WizardControl Control

WizardPane Control

WizardControl Control

WizardPane Control

Activate

When a **WizardPane** is repositioned relative to the activating **WizardControl** control, made **Visible**, and sent to the top of the z-order.

K**Deactivate**

When a **WizardPane's Visible** property is set to **False** by a **WizardControl** control.

K**Sequence**

A grouping of one or more **WizardPane** controls where each control's **PaneIndex** property is included in another group member's **BackButtonPaneIndex** or **NextButtonPaneIndex** properties so that a logical chain of pane index values can be followed.

matching WizardPane

The **WizardPane** control who's **PaneIndex** property is the same value as the **CurrentPaneIndex** property of the **WizardControl**.

standard property
Information on this property can be found in
the standard Visual Basic documentation.
No other information is available (or needed).

Overview

The LDrive WizardX ActiveX control provides all of the elements needed to add professional quality wizard dialogs to any application. LDrive WizardX contains two controls, the **WizardPane** control and the **WizardControl** control. Together, these components team up to help you build powerful, flexible solutions.

The **WizardControl** provides the mechanism for the end user to control and direct the actions of the wizard. Separate events are raised when the user moves back and forth through a sequence, and when the Finish or Cancel buttons are clicked. A comprehensive list of properties offers you complete control over the style and behavior of the control.

Each **WizardPane** control provides a flexible container control to store elements of the wizard into. The control has a number of properties that are read by the WizardControl when the control is activated. The **WizardControl** is reconfigured on a pane-by-pane basis - all without a single line of code.

See Also

How the controls work together

Understanding the event cycle

WizardControl

The **WizardControl** provides the mechanism for the end user to control and direct the actions of the wizard.

WizardPane

Each **WizardPane** control provides a flexible container control to store elements of the wizard into.

New in this release

Version 1.20:

Stretch property is now transferred to the **WizardControl** from the **WizardPane**.

ClientLeft, **ClientTop**, **ClientWidth**, **ClientHeight** properties added. These properties allow easier development when not using **WizardPane**s or in those environments where container properties are not supported.

ShowGroove property added to allow control over the display of the 3D groove on the **WizardControl**.

Version 1.10:

Added **Stretch** property to the **WizardControl** to allow programmatic control over stretching of the Image.

WizardControl Properties

AutoSizeAtDesignTime
BackButtonCaption
BackButtonEnabled
BackButtonPaneIndex
BackButtonVisible
BorderAtDesignTime
Cancel
CancelCaption
CurrentPaneIndex
Default
FinishCaption
Image
IsFinishPane
NextButtonCaption
NextButtonEnabled
NextButtonPaneIndex
NextButtonVisible
Picture
Style
UseWizardPanes

WizardPane Properties

AutoSizeAtDesignTime
BackButtonEnabled
BackButtonPaneIndex
BackButtonVisible
BorderAtDesignTime
Caption
Image
IsFinishPane
NextButtonEnabled
NextButtonPaneIndex
NextButtonVisible
Picture
Style

AutoSizeAtDesignTime Property

Applies To

Returns or sets a value indicating whether or not the control will resize to its runtime size at design time.

Syntax

object.**AutoSizeAtDesignTime** [= *boolean*]

The **AutoSizeAtDesignTime** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the object resizes at design time or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Object enforces runtime size at design time.
False	Object can be resized at design time.

Remarks

Regardless of the **AutoSizeAtDesignTime** property value, both controls automatically set their size to a fixed value at runtime.

BackButtonCaption Property

Applies To

Determines the text displayed in the Back button.

Syntax

object.**BackButtonCaption** [= *string*]

The **BackButtonCaption** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
string	A string expression that evaluates to the text displayed as the Back button's caption.

Remarks

The default caption is "< &Back".

BackButtonEnabled Property

Applies To

Returns or sets a value indicating whether or not the Back button is enabled.

Syntax

object.**BackButtonEnabled** [= *boolean*]

The **BackButtonEnabled** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the Back button is enabled or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Back button is enabled.
False	Back button is disabled.

Remarks

This property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

The *first* **WizardPane** created on a form will have the **BackButtonEnabled** property set to **False**. All subsequently created **WizardPanes** will have the property set to **True**.

Tip

Set the **BackButtonEnabled** property to **True** when the **WizardPane** is the first pane in a sequence.

BackButtonPaneIndex Property

Applies To

Determines the **PaneIndex** of the **WizardPane** to activate when the user clicks the Back button.

Syntax

object.**BackButtonPaneIndex** [= *long*]

The **BackButtonPaneIndex** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
long	A long integer expression that evaluates to the **PaneIndex** of the **WizardPane** to activate when the **WizardControl's** Back button is clicked.

Remarks

This property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

BackButtonVisible Property

Applies To

Returns or sets a value indicating whether or not the Back button is visible.

Syntax

object.**BackButtonVisible** [= *boolean*]

The **BackButtonVisible** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the Back button is visible or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Back button is visible.
False	Back button is not visible.

Remarks

This property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

BorderAtDesignTime Property

Applies To

Returns or sets a value indicating whether or not the control will display a border at design time.

Syntax

object.**BorderAtDesignTime** [= *boolean*]

The **BorderAtDesignTime** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the object shows a border at design time or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Object has a border at design time.
False	Object does not have a border at design time.

Remarks

Regardless of the **BorderAtDesignTime** property value, both controls do not show a border at runtime.

Cancel Property

Applies To

Returns or sets a value indicating whether the **WizardControl** acts as the Cancel button on a form.

Syntax

object.**Cancel** [= *boolean*]

The **Cancel** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the object behaves as the Cancel button or not.

Settings

The settings for *boolean* are:

Setting	Description
True	Object behaves as the Cancel button.
False	(Default) Object does not behave as the Cancel button.

Remarks

If the **Cancel** property is **True**, pressing ESC causes the **CancelClick** event to be raised.

CancelCaption Property

Applies To

Determines the text displayed in the Cancel button.

Syntax

object.**CancelCaption** [= *string*]

The **CancelCaption** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
string	A string expression that evaluates to the text displayed as the Cancel button's caption.

Remarks

The default caption is "Cancel"

Caption Property

Applies To

Determines the text displayed in the **WizardPane** control.

Syntax

object.**Caption** [= *string*]

The **Caption** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
string	A string expression that evaluates to the text displayed as the caption.

Remarks

The default caption is the **Name** of the **WizardPane**.

ClientLeft, Client Top, ClientHeight, ClientWidth Properties

Applies To

ClientLeft - returns the distance (in twips) between the left edge of the control and the pane area.

ClientTop - returns the distance (in twips) between the top edge of the control and the pane area.

ClientHeight - returns the height (in twips) of the pane area.

ClientWidth - returns the width (in twips) of the pane area.

Syntax

object.**ClientLeft**

object.**ClientTop**

object.**ClientWidth**

object.**ClientHeight**

The **ClientLeft**, **ClientTop**, **ClientWidth**, and **ClientHeight** property syntaxes have these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.

Remarks

These properties are new for version 1.2.

CurrentPaneIndex Property

Applies To

Returns or sets the **PaneIndex** of the **WizardPane** that is currentlyactive.

Syntax

object.**CurrentPaneIndex** [= *long*]

The **CurrentPaneIndex** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
long	A long integer expression that evaluates to the **PaneIndex** of the active **WizardPane**.

Remarks

Changing the **CurrentPaneIndex** property while **UseWizardPanes** is **True** causes a series of events to occur.

The **matching WizardPane** control's **Validate** event is raised. If the Cancel flag was not set, the **BeforePaneChange** event is raised. Again, if the Cancel flag was not set, the control is deactivated. If the WizardControl is unable to find a WizardPane with a **PaneIndex** property matching the new index, the **WizardPaneError** event is raised. If a **matching WizardPane** is found, it is activated and its **Reposition** event is raised. Finally, the WizardControl's **AfterPaneChange** event is raised.

If **UseWizardPanes** is **False**, then only the **BeforePaneChange** and **AfterPaneChange** events are raised.

Default Property

Applies To

Returns or sets a value indicating whether the **WizardControl's** Next button acts as the default button on a form.

Syntax

object.**Default** [= *boolean*]

The **Default** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the control's Next button behaves as the Default button or not.

Settings

The settings for *boolean* are:

Setting	Description
True	Control's Next button behaves as the Default button.
False	(Default) Control's Next button does not behave as the Default button.

Remarks

none.

FinishCaption Property

Applies To

Determines the text to be displayed in the control's Next button when **IsFinishPane** is **True**.

Syntax

object.**FinishCaption** [= *string*]

The **FinishCaption** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
string	A string expression that evaluates to the text to be used when **IsFinishPane** is **True**.

Remarks

The default caption is “Finish”

Image Property

Applies To

Returns or sets a graphic to be displayed on the left-hand side of the **WizardControl**.

Syntax

Set *object*.**Image** [= *picture*]

The **Image** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
picture	A string expression specifying a file containing a graphic, as described in Settings.

Settings

The settings for *picture* are:

Setting	Description
(none)	(Default) No picture.
(Bitmap, icon, metafile, GIF, JPEG)	Specifies a graphic. You can load the graphic from the Properties window at design time. At run time, you can also set this property using the **LoadPicture** function on a bitmap, icon, or metafile.

Remarks

The standard image size is 116x224 pixels. You can use the **Stretch** property (new in version 1.10) to ensure irregularly sized pictures fit.

This property is taken from the **WizardPane** and used by the **WizardControl** when the pane is activated. Although the new image is shown, the WizardControl's **Image** property value remains unchanged. You can restore the property value using the **ResetImage** method.

IsFinishPane Property

Applies To

Returns or sets a value indicating whether or not the Next button should behave as the Finish button.

Syntax

object.**IsFinishPane** [= *boolean*]

The **IsFinishPane** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the Next button is the Finish button or not.

Settings

The settings for *boolean* are:

Setting	Description
True	Finish button behavior is enabled.
False	(Default) Finish button behavior is disabled.

Remarks

This property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

When **IsFinishPane** is **True**, the Next button's caption is set to the value of the **FinishCaption** property. When the Next button is clicked, the **FinishClick** event is raised (after a successful **Validate** event).

Tip

Set the **IsFinishPane** property to **True** when the **WizardPane** is the last pane in a sequence.

NextButtonCaption Property

Applies To

Determines the text to be displayed in the control's Next button when **IsFinishPane** is **False**.

Syntax

object.**NextButtonCaption** [= *string*]

The **NextButtonCaption** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
string	A string expression that evaluates to the text displayed as the Next button’s caption.

Remarks

The default caption is “&Next >”.

NextButtonEnabled Property

Applies To

Returns or sets a value indicating whether or not the Next button is enabled.

Syntax

object.**NextButtonEnabled** [= *boolean*]

The **NextButtonEnabled** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the Next button is enabled or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Next button is enabled.
False	Next button is disabled.

Remarks

This property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

NextButtonPaneIndex Property

Applies To

Determines the **PaneIndex** of the **WizardPane** to activate when the user clicks the Next button.

Syntax

object.**NextButtonPaneIndex** [= *long*]

The **NextButtonPaneIndex** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
long	A long integer expression that evaluates to the **PaneIndex** of the **WizardPane** to activate when the **WizardControl's** Next button is clicked.

Remarks

This property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

This property is ignored if **IsFinishPane** is **True**.

NextButtonVisible Property

Applies To

Returns or sets a value indicating whether or not the Next button is visible.

Syntax

object.**NextButtonVisible** [= *boolean*]

The **NextButtonVisible** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the Next button is visible or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Next button is visible.
False	Next button is not visible.

Remarks

This property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

PaneIndex Property

Applies To

Returns or sets a value uniquely identifying this control on the form.

Syntax

object.**PaneIndex** [= *long*]

The **PaneIndex** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
long	A long integer expression that evaluates to the PaneIndex.

Remarks

The uniqueness of this property is enforced by the control. Attempting to set this property to an invalid value raises a runtime error.

This property is used by the **WizardControl** when attempting to activate or deactivate a control or raise its **Reposition** and **Validate** events.

Picture Property

Applies To

Returns or sets a graphic to be displayed in the object.

Syntax

Set *object*.**Picture** [= *picture*]

The **Picture** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
picture	A string expression specifying a file containing a graphic, as described in Settings.

Settings

The settings for *picture* are:

Setting	Description
(none)	(Default) No picture.
(Bitmap, icon, metafile, GIF, JPEG)	Specifies a graphic. You can load the graphic from the Properties window at design time. At run time, you can also set this property using the **LoadPicture** function on a bitmap, icon, or metafile.

Remarks

none.

ShowGroove Property

Applies To

Returns or sets a value indicating whether or not the 3D “groove” is visible on the **WizardControl**.

Syntax

object.**ShowGroove** [= *boolean*]

The **ShowGroove** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the groove is visible or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) The groove is visible.
False	The groove is not visible.

Remarks

This property is new for version 1.2.

Stretch Property

Applies To

Returns or sets a value indicating whether or not the **Image** picture will be stretched to fit the image area.

Syntax

object.**Stretch** [= *boolean*]

The **Stretch** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the picture will be stretched or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) The **Image** is stretched to fit.
False	The **Image** is clipped to fit within the available area.

Remarks

In version 1.2 and higher of the control, this property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

This property is new for version 1.1.

Style Property

Applies To

Returns or sets a value that determines if the **WizardControl's** image will be shown and the **WizardPane** expanded to fill the available area.

Syntax

object.**Style** [= *integer*]

The **Style** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
integer	Determines the style to be used, as described in Settings.

Settings

The settings for *integer* are:

Constant	Value	Description
wxWithPicture	0	(Default) The image is shown.
wxNoPicture	1	The image is not shown. The **WizardPane** control expands to fill the available area.

Remarks

This property is taken from the **WizardPane** and applied to the **WizardControl** when the pane is activated.

UseWizardPanes Property

Applies To

Returns or sets a value indicating whether or not the object will control **WizardPane** controls on the form.

Syntax

object.**UseWizardPanes** [= *boolean*]

The **UseWizardPanes** property syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the object will control **WizardPanes** or not.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Object controls **WizardPanes**.
False	Object does not control **WizardPanes**.

Remarks

If the **UseWizardPanes** property value is **False**, the **WizardPaneError** event will never be raised.

WizardControl Methods

About
ResetImage

WizardPane Methods

About

About Method

Applies To

Displays the About box for the control.

Syntax

object.**About**

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Remarks

This is the same as clicking About in the Properties window.

ResetImage Method

Applies To

Restores the image area of the **WizardControl** to its **Image** property value.

Syntax

object.**ResetImage**

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Remarks

none.

WizardControl Events

AfterPaneChange
BeforePaneChange
CancelClick
FinishClick
WizardPaneError

WizardPane Events

Reposition
Validate

AfterPaneChange Event

Applies To

Occurs after the **CurrentPaneIndex** property has changed.

Syntax

Private Sub *object*_**AfterPaneChange(**[*index* **As Integer**,] *newpane* **As Long)**

The **AfterPaneChange** event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	An integer that identifies a control if it is in a control array.
newpane	A long integer that contains the value of the **CurrentPaneIndex** property.

Remarks

If the **UseWizardPanes** property is **True**, this event occurs after the **Reposition** event of the **matching WizardPane** is raised.

BeforePaneChange Event

Applies To

Occurs before the **CurrentPaneIndex** property is changed.

Syntax

Private Sub *object*_**BeforePaneChange(**[*index* **As Integer**,] *oldpane* **As Long**, *newpane* **As Long**, *cancel* **As Boolean)**

The **BeforePaneChange** event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	An integer that identifies a control if it is in a control array.
oldpane	A long integer that contains the value of the **CurrentPaneIndex** property.
newpane	A long integer that contains the new **CurrentPaneIndex** property value. This value can be changed by the user.
cancel	A Boolean expression expression that specifies whether the change occurs, as described in Settings.

Settings

The settings for *cancel* are:

Setting	Description
True	Cancels the change.
False	(Default) Continues with change.

Remarks

If the **UseWizardPanes** property is **True** and this change is occuring because of the user clicking the Next button, this event will occur after the **Validate** event of the **WizardPane** with a **PaneIndex** property value matching *oldpane*.

If the **UseWizardPanes** property is **True** and no **WizardPane** can be found with a **PaneIndex** matching *newpane*, the **WizardPaneError** event will be raised.

CancelClick Event

Applies To

Occurs when the Cancel button is clicked.

Syntax

Private Sub *object*_**CancelClick(**[*index* **As Integer**,]**)**

The **CancelClick** event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	An integer that identifies a control if it is in a control array.

Remarks

none.

FinishClick Event

Applies To

Occurs when the Next button is clicked and the **IsFinishPane** property is **True**.

Syntax

Private Sub *object*_**FinishClick(**[*index* **As Integer**,]**)**

The **FinishClick** event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	An integer that identifies a control if it is in a control array.

Remarks

none.

Reposition Event

Applies To

Occurs when the control is activated.

Syntax

Private Sub *object*_**Reposition(**[*index* **As Integer**,]**)**

The **Reposition** event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	An integer that identifies a control if it is in a control array.

Remarks

A **WizardControl** will not activate a **WizardPane** until it is first shown or its **CurrentPaneIndex** property is changed.

Validate Event

Applies To

Occurs to the **matching WizardPane** when the user clicks the Next button on a **WizardControl**, before the **BeforePaneChange** or **FinishClick** events.

Syntax

Private Sub *object*_**Reposition(**[*index* **As Integer**,], *cancel* **As Boolean)**

The **Reposition** event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	An integer that identifies a control if it is in a control array.
cancel	A Boolean expression expression that specifies whether the pane change or finish sequences occur, as described in Settings.

Settings

The settings for *cancel* are:

Setting	Description
True	Cancels the sequence. No further events are raised.
False	(Default) Continues with sequence.

Remarks

If the **WizardControl's IsFinishPane** property is **True** and *cancel* is **False**, the **FinishClick** event will be raised.

If **IfFinishPane** is **False**, the following sequence of actions will occur:

The **BeforePaneChange** will occur for the **WizardControl**. The *newpane* argument will be the control's **NextPaneIndex** property value.

If the sequence was not cancelled, this control will be deactivated.

The **WizardPane** who's **PaneIndex** property matches the *newpane* argument will be activated. The control's **Reposition** event will be raised.

The **WizardControl's AfterPaneChange** event will be raised.

WizardPaneError Event

Applies To

Occurs when the **UseWizardPanes** property is **True** and the control cannot find a **WizardPane** with a **PaneIndex** property matching the requested **CurrentPaneIndex**.

Syntax

Private Sub *object*_**WizardPaneError(**[*index* **As Integer**,]**)**

The **WizardPaneError** event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	An integer that identifies a control if it is in a control array.

Remarks

none.

Constants

Style property

Constant	Value	Description
wxWithPicture	0	(Default) The image is shown.
wxNoPicture	1	The image is not shown. The **WizardPane** control expands to fill the available area.

BackStyle property

Constant	Value	Description
wxTransparent	0	Transparent - background color and any graphics are visible behind the control.
wxOpaque	1	(Default) Opaque - the control's **BackColor** property setting fills the control and obscures any color or graphics behind it.

How the controls work together

The **WizardControl** and **WizardPane** communicate with each other to automate most of the chores involved with writing wizard display code -- showing and positioning elements, determining which pane of the wizard is active, and figuring out what to display next.

The **WizardControl** causes the **WizardPane** to raise events based on actions *that occur on the WizardControl*.

For example, if a user clicks the **WizardControl's** Next button, the **WizardControl** will attempt to locate the current **WizardPane** and if it finds it, it will force the **WizardPane** to raise its Validate event.

The **WizardPane** is "read" by the **WizardControl** when it is activated, and many of the **WizardControl's** properties are set to the activated **WizardPane's** property values.

Understanding the event cycle

Examine the WXEVENTS.VBP project to see the event cycle in action.

Notes:
Events are raised when the **CurrentPaneIndex** property is change or the first time that the control is made **Visible**.

When the control is first shown, the **BeforePaneChange** event is raised. Both *oldindex* and *newindex* will be the same as the **CurrentPaneIndex** property.

If the user clicks the **WizardControl's** Next button (and **IsFinishPane** is **False)**, the control will attempt to set the **CurrentPaneIndex** to the **NextButtonPaneIndex** property value. If the user clicks the Back button, the control will try the **BackButtonPaneIndex** property value. The **CurrentPaneIndex** property value can also be changed at runtime.

If the **WizardControl's** Next button is clicked, and **UseWizardPanes** is **True**, then the **matching WizardPane's Validate** event is raised. If no matching **WizardPane** was not found, *an error is not raised.* If the **WizardControl's** Back button is clicked, or the **CurrentPaneIndex** property is changed by the program, no **Validate** event will be raised.

If the action was not cancelled by the event (assuming the Next button was clicked), the **BeforePaneChange** event is raised. *Newindex* is the value of the **NextButtonPaneIndex** property. If the action was not cancelled, the **matching WizardPane** is deactivated. If the new **WizardPane** is not found, the action is cancelled and the **WizardPaneError** event is raised.

If the **WizardPane** is found, the following properties are read and the **WizardControl's** properties are set to their values:

BackButtonEnabled
BackButtonPaneIndex
BackButtonVisible
Image
IsFinishPane
NextButtonEnabled
PaneIndex
NextButtonPaneIndex
NextButtonVisible
Style

After the properties are transferred, the new control is activated and its **Reposition** event is raised. Finally the **AfterPaneChange** event is raised.

If the **UseWizardPanes** property is **False**, then only the **BeforePaneChange** and **AfterPaneChange** events are raised, in that order.

Runtime Distribution Requirements

This release of LDrive WizardX was compiled using Microsoft Visual Basic 5 Service Pack 3. No additional files other than the VB 5 runtime files are needed. A .DEP file was installed by Setup.

Differences Between the VB5 and VB6 Versions

LDrive WizardX is available in both VB5 and VB6 runtime versions. The control license allows either, or both, versions to be shipped with your products. This page explains the differences between the two.

Note: Either version of the controls can be used with *any* OLE-based hosting environment. The term "Version" refers only to the runtime requirements of the control. VB5-based versions require MSVBVM50.DLL and OLE 2.2+, while VB6-based versions require MSVBVM60.DLL and OLE 2.3+. Information on runtime file versions can be found in the .DEP file shipped with the control.

Distinct GUIDs and ProgIDs

Each version uses its own GUID. VB6 versions of the controls have a "6" appended to the VB5 ProgID. For example, the VB5 version ProgID for **WizardX** is "WizardX", whereas the VB6 version ProgID is "WizardX6". In either version, the control name (in the example, "WizardControl" or "WizardPane") remains the same.

Using distinct ProgIDs and GUIDs allows the developer to use either or both controls in a project.

Unique File Names

VB6 versions append a "6" to the control's file name. For example, the VB6 version of the **WizardX** control is "WizardX6.ocx".

Common File Versions

File major, minor, and build version numbers (e.g. 1.10.0056) are shared between all versioned builds of the control. This reduces confusion when updating libraries.

Distinct Library Descriptions

VB6 versions of controls are identified in the **Components** and **References** dialogs with a "(VB6)" designator. Version resource information also identifies a VB6 version control.

Separate Distribution Packages

Each package contains version-appropriate help and samples (if available) created with the control included in the package. VB6 shortcuts created by Setup are identified with a "(VB6)" designator.

Known Issues

Issue

The **WizardPane's Caption** and the **WizardControl's** grooved line do not display when the appropriate control's **BackStyle** property is set to 1 = **wxTransparent**.

Status

This problem is due to the way VB supports transparency. You can add your own controls to duplicate the missing elements.

Registration and Support

For complete information on LDrive software products, including updates, pricing, payment options, and availability visit the LDrive web site at:

http://www.ldrive.com

You can also send check or money order along with your name, company, address and e-mail address to:

LogicDriven Incorporated
49 Alafaya Woods Blvd.
Suite 273
Oviedo, FL 32765

Don't forget to indicate which LDrive product you are purchasing.

Contact us via e-mail at: **info@ldrive.com**

Bug reports, feedback, and suggestions are welcome from anyone, however, support is provided only to registered users. Send your comments, questions, observations, gripes and complaints to:

feedback@ldrive.com

LogicDriven Software License Agreement

This License Agreement is a legal agreement between you (either an individual or a single entity) and LogicDriven, Incorporated (“LogicDriven”). for the software product identified above, which includes computer software and "online" or electronic documentation and may include associated media and printed materials ("the Software"). By installing, copying, or otherwise using the Software, you agree to be bound by the terms of this License Agreement.

LogicDriven Software License

LogicDriven software and the separately copyrighted components included therewith, and any enhancements or modifications thereto, whether delivered electronically or otherwise ("the Software") and the User Documentation accompanying the Software ("the User Documentation") are licensed to you for your personal use only.

Copyright laws and international copyright treaties, as well as other intellectual property laws and treaties protect the Software. The Software is licensed, not sold.

This Agreement grants you the following limited, non-exclusive rights:

You may install and use the Software on any number of computers, and may make any number of copies of the Software, provided that only one copy of the software is in use by you at any one time.

You may modify the sample source code included with the Software to design, develop, and test your Application. You may also reproduce and distribute your derivative works of the Sample Code in source and object code forms, provided that you comply with the Distribution Requirements described below. For purposes of this section, "modifications" shall mean enhancements to the functionality of the Sample Code.

Portions of the Software are designated as "Redistributable". Your rights to distribute the Redistributable Code are subject to the following requirements:

You may copy and redistribute the Sample Code and Redistributable Code (collectively "Redistributable Components") as described above provided that: (a) you distribute the Redistributable Components only in conjunction with and as a part of your Application; (b) you agree to indemnify, hold harmless, and defend LogicDriven from and against any claims or lawsuits, including attorney's fees, that arise or result from the use or distribution of your Application; (c) you grant third parties the rights to redistribute the Redistributable Components, which they acquire from your Application, only pursuant to terms no less restrictive as those contained in this Agreement.

LogicDriven retains all rights not expressly granted.

Limited Warranty

LogicDriven warrants that the Software will conform substantially to published specifications, documentation, and authorized advertising. If the Software and User Documentation was distributed in a non-electronic form, that media shall be free from defects in materials and workmanship for a period of ninety (90) days from the date of delivery to you.

Except for the limited warranty set forth above, the software and the user documentation are provided "as is".

LogicDriven makes no other warranty, either express or implied, with respect to the Software and/or the User Documentation and specifically disclaims the implied warranties of merchantability and fitness for a particular purpose. LogicDriven does not warrant that the Software and/or the User Documentation will meet your requirements or expectations or that the operation of the software will be uninterrupted and/or error free. You are solely responsible for the selection of the software to achieve your intended results and for the results actually obtained.

Some states do not allow the exclusion or limitation of implied warranties, so the above exclusions and limitations may not apply to you. This warranty gives you specific rights, and you may also have other rights that vary from state to state.

Limitation of Remedies and Liability

In the event the Software and/or the User Documentation fail to meet the Limited Warranty, LogicDriven's entire liability and your exclusive remedies shall be:

A) The replacement of the Software and/or the User Documentation not meeting LogicDriven's Limited Warranty, which is returned to LogicDriven with your proof of payment.

B) If LogicDriven cannot or will not replace the Software and/or User Documentation, you may terminate this Agreement by returning the Software and the User Documentation to LogicDriven with your proof of payment, and your license fee will be refunded.

Under no circumstances, and notwithstanding any failure of the essential purpose of any limited remedy provided for herein, shall LogicDriven be liable to you for any damages, claims or losses whatsoever, including but not limited to any claims for lost profits, lost savings or other special incidental or consequential damages arising out of the use or inability to use the Software and/or the User Documentation regardless of the circumstances.

General

In the event it is determined that any provision contained in this license is unlawful, void, or unenforceable, such determination shall solely affect such unlawful, void, or unenforceable provision and shall not affect the validity or enforceability of the remaining provisions of this license. The laws of the State of Florida will govern this agreement.