WebListBox White Paper

This whitepaper describes the concept and features introduced in WebListBox.

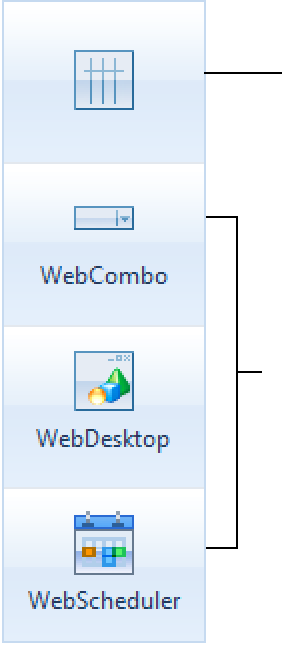
Table of Contents

Overview

WebListBox is a unique control, part of WebEssentials family, which can be used as an input control or a navigation control. It contains a list of items displayed vertically and users can select single or multiple items.

User Interface

Built to support XHTML and HTML doctype, WebListBox provides limitless customizations on the user interface aspect through its comprehensive properties. Furthermore, the UI personalization can be made globally (specified in **LayoutSettings** property) or item specific depending on your need.



DisplayMode property in this specific item is set to *Image*, thus the global *DisplayMode* configuration in LayoutSettings is not used.

DisplayMode property in these items are not specified, so the global *DisplayMode* configuration in LayoutSettings are used instead.

Display Mode

It is possible to display text and images, text, or images only in WebListBox's item. Use the **LayoutSettings >> DisplayMode** property to customize this behavior, or **[Item] >> DisplayMode** property for individual configuration. By default, the display mode is set to *Text*.

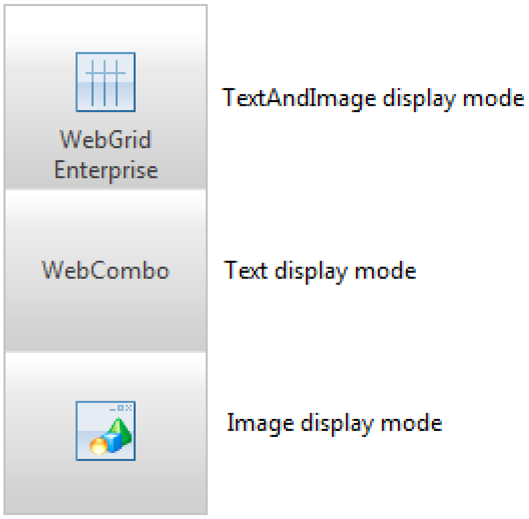


Image Position

When *TextAndImage* display mode is used, the default image positioning is above the text. This behavior can be tweaked further using **ImagePosition** property inside **LayoutSettings**, or**[Item] >> ImagePosition** for individual setting. Four image positioning options are available: *AboveText, BelowText, AfterText,* and *BeforeText*.

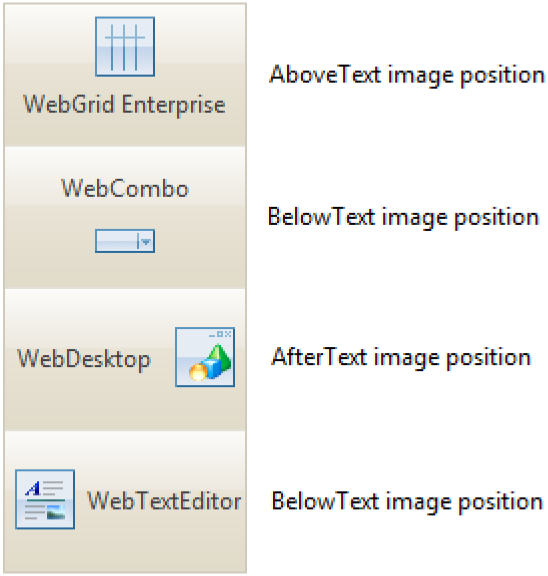


Image Size

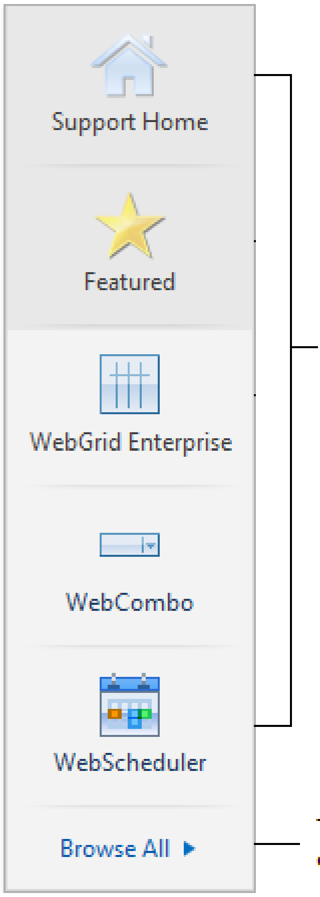
When using *Image* or *TextAndImage* display modes, you can customize the dimension of the images. The default is 40 pixels x 40 pixels. Use the **LayoutSettings >> ImageSize** property for global configuration, or **[Item] >> ItemImageStyle** for individual configuration.

Item Alignment

Item alignment can be specified globally or individually. Use the **LayoutSettings >> ItemAlignment** property for global configuration, or **[Item] >> ItemAlignment** for individual configuration.

Item Height

Item's height can be specified globally or individually. Use the **LayoutSettings >> ItemHeight** property to apply this globally or **[Item] >> ItemHeight** property to set individual item height. By default the height is 80 pixels.



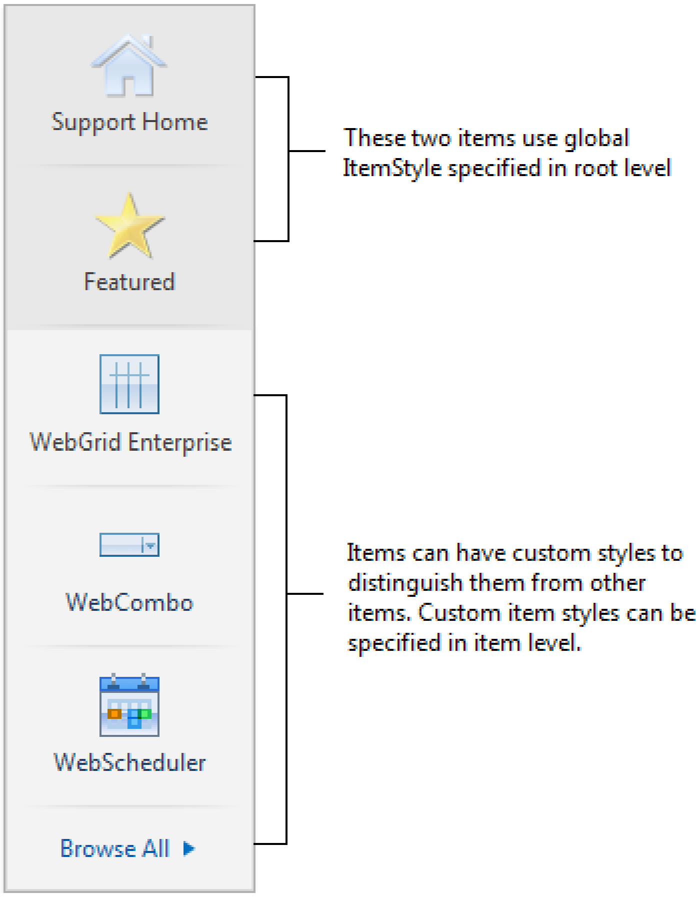
These items use global ItemHeight configuration in LayoutSettings.

This item applies custom ItemHeight value.

Item Style

WebListBox enables you to refine the item styles to match your need. **ItemStyle, ItemTextStyle,** and **ItemImageStyle** properties are provided for global configuration.

Individual item styling properties are also available enabling you to fine-tune the look-and-feel, **ItemNormalClass, ItemHoverClass, ItemActiveClass, ItemTextClass,** and **ItemImageClass**. All you need to do is to simply assign CSS classes to those properties.



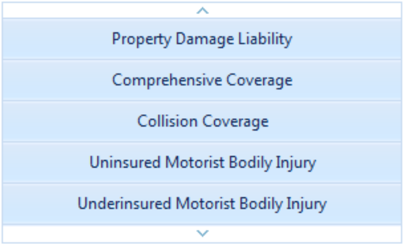
These two items use global ItemStyle specified in root level

Items can have custom styles to distinguish them from other items. Custom item styles can be specified in item level.

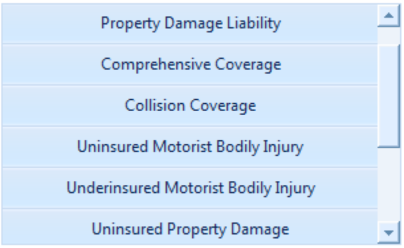
A helpful triangle indicator can be enabled to show the currently active item. Use the **LayoutSettings >> DisplayActiveItemIndicator** property to enable this feature.

Scroll Mode

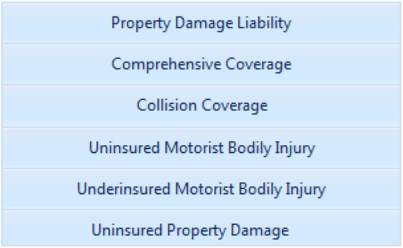
Scroll mode option is provided to anticipate having many items in WebListBox which exceeds the container's height. The common practice is to use scrollbar. WebListBox offers several scroll options, *Scroller, ScrollBar,* or *None,* that you can set in **LayoutSettings >> ScrollMode** property.


Property Damage Liability
Comprehensive Coverage
Collision Coverage
Uninsured Motorist Bodily Injury
Underinsured Motorist Bodily Injury

Scroller
scroll mode


Property Damage Liability
Comprehensive Coverage
Collision Coverage
Uninsured Motorist Bodily Injury
Underinsured Motorist Bodily Injury
Uninsured Property Damage

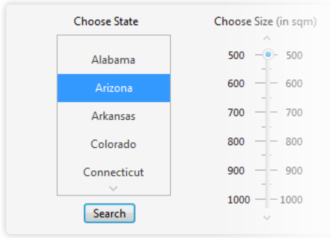
ScrollBar
scroll mode


Property Damage Liability
Comprehensive Coverage
Collision Coverage
Uninsured Motorist Bodily Injury
Underinsured Motorist Bodily Injury
Uninsured Property Damage

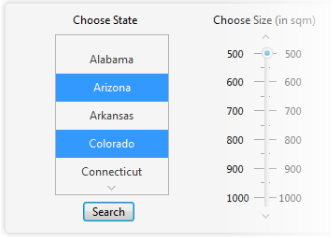
None
scroll mode

The *Scroller* scroll mode displays two directional arrows, up and down, that when users hover on it, all items will be scrolled. The *ScrollBar* scroll mode displays a vertical scrollbar, and the *None* option shows nothing. Hidden items remain invisible.

Selection Mode



Single Selection Mode



Multiple Selection Mode

The nature of WebListBox is as an input control where users can select not only one item, but multiple items at a time. This behavior can be configured in **SelectionMode** property. *Single* mode lets users to select only one item at a time. In *Multiple* mode, users can select multiple items by selecting a range using mouse, or using the keyboard + mouse combo to perform the selections. It works with CTRL and SHIFT keys.

Set Selected Item(s) On Load

Initial item selections feature is essential for certain scenarios. To achieve this, simply set the **[Item].Selected** property to *True* or the **SelectedIndex** property to the selected item's index.

In *Single* selection mode with **SelectedIndex** property set, any items' **Selected** property setting is ignored. In *Multiple* mode, you can either use the **SelectedIndex** property or Item's **Selected** property to make the selection.

Additionally, selected items can be defined using *SetValue* and *SetValues* server-side methods.

Get Selected Value

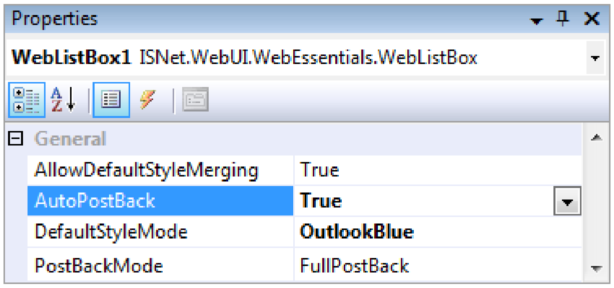
The selected item's index and value can be retrieved using **Value** and **SelectedIndex** property. When *Multiple* selection mode is active, use the *GetValues* server-side event to retrieve all selection.

Enable Keyboard Support

In addition to use mouse, users can use keyboard to select an item(s). Set the **EnableKeyboardSupport** property to *True* and users will be able to swiftly move through the items using up and down directional keys. Item will be automatically selected as users navigate using keyboard. Note that this property does not need to be enabled in order to use Ctrl and Shift keys when *Multiple* selection mode is enabled.

PostBack

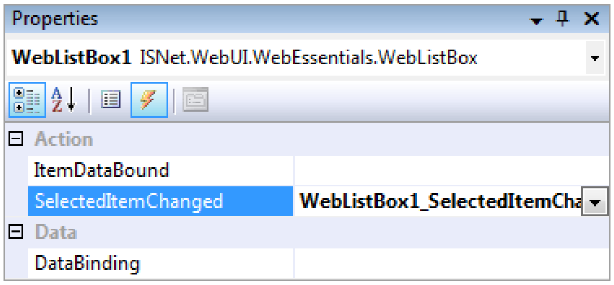
WebListBox has built-in **AutoPostBack** property for client-server communication. When enabled, postback will be trigged as the selected item is changed. You can perform *FullPostBack* or *FlyPostBack*, customizable in **PostBackMode** property.



WebListBox1 ISNet.WebUI.WebEssentials.WebListBox

General

AllowDefaultStyleMerging	True
AutoPostBack	**True**
DefaultStyleMode	**OutlookBlue**
PostBackMode	FullPostBack



WebListBox1 ISNet.WebUI.WebEssentials.WebListBox

Action

ItemDataBound	
SelectedItemChanged	**WebListBox1_SelectedItemCha**

Data

DataBinding	

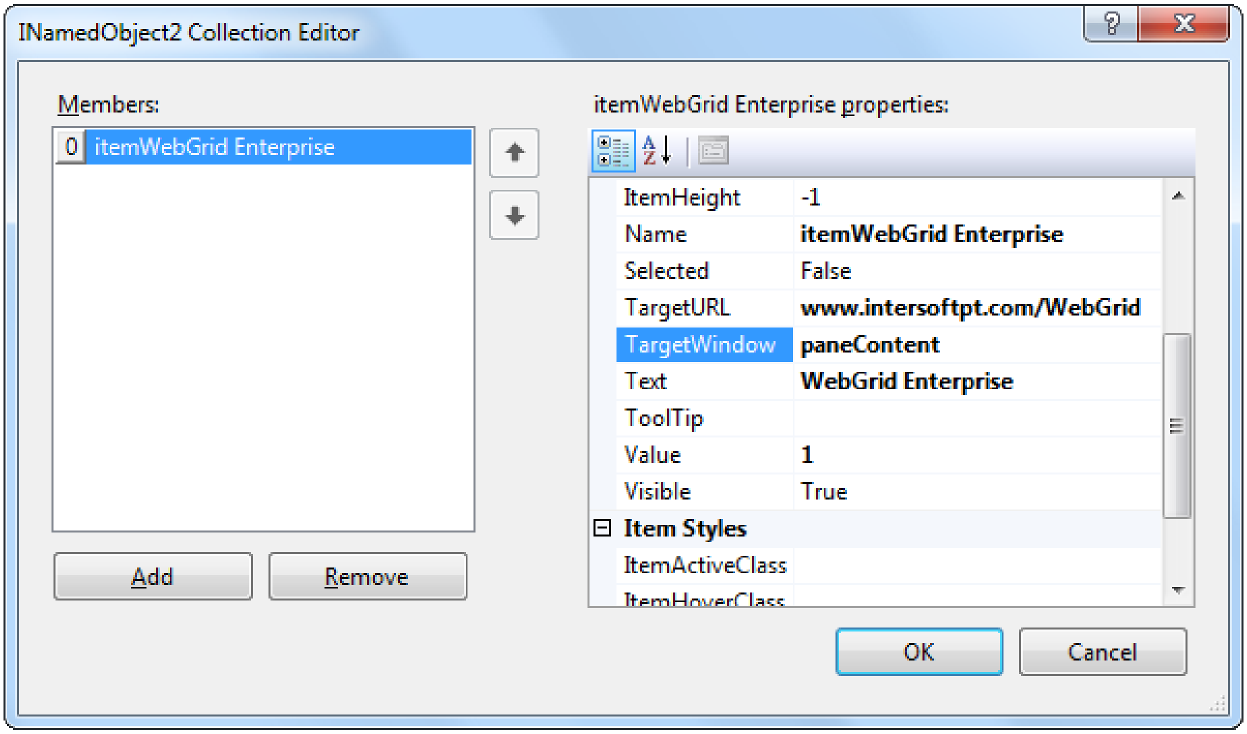
When postback is enabled, *OnSelectedItemChanged* server-side event is raised. This enables you to apply your programmatic processing. It allows you to obtain the selected items' index via the event arguments.

```
protected void WebListBox1_SelectedItemChanged(object sender,
WebListBoxSelectedItemChangedEventArgs e)
{
        int index = e.SelectedIndex;
        WebListBoxItem item = WebListBox1.Items[index] as WebListBoxItem;
        SetData(int.Parse(item.Value));
}
```

Note that this configuration will be ignored when the selection mode is set to Multiple.

Navigation

Despite its nature as an input control, WebListBox can also be used as a navigation control. When an item is clicked, the specified URL will be opened in a new window or target IFrame. The target window can be specified in **LayoutSettings >> TargetWindow** property or **[item] >> TargetWindow** property for individual item. The target URL can be specified in **[item] >> TargetURL** property.


INamedObject2 Collection Editor
Members:
0
itemWebGrid Enterprise
itemWebGrid Enterprise properties:
Add
Remove
OK
Cancel

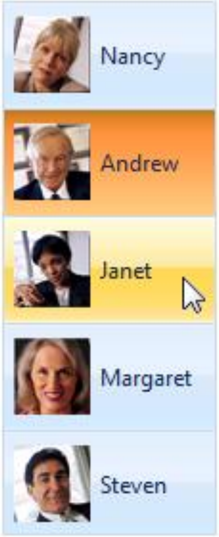
ItemHeight	-1
Name	itemWebGrid Enterprise
Selected	False
TargetURL	www.intersoftpt.com/WebGrid
TargetWindow	paneContent
Text	WebGrid Enterprise
ToolTip	
Value	1
Visible	True
⊟ Item Styles	
ItemActiveClass	
ItemHoverClass	

TargetURL and TargetWindow specified at item level

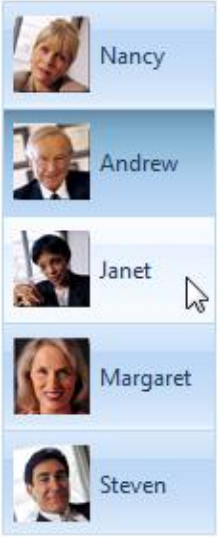
Note that this configuration will be ignored when the selection mode is set to Multiple.

Themes

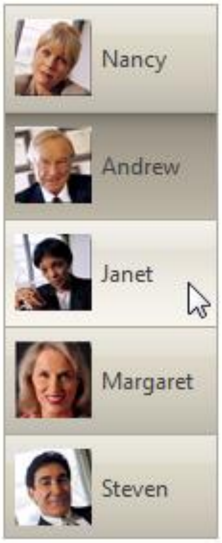
WebListBox sports various colorful and modern looking visual themes that you can select easily.



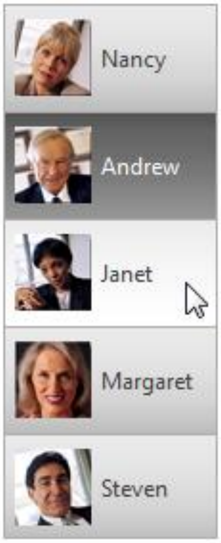
OutlookBlue style



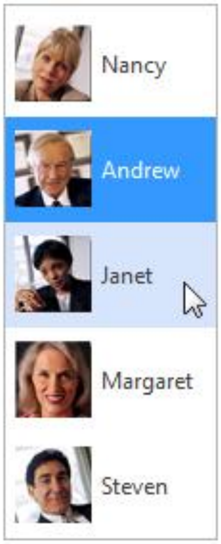
Win7 Style



Natural Style



Silver Style



Minimalist Style

Comprehensive Client-side Events

The following is the list of client side events in WebListBox.

OnInitialize(*controlId*)
This client-side event will be invoked when WebListBox is initialized.

OnItemMouseOver(*controlId, itemObject, itemElement*)
This client-side event will be invoked when an item is hovered.

OnItemMouseOut(*controlId, itemObject, itemElement*)
This client-side event will be invoked when mouse out action is performed on an item.

OnBeforeItemSelected(*controlId, selectedItem*)
This client-side event will be invoked before an item is selected.

OnAfterItemSelected(*controlId, selectedItems*)
This client-side event will be invoked after an item is selected.

OnResponse(*controlId, action*)
This client-side event will be invoked when response is returned after a FlyPostBack action.

OnResponseError(*controlId, action, errorMessage*)
This client-side event will be invoked if error occurs during a FlyPostBack action.

OnKeyUp(*controlId, keyCode*)
This client-side event will be invoked during key up action.

Rich Client-side APIs

The following is the list of client side APIs in WebListBox.

Functions	**Description**
GetElement()	Gets frame element of WebListBox.
GetContentFrameElement()	Gets the content frame element of WebListBox.
GetContentElement()	Gets the content element of WebListBox.
GetIndicatorElement()	Gets the triangle indicator element.
GetTopScrollerElement()	Gets the content frame element of the top scroller.
GetTopScroller()	Gets the top scroller element.
GetBottomScrollerElement()	Gets the content frame element of the bottom scroller.
GetBottomScroller()	Gets the bottom scroller element.
GetItemByValue()	Gets a specific item by value.
GetSelectedIndexes()	Gets the collection of selected items' indexes.
DoResize()	Performs resize in WebListBox.

Data Binding

Targeting enterprise Web applications, WebListBox is enhanced with data binding to data source control capability. WebListBox can be bound using traditional binding and datasource control.

Several properties related to data binding functionality are listed below.

Properties	**Description**
DataSourceID	The data source id of WebListBox.
DataMember	The data member of WebListBox.
NameFieldDataMember	The field data member of Name property.
ImageFieldDataMember	The field data member of Image property.
TextFieldDataMember	The field data member of Text property.

ValueFieldDataMember	The field data member of Value property.
TargetURLFieldDataMember	The field data member of TargetURL property.
ImageFieldFormatString	The format string of Image field data member.

The related properties will be bound automatically when the field data member is specified.

WebListBox also supports unbound mode. You need to add the items manually.