

## headline area

Upper part of a **NewsList** control where text from the **Headline** property of a **NewsItem** object is displayed.

## **byline area**

Part of a **NewsList** control below the **headline area** and above the **body area** where text from the **Byline** property of a **NewsItem** object is displayed.

## **body area**

Lower part of a **NewsList** control where text from the **Headline** property of a **NewsItem** object is displayed.

## **Kheadline footer**

The area just below the **headline area** and just above the **byline area**.

## Kbyline footer

The area just below the **byline area** and just above the **body area**.

**standard property or method**

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

## **LDrive NewsList**

Roll your own news!

LDrive NewsList is an ActiveX control that displays pages of user-defined text in a newspaper-like fashion. It can resize automatically, and rotate through an internal collection of text items on a timer.

[Properties](#)

[Methods](#)

[Events](#)

[Objects](#)

[NewsList layout](#)

## NewsList Properties

AutoSize  
BackColor  
BodyColor  
BodyFont  
BorderWidth  
BylineFont  
BylineColor  
BylineFooter  
CurrentNewsItem  
HeadlineColor  
HeadlineFont  
HeadlineFooter  
Interval  
NewsItems  
Pause

## NewsList Methods

[About](#)

[ShowFirst](#)

[ShowNext](#)

[ShowPrevious](#)

[ShowNewsItem](#)

## NewsList Events

Click

DbClick

ShowItem

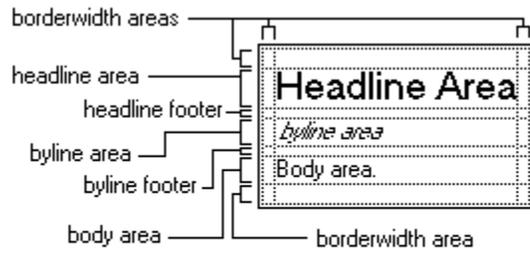
## NewsList Objects

NewsItemCollection

NewsItem

## NewsList layout

The **NewsList** control consists of a series of areas as shown in the diagram below.



## AutoSize Property

Returns or sets a value that determines whether the height of the control is automatically resized to display its contents.

### Syntax

*object*.**AutoSize** [= *integer*]

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

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>boolean</i>	A Boolean expression specifying whether the object resizes at design time or not.

### Settings

The settings for *boolean* are:

<b><u>Setting</u></b>	<b><u>Description</u></b>
<b>True</b>	Automatically resizes the height of the control to display its entire contents whenever a new <b>NewsItem</b> is shown.
<b>False</b>	(Default) Keeps the size of the control constant. Contents are clipped when they exceed the area of the control.

### Remarks

none.

## BackColor, BodyColor, BylineColor, HeadlineColor Properties

**BackColor** - returns or sets the background color of the control.

**BodyColor** - returns or sets the foreground color used to display text in the body area of the control.

**BylineColor** - returns or sets the foreground color used to display text in the byline area of the control.

**HeadlineColor** - returns or sets the foreground color used to display text in the headline area of the control.

### Syntax

*object*.**BackColor** [= *color*]

*object*.**BodyColor** [= *color*]

*object*.**BylineColor** [= *color*]

*object*.**HeadlineColor** [= *color*]

The **BackColor**, **BodyColor**, **BylineColor**, and **HeadlineColor** property syntaxes have these parts:

<b>Part</b>	<b>Description</b>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>color</i>	A value or constant that determines the background or foreground colors of an object, as described in Settings.

### Settings

**NewsList** uses the standard Visual Basic/Microsoft Windows operating environment red-green-blue (RGB) color scheme. The settings for *color* are:

<b>Setting</b>	<b>Description</b>
Normal RGB colors	Colors specified by using the Color palette or by using the <b>RGB</b> or <b>QBColor</b> functions in code.
System default colors	Colors specified by system color constants listed in the Visual Basic (VB) object library in the Object Browser. The Windows operating environment substitutes the user's choices as specified in the Control Panel settings.

The default settings are:

**BackColor** - set to the system default color specified by the constant **vbButtonFace**.

**BodyColor**, **BylineColor**, and **HeadlineColor** - set to the system default color specified by the constant **vbButtonText**.

### Remarks

none.

## BodyFont, BylineFont, HeadlineFont Properties

**BodyFont** - returns the **Font** object used to display text in the body area of the control.

**BylineFont** - returns the **Font** object used to display text in the byline area of the control.

**HeadlineFont** - returns the **Font** object used to display text in the headline area of the control.

### Syntax

*object*.**BodyFont**

*object*.**BylineFont**

*object*.**HeadlineFont**

The object placeholder represents an object expression that evaluates to a **NewsList** object.

### Remarks

Use these properties to identify the specific **Font** object whose properties you want to use. For example, the following code changes the **Bold** property setting of the **Font** object identified by the byline area of a **NewsList** object:

```
nlNewItems.BylineFont.Bold = True
```

## BorderWidth Property

Returns or sets the number of pixels between the control's border and any text displayed.

### Syntax

*object*.**BorderWidth** [= *integer*]

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

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>integer</i>	An Integer expression specifying the border width, in pixels.

### Remarks

none.

## BylineFooter, HeadlineFooter Properties

**BylineFooter** - returns or sets the number of pixels between the byline area and the body area of the control.

**HeadlineColor** - returns or sets the number of pixels between the headline area and the byline area of the control.

### Syntax

*object*.**BylineFooter** [= *integer*]

*object*.**HeadlineFooter** [= *integer*]

The **BylineFooter** and **HeadlineFooter** property syntaxes have these parts:

<u>Part</u>	<u>Description</u>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>integer</i>	An Integer expression specifying the new distance, in pixels.

### Remarks

Valid *index* values range from 1 to **LineCount**. Note that **LineItems** is an array and not a collection, so the **For Each** syntax is not supported.

## CurrentNewsItem Property

Returns the index of the **NewsItem** currently displayed. Read-only.

### Syntax

*object*.**CurrentNewsItem**

The object placeholder represents an object expression that evaluates to a **NewsList** object.

### Remarks

**CurrentNewsItem** returns 0 when there is no **NewsItem** displayed.

Use the **ShowFirstNewsItem**, **ShowNextNewsItem**, **ShowPreviousNewsItem**, and **ShowNewsItem** methods to change the current **NewsItem** manually.

## Interval Property

Returns/sets the number of milliseconds before the next **NewsItem** is shown.

### Syntax

*object*.**Interval** [= *milliseconds*]

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

<b>Part</b>	<b>Description</b>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>milliseconds</i>	A numeric expression specifying the number of milliseconds, as described in Settings.

### Settings

The settings for *milliseconds* are:

<b>Setting</b>	<b>Description</b>
0	Automatic display is disabled.
1 to 65,535	Sets an interval (in milliseconds) before the next <b>NewsItem</b> is shown. For example, a value of 10,000 milliseconds equals 10 seconds. The maximum, 65,535 milliseconds, is equivalent to just over 1 minute.

The default value for *milliseconds* is 5,000, or 5 seconds.

### Remarks

The **Interval** property is ignored while **Pause** is **True**.

## NewsItems Property

Returns a reference to a collection of **NewsItem** objects in a **NewsList** control.

### Syntax

*object*.**NewsItems**

The object placeholder represents an object expression that evaluates to a **NewsList** object.

### Remarks

You can manipulate **NewsItem** objects using collection methods (for example, the **Add** and **Remove** methods) and the **For Each** enumeration syntax. Each member of the collection can be accessed by its index or unique key. The key is stored in the Key property (specified in the **Add** method).

Although the **NewsItems** property looks like a standard collection, it is not. Refer to the [NewsItemCollection](#) topic for more information.

## Pause Property

Returns or sets a value that determines if the control will automatically show its **NewsItems** in sequence according to the value of the **Interval** property.

### Syntax

*object.Pause* [= *boolean*]

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

<b>Part</b>	<b>Description</b>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>boolean</i>	A Boolean expression specifying whether the automatic display of <b>NewsItems</b> is paused, as described in Settings.

### Settings

The settings for *boolean* are:

<b>Setting</b>	<b>Description</b>
<b>True</b>	Automatic display of <b>NewsItems</b> is paused. When the time specified by <b>Interval</b> expires, no action is taken.
<b>False</b>	(Default) Automatic display is enabled.

### Remarks

Use **Pause** to temporarily “freeze” the display.

## About Method

Displays the About box for the control.

### Syntax

*object*.**About**

The *object* placeholder an object expression that evaluates to a **NewsList** object.

### Remarks

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

## ShowFirst, ShowNext, ShowPrevious, ShowNewsItem Methods

Display the first, next, previous, or specific **NewsItem** in the control's **NewsItems** collection, and make that index the **CurrentNewsItem**.

### Syntax

*object*.{**ShowFirst** | **ShowNext** | **ShowPrevious**}

*object*.**ShowNewsItem**(*index*)

The **ShowFirst**, **ShowNext**, **ShowPrevious**, and **ShowNewsItem** method syntaxes have these parts:

<u>Part</u>	<u>Description</u>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>index</i>	Required for <b>ShowNewsItem</b> only. An expression that specifies the position of a member of the collection. If a numeric expression, <i>index</i> must be a number from 1 to the value of the control's <b>NewsItems.Count</b> property. If a string expression, <i>index</i> must correspond to the <b>Key</b> argument specified when the member referred to was added to the collection.

### Remarks

If the value provided as *index* doesn't match any existing member of the collection, an error occurs.

The **ShowNext** and **ShowPrevious** methods will "wrap" around respectively to the beginning and end of the **NewsItems** collection. Neither they nor the **ShowFirst** method will return an error if there are no **NewsItems** in the control's collection.

## Click Event

Occurs when the user presses and then releases a mouse button over the object.

### Syntax

**Private Sub** *object*\_**Click**([*index As Integer*,] *controlarea As Integer*)

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

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>index</i>	An integer that identifies a control if it is in a control array.
<i>controlarea</i>	An integer that contains a number corresponding to the part of the control that was clicked, as described in Settings, below.

### Settings

The settings for *controlarea* are:

<b><u>Value</u></b>	<b><u>Description</u></b>
0	The underlying control was clicked.
1	The <b>headline area</b> was clicked.
2	The <b>byline area</b> was clicked.
3	The <b>body area</b> was clicked.

### Remarks

none.

## DbiClick Event

Occurs when the user presses and releases a mouse button and then presses and releases it again over an object.

### Syntax

**Private Sub *object*\_DbiClick(*[index As Integer,] controlarea As Integer*)**

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

<b><u>Part</u></b>	<b><u>Description</u></b>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>index</i>	An integer that identifies a control if it is in a control array.
<i>controlarea</i>	An integer that contains a number corresponding to the part of the control that was double-clicked, as described in Settings, below.

### Settings

The settings for *controlarea* are:

<b><u>Value</u></b>	<b><u>Description</u></b>
0	The underlying control was clicked.
1	The <b>headline area</b> was clicked.
2	The <b>byline area</b> was clicked.
3	The <b>body area</b> was clicked.

### Remarks

none.

## ShowItem Event

Occurs just after a **NewsItem** is shown.

### Syntax

**Private Sub** *object*\_**ShowItem**([*index* **As Integer**,] *headline* **As String**, *byline* **As String**, *body* **As String**, *itemtag* **As Variant**, *key* **As String**)

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

<b>Part</b>	<b>Description</b>
<i>object</i>	An object expression that evaluates to a <b>NewsList</b> object.
<i>index</i>	An integer that identifies a control if it is in a control array.
<i>headline</i>	A string that contains the value of the text just displayed in the <u>headline area</u> of the control.
<i>byline</i>	A string that contains the value of the text just displayed in the <u>byline area</u> of the control.
<i>body</i>	A string that contains the value of the text just displayed in the <u>body area</u> of the control.
<i>itemtag</i>	A variant that contains the value of the <b>ItemTag</b> property of the <b>NewsItem</b> object just shown.
<i>key</i>	A string that contains the value of the unique key of the <b>NewsItem</b> object just shown, or "" if no key was used.

### Remarks

none.

## NewsItemCollection Object

A **NewsItemCollection** object contains an ordered set of **NewsItem** objects that can be referred to as a unit.

### Remarks

Access to this object is provided by the **NewsItems** property.

Although new members added using the **Add** method can only be added to the end, any member can be removed using the **Remove** method. Specific members can be returned from the collection using the **Item** method, while the entire collection can be iterated using the **For Each...Next** statement.

Items from the collection are displayed by calling the **ShowFirst**, **ShowNext**, **ShowPrevious**, and **ShowNewsItem** methods.

### Properties

Count

### Methods

Add

Clear

Item

Remove

## NewsItem Object

A **NewsItem** object holds a single display's worth of information.

### Remarks

none.

### Properties

The **NewsItem** object has the following properties:

<u>Property</u>	<u>Description</u>
<i>headline</i>	A string that contains the value of the text to display in the <u>headline area</u> of the control.
<i>byline</i>	A string that contains the value of the text to display in the <u>byline area</u> of the control.
<i>body</i>	A string that contains the value of the text to display in the <u>body area</u> of the control.
<i>itemtag</i>	A variant that contains any extra data needed for your program.
<i>key</i>	A string that contains a value unique to this item in its parent collection, or "".

## **Runtime Distribution Requirements**

LDrive NewsList was compiled using Microsoft Visual Basic 5 (With the Service Pack 3 patch). 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 NewsList 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 **NewsList** is “NewsListLib”, whereas the VB6 version ProgID is “NewsListLib6”. In either version, the control name (in the example, “MsgBox”) 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 **NewsList** control is “NewsList6.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**

none.

## 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](mailto: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](mailto: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.

