MagicListbox

A Listbox Component from

Simon Berridge
simberr@gmail.com

RealStudio 2011 R4.3

Table of Contents

Introduction 4

New Properties 5

Property List Editor Properties 5

TableType 5
TextColor 5
BackColor 5
AlternateColor 6
HighlightColor 6
BoxColor 6
ProgBarColor 6
InitialValue 6
HideColumns 6
ColumnTypes 6
ColAlignment 7
CurrencyFormat 8
DateFormat 8
DateTimeFormat 8
TimeFormat 9
IntegerFormat 9
NumberFormat 9

Run-Time Methods and Properties 9

Reset (replaces DeleteAllRows) 9
NewRow (replaces AddRow) 9
ResizeColumns 9
SetDropDownList 10
SetPBMaxValue 10
SetValue (replaces Cell(row, column) =) 10
UpdateValue (replaces Cell(row, column) =) 10
GetValue (replaces x = Cell(row, column)) 11

Exported Methods 11

CellCheckBoxStateChange 11
CellDropDownChanged 11
CellDateChanged 11

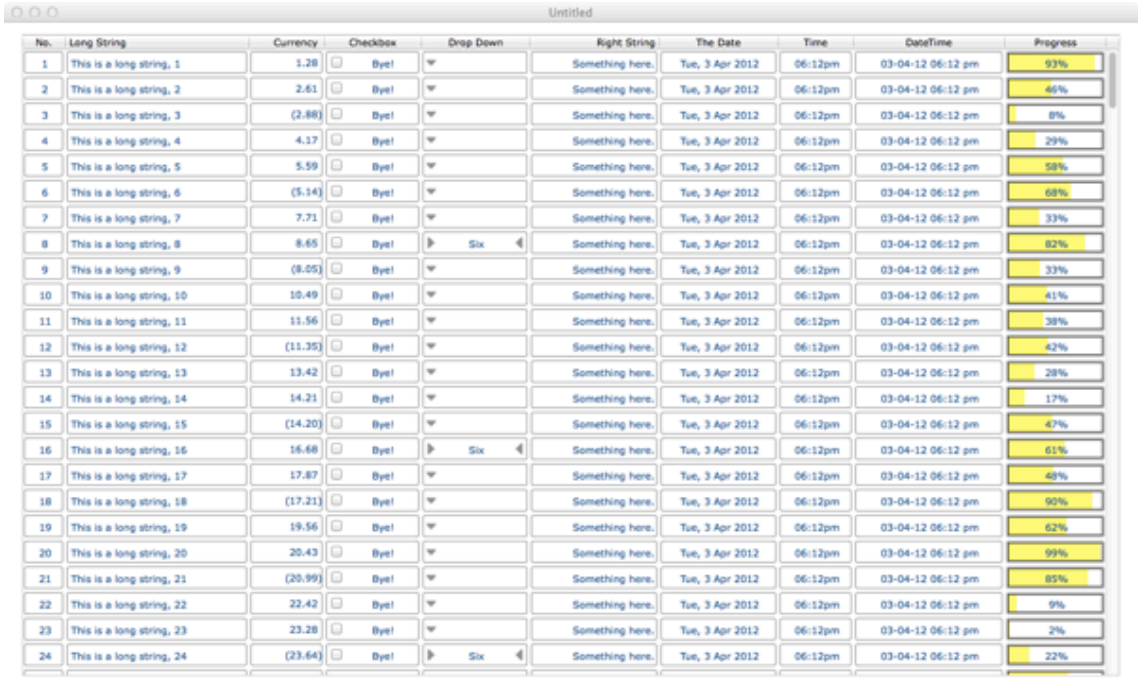
CellTimeChanged 11
CellCurrencyChanged 11
CellIntegerChanged 11
CellNumberChanged 12
CellTextChanged 12

Introduction

This is the help file for the MagicListbox listbox control created by Simon Berridge. You can contact Simon via email at:

simberr@gmail.com

Screenshot:



The basic enhancements to the standard listbox are:

- Ability to set the color of the text displayed
- Ability to set the background color of the table
- Ability to set the highlight color of the table
- Ability to set alternating colors
- Ability to outline the text with a round-cornered box
- Ability to outline the text with a square-cornered box

The control adds the following controls within the listbox:

- Drop Down List
- Check Box (enhanced)
- Progress Bar
- Date Selection drop-down
- Time Selection drop-down

In addition, MagicListbox allows you to specify the the type of column that is within the listbox. The types that can be specified are as follows:

- Integer
- String
- Currency
- Number
- Drop Down
- Check Box
- Date
- Time
- DateTime
- Progress Bar

It is possible to define whether the cells within the column can be edited and allows the user access to the values within the cells.

New Properties

The new properties that have been introduced to MagicListbox are defined below. This section is broken into two separate sections; those properties available within the property list editor (design time properties) and; those properties available within the code.

Property List Editor Properties

TableType

This property offers a drop-down list of three options: Normal; Rounded or; Square. A Normal list box will draw a table as usual with no lines around the cells (unless you specify these as normal). A Rounded selection will draw a round-cornered box around each cell item and a Square selection will draw a square-cornered box around each cell item.

TextColor

This will set the color of the text in the listbox.

BackColor

This will set the color of the listbox background.

AlternateColor

This will set the color of the alternate row listbox background.

HighlightColor

This will set the color of the highlight bar when selecting a row in the listbox.

BoxColor

If you have chosen a TableType of Rounded or Square then this property determines the color of the box that will be drawn around the cell item.

ProgBarColor

This will set the color of the progress bar specified.

InitialValue

This property has changed from the original use. It is now a comma-separated list of values to use for the column headings as opposed to the original tab-separated list.

HideColumns

It is normal to hide a column where you might keep a record key or some hidden data. This is normally achieved by setting its column width to zero. This property allows you to specify which columns should be set to zero by way of a comma separated list of the column numbers (zero based) that should be hidden. Only use this property if there are any columns that you wish to hide, and only specify the hidden columns.

For example: To hide columns 0 and 4 use:

```
0,4
```

ColumnTypes

This is a comma separated list of column types. The values are:

- I - Integer
- S - String
- C - Currency
- N - Number
- Dx - Drop Down. 'x' is L or R for Left or Right side of the cell. The default is Left.
- B - Check Box
- T - Date
- M - Time
- A - DateTime
- P - Progress Bar

If the column can be edited then use the column value plus the letter 'E'

- IE - Integer column, can be edited
- CE - Currency column, can be edited

Column types that DO NOT take the E extension:

- Dx - Drop Down
- B - Check Box
- P - Progress Bar

An example would be:

```
i,se,ce,b,d,s,te,me,ae,p
```

This example defines a 10 column MagicListbox. Each column has been defined as follows:

- i - Integer
- se - String, editable
- ce - Currency, editable
- b - Checkbox
- d - Drop Down, state shown on the left
- s - String, not editable
- te - Date, editable
- me - Time, editable
- ae -DateTime, editable
- p - Progress Bar

ColAlignment

This is a comma separated list of the alignments for each of the columns. The allowable values are as follows:

- F - Default
- L - Left
- R - Right
- C - Center
- Dx - Decimal aligned, the 'x' is a number representing the number of decimal places or characters that will follow the decimal point.

An example would be:

```
c,l,d3,c,c,r,c,c,c,c
```

This example is for a 10 column table. The columns will be aligned as follows:

- c - Centered
- l - Left
- d3 - Decimal aligned, 3 characters following the decimal point
- c - Centered

- c - Centered
- r - Right
- c - Centered
- c - Centered
- c - Centered
- c - Centered

CurrencyFormat

For a column that is defined as a currency column then the values in the cells within that column will be formatted by the format string specified in this property. The default format is: "`#,0.00;(#,0.00);\ `"

DateFormat

For a column that is defined as a Date column then the values in the cells within that column will be formatted by the format string specified in this property. The default format is: "`n, d mmm yyyy`"

Send	Result
n	Mon, Tue, Wed, Thu, Fri, Sat, Sun
nn	Monday, Tuesday, Wednesday...
d	Day in short numeric (i.e. 3 is "3")
dd	Day in "00" numeric (i.e. 3 is "03")
ddd	Day with speech (i.e. 3 - 3rd, 1 - 1st...)
m	Month in short numeric (i.e. 3 is "3")
mm	Month in "00" numeric (i.e. 3 is "03")
mmm	Jan, Feb, Mar ...
mmmm	January, February, March ...
yy	Two digit year (i.e. 2012 is "12")
yyyy	Four digit year (i.e. 2012 is "2012")

DateTimeFormat

For a column that is defined as a DateTime column then the values in the cells within that column will be formatted by the format string specified in this property. The default format is: "`dd-mm-yy hh:tt ap`"

Send	Result
n	Mon, Tue, Wed, Thu, Fri, Sat, Sun
nn	Monday, Tuesday, Wednesday...
d	Day in short numeric (i.e. 3 is "3")
dd	Day in "00" numeric (i.e. 3 is "03")
ddd	Day with speech (i.e. 3 - 3rd, 1 - 1st...)
m	Month in short numeric (i.e. 3 is "3")
mm	Month in "00" numeric (i.e. 3 is "03")
mmm	Jan, Feb, Mar ...
mmmm	January, February, March ...
yy	Two digit year (i.e. 2012 is "12")
yyyy	Four digit year (i.e. 2012 is "2012")
h	Hour in short format (i.e. 9 is "9")
hh	Hour in long format (i.e. 9 is "09")
tt	Minutes
ss	Seconds

```
ap          AM/PM indicator (i.e. "am" or "pm")
```

TimeFormat

For a column that is defined as a Time column then the values in the cells within that column will be formatted by the format string specified in this property. The default format is: "HH:MMap"

Send	Result
hh	Hour in long format (i.e. 9 is "09")
h	Hour in short format (i.e. 9 is "9")
mm	Minutes
ss	Seconds
ap	AM/PM indicator (i.e. "am" or "pm")

IntegerFormat

For a column that is defined as an Integer column then the values in the cells within that column will be formatted by the format string specified in this property. The default format is: "-#"

NumberFormat

For a column that is defined as a Number column then the values in the cells within that column will be formatted by the format string specified in this property. The default format is: "-#0.00"

Run-Time Methods and Properties

Reset (replaces DeleteAllRows)

Useage: LBox.Reset

Use this method rather than DeleteAllRows. Reset initialises all of the variables within MagicListbox.

NewRow (replaces AddRow)

Useage: LBox.NewRow

ResizeColumns

Useage: LBox.ResizeColumns

Used to automatically size the columns so that the data will fit. It attempts to size the columns so that they fit exactly within the allowable space. If the widths of the values combined exceeds the width of the available space then the routine will open the horizontal scrollbar to allow horizontal scrolling.

This method should be called after you have filled the MagicListbox with data.

In the event that the horizontal scrollbar is operated then a user resize of the window will not automatically resize the MagicListbox. If that is of concern then place the command also within the Window.Resizing or Window.Resized method.

SetDropDownList

Useage: LBox.SetDropDownList(row, column, List)

The List parameter is a comma separated list of Menu Items that will appear when the user clicks on the drop-down list icon within the cell.

An example would be:

```
"One,Two,-,Three,Four"
```

The above List would generate a drop-down (pop-up) menu with the four items and a separator between items Two and Three. When the user selects an item it will be displayed within the cell. You can initialise the cell value with either the SetValue or UpdateValue methods.

SetPBMaxValue

Useage: LBox.SetPBMaxValue(row, column, MaxVal)

This method is used to initialise the ProgressBar cell with a specific maximum value. The minimum value is always 0. When the table is initialised the value within the cell is initialised at zero. You may set the initial value with either SetValue or UpdateValue.

To update the progress bar dynamically just use the UpdateValue method. If you send a value to the cell that is greater than the cell's maximum allowable value then the progress bar will just show 100%.

SetValue (replaces Cell(row, column) =)

Useage: LBox.SetValue(Column,Value)

Value is any of the data types. All are accepted by the routine, no formatting is required or conversion to a string. You should ensure that the value passed to the cell is the same as the column type initialised earlier (see ColumnTypes).

This method is only used within the setup of the listbox as it only takes the Column parameter. It assumes that Listbox.LastIndex will be used. If you need to set a cell value without using NewRow first then use UpdateValue (described later).

UpdateValue (replaces Cell(row, column) =)

Useage: LBox.UpdateValue(Row, Column,Value)

Value is any of the data types. All are accepted by the routine, no formatting is required or conversion to a string. You should ensure that the value passed to the cell is the same as the column type initialised earlier (see ColumnTypes).

This method is used outside the setup of the listbox as it takes both the Row the Column parameters. See also SetValue.

GetValue (replaces x = Cell(row, column))

Useage: myVariable = LBox.GetValue(row, column)

This method returns the value associated with a particular cell. It returns a variant so it should be cast correctly.

Exported Methods

CellCheckBoxStateChange

Parameters Sent: row As Integer, column As Integer, newState As Boolean

When the user clicks on a cell that is a checkbox cell then this method is called. The newState parameter returns the state of the checkbox, True being checked and False being unchecked.

CellDropDownChanged

Parameters Sent: row As Integer, column As Integer, newValue As String

When the user selects a menu item from a drop-down list from within a Drop Down list cell then this method is called. The newValue parameter returns the text string of the menu item selected (which is also now shown within the cell).

CellDateChanged

Parameters Sent: newDate As Date, row As Integer, column As Integer

This event is fired if the user changes the date in the cell.

CellTimeChanged

Parameters Sent: newTime As Date, row As Integer, column As Integer

This event is fired if the user changes the time in the cell.

CellCurrencyChanged

Parameters Sent: newVal As Currency, row As Integer, column As Integer

This event is fired if the user changes the currency value in the cell.

CellIntegerChanged

Parameters Sent: newVal As Integer, row As Integer, column As Integer

This event is fired if the user changes the integer value in the cell.

CellNumberChanged

Parameters Sent: newVal As Double, row As Integer, column As Integer

This event is fired if the user changes the number (double) value in the cell.

CellTextChanged

Parameters Sent: newText As String, row As Integer, column As Integer

This event is fired if the user changes the text in the cell.