Conditions

Taking advantage of conditional automation.

Conditions are an excellent way to define the behavior of the SRP Tree Control. Instead of writing your own code to specify when something should occur, you just send the condition logic to the SRP Tree Control to handle it for you. The following properties use conditions:

Name	Description
CheckBoxConditions	Uses conditions to determine when a check box should appear to the left or right of an item, if at all
DragCondition	Uses conditions to determine if the user is allowed to drag and item
DropCondition	Uses conditions to determine if the user is allowed to drop other tree items onto the target item
Filter	Uses conditions to determine if an item is visible
ImageConditions	Uses conditions to determine which global image to use for an item
ImageEffects	Uses conditions to determine which effect to apply to an item's image at any given time

You will pass condition strings to these properties to instruct the SRP Tree Control on how to proceed with certain tasks. Properties using the plural form accept multiple sequential conditions, like a case statement. Properties using the singular form accept a single condition that is used to make a simple decision. Refer to the individual properties for details.

A condition string is a specially formatted string that the control will parse and execute. The format is that of a Boolean expression, that is, a mathematical expression that ultimately evaluates to true or false. Here are some sample expressions:

```
"Selected"
"Selected AND Not(HasChildren)"
"HasOnlyLeaves AND (Selected OR Checked)"
"Level LT 1"
"Class EQ 'Group'"
```

Notice that the condition strings take on the BASIC+ language syntax, although you may also use operators from other familiar languages, such as C++. Here are the same sample conditions using C++ operatores:

```
"Selected"
"Selected && !HasChildren"
"HasOnlyLeaves && (Selected || Checked)"
"Level < 1"
"Class == 'Group'"
```

The following operators are available to you when writing conditions:

Name	Operator(s)	Example
Logical OR	OR, ||	"Selected OR Checked"
Logical AND	AND, &&	"Hot AND Selected"
Equals	EQ, =, ==	"Class = 'Group'"
Not Equals	NE, #, !=, <>	"Class # 'Group'"
Greater Than	GT, >	"Level > 1"
Greater Than or Equal To	GE, >=	"Level >= 2"
Less Than	LT, <	"Level < 2"
Less Than or Equal To	LE, <=	"Level <= 1"
Not	*Not(expr), !expr*	"Not(HasOnlyLeaves)"

The SRP Tree Control allows the use of a limited set of keywords in your conditions. Keywords identify a particular state of a tree item. You may use the following keywords in your condition statement:

Keyword	Type	Description

Checked	Boolean	The item's checked state
Hot	Boolean	The item's hot state. An item is hot when the mouse hovers over it
Enabled	Boolean	The item's enabled state
Selected	Boolean	The item's selected state
Expanded	Boolean	The item's expanded state
HasChildren	Boolean	True if the item has children
HasOnlyBranches	Boolean	True if at least on child items has children
HasOnlyLeaves	Boolean	True if no children have children
Level	Integer	The item's indent level. Level 1 items are at the root level. Level 2 items have at least one parent, and so on.
Class	String	The item's class name.
ParentClass	String	The item's parent's class name.
Key	String	The item's unique key ID.
ParentKey	String	The item's parent's unique key ID.
Text	String	The item's text.

In addition to the above pre-defined keywords, conditions can also reference any user defined field as defined in the ItemFields or ItemField properties.

Using equality operators with Boolean keywords is optional, so "Selected" and "Selected EQ 1" are equivalent. The integer and string keywords require the use of the equality or comparison operators. So use "Level > 1" or "Class = 'Group'" instead of just "Level" or "Class".

If your conditions do not seem to working, double check your syntax. The condition is case insensitive, but you must spell the keywords correctly. Make sure your string values are incased in quotes and that your integer values are not. Also make sure that you are not using unary operators on integer or string keywords.

To make a condition that evaluates to true in all cases, just set it to "1".

See Also

CheckBoxConditions, DragCondition, DropCondition, ImageConditions, ImageEffects