



Aspect Control Center Form Developer User's Guide

Aspect Control Center
Form Developer User's Guide
© **INFOTECH Soft, Inc.**
1201 Brickell Avenue • Suite 220
Miami • Florida • 33131
Phone 305.371.5111 • Fax 305.371.5112
Email solutions@infotechsoft.com
Web <http://www.infotechsoft.com>

Last Updated May 4, 2010

The information of this document may change without notice.

In case of any problem with this document, please send an email to solutions@infotechsoft.com

| | |
|--|----|
| Aspect Control Center | 4 |
| Form Developer Plug-In..... | 5 |
| Create Form Definition Action | 6 |
| Open Form Definition Action | 8 |
| Load Form Definition Action..... | 9 |
| Close & Save Form Definition Action | 9 |
| Save As Form Definition Action..... | 9 |
| Enabling/Disabling Auto-Refresh Action | 10 |
| Form Structure Navigation | 10 |
| Collapse and Expand Tree Action | 10 |
| Modify Form Structure Action | 10 |
| Form Code List View..... | 11 |
| Form Layout View | 11 |
| Attribute Editor | 12 |
| Form Attribute Editor | 12 |
| Item Group Attribute Editor..... | 15 |
| Item Attribute Editor..... | 15 |

Aspect Control Center

The Aspect Control Center is the main administrative interface into the Aspect System. As such, the Control Center supports plug-in components for form design, security settings and user management, synchronization configuration, conflict resolution, and system auditing. The Control Center also takes care of user authentications.

- **User Management:** The user management task takes care of creating user accounts in the Aspect System, configuring the users' permissions, and managing each user's profile, including signing certificate and system settings.
- **System Auditing:** The system-auditing task provides the means of easily analyzing the audit information of the Aspect System. Additionally, the system-auditing task includes an interface to customize the amount of auditing information to collect within the Aspect System.
- **Synchronization Configuration:** The synchronization configuration task manages the process of defining ClinicalSync settings.
- **Form Design:** One of the main tasks of the Control Center is the creation and management of portable electronic form definitions to be used throughout the Aspect System.
- **Login:** When started, the Control Center prompts you for username and password in order to authenticate you. Since the login is common to all Control Center plug-ins, you will not be ask to log-in again when switching for one plug-in to the next. If you cannot be authenticated by the system, the plug-ins will not load.
- **Logout:** When you exit the Control Center, you are automatically logged out of the Aspect System. The Control Center also provides you with a way to logout out of the system without closing the control center. When you logout, the plug-in is disabled and replaced by a blue screen. Similarly to the login, the logout is common to all plug-ins; once logged out, all the plug-ins that requires a connection to the system are disabled.



You must be an administrator to access the Control Center



The Login/Logout button is accessible under the File menu in both the Security Manager and the Data Manager.

Form Developer Plug-In

The Form Developer plug-in provides a platform-independent, graphical development tool that assists form developers with the operations of defining new clinical forms and modifying existing forms. Its easy-to-use graphical user interface provides "point-and-click" ability to define the content, the layout, and the look-and-feel of forms, conforming to the *form definition specification*. Upon user inputs, the Form Developer automatically generates the appropriate XML file that will be synchronized with the Clinical Forms Database in order to be used within the Forms Administrator.

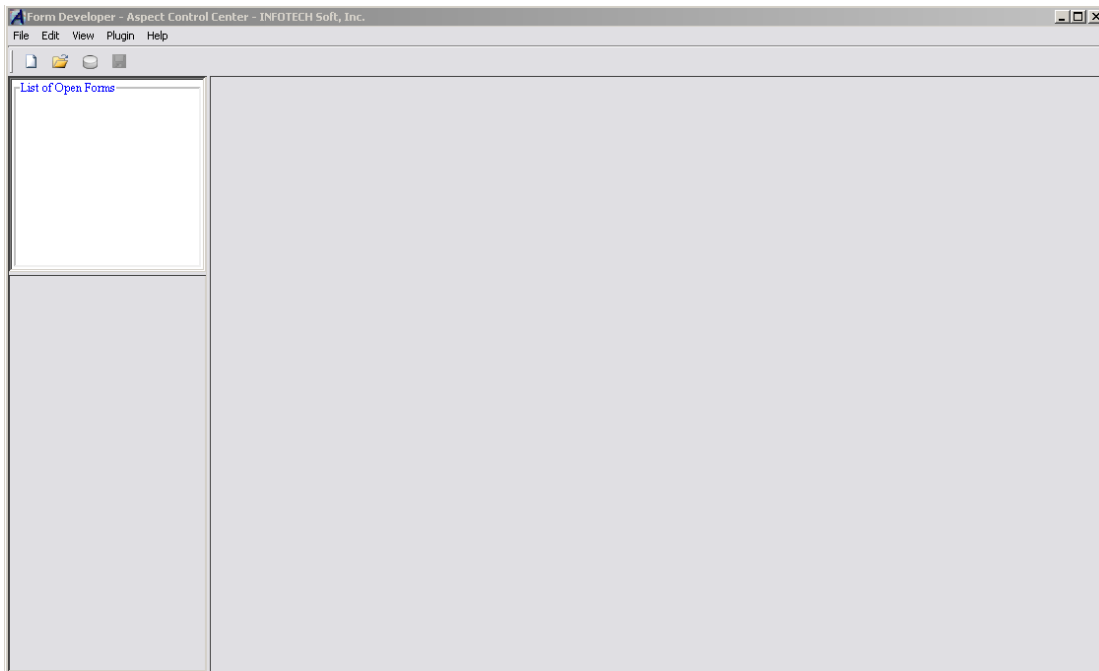
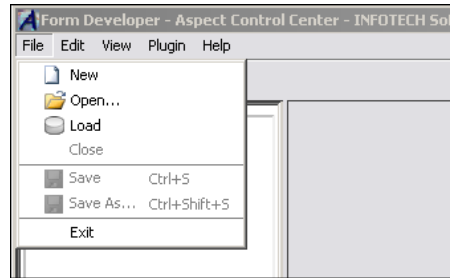


Figure 1 Form Developer Plug-in

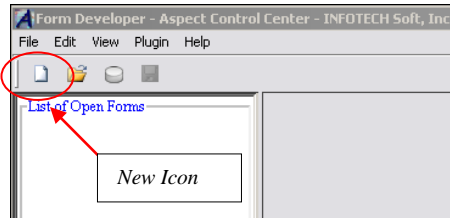
Create Form Definition Action

You can create a new form definition:

- I. By clicking *File*, then *New*



- II. By Clicking the *New Icon* on the toolbar



After selecting one of the options listed above, a dialog box appears, prompting you for a form ID (an alpha-numerical value that may include dashes and underscores), a version (an integer indicating the version of the form), a language (the language in which the content of the form will be), a name (the name of the form), and a layout ID (the id of the default layout attached to the new form). Fill in the dialog box spaces. The dialog box is shown in figure 2.

The 'New Form' dialog box contains the following fields and controls:

- Form ID:** Text input field containing 'sample-form_1a'.
- Version:** Spin box containing '0'.
- Language:** Dropdown menu showing 'English'.
- Name:** Text input field containing 'Sample Form'.
- Layout ID:** Text input field containing 'default'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

Figure 2 New Form Dialog

Once the form is created, a reference to it will appear in the *form selection list* on the upper left side of the Form Developer (see Fig. 3). The reference will read: '*form name*' [*'form id'* v'*version#*] [*'language code'*, '*layout id*']. The reference selected dictates which form will be shown. Below the *form selection list*, the *form structure navigation* provides a tree-like layout, describing the current form's content and structure. Each item group and item contained in the form can be easily selected and edited; when a node of the navigation tree is selected, its corresponding graphical element is indicated within the graphical form and its attributes are displayed in the *form attribute editor* located on the right side of the Form Developer. The layout components and the code lists can be edited from the tabs located also on the left side of the developer. The title on the Form Developer will now indicate which form

definition is displayed and the save status of that form. Since the form has just been created, it is not yet saved; thus, 'Not saved' will be shown. (If you opened a saved form, the path of the form definition file will be shown. Furthermore, a star indicates that a saved file was modified after the last time it was saved.) Figure 3 shows the Form Developer after creating a new Form.

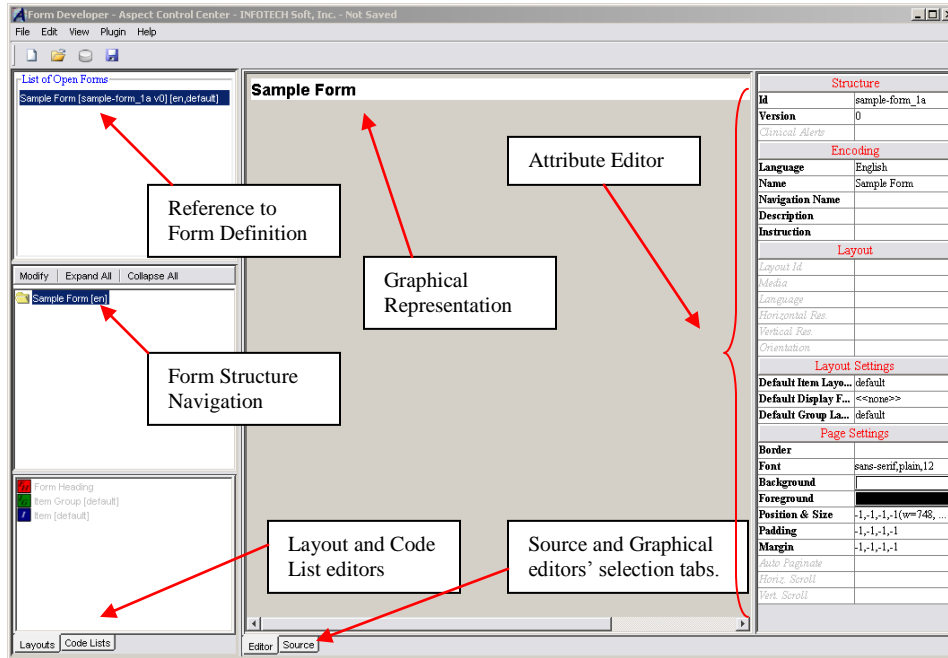


Figure 3 Form Developer after creating a new Form

If you are an advanced user, the Form Developer gives you the option of working directly with the XML format of the form definition through the source editor (see Fig. 5). Since the XML text can be very long, a tree navigation situated on the bottom left side of the Form Developer enables you to jump quickly to key locations of the text. A few tools are available to render the source editing easier: you can check if the XML is well-formed, reformat the text, undo and redo, and reload the last updated text by clicking on the buttons placed in the source toolbar. Any changes made in the source editor will apply to the form definition only after you click on apply changes. If you want to refresh the tree navigation without applying any changes, click on the refresh tree button. If the XML text is not well-formed, you will not be able to reformat it or to apply the changes made; an error message will popup indicating the reason for the mistake and its location. After closing the error message, the mistake will be highlighted within the source editor. Figure 4 shows a sample error message.

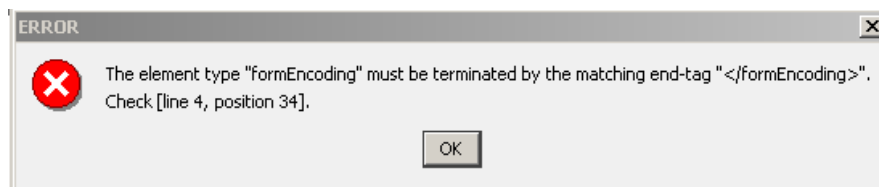


Figure 4 Error Message

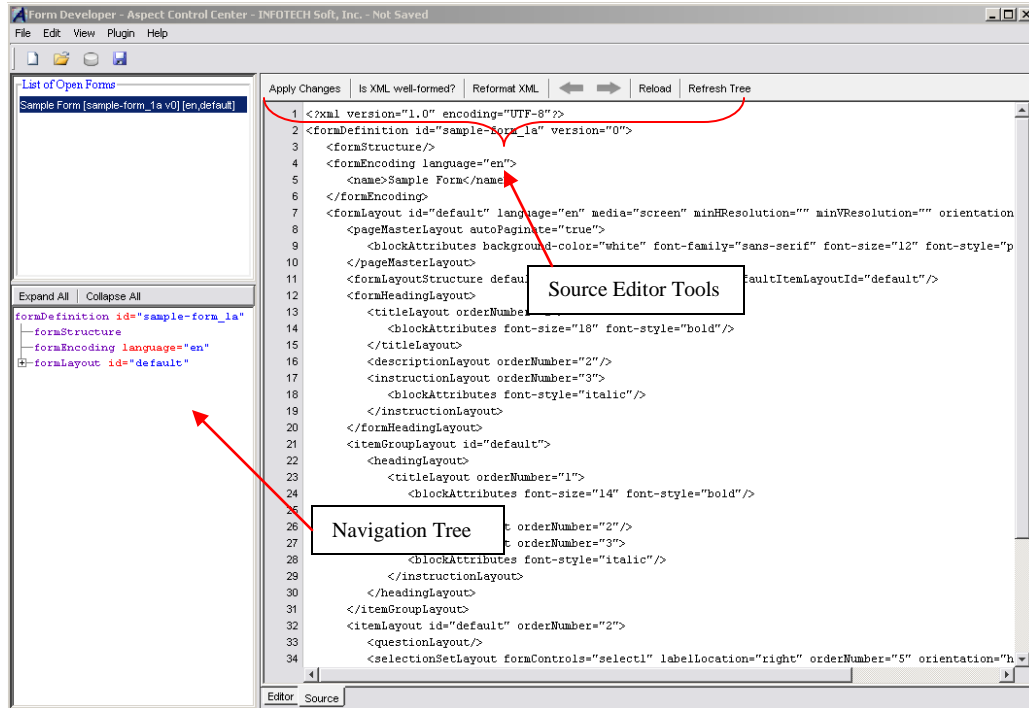


Figure 5 Form Developer – Source Editor

Open Form Definition Action

Opening a form definition is as simple as creating one. Clicking *File* then *Open*, or clicking the *Open Icon* on the toolbar will cause an Open File dialog to popup. From there you will be able to navigate through directories and select previously saved form definitions from XML files. Since the name of the file might not describe the form it contains, the open file dialog is equipped with a preview area that lists a brief description of the form (name, id, version, description, languages, and layouts) as shown in Figure 6.

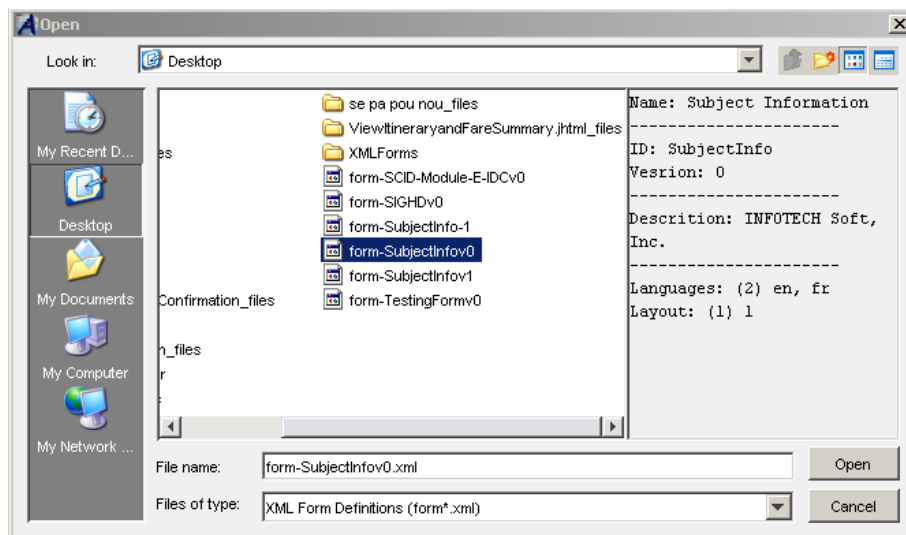



Figure 6 File Dialog

Load Form Definition Action

Loading a form is very similar to opening one. The only difference is that when opening a form, you select a file that was previously saved on your local machine whereas loading a form retrieves a currently stored form from the system's database. Clicking *File* then *Load*, or clicking the *Load Icon* on the toolbar gives you access to the *Form Chooser* dialog (see Figure 7). You can enable the display of a brief description of the form definition by checking the details checkbox.

 Before loading a form, the system will prompt you for a username and password if you are not logged in. You must have the appropriate permissions to be able to retrieve a form.

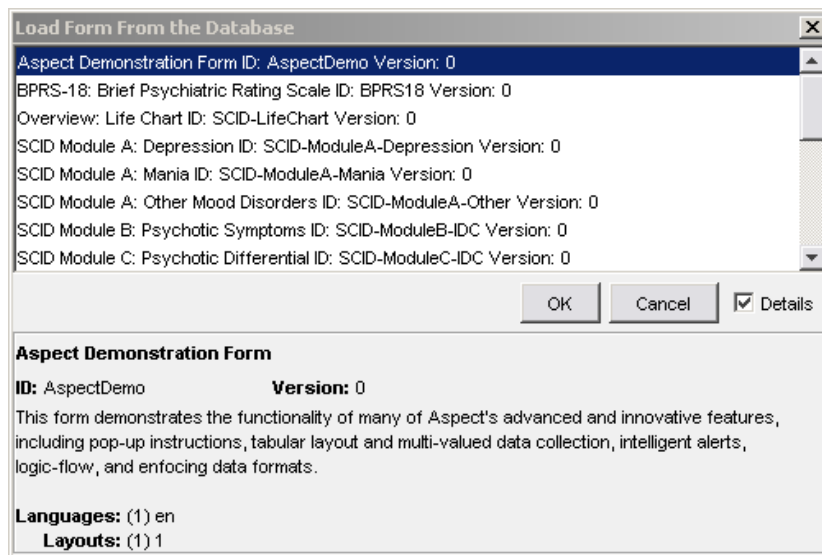


Figure 7 Form Chooser Dialog

Close & Save Form Definition Action

If you are done working with a given form definition, you can close it to remove the form from the Form Developer user interface by clicking *File* then *Close*. If the form was modified since the last time it was saved, you will be prompted to save before closing. You can choose not to save or to cancel the closing process. If you opt to save, keep in mind that the name of the file is computer generated; all you have control over is the selection of the directory where the file will be saved. The file name scheme is: [form-*'form id'* v *'version#'* .xml].

Save As Form Definition Action

After applying modifications to a previously saved form, you can opt to save in a different file; this can be done by saving as. If you choose to save in the same directory as the previously linked file and neither the id nor the version of the form has changed, the existing file will be overridden; otherwise, a new file will be linked to the form definition.

Enabling/Disabling Auto-Refresh Action

Auto-refresh causes an automatic update of the graphical representation of the form upon user interaction. For a long form, this feature can render the system slow since the longer the form, the longer the refresh time. You can disable or enable the auto-refresh feature under the *View* menu or by hitting *F4*. When auto-refresh is disabled, you can manually refresh by clicking on *Refresh* under the *View* menu or by hitting *F5*.

Form Structure Navigation

Other than accessing the different item groups and items of a form (as described in *Create Form Definition Action*), the *Form Structure Navigation* also gives you the ability to easily modify a form structure, and expand and collapse the tree navigation.

Collapse and Expand Tree Action

A tree can be very long depending on the number of item and item groups. The *'Expand All'* and *'Collapse All'* buttons enable you to respectively expand and collapse all nodes of the tree navigation.

Modify Form Structure Action

By clicking on the *'Modify'* button, a dialog that enables you to modify the structure of a form appears (see Figure 9). Through this *Structure Editor* dialog, you can add or remove item groups and items to and from various locations of the form. Depending on the node selected from the tree on the left, you have the option of inserting item groups and items before, after, or in the node in question. Child item groups and items are always placed after the last child of the selected node. Moving item groups and items after insertion is made easy; you can drag, copy (Ctrl + C), cut (Ctrl + X), and paste (Ctrl + V) the nodes of the tree. You can also move the node one position up or down by using the arrow buttons on the right side. The *'Expand All'* and *'Collapse All'* enables you to expand or collapse the tree for easier navigation.

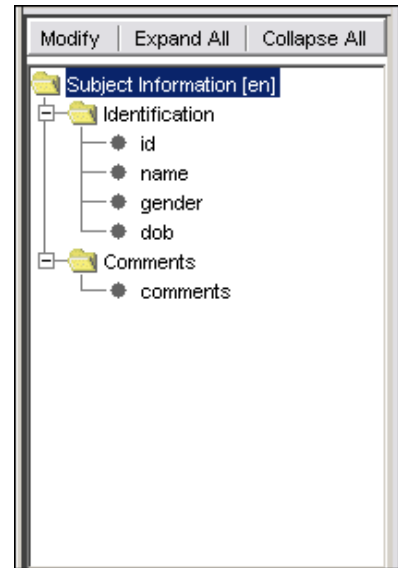


Figure 8 Form Structure Navigation

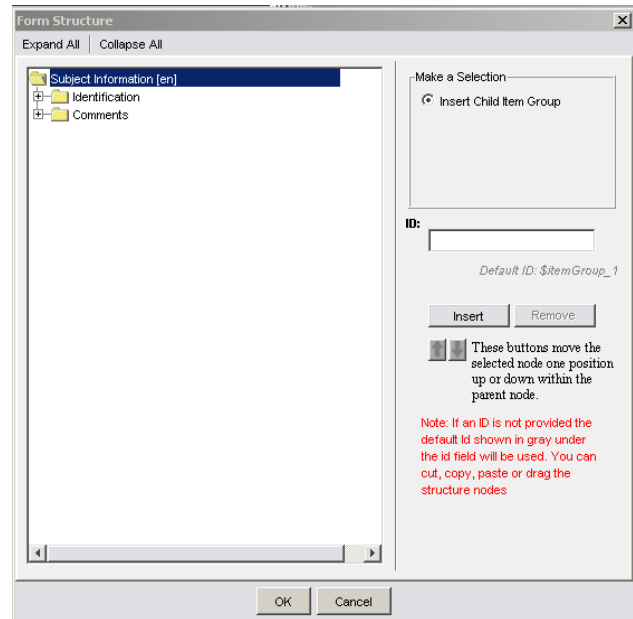


Figure 9 Form Structure Editor Dialog

Form Code List View

Below the *Form Structure Navigation* lies the layout and code list tabs. Selecting the Code List causes the *Code List View* (Figure 10) to show where you can view, modify, and delete existing code list or add new ones through the *Code List* dialog shown in figure 11a. A code list contains three attributes and a list of codes; out of the attributes, only the id is required. When editing or adding a code list, the code list id and code ids are required. Each code of a code list also has three attributes: an id which uniquely identifies the code, a text, which is what the form displays, and a description. The codes can be edited or deleted simply by selecting one of them in the list of codes and respectively clicking on the update or delete buttons. To add a code, fill in the code's attributes and click the add button. Just like the code list, only an id is required. Figure 11b shows how a code list from figure 11a is displayed in a form.



Figure 10 Code List View

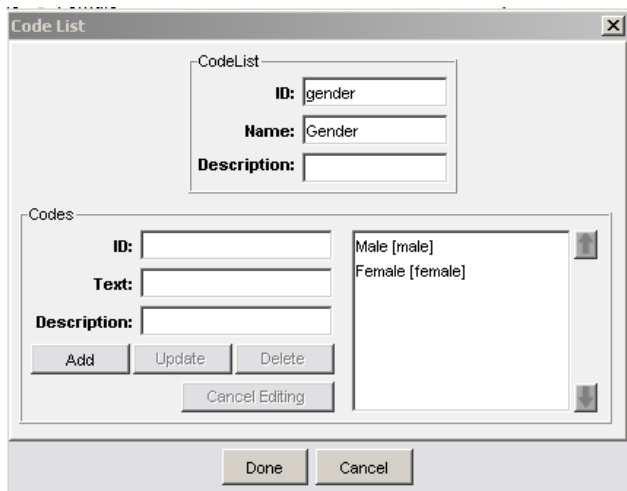


Figure 11a Code List Dialog

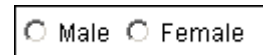


Figure 11b Graphical Representation of Code List

Form Layout View

The other tab below the *Form Structure View* leads to the *Form Layout View* (Figure 12) which at this point only displays the layout components of the form.



For this version of the form developer, the layout can only be edited through the Source Editor.



Figure 12 Form Layout View

Attribute Editor

Through the Attribute Editor, you can modify the form's attributes, item groups' attributes, and items' attributes.

Form Attribute Editor

Presently this attribute editor allows you to manipulate the form structure's attributes, the form encoding, and the layout and page settings as shown in Figure 13.

The editable form structure's attributes are the form id and the form version.

The form encoding section enables you to add or remove languages through the *Language Editor* dialog (see Figure 14b) that appears after selecting the language attribute by clicking on the edit column (right-hand column) then clicking on the button that shows up. The language edit column also enables you to set the language in which the form is displayed. Changing the language will cause the encoding section of all the attribute editors to be populated with text from the selected language. The name, description, and instruction attributes are texts that show in the form depending on the layout selected. The navigation name, if set, is the text that identifies the form node from the Form Navigation.

Through the Layout settings, you can set the default item layout, the default display format, and the default Item Group layout, which are values used by child item groups and child items if their layout attributes are not set.

Editing the page setting is more complex; each attribute is associated with a dialog accessible in the same manner as the *Language Editor* dialog.

The border attribute is editable through the *Border Editor* dialog through which borders can be added to the form. There are five types of borders: a box, or just a line on the left, top, right, and bottom. For each type, attributes (color, size, and style) can be edited. The border attributes for a border type can only be set if that type of border is added to the form. In order to add a border type, you must click on the corresponding button in the preview area. Figure 14a shows the *Border Editor* dialog. The font is editable through the Font Editor (Figure 14c), where a family, a size, and a style can be selected.

The background color and text color (foreground) can be set either by selecting one of the predefined colors of a drop-down selection (Figure 14d) that appears when you

| Structure | |
|----------------------------|----------------------|
| Id | SubjectInfo |
| Version | 0 |
| <i>Clinical Alerts</i> | |
| Encoding | |
| Language | English ▾ ... |
| Name | Subject Information |
| Navigation Name | |
| Description | INFOTECH Soft,... |
| Instruction | |
| Layout | |
| <i>Layout Id</i> | |
| <i>Media</i> | |
| <i>Language</i> | |
| <i>Horizontal Res.</i> | |
| <i>Vertical Res.</i> | |
| <i>Orientation</i> | |
| Layout Settings | |
| Default Item La... | 1 |
| Default Display ... | <<none>> |
| Default Group L... | 1 |
| Page Settings | |
| Border | |
| Font | sans-serif,plain,12 |
| Background | |
| Foreground | |
| Position & Size | -1,-1,-1,-1(w=738... |
| Padding | -1,-1,-1,-1 |
| Margin | -1,-1,-1,-1 |
| <i>Auto Pagnate</i> | |
| <i>Horiz. Scroll</i> | |
| <i>Vert. Scroll</i> | |

Figure 13 Form Attribute Editor

click in the edit column or by using the *Color Editor* dialog (Figure 14e) to create custom colors.

The *Position & Size Editor* dialog shown in Figure 14f allows you to set the size of the form. If the width or height is set to -1, the width or height of the form will be dictated by the width or height of its items and item groups. If both the left and right attribute are set to a value other than -1, the width of the form will be right – left; the top and bottom attributes affect the height similarly.

The margin and padding are set respectively through the *Margin* and *Padding Editors* (Figures 14g and 14h, respectively). The margin is the space between the form and its borders, whereas the padding is the space before the border of a form.

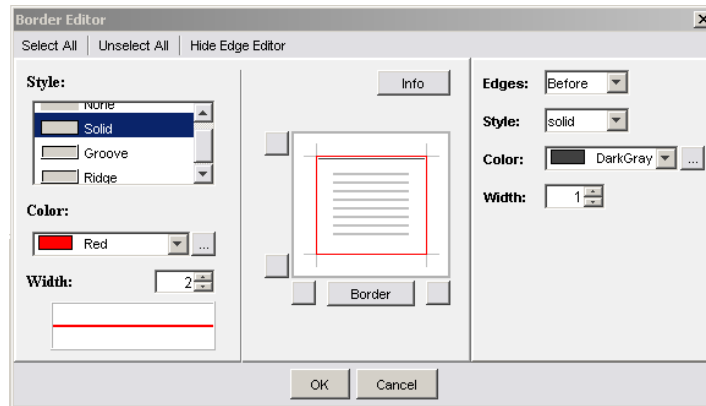


Figure 14a Border Editor Dialog

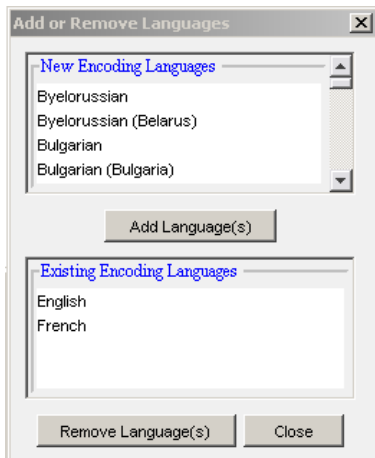


Figure 14b Language Editor Dialog

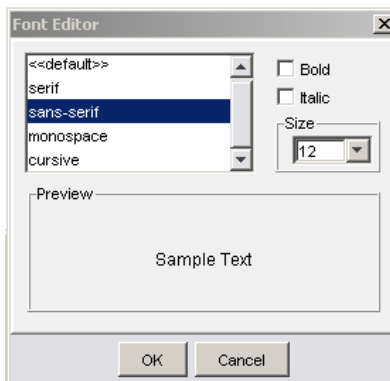


Figure 14c Font Editor Dialog

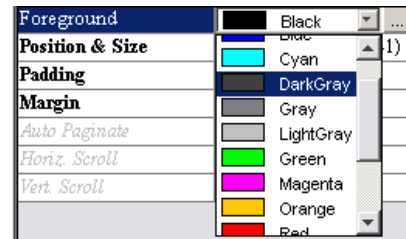


Figure 14d Drop-Down Color Selection

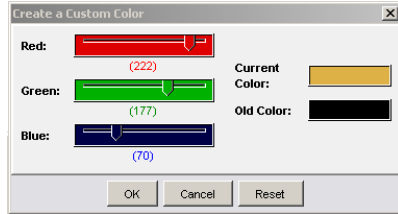


Figure 14e Color Editor Dialog

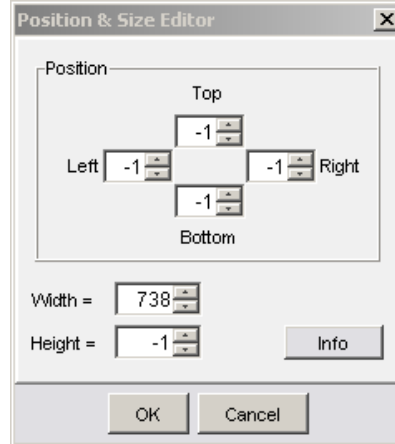


Figure 14f Position & Size Editor Dialog

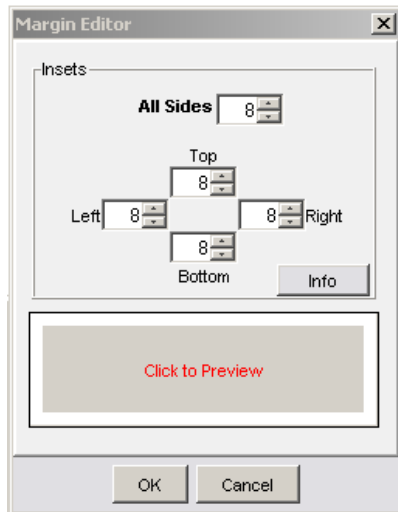


Figure 14g Margin Editor Dialog

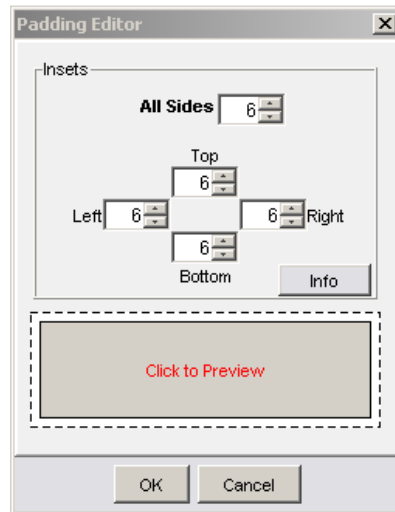


Figure 14h Padding Editor Dialog

Item Group Attribute Editor

Similarly to the *Form Attribute Editor*, the *Item Group Attribute Editor* allows you to set the structure, the encoding, and the layout. The editable attributes of the structure are the group id and the multi-value attribute. You can set the name, navigation name, description and instruction using the encoding part of the editor. Through the layout section, you can set the layout of the selected item group as well as the default layout and display format of child item groups and child items. Figure 15 shows the *Item Group Attribute Editor*.

| Structure | |
|------------------------|----------------|
| Group Id | demo |
| Multi-valued | false |
| <i>Clinical Alerts</i> | |
| Encoding | |
| Group Id | demo |
| Name | Identification |
| Navigation Name | |
| Description | |
| Instruction | |
| Layout | |
| Group Id | demo |
| Group Layout | 1 |
| Occurrence Lay... | <<none>> |
| Default Item La... | <<none>> |
| Default Display ... | <<none>> |
| Default Child ... | <<none>> |

Figure 15 Item Group Attribute Editor

Item Attribute Editor

The structure section of the *Item Attribute Editor* (Figure 16) has multiple attributes: an id which uniquely identifies the item within its parent, a form control attribute that is dictating how user input is collected for that item, a data type attribute which sets the data type of the input, a data mask attribute which puts restriction on the data type, and a code list attribute which selects the code list to use if select or select1 is selected as form control. Through the encoding section, the name, navigation name, question, description, and instruction can be set. The layout section enables you to select a layout and display format.

| Structure | |
|------------------------|----------|
| Item Id | comments |
| Form Control | textarea |
| Data Type | string |
| Data Mask | |
| Code List | <<none>> |
| <i>Output</i> | |
| <i>Clinical Alerts</i> | |
| Encoding | |
| Item Id | comments |
| Name | |
| Navigation Name | |
| Question | |
| Description | |
| Instruction | |
| Layout | |
| Item Id | comments |
| Item Layout Id | <<none>> |
| Display Format | <<none>> |

Figure 16 Item Attribute Editor