IconEffects

Conditions to determine which special effect to apply to the buttons' icon.

Usage

```
Set_Property(OLECtrlEntID, "OLE.IconEffects", StringValue)
```

Values

StringValue can be any string meeting the following format requirements:

Syntax: @FM delimited list of conditions

Default: ""

Remarks

The IconEffects property allows you to automate which special effect should be applied to the button's image. You provide an ordered list of specially formatted condition statements telling the button what state it must be in order to apply a particular effect. When the SRP Button Control is about to display its icon, it evaluates each condition in order and, if one of them is true, applies the associated effect.

Each field has two values: Effect and Condition. The Effect value identifies one of the available effects.

Effect	Description
Saturate	Saturates the image slightly so that normal images stand out more
Grayscale	Displays the image in grayscale instead of full color
Levitate	Displays the image slightly raised with a drop shadow
Faded	Displays the image translucently over the background
TextColor	Displays the image as in monochrome matching the text's current color (See Forecolor property).
Default	The control draws the image normally if enabled or grayed out if disabled

The second value of each field, the Condition value, is the condition in which the effect is to be used. Each field in this property as a like case statement. So, the property value might look something like this:

```
Conditions<1> = "Levitate" :@VM:"Hot"
Conditions<2> = "Default"  :@VM:"Pressed"
Conditions<3> = "Saturate" :@VM:"1"
```

However, you can visualize the property to mean this:

```
Begin Case
   Case Hot
       ApplyLevitateEffect()
   Case Pressed
       ApplyDefaultEffect()
   Case 1
       ApplySaturatedEffect()
End Case
```

The SRP Button Control will first check to see if the button is hot, and if it is, it will apply the Levitate effect to the icon. If it is not hot, then it will check to see if it is pressed, and if it is, it will apply the Default effect (which is to say, no effect at all). If it is not pressed, then it will apply the Saturated effect.

The following operators are available to you when writing conditions:

Name	Operator(s)	Example

Logical OR	OR, ||	"Tristate OR Checked"
Logical AND	AND, &&	"Hot AND Focused"
Not	Not(expr), !expr	"Not(Focused)"

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

Keyword	Type	Description
Checked	Boolean	The button's toggle state. Checked means it's toggled on.
Hot	Boolean	The button's hot state, which means the mouse is hovering over it
Enabled	Boolean	The button's enabled state.
Pressed	Boolean	The buttons's pressed state.
Focused	Boolean	The buttons's focus state
Tristate	Boolean	The buttons's toggle state. Tristate means it's neither checked or unchecked. That is, it's unknown.

If your conditions do not seem to working, double check your syntax. The condition is case insensitive, but you must spell the keywords correctly.

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

Example

```
// Icon effects base on button states
Conditions = ""
Conditions<1> = "Levitate":@VM:"Hot"
Conditions<2> = "Default":@VM:"Pressed"
Conditions<3> = "Saturate":@VM:"1"
Set_Property(@Window:".OLE_BUTTON", "OLE.IconEffects", Conditions)
```

See Also

Icon, IconList