ItemList

The list of all tree items in the control

Usage

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

Values

StringValue can be any string meeting the following format requirements:

Syntax: @FM delimited list of tree items

Default: ""

Remarks

The ItemList property gets and sets all the tree items in the entire control. Setting this property replaces all existing tree items with the new ones. Therefore, this property is ideally suited for initialization purposes. If you simply need to add new items, use the AddItems method instead.

This property is formatted as an @FM delimited array of tree items. Each field in the array represents a single item, and the multivalue structure for the field is as follows:

Pos	Name	Type	Description	Default
<x, 1>	Level	Integer	The item's level in the tree	
<x, 2>	Key	Text	The item's unique key	
<x, 3>	Data	Text	The item's data	""
<x, 4>	**Data Type**			
<x, 4, 1>	Type	Option	The item's data type, used to make sorting accurate and efficient	Text
<x, 4, 2>	Format	Formatted String	The item's data format, used to present the data to the user	""
<x, 5>	Class	Text	The item's class, an application specific string used for classification purposes	""
<x, 6>	Image	Formatted String	The item's image, either a file or a key pointing to a global image	""
<x, 7>	**Colors**			
<x, 7, 1>	Text Color	Color	The color of the item's text	WindowText
<x, 7, 2>	Background	Color Fill	The item's background	Window
<x, 8>	Font	Font	The font used to render the item's data	Tahoma, 8 pt.
<x, 9>	**Alignment**			
<x, 9, 1>	Horizontal	Option	The horizontal alignment of the item's data	Left
<x, 9, 2>	Vertical	Option	The vertical alignment of the item's data	Center
<x, 10>	Sort	Option	The item's sort setting	None
<x, 11>	Check Box	Option	The item's check box setting	None
<x, 12>	Expandable	Boolean	Whether or not the item can be expanded or collapsed by the user	1

<x, 13>	Height	Integer	The item's height	16
<x, 14>	Hyperlink	Boolean	Whether or not the item displays hyperlink feedback to the user	0
<x, 15>	Selectable	Boolean	Whether or not the item can be selected by the user	1
<x, 16>	Swatch Color	Color Fill	A custom colored rectangle that appears at the beginning or end of an item	None
<x, 17>	Swatch Position	Option	The position of the swatch, if there is one: Left or Right	Left
<x, 18>	**Swatch Size**			
<x, 18, 1>	Width	Integer	The swatch width, in pixels, if there is one	13
<x, 18, 2>	Height	Integer	The swatch height, in pixels, if there is one	13
<x, 19>	**Fields**		List of user defined fields and their values: each value has the following format:	""
<x, 19, y, 1>	Field Name	Text	The unique field name	""
<x, 19, y, 2>	Field Value	Text	The field's value	""

The first two values, Level and Key, **must** be included for every item. All other values are optional.

Level

The first value of every item is its level within the tree. The level will be used to determine where the item fits within the hierarchy. If an item's level is greater than the item preceding it, then it becomes that item's child. Since the SRP Tree Control is a true hierarchy, the levels will be interpreted in terms of relativity, not actual value. For example, if you set item 1 to level 1 and item 2 to level 8, item 2 will still be a direct child of item 1.The first two values, Level and Key, **must** be included for every item. All other values are optional.

Key

Every item has a unique key so it can be accessed directly later. This reduces the amount of tree traversal necessary to locate a particular item. **An item's key can be any alphanumeric string, but it cannot be "" or "All" and it cannot contain delimiters, periods, or commas.** If an item with the key already exists, then this item will not be added to the control.

Item Settings

The remaining values in each field initialize the new item. They are all optional, so set only those that are necessary. Usually, you will only set the Data and Class values while letting the default property settings take care of the rest. See Default Properties for more information.

See the Item property for more details on each of the possible values in the data structure.

Example

```
// Initialize the tree to boys and girls organized into teams
// Note that we're setting only the Level, Key, Data, and Class values
Items = ""

// Blue Team
Items<-1> = 1:@VM:"BlueTeam":@VM:"Blue Team":@VM:@VM:"BlueTeam"
Items<-1> = 2:@VM:"Emma":@VM:"Emma":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Ava":@VM:"Ava":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Isabella":@VM:"Isabella":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Sophia":@VM:"Sophia":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Aiden":@VM:"Aiden":@VM:@VM:"Boy"
Items<-1> = 2:@VM:"Ethan":@VM:"Ethan":@VM:@VM:"Boy"
Items<-1> = 2:@VM:"Matthew":@VM:"Matthew":@VM:@VM:"Boy"
Items<-1> = 2:@VM:"Noah":@VM:"Noah":@VM:@VM:"Boy"

// Red Team
Items<-1> = 1:@VM:"RedTeam":@VM:"Red Team":@VM:@VM:"RedTeam"
Items<-1> = 2:@VM:"Madison":@VM:"Madison":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Emily":@VM:"Emily":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Kaitlyn":@VM:"Kaitlyn":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Olivia":@VM:"Olivia":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Jacob":@VM:"Jacob":@VM:@VM:"Boy"
Items<-1> = 2:@VM:"Ryan":@VM:"Ryan":@VM:@VM:"Boy"
Items<-1> = 2:@VM:"Jack":@VM:"Jack":@VM:@VM:"Boy"
Items<-1> = 2:@VM:"Nicholas":@VM:"Nicholas":@VM:@VM:"Boy"

// Unassigned
Items<-1> = 1:@VM:"Unassigned":@VM:"Unassigned":@VM:@VM:"Unassigned"
Items<-1> = 2:@VM:"Abigail":@VM:"Abigail":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Hailey":@VM:"Hailey":@VM:@VM:"Girl"
Items<-1> = 2:@VM:"Joshua":@VM:"Joshua":@VM:@VM:"Boy"
Items<-1> = 2:@VM:"Logan":@VM:"Logan":@VM:@VM:"Boy"

// Set the items
Set_Property(@Window:".OLE_TREE", "OLE.ItemList", Items)
```

See Also

Item, AddItems