

Raintree Systems, Inc.



RAINTREE TEMPLATE EDITOR

Revised 8/21/2009

TEMPLATE EDITOR

If you're viewing this document online, you can click any of the topics below to jump directly to that section.

Template Editor

- Getting Started 4
 - Opening the Template Editor 4
 - Layout 4
 - Template Editor Modes 5
- Menu Editor 5
- Menu Bar 6
- Toolbar 7
- Editing Widgets 7
 - Selecting Widgets 7
 - Cut, Copy and Paste 7
 - Other 8
- Right-click Menu 8
- Object Search 8
- Template Window 10
- Template Window Properties 10

Widgets

- Widgets 11
 - Widget Types 12
 - Inserting Widgets 12
- Widget Properties 13
 - Name 13
 - Read Only 13
 - Tab Control 13
- Multiline Label 14
 - Properties 15
- Field 15
 - Properties 15
 - Examples of Mask Values 15
 - Linking Masks 17
- Line Widget 19
- Image 19
 - Adding an Image 19
 - Properties 19
- Image Button Widget 20

Properties	21
RTF Memo	21
Properties	21
Shrinker	21
Properties	22
Checkbox	22
Properties	22
Radio Buttons	22
Properties	23
Tab Control Widget	23
Using the Tabs Editor	23
Properties	25
List Box	25
Adding a list box	25
Using the list box	26
List Editor	23
Properties	28
Combo Box	28
Audio Widget	28
Panel	29
Properties	29
Repeater Widget	29
Properties	29
Link Button Widget	29
Properties	30
Edit Button Widget	30
<u>Designing Screens</u>	
Screen Design Guidelines	31
Organize widgets	31
Use the correct widgets	31
Widget Naming Conventions	31
Using Formscript Variables	32
EMR Examples	32
Notes	32
Including the Value in the Note	33
Note Collections	33
Overview	33
Defining Collections	34
Collection Items	35

TEMPLATE EDITOR

GETTING STARTED

Template Editor is a utility designed for creating and editing menus and input screens.

The Template Editor can be used in two **modes**, the Custom Screen Mode for editing custom screens, and the WCR Mode for editing menus. Depending on the mode, the Template Editor acts differently: some **widget properties** differ, and some **widgets** are disabled in the WCR Mode.

Note that most of the behaviour described in Template Editor help will apply to how the screen acts like in the real application. Widgets are actually abstract objects, based on which the real screen with real components is created in Raintree.

Template Editor is **shadowable** (i.e. several users are able to engage in the same Raintree session simultaneously). All users see the same **Template Window**, its appearance and the selected property in the left part of the screen. The changes made by one user will be reflected to the other user with a slight delay (approximately one second) in order not to overburden network traffic. Because of that it is recommended that users do not prompt changes at the very same moment, because synchronization problems might occur.

OPENING THE TEMPLATE EDITOR

There are several ways to open the Template Editor:

- If the screen or menu that you want to edit is currently open, you can press **Ctrl+K** to open it in the Template Editor. This requires the DEBUG E security right. Note that some menu items may be added through scripting, so they cannot be edited in the Template Editor.
- To edit custom input screens, you can open the screen from the **Ad Hoc Notes & Records** table.
- To edit a screen used for an EMR, you can also open the **EMR Templates Table**, select the template and then select **Edit input form**.
- To edit screens that belong to a custom module, open the **Module Configuration Panel**, select the module, select the Screens/Records tab, select the screen and click **Edit** to edit the RTM. Note that editing a screen may cause errors in the module if the respective scripts are not updated.
- To create a new screen or menu, you can also open Ride, open the **File** menu and select **New RTM**. The Template Editor opens automatically.

LAYOUT

At the top of the Template Editor window are the **menu bar** and the **toolbar**.

Below them on the is the property list. When a widget is selected, it displays the properties of that widget; otherwise, it displays the properties of the Template Window.

The largest part of the Template Editor window is the editor pane. This also contains the **Template Window** - the screen or menu that you are currently editing.

Between the Template Window and the property list is a vertical bar containing all widgets that you can add to the screen. To add a widget, click it on the object bar, and then click on the Template Window where you want to add the widget.

TEMPLATE EDITOR MODES

The Template Editor can be used in two modes:

- **Custom Screen Mode** for editing custom screens (.RTM files).
- **WCR Mode** for editing menus.

When editing menus, you should use the **Edit As Menu** button on the menu bar. Note that some menus are in the RTM format. In that case, the Template Editor opens in custom screen mode and the menu editor is not available.

Depending on the mode, the Template Editor acts differently: some widget properties differ and most widgets are disabled in the WCR mode.

MENU EDITOR

The Menu Editor is a tool for editing and creating menus quickly and easily. Open the WCR file you want to edit and click the **Edit As Menu** button on the top toolbar.



The Edit Menu window opens:

Edit menu

Menu title:
Batch Billing & Forms

Menu items:
Forms and Bills
Billing Labels / Charts
Tracers / Rebilling

Move up
Move down

Delete

Add

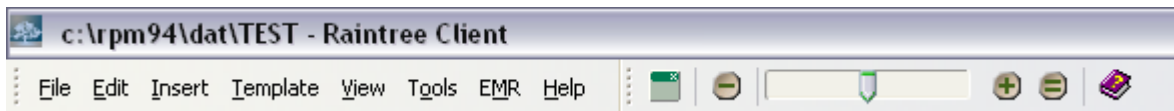
OK Cancel

If the file you are editing was initially not a menu, then the Menu Editor collects all text widgets in the file and lists them as menu items. Screen title will be used as the menu title. You can use the **Move Up** and **Move Down** buttons to change the order of the items. Use the Add and Delete buttons to add and delete menu items. Clicking the **OK** button will generate a new WCR screen from the created text rows, designed as a menu.

Note that if the & sign is used within a text row, then that sign will not appear in the created text widget, instead the character next to it is underlined (and used as the shortcut key for this menu option). To use the & sign as a menu element, insert /& in the Menu Editor. For example, inserting 'Element &1' will create the following widget: 'Element 1'. Insert 'Element /& 1' to create the widget 'Element & 1' with no character underlined.

MENU BAR

In the Template Editor menu bar, there are seven menus, zoom controls and the Help button.



- The **File** menu contains options for creating new screens and menus (see **Modes** for more information), saving and printing files, as well as an option for saving the **Data Dictionary**.
- The **Edit** menu contains Undo and Redo options, options for selecting, copying, pasting and deleting widgets, as well as several special options:
 - **Optimal Window Size**
 - **Optimal Text Label Size**
 - **X-Boxes to Checkboxes** - converts any **Fields** with the mask **X** to checkboxes
 - **Labels to Multiline Labels** - converts any text labels to **Multiline Labels**
 - **Copy Names to Clipboard** - copies the names of the currently selected widgets to clipboard, so that you can paste them somewhere
- The **Insert** menu contains shortcuts to insert the available widgets (duplicating the **Widget Bar**)
- The **Template** menu contains shortcuts to edit **Notes** and **Note Collections**, a shortcut to the **Image List** and two special options:
 - Hidden Fields
 - **Edit Prescript**
- The **View** menu contains the following options:
 - **Show Field Masks** - if set, the mask of each field is displayed inside the field
 - **Show Field Names** - if set, the name of each field is displayed inside the field
- The **Tools** menu contains options to edit the **Tab Order**
- The **EMR** menu contains options for the **EMR** Editor:
 - **No Editor** - hides the EMR editor.
 - **Editor** - opens the EMR editor as a separate pane. The keyboard shortcut for this is **Ctrl+Shift+E**.
 - **Editor and Insert Mode** - opens the RTF editor in Insert mode. This allows you to insert note tags simply by clicking on the respective element on the custom screen.
 - **Associated NARs** - this option allows you to define the .nar files associated with this screen. When an EMR template is defined, a .nar file is created with the same name. Note that even if you create a .nar file here, you still have to add an entry with the appropriate code in the **EMR Templates table**.
 - **Align Editor to Bottom/Align Editor to Right** - by default, the editor is aligned to the right side of the screen. This option allows you to toggle between aligning the EMR editor to the right/to the bottom of the Template Editor window.
- The **Help** menu allows you to view the Help.

TOOLBAR

Below the **Menu Bar** is a toolbar containing shortcuts for most common actions (note that depending on the **Mode** you are in or the widget you are focused on, some options may be grayed out):



- **New** - creates a new screen
- **Open** - lets you select a screen or menu file to open
- **Save** - saves the screen that you are editing
- **Optimal Window Size**
- **Optimal Text Label Size**
- **Copy Selected Objects**
- **Paste Objects**
- **Delete Selected Objects**
- **Edit As Menu**
- **X-Boxes to Checkboxes** - converts any **Fields** with the mask **X** to checkboxes
- **Set Tab Order Manually**
- **Image List**
- **Insert VIS Line**
- **Insert Checkbox and Label**
- **Insert Radio Button and Label**

EDITING WIDGETS

SELECTING WIDGETS

To **select a single widget**, click the widget.

To **select several widgets**, hold **Shift** and click the widgets. You can also hold **Shift** and drag the mouse to select widgets.

To select all widgets **inside a panel** (the **Panel** or **Tab Control** widget), hold **Alt** and drag the mouse. This selects all widgets inside the panel, but not the panel itself.

To **select all widgets** at once, press **Ctrl+A**. Alternatively, go to the **Edit** menu and click **Toggle Select All**.

CUT, COPY AND PASTE

To **copy** widgets, select the widget (you can also hold down **Shift** and select several widgets) and press **Ctrl+Shift+C**. Alternatively, select the widget(s), open the **Edit** menu and click **Copy Selected Widgets**.

An alternative is to select the widget(s) you want to copy, then press and hold **Ctrl** and drag the widgets to the desired location with the mouse. This makes it easy to copy a group of widgets with their positions.

To **paste** widgets, press **Ctrl+Shift+V**. Alternatively, go to the **Edit** menu and click **Paste Selected Widgets**.

To **delete** a widget, click on it and press **Ctrl+Delete**. You can also delete several widgets at once. To do that, hold down **Shift** while selecting the widgets, then press **Ctrl+Delete**. Alternatively, select the widget(s) you want to delete, right-click and select **Delete Selected Widgets** in the pop-up menu that is opened.

OTHER

To **resize a widget**, position the mouse cursor over the right side/at the bottom of the widget until the cursor turns into a two-point arrow. Hold down the left mouse button and drag. Note that various widgets may have size constraints. For example, the height of a Text Widget can only be 1.

To **move** a widget, drag it with the mouse to the desired position.

To **assign** widget(s) to a Tab Control, Panel or Repeater widget, select the widget(s) and click the **middle** mouse button (or the scroll wheel) on the Tab Control you want to assign the widget(s) to.

To **undo** recent changes, press **Ctrl+Z** or select **Undo** from the **Edit** menu. You can undo up to 10 most recent steps.

To **redo** the changes, press **Ctrl+Shift+Z** or select **Redo** from the **Edit** menu.

RIGHT-CLICK MENU

This menu appears when you right-click on the screen in the **Template Window**. Note that depending on where you click, some of the options may not be available.

- **Copy Selected Widgets** - copies the selected widgets to clipboard.
- **Paste Objects** - pastes any widgets in the clipboard.
- **Delete Selected Widgets** - deletes the selected widgets.
- **Create Note Group from Selected** - creates a note **collection** from the selected widgets. You can create a new collection or add the widgets to an existing collection.
- **Create Widget Group from Selected** - creates a widget **collection** from the selected widgets. This is similar to the previous option, with the difference that the collection type is defaulted to SETW.
- **Name/Change Value of Selected Widgets** - names the selected widgets. The difference with the next option is that this option allows you to use the adjacent labels for widget names. You have to specify a prefix and optionally specify where the labels are located.

You can also use this option to change the values of radio buttons, etc.

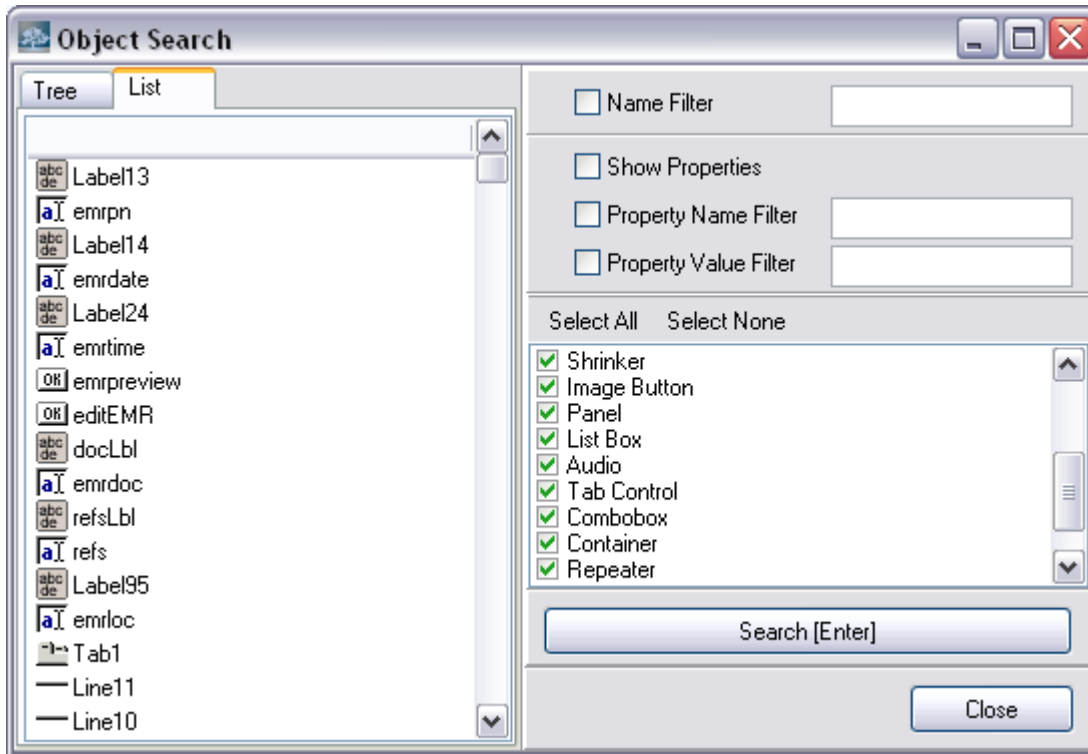
- **Name Widget Sequence** - names the selected widgets in sequence. You have to specify the prefix for the name.

For example, if you specify the prefix as "foo", the widgets will be named foo1, foo2, foo3, etc.

- **Belongs To Groups** - displays a list of all **collections** on this screen. Collections where the selected widget belongs are indicated by a check mark.
- **Insert** - displays a list of widgets to insert.
- **VIS Line** - inserts a **VIS Line**.
- **Help** - displays help about the selected widget.

OBJECT SEARCH

When editing large screens, it will become hard to find the location of a particular widget on the screen. You can press **Ctrl+F** to open the Object Search window and search for widgets.



On the left are two tabs - **Tree** and **List**, displaying all widgets that match the filters you define. When you click on a widget in the list, it is automatically focused on the screen, so that you can close the search window and start editing it.

The **Tree** tab shows all widgets as a tree, meaning that all widgets inside a panel or tab control are contained in a sub-list inside it. To expand that list, you have to click on the panel name. The **List** tab shows all widgets as a single list. Note that you cannot sort the list by widget name - use the filters to narrow down the list instead.

On the right are various filters you can apply. Note that if you change a filter, you must press **Enter** to apply them - they are not applied automatically.

- The **Name Filter** is useful when you remember the widget's name or a part of it. Enter the name in the field. The checkbox is automatically selected. You can clear the checkbox to remove the filter without losing the value you entered. For example, entering "label" (without the quotes) will display all widgets whose name contains "label".
- **Show Properties** toggles displaying the widget properties in the list. Note that if you select either of the following two filters, this is automatically selected.
- **Property Name Filter** is used to search for common properties. An example would be the **Read Only** property - enter ReadOnly to view all widgets that **can be** set as read-only.
- **Property Value Filter** searches for a particular property value. It is mostly used together with the Property Name filter. For example, to find all widgets that **are** set to read-only, enter "ReadOnly" as the Name filter and "True" as the Value filter. This will display all widgets that have the ReadOnly property selected.
- Below the filter fields is a list of all widget types. The selected widgets are displayed in the list, so that to view only a particular type of widget, clear the checkboxes of all other widgets. Above the list are two buttons to select and clear all widgets at once.

There is a special widget type in the list - **Container**. This displays all widgets that can contain other widgets, meaning **Panel**, **Tab Control** and **Repeater**.

Below the filters are two buttons - **Search** to apply the current filters and **Close** to close the search screen.

TEMPLATE WINDOW

In short, the Template Window represents the screen that is currently being edited. You can build the template by placing and arranging different **widgets** on the Template Window. (There is a vertical widget bar to the left of the Preview area.) Keep in mind that the Template Window **is not** an exact representation of the real screen. Widgets are actually abstract objects, based on which the real screen with real components is created in the main application.

The grid in the Template Window consists of points, each a certain distance from the other, that are used to control the size and placement of the widgets. One grid point is equivalent to one character on the screen.

To access the **properties of the Template Window**, click anywhere on the Template Window where there are no widgets. The Window property list will be displayed on the left part of the screen.

There is a vertical widget bar between the property list and the Template Window, which contains a button for each widget type.

TEMPLATE WINDOW PROPERTIES

To access the properties of the **Template Window**, click anywhere on the Template Window. The 'Window' property list will be displayed on the left part of the Template Editor.

The following properties apply to the Template Window:

- **Caption** - specifies the screen title
- **CHeight** and **CWidth** - control the height and width of the Template Window
- **CLeft** and **CTop** - control the screen position in Raintree
- **Left** and **Top** - control the screen position in the Preview Area
- **Global Refresh Keys** - defines the keys that issue an RSL refresh when pressed
- **Hide Default Function Buttons** - hides the default function buttons normally displayed on the **Menu Bar**
- **Menu** - tells Raintree to handle this screen as a menu
- **RefreshTimeout** - if set to something else than 0 then an RSL refresh is triggered after the indicated amount of seconds.
- **Use Tab Order** - indicates whether to look at the **Tab Order** property of the widgets when deciding where to focus next

WIDGETS

WIDGETS

There is a vertical widget bar between the property list and the **Template Window**. It contains a button for each widget. Below is a list of all buttons and the corresponding widgets:



Multiline Label



Field



Horizontal Line



Vertical Line



Text Label



Image



RTF Memo



Shrinker



Checkbox



Radio Button



Image Button



Tab Control



List Box



Combo Box



Audio recording device



Panel



Repeater



Link Button



Edit Button

WIDGET TYPES

Widgets can be divided into three categories according to their type:

- Data elements - allow entering data that is stored in the database
- Interactive UI elements - enable user interaction
- Static UI elements - view only

The **data elements** are:

- **Field**
- **Image**
- **RTF Memo**
- **Checkbox**
- **Radio Button**
- **List box**
- **Combo box**
- **Audio Widget**
- **Link button**
- **Edit button**

The **interactive UI elements** are:

- **Shrinker**
- **Image Button**
- **Tab Control**
- **Panel**
- **Repeater**

The **static UI elements** are:

- **Label** (note that labels can also be made dynamic, thus belonging to interactive elements)
- **Lines**

INSERTING WIDGETS

Click a widget button, and you will see that it stays switched on. Click again anywhere within the Preview Area, and the widget will be created in that spot. Alternatively, right-click anywhere in the Preview Area and select the widget from a list.

To create several widgets of the same type at once, hold down **Shift** before clicking on the widget button. Then you can click as many times as you want in the Preview Area, and each time a copy of the widget will be created. When you release **Shift**, the button selection will be canceled.

To cancel a button selection without creating a widget, click a different button or press **Esc**.

To speed up creating checkboxes and radio buttons, there are two extra options in the Insert menu (and on the toolbar):



Checkbox and Label



Radio Button and Label

Since usually each checkbox and radio button has a corresponding label, this saves you the need to create the two widgets separately. You can just click the respective button, click in the Template Window and enter the label text in the message box that appears.

WIDGET PROPERTIES

The left part of the Template Editor screen is occupied by the Property List.

When you create or select a widget in the Template Window, all the properties of that widget are displayed in the list. Depending on the type of the property's value, entries in the list can have check boxes, dropdown boxes, fields, etc next to them. The window itself has certain properties as well - these are displayed in the list when no widgets are selected. (See **Template Window Properties** for more information).

Note that properties differ depending on the **mode** you are in.

NAME

This property contains the identifying name for the widget. For widgets that contain user input this is usually also the field name of the database field.

Note that for **Radio Buttons** this name is used only for identifying the radio button on the screen. The **Group Name** is used to save the value instead.

READ ONLY

If this property is selected, the user is not able to change the widget contents.

This property applies to: **Field, Edit Button, Checkbox, Radio Button** and **RTF Memo**.

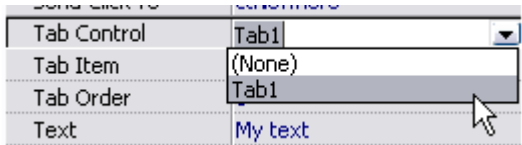
TAB CONTROL

This property specifies the **Panel, Tab Control** or **Repeater** widget under which this widget belongs.

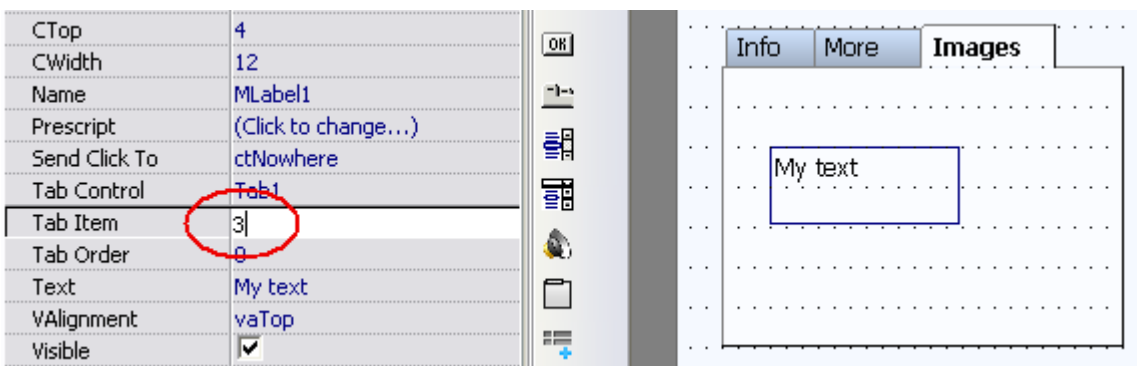
To link a widget to a Tab Control:

1. Add a widget, e.g. a field, a text box, an image - you can link any Template Editor object, including other Panel, Tab Control and Repeater widgets. In this example, a label was added.

- Click on the widget.
- By default, the **Tab Control** property value is blank, meaning that the widget is always visible, no matter which tab is selected.
- Open the dropdown list with property values. If you created just one set of tabs (as in this example - the three tabs belonged to one Tab Control), then Tab1 appears in the dropdown list. (If you change the name of the Tab Control, then that name appears instead). Change the value to Tab1. If you created more than one set of tabs, select the desired set from the list.



- Set the **Tab Item** property value equal to the sequence number of the tab that you want to link the widget to. For example, in the illustration the widget is linked to the last tab, and there are 3 tabs, so the number is 3.



The label widget will now be visible only when the user clicks on the Images tab.

Note: you can set the Tab Control property by selecting the widget(s) and then clicking the **middle** mouse button (or the scroll wheel) on the Tab Control you want to assign the widget(s) to.

MULTILINE LABEL

This widget can be used for displaying text on the screen. The displayed text can not be edited. This is used for the labels next to fields, radio buttons, etc. Note that the **RTF Memo** can also be displayed as read-only.

To insert a Multiline Label:

- Click the **Insert Multiline Label** button.
- Click anywhere in the Preview Area. The label will appear.
- Click and drag the label edges to extend it.
- Click on the **Text** property box and type or paste the contents of the field. The text is wrapped automatically. Note that if more text is inserted than fits in the Label area, then the label should be extended manually.
- By default, the widget appearance is set to equal the Raintree default font and window color.
- To change the default background, clear the **Default Background** checkbox, then click the **Background Color** property and select the desired color from the dropdown list.
- To change the default font, uncheck the **Default Font** checkbox. The **Font Name**, **Font Size**, and **Font Style** properties appear. Click the **Font Name** property and select the desired font from the dropdown list.
- Click the **Font Size** property and type the desired font size.
- Select/clear the Bold, Italic, Strike Out and Underline checkboxes to switch those font style options on or off.

10. To change the vertical alignment of text, click on the **VAlignment** property box and select the preferred option.

PROPERTIES

The **Multiline Label** has several properties for changing the label appearance - you can edit the font, color and alignment of the text.

The **Text** property specifies the text displayed on the label. You can use `\n` to insert a line break, for example `Line1\nLine2` will be displayed as

Line 1

Line 2

An important property is **Send Click To**. It is used when the label is next to a checkbox or radio button. When the user clicks on the label, then if this property is set, the click is set to the widget on the left or right of the label. The key advantage is that the user can click the label to select/clear an option, instead of having to point at the small checkbox itself, meaning that it will be much easier to use.

FIELD



The Field widget is a single line edit box where the user can enter text. The text inserted will be put into the record handled by RSL or PHP under the same name as the Name property specifies.

PROPERTIES

One of the most important properties of the **Field** widget is **Mask**. Using the mask, you can turn the field into a date field, specify a format for the value or link it to a table. See **Examples of Mask Values** for details.

The **Auto Tab** property can be used to speed up entering data. When this is set, then upon entering the field, the linked table (or in case of a date field, the calendar) opens. The user can then select the desired value without having to press **Tab** to open the table first. However, often it is easier to type the date or the desired code manually, instead of having to select it with the arrow keys.

The Field widget also has several properties that can be used to trigger a RSL refresh. The RSL refresh can populate the field with some predefined data, update some other element on the screen (for example when the field is used as the filter for an EMR list), etc. You can trigger the refresh when the user enters the field, changes the field content or exits the field.

EXAMPLES OF MASK VALUES

Mask format	The field value in the input screen
<code>_</code> (underscore)	Any symbol (characters, signs, digits) can be inserted in the field position where the '_' sign appears in the mask.
<code>----</code>	Creates a field where maximum 4 symbols can be entered.
<code>14_</code>	Creates a field where maximum 14 symbols can be entered.

Mask format

The field value in the input screen

Note that this option can be used if only one type of characters are included in the mask, in case of a more complex mask the first option (lining up the characters) should be used.

n	Only numbers are accepted in the field position where 'n' appears in the mask.
10n	Creates a 10-digit field, where maximum 10 numbers can be entered.
nnnnnnnnnn	The same as 10n, i.e. creates a field, where a certain number of digits can be inserted, the number being equal to the number of 'n'-s in the mask.
(nnn) nn- nnn	It is possible to include brackets, dashes, etc. in the mask.
d	d-format is the date format, equivalent to the nn- nn-nn format. The current date can be automatically inserted in this field by typing 0 in the first position of the field and pressing 'Enter'.
b	The b-format creates a blocked field. 11b, for example, creates a field with 11 positions, which cannot be changed.
c	The field position corresponding to 'c' in the mask forces text into uppercase, but allows also numbers and signs to be inserted For example, both '4c' and 'cccc' create a field where maximum 4 symbols can be inserted. All the letters that are inserted will be capitalized, and it is possible to insert signs and/or numbers.
u	The field position corresponding to 'u' in the mask accepts characters only, the letters inserted are capitalized. For example, '6u' and 'uuuuuu' create a field that accepts maximum 6 characters, neither numbers nor signs are accepted. The mask 'u____' (four lowered dashes) creates a field with 5 positions with the first letter capitalized automatically.
?	A question mark in the beginning of a mask creates a password field: the value will be formatted in the exact same way, but only the * signs are displayed on the input screen. For example "? _____" (five lowered dashes) creates a password field that is six characters long. Note that if you specify the mask as "??????", it will be converted to "? _____" automatically.
x	x-format creates a check box. A mask can contain only one x (things like xx or xxxxx should not be included in the mask). It is recommended to use checkboxes instead (see Checkbox Widget)
Y	y-format creates a yes/no field, i.e. only 'Y' or 'N' can be inserted in the field.
i	i-format creates an invisible field.
iiii	creates an invisible field with 4 positions.
.	Decimal point, used together with the '#' sign. Determines the location of the decimal separator in the field.
#####.##	Creates a 7-digit field where real numbers can be inserted and where the centered point (.) determines the location of the decimal separator in the field. If the value inserted in the field has the same number of digits as there are the '#' signs in the mask, then all the digits will appear in the input screen. If the value has less digits, the remaining '#' signs are replaced with '0'-s. If the number of inserted digits exceeds the number of '#' signs, then the last part of the value is cut off.
\$	the same as #####.##
_###.##	Creates a 7-digit field similar to #####.##, the difference being that also letters are accepted.

Mask format

The field value in the input screen

An additional position is added in the beginning of the field, which accepts a '+' or '-' sign.

r Creates a field for inserting real numbers. Only the 'r' characters can be included in the mask (for example, no decimal points can be included).

For example, the mask 'rrrr' would create a 4-digit field for real numbers. If the value inserted in the field has less digits than there are the 'r' signs, no '0'-s will be forced upon the remaining 'r'-s. If the value '45' is inserted in the field, then nothing will be displayed in the position of the two extra 'r'-s in the mask.

_r Creates a field similar to r, the difference being that also letters are accepted.

nn:nna Creates a current time field.

The current time can be automatically inserted by typing 0 in the first position of the field and pressing enter.

***** To create a link, a Text Label should be created and a mask added to it. To do that, asterisk (*) should be entered in the beginning of the mask property value. The masks affect the Text Label only if the asterisk is added in front of the mask.
If nothing except the asterisk is included in the mask, the Text Label will be turned into a link and the refresh function is activated on that link in the input screen. (See also the **RefreshStyle** property).

If a screen name is inserted next to the asterisk, the Text Label will be linked to that screen.

Other symbols If a symbol not defined as a format specifier is included in the mask, the symbol will appear blocked in the final input screen.

For example, the mask for the Date of Birth could be nn-nn-nnnn, in which case the '-' signs will appear in the input screen and cannot be altered.

linking Linking applies to **Field** masks only. You can link the field to a specific table. See **Linking Masks** for details.

LINKING MASKS

You can link a field to a specific table in Raintree. Use an ampersand & followed by the table code as the **mask**. The user can then select a record from that table as the field value. In addition to the code, you can also use descriptions, for example `&DOCTO:desc1:20` links the field to the provider table and inserts the first description of that table into the field (up to 20 characters). Below is a complete reference of all table masks:

Mask	Table	desc1	desc2	desc3	desc4
&ADJ	Adjustment	Description			
&APTYP	Appointment type	Description			
&APVIE	Appointment view	Description			
&BC	Billing code	Description			
&BGRP	Billing group	Description			
&CHARG	Charge	Code + description + billed amount	RVS/CPT code	Description	Code + RVS/CPT code + description + billed amount

Mask	Table	desc1	desc2	desc3	desc4
&CLINI	Clinic/Ledger	Name	Name	Name	
&DB	Database	Description			
&DIAG	Diagnosis	ICD9 code + description	ICD9 code	Description	
&DOCTL	Provider	First + last name + phone	License 1	License 2	License 3
&DOCTO	Provider	First + last name	Address	City + state + ZIP	Phone
&EMP	Employer	Name	Address (both lines)	City + state + ZIP	Phone
&EMRCA	EMR category	Description	Code + description		
&EMRLI	EMR Lists				
&EMRTE	EMR template	Category	Description		
&FC	Financial class	Description			
&FORMS	Formscripts	Script name			
&FORM	Same as FORMS.				
&INS	Insurance	Name	Address	City + state + ZIP	Phone
&INSUR	Patient Insurance	Program code	Insurance code	Name	From date
&INV	Inventory	Description			
&INVSU	Inventory supplier	Name	Address line 1	Address line 2	
&LETTE	Letter	Description	Category		
&LOC	Location	Name	Address	City + state + ZIP	
&LSUP	Ledger supplier	Line 1	Line 2		
&MA	Appointment flag	Description	Code + description		
&MCAT	Ad Hoc Notes & Records	Description			
&NOTE	Ledger note	Description line 1	Description line 2		
&OFAC	Other facilities	Name	Address	City + state + ZIP	Code
&PATIE	Patient	PN + first + last name	Address	City + state + ZIP	Phone
&PAY	Payment	Description			
&PDIAG	Patient diagnosis	ICD9 code	Diagnosis code	Date	Provider
&PROGR	Program	Description			
&REASO	Reason code	Description			

Mask	Table	desc1	desc2	desc3	desc4
&REFS	Referral source	Last + first name + initial	Address	City + state + ZIP	Phone
&SEC	Users	First + last name	ID + login name	ID + description	ID + email
&SPEC	Speciality	Description			
&STAT	Patient status	Description			
&TRANS	Transfer	Description			
&ZIP	ZIP code	City	County	State	Code

LINE WIDGET

This widget is used for displaying lines on the screen. The **Vertical** property controls whether the line is vertical or horizontal. If the Vertical property is not selected (meaning that the line is horizontal), you can use the **Vertical Alignment** property to control how the line is aligned.

Both



Horizontal Line and



Vertical Line create a line widget, the difference is only whether the Vertical property is selected.

Note: if you want to draw a box around some widgets, then the **Panel Widget** might be a better option - in addition to visually separating the widgets, it also groups widgets logically, so that if the panel is moved or hidden, all widgets inside the panel are moved or hidden as well.

IMAGE



The Image Widget enables you to store an image or a group of images (also known as an image book) in the custom record.

ADDING AN IMAGE

1. Click the **Image** button, then click on the area where you want the image area to appear on the final screen.
2. To resize the frame area, click and drag the edges of the frame.
3. Edit the properties if necessary.

When the end user clicks the image frame, the Image Upload Manager will open and the user can insert images from file, scan images or copy and paste an image. See **Image Upload Manager** help for details.

PROPERTIES

When the **Image** widget is used to allow the user to upload an image, there are two properties that can be used to customize the widget.

The **Empty Text** property specifies the text that is displayed when no image has been loaded. By default this is set to "Click to Add Image...", but you can edit this as needed. For example when the user needs to upload a scanned copy of the patient's insurance card, the Empty Text might be "Scan of ins. card".

The **Use Background Transfer** property allows uploading the image as a background process, so that the user can continue to work while the image is being uploaded. When this is not set, Raintree waits until the image is uploaded before allowing the user to continue.

IMAGE BUTTON WIDGET



Image button is used for showing a highly customizable button on the screen. Pressing the button issues an RSL refresh. You can create a **Save** button to save the screen or a **Print** button for printing the data.

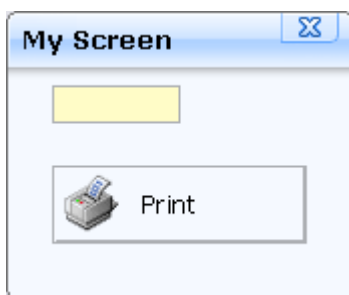
When the **Drawable** property is set, the button becomes an image widget. Now, when the user clicks the image widget, the **Freehand Drawing Pad** opens where the user can draw with the mouse or a handwriting input device. This is often used to allow users to add signatures to the screen. As the button can also have a background then alternatively this control can be used to show a medical image onto what the doctor can add free hand notes.

There are two ways to add an image:

- If you want to use the same image on multiple buttons, add the image in the **Image List** and specify its index in the **Image Index** property. Note that you can use the Image List even if you use the image only once.
- If you want to use the image only once, you can use the **Image** property.

To add an image button,

1. Click the **Insert Image** Button.
2. Click on the Template Window to place the button. To move the button, drag it to the desired location.
3. The Image Button Properties will appear on the left.
4. Type the text that should appear on the button in the **Text** field.
5. Specify an image using either the Image or the Image Index property.
6. Press **F10** when done.



Now, every time a user clicks the button, a RSL refresh is issued. To click the button, users can either click it with the mouse or navigate to it using the Arrow Keys and then press **Enter**.

Note that you can also assign a keyboard shortcut to an image button using the **Shortcut Key** property so that users can select the button by pressing the shortcut key.

PROPERTIES

The **Image Button** has many properties to customize its appearance and behavior.

One of the key properties is **Drawable** - if this is set, the image button turns into a freehand drawing widget, where you can draw using the mouse or a handheld input device. If an image has been specified on the button, it will be displayed as the background for the drawing.

You have two options to specify the image - you can browse for a specific image using the **Image** property or if the same image needs to be used on multiple buttons on the same screen, specifying the **Image Index** from the **Image List**.

There are several properties to control the button appearance - you can enter some text to be displayed, customize where the text should appear in relation to the image, etc. The **Shrunked Height** property is useful if the screen does not have much space available - it shrinks the image button, leaving some extra space at the top and bottom of the button.

If users are used to working with the mouse, you may want to create a save button for the screen. Users can then click this button to save the screen, instead of having to press **F10**. To do this, create the button and select the **Save Button** property.

You can also assign shortcut keys to buttons. Enter the character you want to use in the **Shortcut Key** property. Now users can focus the button by pressing **Alt** and the specified character. Note that if the screen contains no focusable fields, you can set the **Shortcuts Without Alt** property to enable focusing the button by pressing only the shortcut key.

RTF MEMO



RTF text field accepts and shows RTF content. It can be used as a multiline text field. You can change fonts, add files to it, etc. The text is stored as RTF, therefore the RTF Memo is not as fast as a simple field.

PROPERTIES

Because the **RTF Memo** is designed for entering rich text, there are no properties to customize its appearance. You can however adjust its behavior.

The **Category Filter** property lets you filter the list of EMR templates that is displayed when you press **Alt+L**. Enter a category code here and only templates in that category will be displayed.

If the memo contents generally will not be very long, but rich text formatting is needed, you can set the **Edit Bar** property to display a small edit bar on top of the RTF memo. It contains the basic formatting controls - font size and style, alignment, etc.

Similar to fields, there are three properties to trigger RSL refreshes when the user enters, exits, or changes the RTF memo.

SHRINKER



Shrinker is a nonvisual widget for hiding/collapsing horizontal areas on the screen. If **Visible** is set to false for this control,

then all widgets on the same height as the shrinker will be made invisible. Optionally, the area will be collapsed and the screen/tab will become shorter. This widget is useful for managing big screens that may have a lot of controls that the user should not always see.

Another alternative is to add the widgets to a **Panel** and show/hide the panel by toggling the Visible property of the panel.

PROPERTIES

The **Shrinker** has two properties to control how the screen behaves when the shrinker is hidden:

- First, the **Leave Original Height** property determines whether to collapse the area. If set, the widgets are replaced with a blank space; otherwise, the widgets below the shrinker are moved up.
- If that property is not set, you can use **No Window Resize** to prevent resizing the screen when the shrinker is hidden.

CHECKBOX



This widget is equivalent to a **Field** with two possible values - set or unset. The saved values are 'X' and '' respectively.

If a text label has been inserted to the left of the checkbox, then clicking the label is equivalent to clicking the checkbox itself.

Note that there is a shortcut to insert both a label and a checkbox together - just click the respective button



on the toolbar.

PROPERTIES

The **checkbox** has only one specific property - **Refresh Style**. This determines if selecting/clearing the checkbox will issue an RSL refresh.

RADIO BUTTONS

Radio buttons are used when the user needs to select only one option from a group. This works similar to a **Combo Box**.

A radio button by itself does not represent a screen control. The radio buttons belonging to one group are regarded as one widget. In a radio button group only one button can be selected. On the other hand, selecting none of the buttons is also allowed.

If a record is saved then instead of the usual "name->value" saving method the Group Name property is considered as the field name and the Value property of the selected radio button is used as the field value. For example, if you have two radio buttons belonging to group rsex and their values are male and female, then if the second radio button is selected, the widget value is saved as follows: `rsex=female`.

1. To insert a new radio button, click the radio button icon, then click on screen.
2. Repeat step 1 to create a new radio button in the same group.

3. To create a new group, insert a new radio button and change the Group Name. From now on, all radio buttons that are inserted belong to this new group by default. Note that if necessary, you can always change the group name.
4. Each radio button has a unique **value**. By default, this value is set equal to the sequence number of the radio button in its group. You can change this value if desired, simply click on the radio button and type the new value in the Value field. This value is saved as the widget value.

Note that there is a shortcut to insert both a label and a radio button together - just click the respective button



on the toolbar.

PROPERTIES

For radio buttons, the **Group Name** and **Value** properties are the most important ones. Group Name determines the radio group where the radio button belongs; Value determines the radio button value that is saved as the widget value.

Similar to checkboxes, radio buttons also have the **Refresh Style** property.

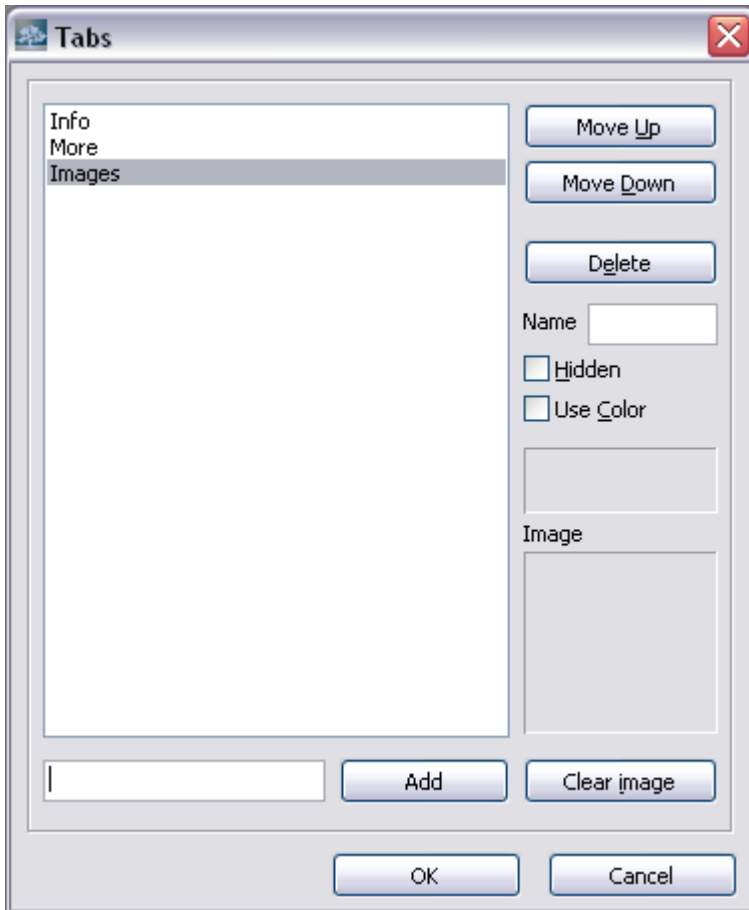
TAB CONTROL WIDGET

Tab control widget is similar to the **Panel** widget. The difference is that it has more than one page to hold different widgets. A widget can be linked to the tab control via **Tab Control** property. When linked, the **Tab Item** property of the widget will show the tab number onto which it is linked. If the user selects a tab, then widgets that are connected to the tab control and point to the clicked tab page will be displayed and others hidden.

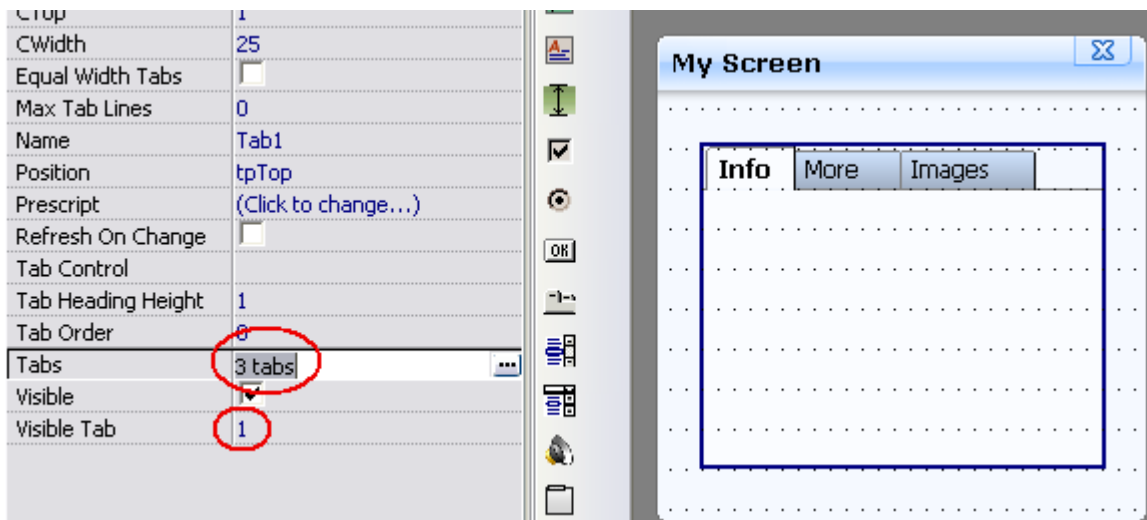
USING THE TABS EDITOR

Say you need to create a Tab Control Widget with three tabs. To do that:

1. Click the **Insert Tab Control** button, then click the Template Window. The first tab is inserted.
2. Click the **Tabs** property, then click the button with three dots that appears next to the property field.
3. The Tabs editor appears. Type the names of the tabs below each other. In this example we have 3 tabs, so we type 3 names.
4. Click on "Default", then click the edit box at the bottom of the screen and type "Info". Press **Enter**.
5. Type "More", press **Enter** to add.
6. Similarly, add the third tab "Images".



7. Click **OK** to save the changes.
8. The **Tabs** property now shows that there are 3 tabs and that the visible tab is 1.



The Tabs editor also allows you to customize individual tabs.

- To hide a tab, select it in the Tabs editor and check the **Hidden** box.
- To set the tab heading color, check the **Use Color** box, then click the field below it to select a color.
- To add an image to the tab heading, click the field below Image and select an image. This image will appear in the tab heading to the left of the text. To remove the image, click **Clear Image**.

PROPERTIES

For **Tab Control**, the most important property is **Tabs** that determines the number of tabs and their individual appearance. The **Visible** property determines which tab is focused when you open the screen.

Most properties specific to the Tab Control widget customize the appearance of the tab headers as a whole:

- **Equal Width Tabs** sets the width of all tab headers to that of the widest tab. This helps to create a sense of consistency.
- When there are many tabs, the tab control creates multiple lines of tab headers. **Max Tab Lines** can be used to limit the number of lines - if the limit is reached, the tab widths are decreased to fit them in the specified number of lines. By default this is set to 0, meaning that the number of lines is not limited. Note that creating more than two lines of tabs is not recommended - this makes it hard to find the correct tab.
- **Position** determines if the tab headers are at the top or at the bottom of the tab control.
- **Tab Heading Height** can increase the tab header height.

LIST BOX

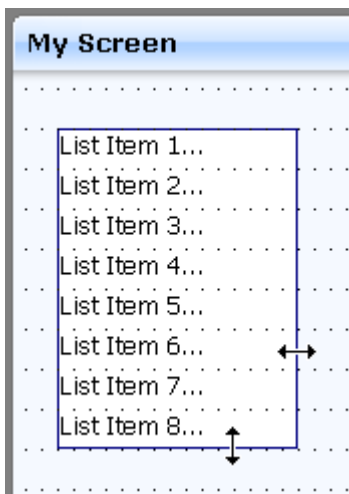


The List Box is a single selection list view that supports multiple columns, sorting etc. In custom record terms this is just one field with one value. Every row in a listview has two elements - the column values for displaying and a real value that is hidden from the user and is saved to the database.

One of the most important features of the list box is displaying an **EMR List**. To do that, specify the list name in the **EMR List** property.

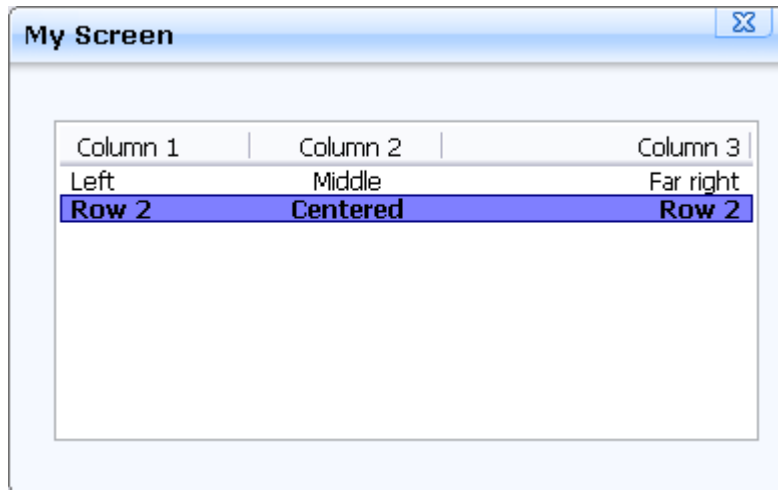
ADDING A LIST BOX

1. To insert a List Box, click on the respective button, then click in the Template Window.
2. Drag the edges of the List Box to resize it.



3. Click and drag the List Box to move it.
4. Set the Refresh and Supported Operations properties.
5. Content of the List Box can be changed either in Template Editor using the **List Editor** or dynamically using the **LISTVIEW** command. Generally the list view is filled using a script. You can also link an **EMR List** to the list box using the **EMRList** property.

USING THE LIST BOX

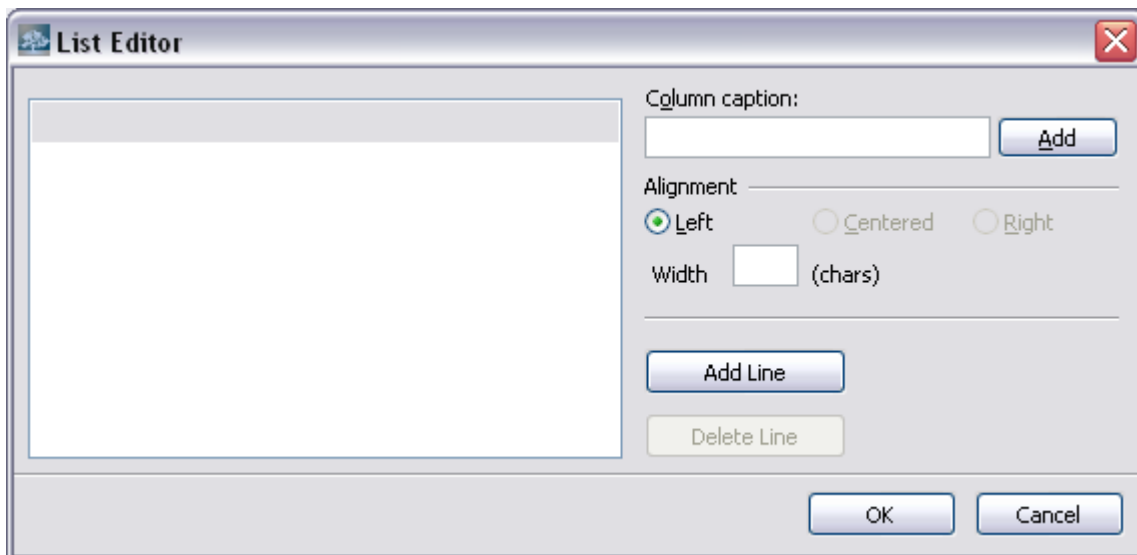


When this screen is now used to edit something, then after the screen is saved, the key of the selected row (Row2 in this illustration) is saved and retrieved as a variable **&screenname:listboxname**.

- If content was added in TE, then the key assigned to a row is its sequence number (e.g. the key of the 2nd row is 2).
- If content was added through script, then the keys can be defined by the script.

LIST EDITOR

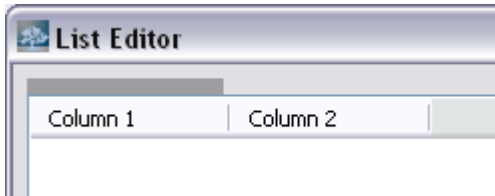
1. To change content in Template Editor, click the **Content** property at first. A button with three dots appears next to it, click the button.
2. The List Editor opens.



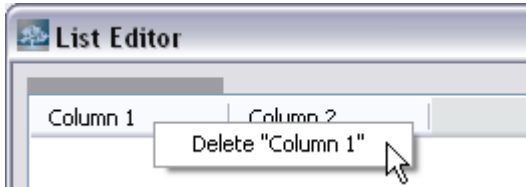
3. Type the caption of the first column in the **Column caption** field and press **Enter** or click the **Add** button to add the column. Repeat the same procedure to add more than one column. The first column is always left aligned, the alignment of others can be changed by clicking on the respective buttons.

After you have added the column, the Width field displays the current column width. The units are set by the **Width Units** property. To resize the column, drag the right edge of the column label.

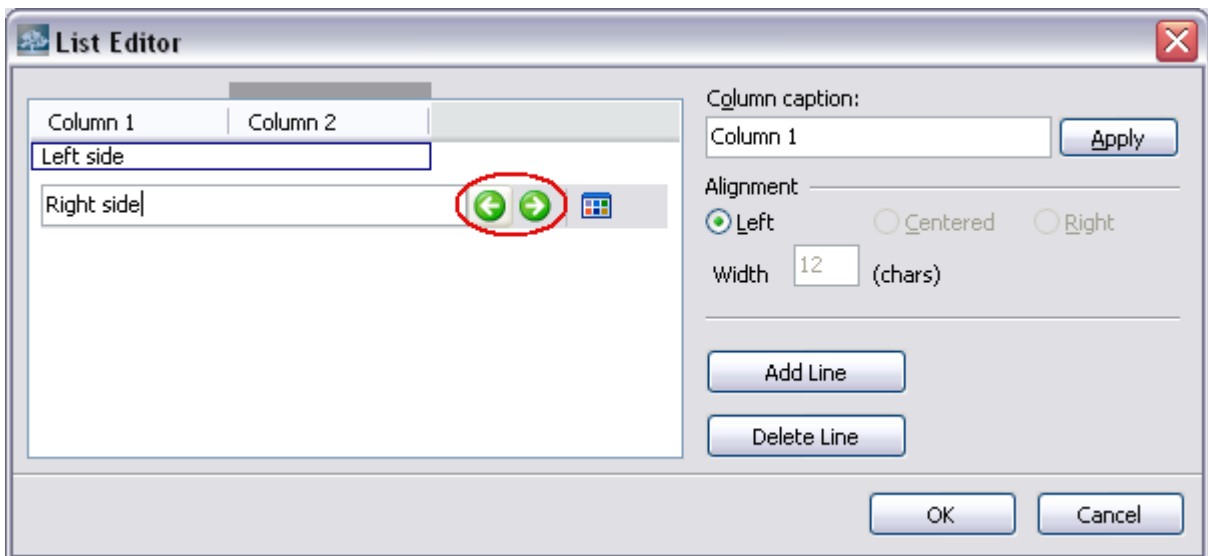
4. Click on a column to select it. The grey bar on top indicates, which column is selected at the moment.



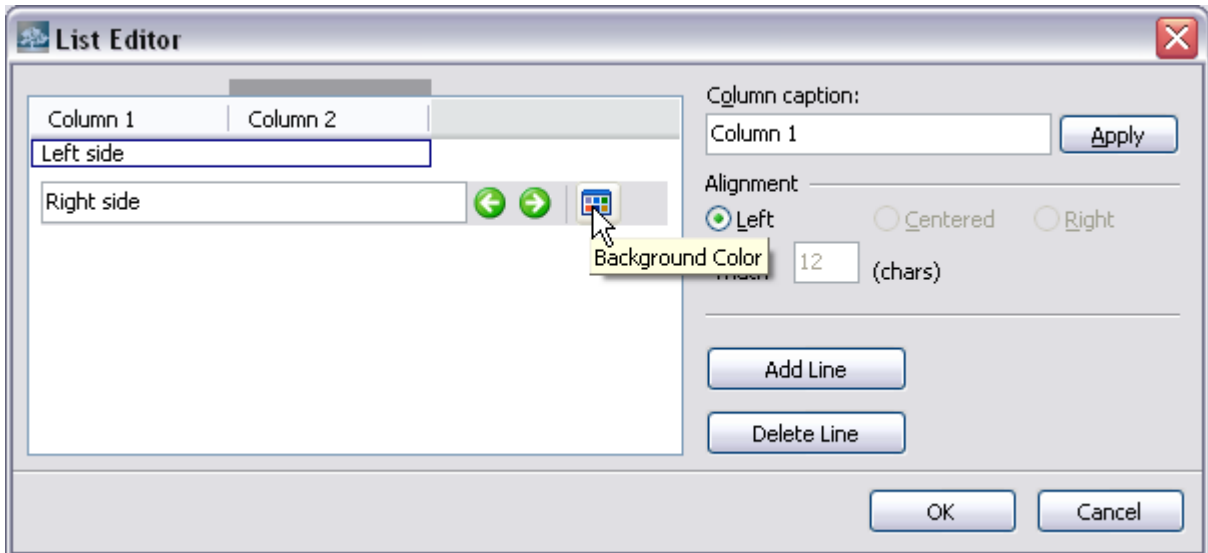
5. To delete a column, right-click on the column label and select the "Delete" option.



6. Click the **Add Line** button to add a row. Type the row text, and press **Enter** to continue. If you now type something and press **Enter**, then by default this text is added to the next column on the same row. To add text to a new row, click the Add Line button first and then type the text.
7. Use the arrows to move between columns on the same row.



8. Use the **Background Color** button to define the text and background color for the row.



9. Double-click on a row to edit it.
10. When done, click on the **OK** button.

PROPERTIES

The **List Box** is a good way to display large amounts of data. Because of that, there are several properties that help you to organize the data.

The **Content** property opens the List Editor to allow manually adding content. This is not commonly used, as the list contents are mostly specified either through script or by an EMR List (using the **EMRList** property).

Supported Ops determines which operations are allowed in the list - finding, printing, etc.

There are also several properties to set up RSL refreshes - when the selection in the list is changed, when the user right-clicks the listbox, or when the list is double-clicked (or an item is selected by pressing **Enter**).

The **Show as Graph** property applies only if the listbox displays an EMR list and the EMR list has the Graph defined. In that case you can set this property to display the graph by default.

COMBO BOX



The Combo Box works like the **Field** widget but instead of the user being able to add any text in the field, the user is limited to values provided by the combo box.

To define the available selections, use the **Lines** property.

AUDIO WIDGET



The Audio Widget is a tool for storing and playing audio stored in the custom record.

PANEL



The Panel Widget is a non-interactive container of widgets that has a border around it. This widget allows grouping widgets together both visually because of the borders and also logically, so that if the panel is moved or hidden then all of the widgets connected to it will also be moved or hidden.

To link a widget to a panel, the **Tab Control** property of that widget needs to be set to point to the name of the panel widget.

Note that you can also have panels inside panels, etc.

PROPERTIES

Most properties specific to the **Panel** widget allow customizing the panel caption.

To create a caption for the panel, type it in the **Caption** property. Now you can edit the caption text color and background color or just display it in bold formatting.

REPEATER WIDGET



The Repeater widget is useful if you need to add many similar items to a screen. Any widgets that are assigned to the Repeater (using the **Tab Control** property) are "repeated" the specified number of times (actually, the Repeater creates copies of the widgets and names them sequentially).

The basic workflow:

1. Create a Repeater.
2. Assign the Repeater to a Panel or Tab Control (optional).
3. Using the **Repeat Count** property, specify how many times the widgets should be repeated.
4. Specify the **Naming Rule**.
5. Specify the **Filling Style** and **Direction**.
6. Create the widgets you want to repeat and assign them to the Repeater.

PROPERTIES

There are several important properties that control the **Repeater** widget behavior.

- **Repeat Count** determines how many times the widget is repeated.
- **Naming Rule** controls how the copies are named.
- **Filling Style** and **Direction** specify how the repeated widgets are positioned.

LINK BUTTON WIDGET

This widget is designed for attaching files. When the user clicks the button, a new window is displayed where the user can select a file to upload. When a file has been uploaded, the button displays the date of the upload and clicking the button opens the file for viewing.

Note that you can also use the **Image** widget to store images.

PROPERTIES

The **Link Button** has two properties to control how the Open File dialog opens when the user clicks the button:

- **Initial Dir** property determines which folder is opened by default.
- **Filter** controls which file types the user can select for the upload.

EDIT BUTTON WIDGET

This widget replaces part of the **Text** widget's functionality. Clicking the button on the screen opens the **RTF Editor**, where the user can enter text. After the text has been saved, the button displays the date of the last update to the text.

This widget is similar to the **RTF Memo**, in that they both store rich text. The key difference is that the RTF Memo takes up more space on the screen, but is easier to access, especially when entering small amounts of text.

DESIGNING SCREENS

SCREEN DESIGN GUIDELINES

The following topics cover some basic design guidelines and provide advice on designing efficient and easy-to-use screens.

ORGANIZE WIDGETS

- Order the widgets according to the order they are filled out. This creates a flow of information that is easy to follow.
- Use tab controls to split the screen into several steps. Note that you can create tab controls inside a tab, thereby creating multiple levels of tabs. However, nesting too many tab controls results in a complicated screen.
- Use panels to group related information.
- Use sufficient space between widgets. If the screen is too crowded, it is hard to follow and use.
- Avoid using too many or uncommon abbreviations. While abbreviating terms and phrases may save screen space, it may cause confusion.
- Automate parts of the process. For example, entering a patient number should automatically display the patient name.

USE THE CORRECT WIDGETS

There are several widgets available for defining options that the user can select. Each of them has its proper use:

- **Radio buttons** are used when there is a list of two or more options that are **mutually exclusive** and the user must select exactly one choice. In other words, clicking a non-selected radio button will deselect whatever other button was previously selected in the list. Note that Raintree allows selecting none of the options.
- A **combo box** is similar to a radio group - it allows selecting only one of several options. The combo box will use less space on the screen, but selecting the option will be a bit slower, because the user has to first open the combo box to see the available options and then select one.
- You can also create a **field** and link it to a custom table. It generally takes even less space than a combo box or a group of radio buttons, but using it is even slower - the user has to open the custom table, find the appropriate entry and then select it. On the other hand, if there are many options available, the custom table can support searching for the appropriate code.
- **Checkboxes** are used when there is a list of options and the user may **select any number of choices**, including zero, one, or several. In other words, each checkbox is independent of all other checkboxes in the list, so checking one box does not clear the others.
- If you need to create a simple yes/no option, use a **stand-alone checkbox**. This takes less space than a pair of radio buttons and is easier to handle. Note that using radio buttons is justified in cases where the options are not simply yes/no, for example Male/Female.

Selecting the correct type of widget is generally a compromise between available screen space, speed and ease of use. Consider which type of widget to use based on your current needs.

WIDGET NAMING CONVENTIONS

These conventions are used by Raintree developers when designing screens. Following them helps preserve clarity and maintainability when customizing screens.

All **data elements and interactive UI elements** must be named. The following conventions apply:

- The widget name should describe its content or purpose.

- The widget name should be limited to 15 characters. Use acronyms if needed.
- Use medial capitals to separate parts of the name. Example: docFirstName, btnPost.
- Do not use any special characters (including underscores or ampersands) in field names.
- If a group of widgets belong in the same category (e.g. charges, diagnoses, clearances) use the prefixes to indicate the category. Examples: chgCode, dxDesc, clrConsent.
- Use the following prefixes to indicate the type of interactive UI widgets:

Widget	Prefix	Example
Image Button	btn	btnPost
Shrinker	shr	shrPatient
Tab Control	tab	tabMain
Panel	pan	panCharges
Repeater	rep	repSymptoms

USING FORMSCRIPT VARIABLES

You can now also use formscript variables on the screen, for example to display the patient name. To do that, enter the variable surrounded by square brackets [] in a multiline label's Text property, for example [time] to display the current time.

EMR EXAMPLES

This functionality is often used when designing EMR input screens to display information about the EMR. Below are three common sections:

Patient Information			
Patient	[first . " " . last]	MR #	[emrpn]
DOB	[dob]	Age	[age]

Record Information			
Date	nn-nn-nn ◀	MD	&docto ◀
Time	nn:nna	By	&sec ◀

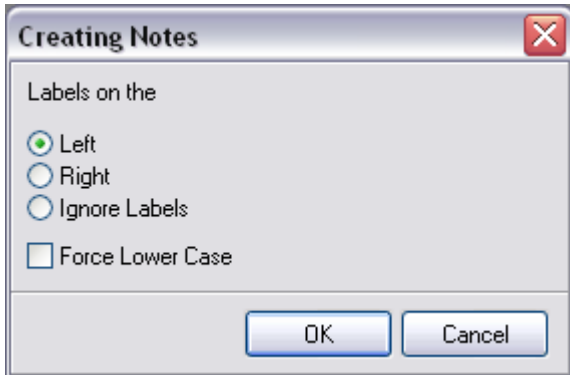
Created and Modified By					
Created By	[emrcreatebyname]		Modified By	[emrmodifbyname]	
Date/Time	[emrcreatedate]	[emrcreatetime]	Date/Time	[emrmodifdate]	[emrmodiftime]

NOTES

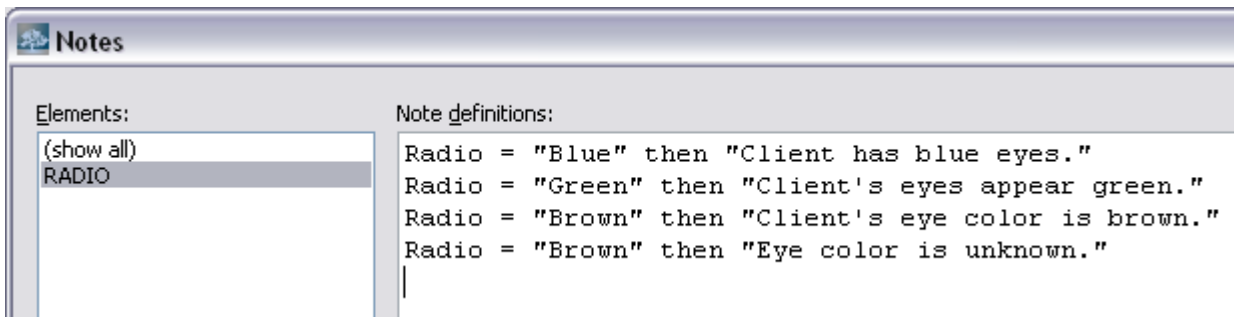
The notes property allows you to specify text that can be inserted into EMR templates for a field depending on its value. The advantage of this approach is that your EMR templates do not have to contain many @if/@endif statements for text that should change depending on the value of a field.

To edit the notes for a widget:

1. Select or add a new widget.
2. Double-click the **Notes** property. You can also click the button with the ellipsis (⋮).
3. For checkboxes and radio buttons an additional window appears, allowing you to specify where the labels for the elements are located.



4. The Notes window opens with the current element selected.
5. Specify notes to appear corresponding to different values of the element.



In this example, different notes are specified for the values of a radio button group. Also, a special note is added for the case when no option is selected.

INCLUDING THE VALUE IN THE NOTE

You can use the variable `%val%` in an item's note. Thus, an example similar to that shown above could have been:

```
Radio <> "" then "Client's eyes are %val%."
Radio = "" then "Client's eye color unknown."
```

You can also compare the value of a field in the notes to insert a note that is context-sensitive, such as a field (in this example TEMP) that recorded a client's temperature. The notes for that field might include:

```
temp > 100 then "High body temperature at %val%."
temp < 97 then "Low body temperature at %val%."
temp ((temp > 97) and (temp < 100)) then "Normal body temperature at %val%."
```

NOTE COLLECTIONS

OVERVIEW

Collections allow you to logically group independent fields together, define how the group contents should be displayed, add some text before or after the group, define the order of the group items, etc.

For example, you might want to display a list of symptoms and start the list with "The symptoms include". The symptoms are defined as checkboxes. If you use only regular notes, you have to reference the notes for each checkbox separately and separate the symptoms by commas. But if some of the symptoms do not appear, the list in the EMR will have several commas following each other. Adding the commas in the notes is also not an option, since each symptom might be the last one in the list and therefore does not need a comma. Also, if none of the symptoms appear, you will be left with a sentence "The symptoms include ." One way to solve this would be using multiple conditionals to control the prefix text and commas. The other, much easier way is to define a note collection, defining the list type as AND and specifying "The symptoms include" as the prefix.

To access collections:

1. In the Template Editor - select **Templates** from the menu bar, then select **Note Collections**.
2. The Note Collections window opens.

DEFINING COLLECTIONS

The upper part of the window allows you to define and delete collections. On the left, the collections are displayed. To create a new collection, click the **New** button and type the collection name in the Name field. The collection name is used to identify the collection.

Style: specifies the final written note format. There are seven styles available:

1. **OR** - item1, item2 or item3
2. **AND** - item1, item2 and item3
3. **Bullets** -
 - item1
 - item2
 - item3
4. **Number List** -
 1. item1
 2. item2
 3. item3
5. **Alpha List** -
 - a. item1
 - b. item2
 - c. item3
6. **Names** - this special option prints the **names** of all collection elements in the same order as they are in the "Selected Items" pane. This option is for programming purposes.
7. **Normal List** -
 - item1
 - item2
 - item3

In case of number or alpha lists, you can optionally specify the first element. For example, if you want to start the numbering with 3, type 3 in the **First Element** field. For alpha lists, this field also controls whether the characters are uppercase or lowercase (the default is lowercase).

Hint: if you define the collection as any type of list, add a line break at the end of the prefix. Otherwise, the first list item will follow the prefix on the same line.

On the right the Prefix and Suffix fields allow you to define text to be added before and after the collection. Click the field to open the RTF editor. This allows you to **insert fields** into the prefix and suffix. **Note that** if the prefix or suffix is in a different font than the rest of your **narrative**, it will override the narrative font.

COLLECTION ITEMS

The lower part of the window allows you to define the items in the selected collection. The **Available Items** pane contains all items on the screen that can be added into a collection. You can check the **Sorted** checkbox to sort the items in alphabetical order; however, this is generally undesired as you might want to keep the collection items in the same order as they appear on the screen.

In the **Selected Items** pane, the items in the currently selected collection are displayed. You can move individual items up and down by using the respective buttons on the right. This allows you to define the order in which the items appear on the screen.

You can add or remove items from the **Selected Items** pane with the buttons in the middle - single arrows ('>' and '<') apply to the currently highlighted items, double arrows ('>>' and '<<') add or remove all items.

TAB ORDER

When building custom screens you may want to create a screen with a specified focusing order of widgets. The sequence in which the user advances from one widget to another can be predetermined using this property.

1. Open a custom screen in Template Editor.
2. Check the **Use Tab Order** screen property (switched off by default) in the **Template Window properties**.
3. Next, set the Tab Order for each widget. To define the Tab Order manually, just click on each widget and set the Tab Order property value for that widget. If a duplicate number is assigned accidentally, all consecutive numbers are automatically changed by one integer so that no duplicates are created.
Alternatively, you may use the **triple arrow button** to add the values to all widgets quickly and comfortably.

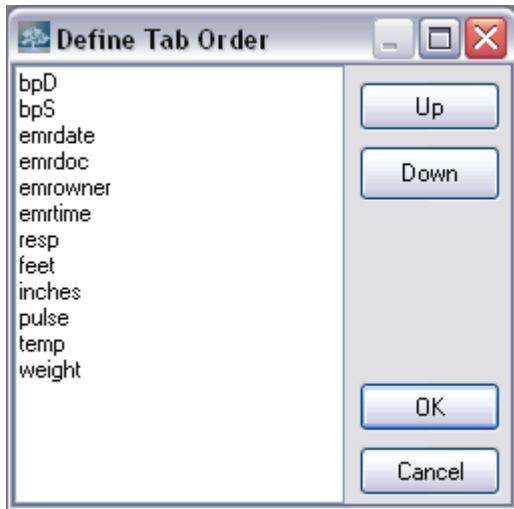
The Tab order has been set now. When a user is advancing from one widget to another on that screen (using Tab or the up and down arrows), then the sequence specified by the tab order is used.

Note that non focuseable widgets (e.g. the text widget) have this property as well, but for these objects the value of this property can be ignored.

The **Tools** menu has two additional options:

- **Reset Tab Order** - resets the tab order, so that widgets will be focused in the order that they appear on the screen.
- **Edit Tab Order List** - displays a list of the focusable items, allowing you to edit their order

The **Tab Order List** is an alternative way to set the focus order - while the Triple Arrow Button method presents a more visual way to assign the focusing order, this list contains all focusable items on the screen, this way making sure that you don't miss any.



To edit the order, use the **Up** and **Down** buttons.

Note that you can also open the editor by clicking the **Tab Order** property, then clicking the button with the three dots that appears.

