Template List

Description

A Template List can be used to create a list of template instances, where each template is used to control different parameter scopes. You can choose any template in the template property. The look of each list view row or column, depends on the selected template. All templates are added as line entries in a column or as entries in a row without spaces in between. If you need an optical space between two entries, position and resize the referenced template content accordingly or add separating graphical elements.

On this page:

Properties

Property	Description
Name	The name of the Template List.
Position /Size	Position and Size of the element in pixels. Pos x, Pos y, Width, Height (Pos = Position of upper left corner)
Attach	• **Left:** If the parent is resized, the control remains attached to left edge, with the specified ratio**.** • **Right:** If the parent is resized, the control moves relatively to the right edge, with the specified ratio. • **Top:** If the parent is resized, the control remains attached to top edge, with the specified ratio. • **Bottom:** If the parent is resized, the control moves relatively to the bottom edge, with the specified ratio. • **Left + Right:** If the parent is resized, the control is resized horizontally relatively to the left and right edges, with the specified ratio. • **Top + Bottom:** If the parent is resized, the control is resized vertically relatively to the top and bottom edges, with the specified ratio.
Layout *(Minimum HA Lion 6.1)*	• **Horizontal:** Activate this option to align the child templates horizontally in a row. • **Vertical:** Activate this option to align the child templates vertically in a column. • **Grid:** Activate this option to align the child templates in a 2-dimensional grid.
Alignment *(Minimum HA Lion 6.1)*	Here, you can adjust how the referenced child templates are postioned in the available templatelist view area. • **Center:** All child templates are added to the list in their original size. The row, coloumn, or grid is then centered inside the template list area. • **Left:** All child templates are added to the list in their original size. The row, coloumn, or grid is then positioned to the left of the template list area. • **Right:** All child templates are added to the list in their original size. The row, coloumn, or grid is then positioned to the right of the template list area. • **Fill:** All list cells are resized equally to match the size of the template list. The content of the child templates is then positioned and resized according to the Scale and Attachment settings of the contained controls.
Tooltip	Text that appears as a tooltip when the mouse hovers over the element.
Value	Allows you to read out the index of the focused template instance.

Style	- **H-Scroll:** Shows a horizontal scrollbar. - **V-Scroll:** Shows a vertical scrollbar. - **Fill:** The template is scaled to fill the list entry space completely. *(Only HALion 6.0)* - **Horizontal:** By default, the templates are listed vertically in a column. Activate this option to align them horizontally in a row. *(Only HALion 6.0)* - **Order:** Set this option if you want to be able to change the order of items via drag and drop. This is only a graphical reordering option. If you want the order change to have an effect on the program, for example, if you want to change the order of alternating layers, this functionality must be scripted. - **Custom SB:** Set this option if you want to use custom scrollbars. For custom scrollbars, you must assign various bitmap resources in the Scrollbars section. The height of the horizontal scroll bar and the width of the vertical scroll bar depend on the maximum width/height of the Hor and Ver Back bitmaps. (Example: If Hor Back is set to a height of 30 pixels and Ver Back to 36 pixels, both will use 36 pixels. Note: Scrollbars can only be scaled in their original orientation, that is, vertical scrollbars scale only vertically and horizontal scrollbars only horizontally.
Focus Var	Allows you to specify a variable (@MyFocusVariable) that can be used inside the source template, for example to switch the focus within a stack view or an Animation.
Index Var	Allows you to specify a variable (@MyIndexVariable) that can be used inside the source template to display the index number of the list entry.
Scrollbars	Allows you to assign the resources that are required to draw custom scrollbars - **Up:** Up button bitmap - **Up Press:** Up button Pressed bitmap - **Down:** Down button bitmap - **Down Press:** Down button Pressed bitmap - **Ver Back:** Vertical scrollbar background bitmap - **Ver Handle:** Vertical scrollbar handle bitmap - **Handle Top:** Top section of the vertical handle - **Handle Bot:** Bottom section of the vertical handle - **Left:** Left button bitmap - **Left Press:** Left button Pressed bitmap - **Right:** Right button bitmap - **Right Press:** Right button Pressed bitmap - **Hor Back:** Vertical scrollbar background bitmap - **Hor Handle:** Vertical scrollbar handle bitmap - **Handle Left:** Left section of the horizontal handle - **Handle Right:** Right section of the horizontal handle

Examples

To create a Template List:

- Add Template List to your macro page.
- Select the template that you want to use for each list entry.
- Check the Template parameters which are added to the Template List.
- Create a String List variable for each template parameter that you want to connect.
- Connect the template parameters to the variables.

As soon as the first variable is connected, the list is filled with a template for each string list entry.

Example 1

You can use a Label control inside the template which exports its text property to the template, to name each row.

- Create a string list variable with all the labels that are needed and in the order in which you want them to appear in the list.
- Connect the variable to the Text parameter of the Template List.

Example 2

There is a control inside the template, for example, a switch with an exported Value parameter.

- Create a string list variable with the different parameter scopes that you want to control, for example, the Mute state of different layers within a program.
- Connect the variable with the switch Value parameter of the Template List.

Switching the scope for controls outside the Template List

One useful usage of the Template List is to manage different articulations of an instrument and to use the list to change the scope for controls that exists only once one the macro page but need to control the different articulations, for example. In this case, you can use the Value property of the Template List which delivers the index of the selected row.

- Create a string list "LayerSelect" variable with an entry for each articulation layer.
- Add the different scopes for each articulation to the string list.
- Add a Template List.
- Specify the source template for the Template List.
- Connect the template parameters (delivered by the source template) using further string list variables defining the different parameters.
- Set the Value property of the Template List to the string list variable (@LayerSelect). This way, the list index can change the scope.
- Add a group and set the Scope to "LayerSelect". Now the group scope changes with the the selection in the Template List.
- Add controls to the group and connect them to HALion parameters. Use only the parameters, not their path information, which is delivered by the group scope.
- When clicking inside a Template List row, the selection changes the focus of the connected control group and you modify each articulation individually.

Showing the Focus in the list

To visualize which entry of the Template List currently has the focus you can use the Focus Var property. It works in combination with an element that needs to be set up inside the source template.

Example 1

An animation to show a focus rectangle

- Set Focus Var of the Template List to "@SelectedRow".
- Add an animation with two frames to the source template. The second frame should show a colored outline rectangle, for example.
- Set the Value property of the animation to @SelectedRow", too.
- When clicking inside a Template List row, the animation of the corresponding row switches to the second frame which shows the colored rectangle.

Example 2

A stack to change the whole appearance of a Template List row

- Set Focus Var of the Template List to "@SelectedRow".
- Add a Stack.
- Group the existing controls of the source template and place the group inside the stack.
- Duplicate the group and modify the appearance of the controls of the second group.
- Set the value property of the stack to "@SelectedRow", too.
- When clicking inside a Template List row, the stack of the corresponding row switches to the second group and shows the modified version.