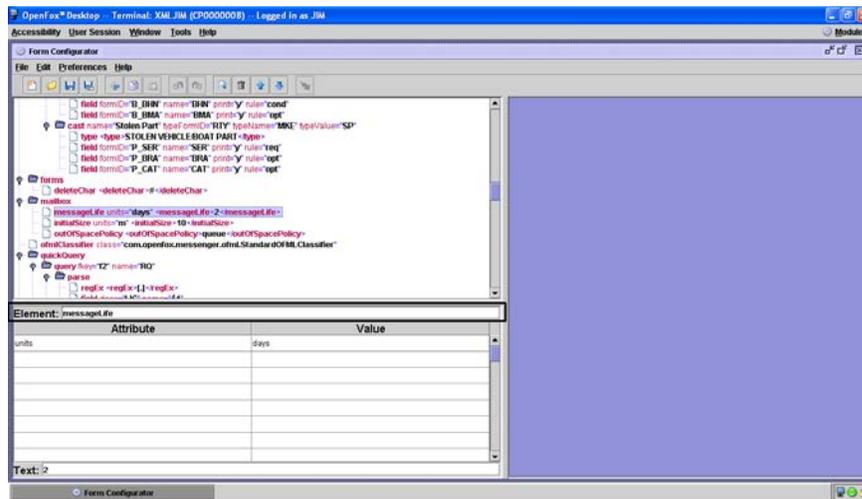
This field contains the name of the current selected XML element in the ‘XML Tree’ screen-area. The value in the ‘Element’ field can be edited using any combination of lower and upper cased alphanumeric characters.

5.4 Change to Behavior of ‘XML Attribute-Value Table’ [Screen-area]

There are three changes to the ‘XML Attribute-Value Table’ screen-area when handling the resource types listed in section 5.1 of this chapter. First, the ‘Attribute’ column is now editable. Secondly, all cells in the table are now treated strictly as editable text data. There are no drop-down-list cells, number cells, check box cells or constant cells. Lastly, new attribute-value pairs of data can be entered by just entering the data in a blank row. Note that it is an error to try and add data in a ‘Value’ column cell without first adding data in the associated ‘Attribute’ column cell on that row. If attempted, the data will be automatically ignored and hence removed. The logical reasoning behind this is that there can not be an attribute-value without having an existing attribute with which to associate the attribute-value.

5.5 Text Field [Screen-area]

The ‘Text’ field is displayed on the left-hand side of the screen just below the ‘Attribute-Value Table’ screen-area.



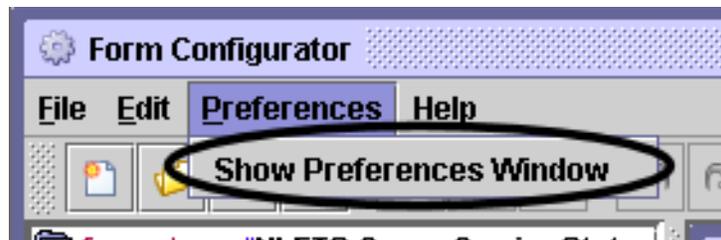
This field contains the text associated with the current selected XML element in the ‘XML Tree’ screen-area. The value in the ‘Text’ field can be edited using any combination of lower and upper cased alphanumeric characters.

Chapter 6: Preferences

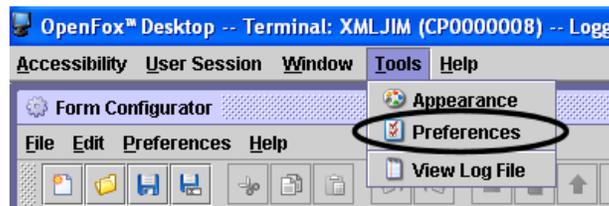
6.1 General Description

All Form Configurator preference options are persistent. This means that if one or more preference options are changed and the OpenFox™ Desktop is logged out, the ‘changed’ preferences are remembered and remain the same when logging back in. In other words, the preference options do not revert back to their default values when logging out and then logging back into OpenFox™ Desktop. Furthermore, preferences settings are saved separately for each user. So if user ‘Jim’ logs into a terminal his Form Configurator preferences can be different then if user ‘Ryan’ logged into the same terminal.

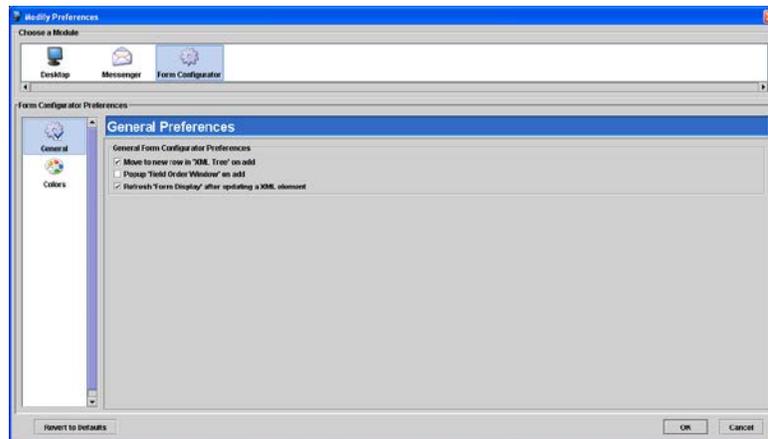
All preference options for Form Configurator can be found in one of two ways. One method is to click on the ‘Preference’ option on the Form Configurator menu-bar and then click on ‘Show Preferences Window’.



Another method is to click on the ‘Tools’ option on the OpenFox™ Desktop menu-bar and then click on ‘Preferences’.

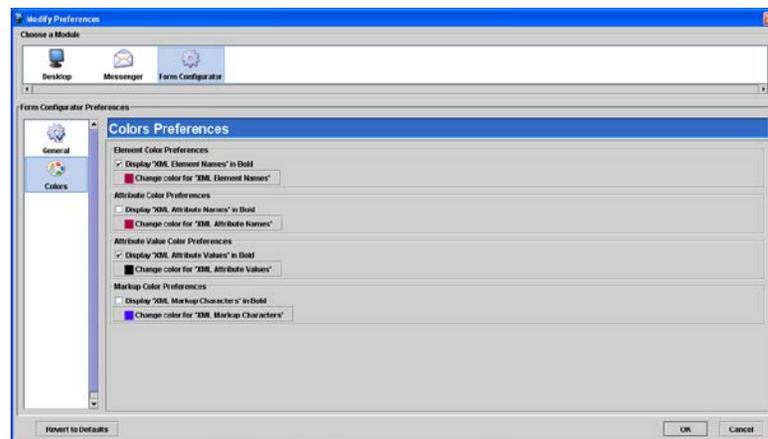


If you followed the steps listed in the first method, the ‘Preferences’ window that pops up should look something like the following:



The only difference between the two methods is that the first method brings up the 'Preferences' window with Form Configurator preferences already selected, while the second method brings up the 'Preferences' window with OpenFox™ Desktop's preferences selected. In which case you can easily display Form Configurator preferences by clicking on the Form Configurator icon displayed in the top section of the 'Preferences' window.

Once you have selected Form Configurator in the top section of the 'Preferences' window, Form Configurator preferences are shown in the lower section of the 'Preferences' window. However, there are two sections in Form Configurator preferences, 'General' and 'Colors'. You can pick the desired section by clicking on the corresponding 'General' or 'Colors' icon shown on the left-hand side of the 'Preferences' window. If you click on the 'Colors' icon, your 'Preferences' window should look something like the following:

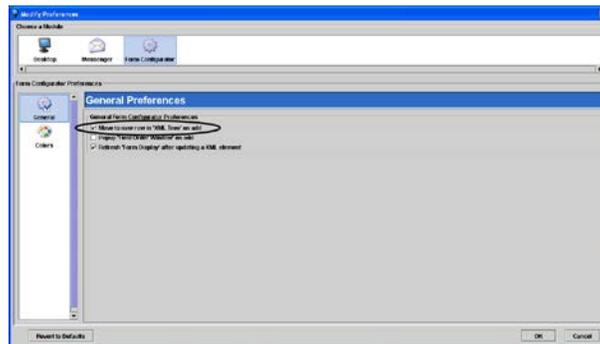


Note that the ‘OK’ button located on the bottom of the ‘Preferences’ window must be clicked after making preference changes; otherwise the changes will be discarded.

The different specific Form Configurator preference options are discussed in the following sections of this chapter. For information about non-Form Configurator preference options please refer to other OpenFox™ documentation.

6.2 Move to New Row on Ad

The ‘Move to New Row on Add’ preference option is found under the ‘General’ section of ‘Preferences’ window. The ‘Move to New Row on Add’ preference option effects the behavior of an add action in the ‘XML Tree’ screen-area. This option is by default turned on.

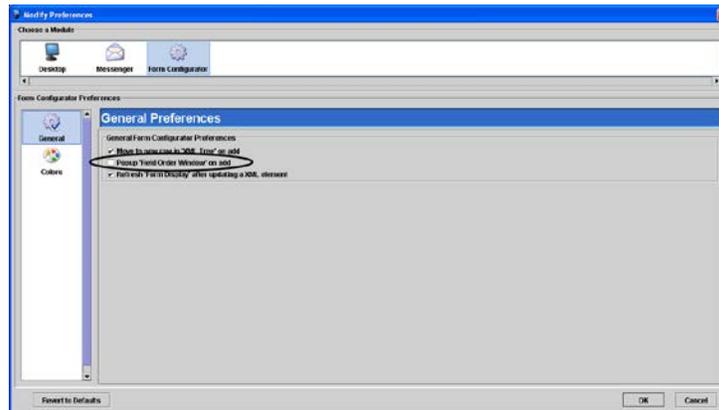


And hence after an element in the ‘XML Tree’ screen-area is added, the new row is automatically selected. If this option is turned off, the parent element will remain as the selected row after you add a new element.

Note that all Form Configurator preference options are persistent. Meaning that if you change one or more preference options, and then log out of OpenFox™ Desktop, when logging back in, the ‘changed’ preferences are remembered and remain the same. In other words, the preference options do not revert back to their default values when you logging out and then logging back into OpenFox™ Desktop. Furthermore, preferences settings are saved separately for each user. So if user ‘Jim’ logs into a terminal his Form Configurator preferences can be different then if user ‘Ryan’ logged into the same terminal. Note that the ‘OK’ button located on the bottom of the ‘Preferences’ window must be clicked after making preference changes; otherwise the changes will be discarded.

6.3 Popup ‘Field Order Window’ on Add

The “Popup ‘Field Order Window’ on add” preference option is found under the ‘General’ section of ‘Preferences’ window. The “Popup ‘Field Order Window’ on add” preference option effects the behavior of an add action in the ‘XML Tree’ screen-area. This option is by default turned off.

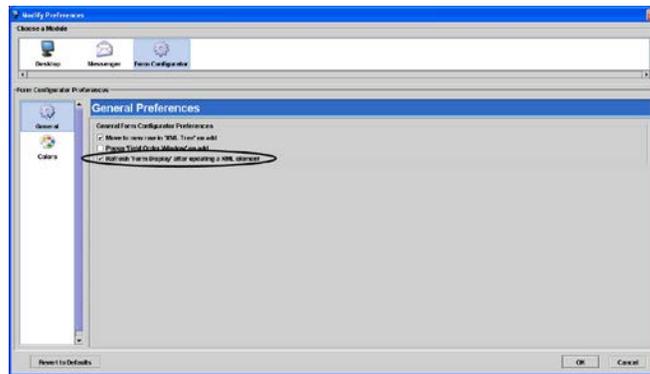


Hence, after adding an element in the ‘XML Tree’ screen-area, the ‘Field Order Window’ will not popup. If this option is turned on, the ‘Field Order Window’ will popup every time a new element is added.

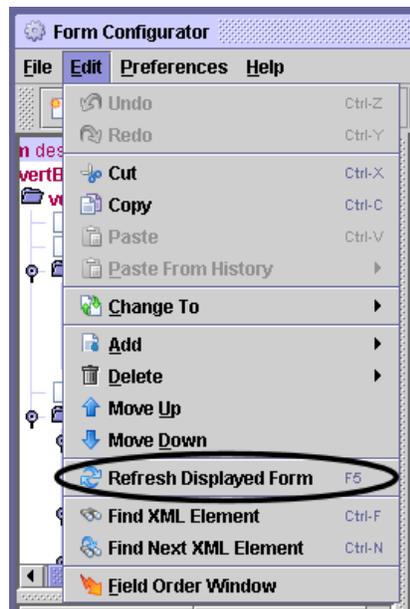
Note that all Form Configurator preference options are persistent. This means that if one or more preference options are changed, and then OpenFox™ Desktop is logged out, the ‘changed’ preferences are remembered and remain the same when logging back in. In other words, the preference options do not revert back to their default values when logging out and then logging back in to OpenFox™ Desktop. Furthermore, preferences settings are saved separately for each user. So if user ‘Jim’ logs into a terminal his Form Configurator preferences can be different then if user ‘Ryan’ logged into the same terminal. Note that the ‘OK’ button located on the bottom of the ‘Preferences’ window must be clicked after making preference changes; otherwise your changes will be discarded.

6.4 Refresh ‘Form Display’ After Updating a XML Element

The “Refresh ‘Form Display’ after updating a XML element” preference option is found under the ‘General’ section of ‘Preferences’ window. The “Refresh ‘Form Display’ after updating a XML element” preference option affects the behavior of the ‘Form Display’ screen-area. This topic is also covered in section ‘2.15 Menu-Item: Refresh Displayed Form’. By default this option is turned on.



Although it is desirable to always have this option turned on, it may be wise to turn off the “Refresh displayed ‘Form’ after updating an XML element” preference option whenever working on huge resources because the performance of Form Configurator can start to slow down for these resources. The speed of Form Configurator can be increased by not updating the ‘Form Display’ screen-area after every change to the underlying XML code. In this way, changes can be made to the resource quickly and without lag-time. Hit the F5 hotkey to “Refresh Displayed Form” and view the changes. Note that this functionality can also be executed by clicking on the ‘Edit’ option on the Form Configurator menu-bar and navigating to the ‘Refresh Displayed Form’ option.



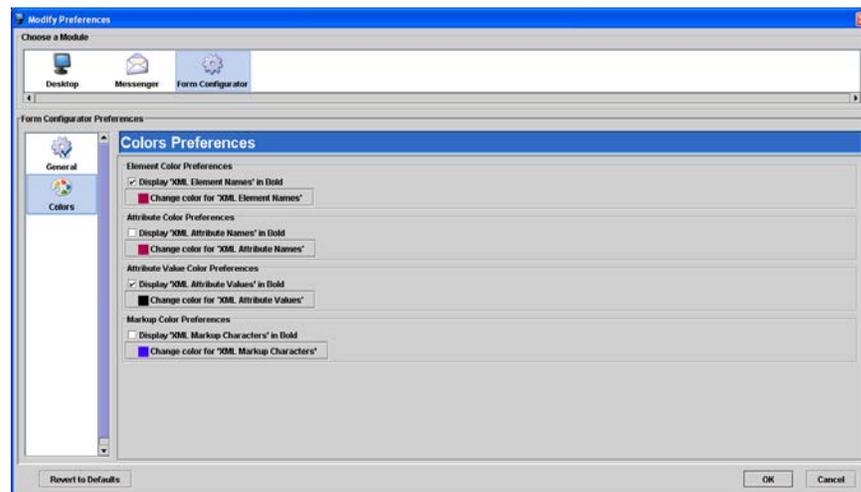
The “Refresh displayed ‘Form’ after updating an XML element” check box option remains checked or uncheck even when another resource is opened. Thus, when un-

checking this option while working on a huge resource, don't forget to re-check it before opening another 'regular sized' resource.

Note that all Form Configurator preference options are persistent. This means that if one or more preference options are changed and OpenFox™ Desktop is logged out, the 'changed' preferences are remembered and remain the same when logging back in. In other words the preference options do not revert back to their default values when you log out and then log back into OpenFox™ Desktop. Furthermore, preference settings are saved separately for each user. So if user 'Jim' logs into a terminal his Form Configurator preferences can be different then if user 'Ryan' logged into the same terminal. Note that the 'OK' button located on the bottom of the 'Preferences' window must be clicked after making preference changes; otherwise the changes will be discarded.

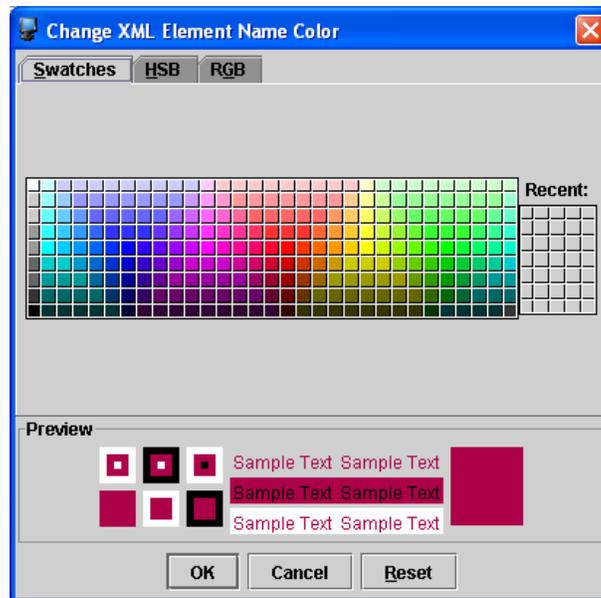
6.5 Change Colors in 'XML Tree' [Screen-Area] and for Printing

The preference options that deal with the color schema of the 'XML Tree' screen-area as well as the color schema used for printing to a color printer can be found under the 'Colors' section of the 'Preference' window.



Note that the colors that are set by these preference options are remembered by Form Configurator. Thus, they remain unchanged even when logging off of OpenFox™ Desktop and then logging back into OpenFox™ Desktop and bringing up Form Configurator. The color schema set will remain unchanged as long as OpenFox™ Desktop runs with the same 'User Id'. If the colors are changed and then the default color schema is wanted back some time later, it can be restored by clicking on the "Revert to Defaults" button and then clicking on the OK button. Both buttons are located on the bottom of the 'Preferences' window.

When clicking on any of the four options that are labeled “Change color for ‘XML ...’” the following window will popup.



If any of the colors displayed in the grid of colors is clicked and then the ‘OK’ button is clicked, the selected color will be shown in the ‘Colors’ section of the Form Configurator ‘Preferences’ window. Clicking the ‘OK’ button located on the bottom of the ‘Preferences’ window, will save all your color changes and show them in the ‘XML Tree’ screen-area. Note for sophisticated users who want to change a color to a specific value of red, green and blue, use the ‘HSB’ or ‘RGB’ tabs shown in the ‘Color’ popup window. Note that the ‘OK’ button located on the bottom of the ‘Preferences’ window must be clicked after making preference changes; otherwise your changes will be discarded.

* * * THIS PAGE LEFT INTENTIONALLY BLANK * * *

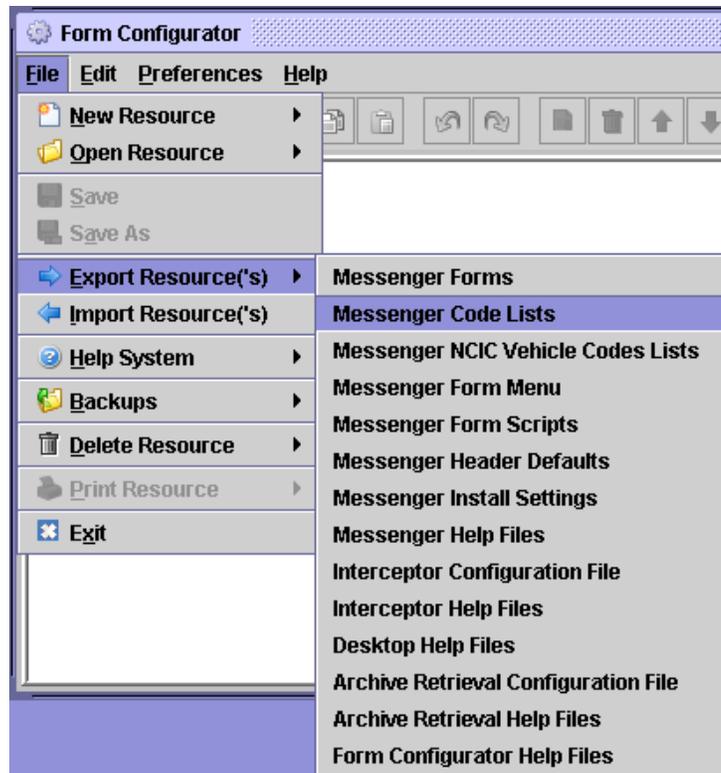
Chapter 7: Export/Import Functionality

7.1 General Description

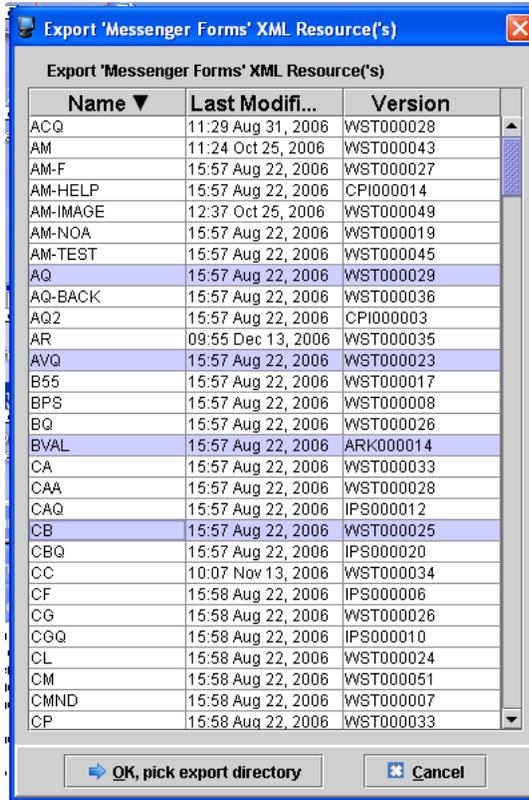
The 'Export/Import' functionality is useful for moving resources from one OpenFox™ system to another. For example if you have a production system and a test system. You could make changes to a form on the test system with Form Configurator and then use Form Configurator to export the resource to your PC. Now exit your Desktop (and Form Configurator) application from the test system and re-launch Desktop and login to your production system. Lastly, use Form Configurator to Import the form from your PC into your production system. This functionality is very useful for CPI personnel who often move Messenger XML resources from one system to another.

7.2 Export Functionality

When you use the 'Export' functionality one or more XML resources are copied from the OpenFox™ system and written to files that are stored locally on your computer. The 'Export' functionality can be accessed by clicking on 'File' in the Form Configurator menu-bar and navigating to 'Export Resource(s)'.

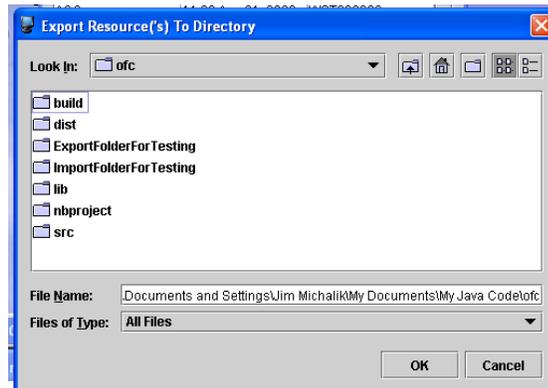


After clicking on ‘Export Resource(s)’ choose one of the resource types that appear in the list to the right. After picking a resource type you will be presented with a dialog box from which you can pick the actually resource or resources you want to export.

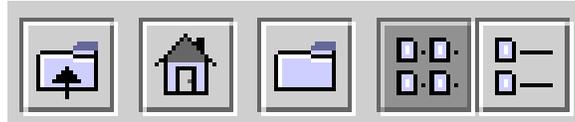


Notice how 4 separate rows have been selected in the above dialog box. You can select more than one row by holding down the Ctrl key before you click over a row with the left mouse button. You can select a range of rows by using the combination of the shift key and the left mouse button. For example to select all rows (or resources) click on the topmost row in the dialog box. Then scroll down all the way to the bottom. Now hold down the shift key then click the left mouse button over the last row in the dialog box.

After you are done selecting the resource(s) you want to export click on the ‘OK pick export directory’ button. Use this dialog box to pick the directory to which you want to export the resource(s).



This dialog box only shows directories, not files. Click on a directory name to display any sub-directories in that directory. Notice the buttons in the upper right-hand corner of the dialog box.



From left to right the functionality of these buttons is as follows:

- ➔ Click to go up one directory level.
- ➔ Click to go to your PC's Desktop directory.
- ➔ Click to create a directory (i.e. folder) under the current selected directory.
- ➔ Click to just display the directory names.
- ➔ Click to display the directory names and standard statistics such as file type and time last modified.

When you are done choosing your directory click the 'OK' button to begin exporting your resource(s). Once the export process starts a status window will appear to display the progress of the export operation.



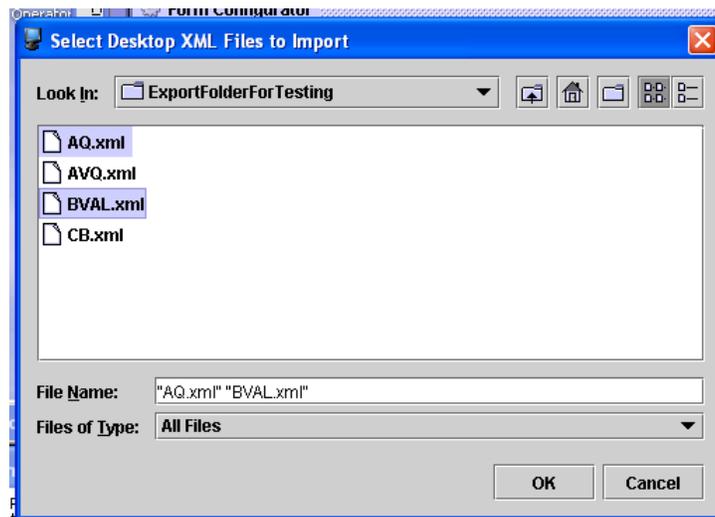
After completion another status window will appear reporting the results of the export operation.

7.3 Import Functionality

The 'Import' functionality is designed to only be used on OpenFox™ XML resource files that have previously been exported from an OpenFox™ system using the 'Export' functionality. The 'Import' functionality will import one or more OpenFox™ XML resource files in one operation. To access the 'Import' functionality click on 'File' on the Form Configurator menu-bar and navigate to the 'Import Resource(s)' menu-item.

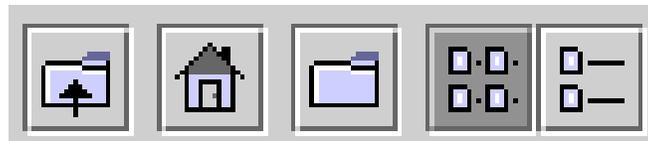


After selecting 'Import Resource('s)' you will be presented with a dialog box that displays directories and the files they contain.



Use this dialog box to select one or more OpenFox™ XML resource files you want to import into an OpenFox™ system. Notice that two filenames have been selected in the above dialog box. You can select more than one file by holding down the Ctrl key before you click over a filename with the left mouse button. You can select a range of filenames by using the combination of the shift key and the left mouse button. For example select the topmost file name in the dialog box. Then hold down the shift key and then click the left mouse button over the last filename in the dialog box.

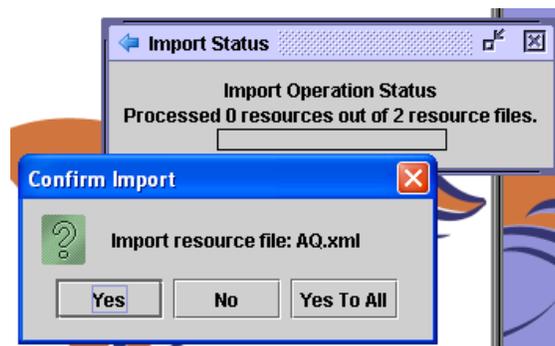
Notice the buttons in the upper right-hand corner of the dialog box.



From left to right the functionality of these buttons is as follows:

- ➔ Click to go up one directory level.
- ➔ Click to go to your PC's Desktop directory.
- ➔ Click to create a directory (i.e. folder) under the current selected directory.
- ➔ Click to just display the directory names.
- ➔ Click to display the directory names and standard statistics such as file type and time last modified.

After you are done picking the OpenFox™ XML resource files you want to import click the 'OK' button. You will now be presented with two windows, a status box and a confirmation box.



Simply confirm the OpenFox™ XML resource files you want to import to complete the import operation.

* * * THIS PAGE LEFT INTENTIIONALLY BLANK * * *

Chapter 8: Running Help System Commands

8.1 General Description

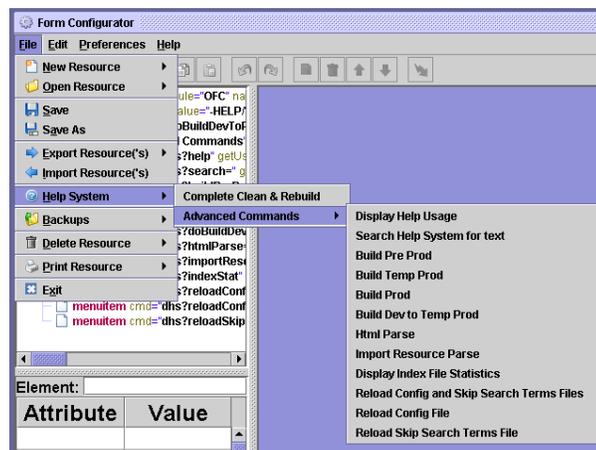
The Desktop Help System is accessible by clicking on Help in the Desktop menu-bar and choosing 'Run Help'. This section discusses how you can use Form Configurator to maintain the data in the Desktop Help System.

You can run all the Desktop Help System maintenance commands from Form Configurator. The following are examples of when and why you would want to run Help System commands:

- You updated some of the Messenger Code List information (such as the NCIC Vehicle Codes) and you want the updated data to be reflected in the Desktop Help System so all users will see the new data when they access the Desktop Help System.
- You have updated, added or deleted a XHTML file contained in the Desktop Help System and you want the changes you made to be reflected so all users will see the changes when they access the Desktop Help System.

8.2 Help System Introduction

To access the Desktop Help System commands go to the Form Configurator menu-bar and click on file, and then choose 'Help System'.



The primary command that most users will run is 'Complete Clean & Rebuild'. This command uses the development Desktop Help System HTML files and the production Messenger codes data (such as the NCIC Vehicle Codes) to build all the production Desktop Help System XHTML files. These production files are what users see when they access the Desktop Help System.

This is a Form Configurator manual and is not intended to be a manual on the Desktop Help System but I will provide a very basic description of some aspects of this system here to provide you a better understanding of the Desktop Help System commands and their usage. For a more detailed explanation of the Desktop Help System please refer to the Desktop User Manual.

The Desktop help system basically uses 3 sources of data:

- XHTML (which means a file is both XML and HTML compliant) text files. These are the XHTML files displayed to users when they use the Desktop Help System.
- Help XML resource files. Like all Desktop/Messenger resources these are XML files that you can edit with Form Configurator. Each Desktop module (Messenger, Form Configurator, etc.) has a Help Table-Of-Contents file (if help is available for that module). The primary purpose of these files is to provide a name and an URL pointing to the location of each XHTML file in the Desktop Help System.
- The Messenger Codes files. Like all Desktop/Messenger resources these are XML files that you can edit with Form Configurator. Various Help System build commands use these files to add the 'codes' data they contain into the Desktop Help System XHTML files. So if one or more Messenger Codes files are modified (using Form Configurator), you could run the appropriate Desktop Help System command from Form Configurator to update all Desktop Help System XHTML files to use all the latest Messenger Codes data.

The XHTML files in the Desktop Help System reside under 2 primary directories and several staging directories on your production server.

- /CPI/web/help/DevHtmlDir – This is the directory under which all editing and modifications to the XHTML files should take place.
- /CPI/web/help/ProdHtmlDir – **These are the XHTML files that are actually displayed when you use the Desktop Help System. These files should not be directly edited. All changes to Desktop Help XHTML files should take place under the DevHtmlDir directory.**
- /CPI/web/help/PreProdHtmlDir – The contents of this directory should be ignored, it is used internally by the Desktop Help System.
- /CPI/web/help/TempProdHtmlDir – The contents of this directory should be ignored, it is used internally by the Desktop Help System.

If you wish to edit an XHTML file in the Desktop Help System you would first identify the name of the file and under which directory path (under the DevHtmlDir directory) the file resides. For example say you wanted to modify the 'NCIC Gun – Introduction' XHTML file. The directory under which this file is located is /CPI/web/help/DevHtmlDir/MESSENGER/NCIC/GUN the name of the file is Gun_Introduction.html. All you need to do is use any text editor or an HTML editor to edit this file, and then execute a 'Complete Clean & Rebuild'. Note that it usually takes several minutes before this command completes running (this is because there are currently at least 425 XHTML files in the Desktop Help System). The command processes all files under DevHtmlDir, and then moves all files to ProdHtmlDir. The changes you made are now viewable by all users.

To read more detailed information about the various Desktop Help System commands, run the ‘Display Help Usage’ command. The output from this command provides a high level description for each step executed, for all Desktop Help System commands.

8.3 ‘Complete Clean & Rebuild’ Command

This command actually runs 3 advanced Desktop Help System commands, one after the other.

- buildPreProd
- buildTempProd
- buildProd

It typically can take a few minutes for the ‘Complete Clean & Rebuild’ command to complete (this is because there are currently at least 425 XHTML files in the Desktop Help System). A summary of the primary actions that are executed during a ‘Complete Clean & Rebuild’ are listed below.

- Backup the current contents of the PreProdHtmlDir and ProdHtmlDir directories to Backup directory.
- Read all XHTML files and validate that they are valid XML (XHTML files are both XML and HTML compliant). Report files that are not valid XML. Note that this step is equivalent to running a ‘HTML Parse’ on all the files under the DevHtmlDir directory.
- Resolve all importResource tags in all XHTML files. The importResource tags contain information that tells the Desktop Help System to insert Messenger Codes data into a XHTML at the location of the importResource tag (replacing the importResource tag). The resulting XHTML file (with the inserted Messenger Codes data) ends up being copied to both the TempProdHtmlDir and ProdHtmlDir directories.
- Read all the text in all the XHTML files and build an index file (DesktopHelp-Index.xml) that is used to support the ‘Search’ functionality in the Desktop Help System. For example after the ‘Complete Clean & Rebuild’ is complete, a user could search for any text, say ‘NCIC’. And be returned a list of all the XHTML files that contain that text.

A way to view a detailed description of all the Desktop Help System commands is by running the ‘Display Help Usage’ command. The output from this command provides a high level description for each step executed, for each Desktop Help System command. A copy of the descriptions for each of the above three advanced Desktop Help System command’s is provided below.

buildPreProd

- 1.) Block all other build operations from running, abort if can not.
- 2.) Backup (by moving) PreProdHtmlDir directory to the Backup directory.
- 3.) Copy all files in DevHtmlDir to PreProdHtmlDir directory.

- 4.) Unblock all other build operations from running.

buildTempProd

- 1.) Block all other build operations from running, abort if can not.
- 2.) Block the FM server from completing Form Configurator save requests, abort if can not.
- 3.) Delete TempProdHtmlDir directory if it exists.
- 4.) Copy all files from PreProdHtmlDir directory to TempProdHtmlDir directory.
- 5.) Use XSL style sheet to resolve all links (to Desktop resource files) in XHTML files.
- 6.) Unblock FM server from completing Form Configurator save requests.
- 7.) Build the Temp-DesktopHelp-Index.xml file.
- 8.) Unblock all other build operations from running.

buildProd

- 1.) Block all other build operations from running, abort if can not.
- 2.) Block all search's from running.
- 3.) Backup all the production tableOfContentsFiles (located under the FM server's forms directory) to the Backup directory.
- 4.) Backup (by copying) ProdHtmlDir directory to the Backup directory.
- 5.) Sync ProdHtmlDir with TempProdHtmlDir.
 - a.) Delete files/dir's in ProdHtmlDir that don't exist in TempProdHtmlDir.
 - b.) If file/dir exists in TempProdHtmlDir but not in ProdHtmlDir then copy file/dir to ProdHtmlDir.
 - c.) If a file exists in both TempProdHtmlDir and ProdHtmlDir but their contents are different. Delete file in ProdHtmlDir and then copy file from TempProdHtmlDir to ProdHtmlDir.
- 6.) If exists move Temp-DesktopHelp-Index.xml to DesktopHelp-Index.xml (will also first backup DesktopHelp-Index.xml file to the Backup directory).
- 7.) Copy all DevTableOfContentsDir TABLEOFCONTENTS.xml files to their production locations (i.e. under the FM server's forms directory).
- 8.) If one or more tableOfContents XML file('s) were moved to production, ask FM server to reload it's version-cache data.
- 9.) Make all files in ProdHtmlDir have read only permissions.
- 10.) Unblock search operations from running.
- 11.) Unblock other build operations from running.

UPON COMPLETION THE FOLLOWING ACTIONS SHOULD TAKE PLACE:

- ALL USERS SHOULD CLOSE AND REOPEN DESKTOP HELP WINDOW

TO PREVENT

ERRORS

- REMEMBER TO REMOVE ALL TABLEOFCONTENTS.xml FILES UNDER DevTableOfContentsDir BEFORE RUNNING buildProd AGAIN.

Chapter 9: Backups

9.1 General Description

Whenever a resource (XML file) is saved in Form Configurator a copy of that XML resource file is created and copied to a backup directory. You can access and open all backup files at any time using the Backup functionality.

To restore a backup file, that is to make a backup the current version of a resource. You first open a backup file with Form Configurator, then you save the file. If the save action is successful, then your copy of the backup file automatically becomes the current version of that resource.

To help better understand how the backup functionality works, it is helpful to understand what happens when you save a file. Let's list the primary actions that take place when you successfully save a XML resource file with Form Configurator.

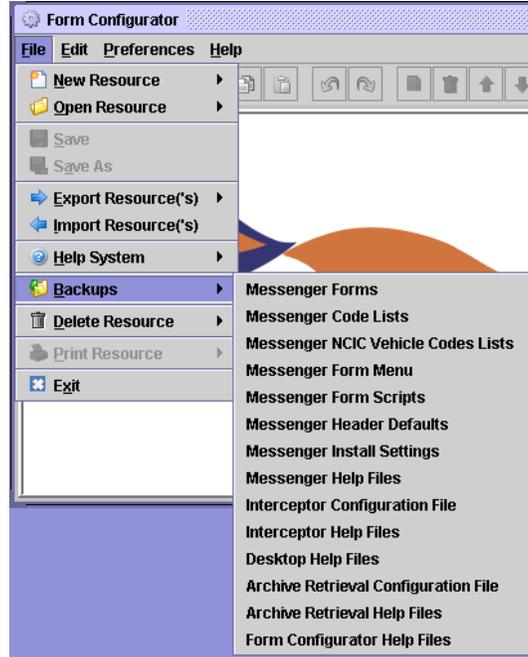
- Form Configurator sends a copy of the current resource you have open (whether or not it's a backup file version) to the FM server (Form Manager Server).
- The FM server calculates the next version number value for that resource, and modifies the version number field in the XML resource with that value.
- The FM server writes the XML resource to file, this file now becomes the current version for this XML resource. Anyone using Messenger, Form Configurator, etc. who tries to access or open this resource will be sent a copy of this version of the resource.
- The FM server writes a copy of the XML resource to the backup directory.
- The FM server sends back a successful acknowledgment message to the Form Configurator client. Imbedded in this message is the new version number for this resource. Form Configurator will use this new version number to update the version number field of the resource, which it is currently displaying on screen.

Likewise, when you open a resource with Form Configurator you are sent a copy of that resource by the FM server. Therefore if you open a backup file and make changes to the resource. When you save the resource it becomes the current version of that resource, but notice that you never actually modified the backup file that's stored on the server. In this way the integrity of the backup files is maintained. That is you can't actually edit a backup file, you can only edit your **copy** of a backup file and make it the current version for that resource.

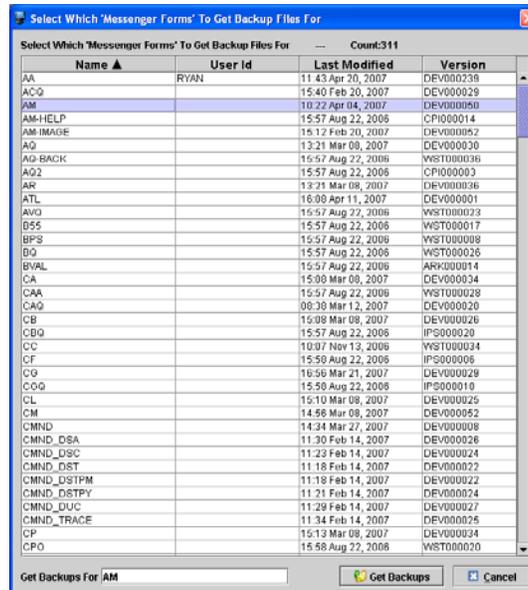
The number of backups allowed for each XML resource is configurable via an XML configuration file located on the server (see CPI personnel for assistance).

9.2 Restoring A Backup

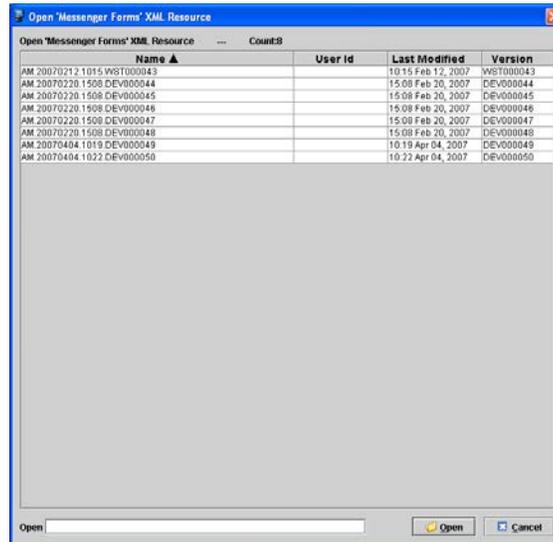
Let's walk through an example, let's restore the AM form to an old version. First go to the Form Configurator menu-bar and select 'File', then select 'Backups'.



Now select 'Messenger Forms'. A list of all the Messenger Forms will appear. Select AM from this list and click on the 'Get Backups' button.



You should now see a window listing all the backup files that are available for resource AM.



Choose a backup file version and click on the 'Open' button. The AM form will now be displayed. If you now 'Save' this form you will make this backup version the current version of the AM resource.

* * * THIS PAGE LEFT INTENTIONALLY BLANK * * *

Chapter 10: Miscellaneous

10.1 Error Handling and Reporting

When an error occurs during the operation of Form Configurator one of 2 things will happen.

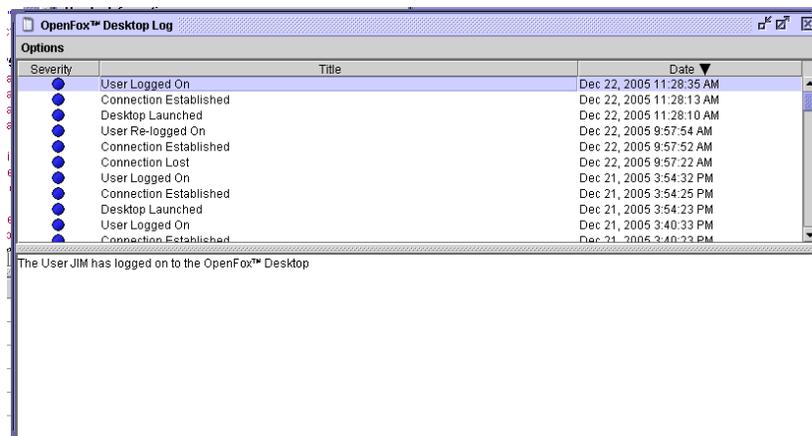
- Error message will popup on screen.
- Error message will popup on screen and the error message will also be written to the OpenFox™ Desktop ‘Log File’.

By far most users should not need to access the OpenFox™ Desktop ‘Log File’. However, this topic is addressed for those rare cases where this functionality is desired. Please refer to the OpenFox™ Desktop documentation for more information on this subject.

The OpenFox™ Desktop ‘Log File’ can be accessed by clicking on the ‘Tools’ option on the OpenFox™ Desktop menu-bar and then navigating to the ‘View Log File’ option.



Clicking on this option causes the OpenFox™ Desktop ‘Log File’ to be displayed. A sample log file is shown below.



When viewing the log file, clicking on an item/row displayed in the ‘Log File’ table will display the message that log file entry contains in the lower half of the window. It is possible to save this information to a file, that can then be emailed to the system support personnel to help trouble shot any problems that may occur. As mentioned above, for more information on the OpenFox™ Desktop ‘Log File’ please refer to the OpenFox™ Desktop documentation.

10.2 Operation Using Keyboard Only

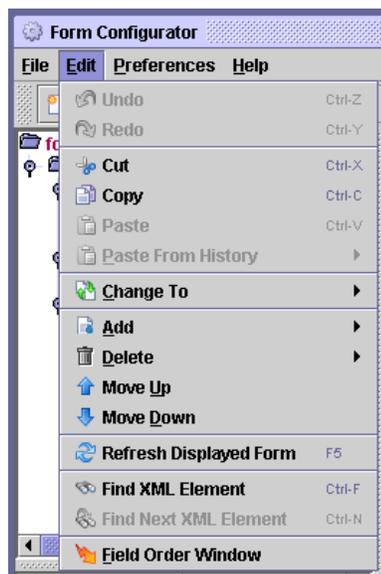
Form Configurator is completely operational using only the keyboard. This topic is broken up into the following sub-sections:

10.2.1 Using Alt-Keys and Ctrl-Keys

There are various hotkeys that can be utilized at any time from within Form Configurator and OpenFox™ Desktop. The way to identify which Alt hotkeys are available is to look for the underlined letter in the name of the button, menu-item or tab that is displayed on screen. For example, if the ‘NCIC Vehicle Codes’ screen is displayed, look closely at the tab’s.

Notice that the ‘i’ in one tab and the ‘S’ in the other tab are underlined. This means that if at any time from within the ‘NCIC Vehicle Codes’ screen, hitting the Alt-I will open the ‘Edit Make and/or Automobile Models’ tab, and hitting the Alt-S will go to the ‘Edit Model and/or Style’ tab.

The best way to identify the Ctrl-hotkey or function-keys that are available is to view the menu-items found under the menu-bar. For example, click on the ‘Edit’ option on the Form Configurator menu-bar.



Notice that for some of the menu-items, Ctrl-<letter> items are shown on the right-hand side. For example, for the ‘Cut’ menu-item, Ctrl-X is shown, or F5 is shown for ‘Refresh Displayed Form’ menu-item. These are the hotkeys for the corresponding menu-item actions. One important point that must be mentioned for the Ctrl-<letter> and function hot keys is that they are available even if they are not shown on screen. For example, if the cursor was in the ‘XML Tree’ screen-area on some selected element and the Ctrl-X keys were hit, that element would be cut from the underlying XML document and removed from the ‘XML Tree’ screen-area. So, the Ctrl-<letter> and function hotkeys are effectively globally active hotkeys, unlike the Alt-<letter> hotkeys where the underlined letter must be seen on screen for them to be active.

One important point is that while the menu bar in Form Configurator is visible on screen, the OpenFox™ Desktop menu-bar is also visible on screen. Hence, the visible Alt-<letter> hotkeys in the OpenFox™ Desktop menu-bar are active as well as all the Ctrl-<letter> and function hotkeys contained in the OpenFox™ Desktop menu-bar menu-items.

10.2.2 Accessibility Menu

The Form Configurator ‘Accessibility’ menu is actually contained in the OpenFox™ Desktop menu-bar.



The goal of the ‘Accessibility’ menu is to provide a non-mouse oriented option for the Form Configurator user to navigate between the major screen-areas in the Form Configurator screen. Notice that there are Ctrl-<letter> hotkeys available for each of the 3 major screen-areas. As mentioned in the above section “7.2.1 Using Alt-Keys and Control-Keys” these hotkeys are available at any time while Form Configurator is running.

10.2.3 Using Tab and Shift-Tab

The tab key and shift-tab key combination are used for navigating around the primary screen-areas in Form Configurator. The primary screen-areas in Form Configurator are:

- XML Tree [screen-area]
- Attribute Value Table [screen-area]
- Form Display [screen-area]

The tab button will allow movement in a counter-clockwise direction between these 3 screen-areas. Movement in a clockwise direction between these 3 screen-areas can be done by hitting a shift-tab.. Note that when using tab or shift-tab while in the Form Display, the cursor will first move over all fields and buttons in the form before moving to either the 'XML Tree' screen-area or the 'Attribute-Value Table' screen-area.

When in the 'NCIC Vehicle Codes' screen-area, tab and shift-tab still work and can be used to navigate between tables, fields, buttons, radio buttons and check boxes.

10.2.4 Keyboard Only Behavior of 'Attribute-Value Table' [Screen-Area]

Navigating around the 'Attribute-Value Table' screen-area without the use of the mouse can be done by using the arrow keys and/or the enter key. In this way it is possible to move from cell to cell in the table.

As stated in section "2.5 Attribute-Value Table [Screen-Area], there are 4 different cell types in the 'Value' column that are used to edit data. The following key strokes should be used to start editing for each of the different cell types. Note the additional keyboard operations that can be used once editing is begun on each of these cell types.

- Text – To start editing hit the space bar. Now if you type characters they will be appended at the end of the current text string in this field. You can enter all standard characters into these cells. You can use the left/right arrow keys, delete and backspace when editing this cell type. Hit the enter key to commit your changes. Depending on the associated attribute, some text fields only allow upper cased letters, and some have limited lengths as well.
- Check Box – To change the value of this cell hit the space bar. You can toggle between checked and unchecked with the space bar. These are cells that are shown with a small box inside. With

or without a check in the box. They represent a true / false value. If checked, means set to true. False if unchecked. A Check Box cell is shown below.



- **Number** – To start editing in this cell (without the use of a mouse) you must hit the space bar followed by the tab key. Once editing is started you can just type digits or use the up/down arrow to increment/decrement the current number value. Hit the enter key twice to commit your changes, or just hit the enter key once if you made no changes. Shown with small ‘spinner’ up/down arrow buttons on right-hand side of the cell. These cells only allow number characters to be entered. Depending on the attribute that’s associated with the cell, negative values can be entered as well. A number cell is shown below.

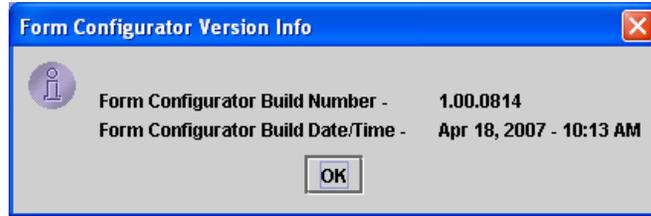


- **Drop Down List** – To start editing in this cell hit the space bar followed by the tab key followed by the down arrow key. Use the up/down arrow keys to pick values from the list, or you can also just type alphanumeric characters if this is an editable Drop Down List. Hit the enter key to commit your changes. An editable Drop Down List is shown below.



10.3 About Window

If you would like to know what version of Form Configurator you are using go to the Form Configurator menu-bar and click on 'Help'. Now click on 'About', you should see a window displaying the Form Configurator version information.



Appendix

A. Definition of Terms

Attribute – An XML attribute is associated with an XML element and provides additional information about the XML element. Each attribute can have an attribute-value associated with it. There can be zero to an unlimited number of attributes associated with a XML element.

Check Box – A graphical cell that is shown as a small box, with or without a check mark in the box. Check Boxes are shown to users of screen based programs (i.e. GUI programs). They represent a true / false value. If checked, means set to true, false if unchecked.

Client Application – A client application is a software application that is intended for distribution to multiple or remote computers. Client applications typically communicate to a main or central computer or server.

Combo Box – Combo Boxes can be editable or non-editable. Combo Boxes provide a means for a user of a screen based program (i.e. GUI program) to select from a list of values. Additionally editable Combo Boxes allow the user to type in a value rather than pick from the list of values. Every Combo Box has a Drop-Down List associated with it that displays the list of possible values that are associated with the Combo Box, whenever the Combo Box is selected by the user.

Drop-Down List – A Drop-Down List is associated with a Combo Box. Typically when the user selects the Combo Box (i.e. causes the program's focus to go to the Combo Box for example by clicking on the Combo Box with a mouse) the Drop Down List is displayed. Drop-Down Lists provide a means of associating a list of different valid values to one Combo Box field. For large lists the Drop-Down List is displayed with scrollbars.

Element – An XML element is a unit of XML data, delimited by tags. An XML element can enclose other elements and can have attributes associated with it.

Field – A text area displayed to users of screen based programs (i.e. GUI programs). It provides a means for users to enter text in screen based programs.

Graphical User Interface (GUI) – A graphical user interface is an interface for issuing commands to a computer utilizing a pointing device, such as a mouse, and/or a keyboard that manipulates and activates graphical images on a display.

Hot Keys – A term used in this document to refer to Ctrl-<letter> key combination, Alt-<letter> key combination and function keys which are all used to execute some associated action in a screen based program (i.e. GUI program).

Java – A high-level programming language developed by Sun Microsystems. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (VM's), exist for most operating systems, including Unix, the Macintosh OS, and Windows. Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web.

Radio Button – A Radio Button is a small graphical circular outlined area shown in screen based programs (i.e. GUI program) that the user can click to select an option or command. When a Radio Button is selected a small black circular dot is displayed inside the Radio Button. Radio Buttons are often presented to users in groups such that the user is asked to select only one of the Radio Buttons in the group. This means that the text associated with the Radio Button with the black dot (the Radio Button that was selected by the user) logically becomes activated or true.

Tab – A means for presenting multiple screen interfaces to users in a screen based program (i.e. GUI program). Typically several Tabs are displayed together in a Tabbed Pane. Each Tab represents a screen layout that the user can choose. The user selects one tab, typically with a mouse click, and then the screen associated with that Tab appears.

XML – Extensible Markup Language, which allows you to define the elements that you need to identify the data and text in XML documents.