axGrid Control

Properties Methods Events

axGrid is an standalone unbound grid which allows the user to edit cells, mask cells, and provide listboxes for selections.

File Name
axGrid.OCX

Remarks

Distribution
- Unzip the source code with directories intact.
- Load the sample Project1 with Visual Basic. This should register the OCX for use with other VB applications. If it does not load correctly, then from the DOS prompt, move to the install directory and type: **regsvr32 axGrid.ocx** (you may need to put **c:\windows\system\** in front of regsvr32 if you don't have a path.
- Whenever you want to add the control to a VB application, go to the Project/Components menu and select "ActiveX Grid Control".

Revisions
License
Tech Support

Revisions

2.0

- Initial release

2.0.16

- Added Remove row capability
- Added checkbox mask

2.0.18

- Dropdown button shows at bottom of grid when first placed on a form, if grid is tall enough

License

This control was developed and published by Software Solution. You may use it freely for development with Microsoft Visual Basic 5.0. This product is freeware and includes source code which you may change to suit your purposes.

Tech Support

If you have any problems installing or using this control, please feel free to contact our technical support department at one of the following:

Internet:
kirkq@execpc.com

Telephone:
414-251-0915

Snail Mail:
N92W17053 Roger Ave
Menomonee Falls, WI 53051

HEY! Check out our world wide web page at:
HTTP://www.execpc.com/~kirkq

Properties

All of the properties for this control are listed below:

Standard
BackColor
DragIcon
Dragmode
Enabled
Font
ForeColor
Height
Index
Left
MousePointer
Name
TabIndex
Tabstop
Tag
ToolTipText
Visible
Width

Control Specific
AllowSelection
AllowUserResizing
AutoNewRow
BackColorBkg
BackColorFixed
BorderStyle
Col
ColAlign
ColAllowEdit
ColHeader
ColMask
Cols
ColWidth
FixedStyle
FontFixed
ForeColorFixed
FormatString
GridLineColor
GridSolid
LeftCol
ListBoxRows
RowHeader
RowHeight
Rows
SelectionMode
ShowGrid
Text
TextMatrix
TopRow

Events

All of the events for this control are listed below:

Standard
Click
DblClick
DragDrop
DragOver
GotFocus
KeyDown
KeyPress
KeyUp
MouseDown
MoveMove
MouseUp
LostFocus
Scroll

Control Specific
AfterAddRow
AfterDeleteRow
AfterEdit
BeforeAddRow
BeforeDeleteRow
BeforeEdit

Standard Property/Method/Event

Depending on your host environment, this property/method/event may be referred to by a different name or may not apply to this control. Refer to your host environments documentation or help file on MSFlexGrid for further information.

Methods

All of the methods for this control are listed below:

Standard
DblClick
Drag
Move
Refresh
SetFocus
ShowWhatsThis
Zorder

Control Specific
AddLookup
AutoSetup
ClearAllLookups
ClearLookup
ColHasLookup
GetColWidth
RemoveLookup
Remove
RowEmpty
ShowAboutBox

AutoNewRow Property

Returns or sets the AutoNewRow property for an object

Syntax:

object.**AutoNewRow** [= boolean]

The AutoNewRow property syntax has these parts:

Part	Description
object	An object expression that evaluates to a axGrid
boolean	A boolean expression that evaluates to True/False to indicate if the grid will autocreate new rows

Remarks

If this property is True, then the grid will autocreate a new row if the last row is empty and the user tries to edit a cell in the last row.

ColAllowEdit Property

Returns or sets the ColAllowEdit property for an object

Syntax:

object.**ColAllowEdit**(col as integer) [= boolean]

The ColAllowEdit property syntax has these parts:

Part	**Description**
object	An object expression that evaluates to a axGrid object
col	**An integer expression that evaluates to a specific column number for the grid. This value is zero based.**
boolean	**A boolean expression that evaluates to whether or not the column allows editing**

Remarks

This property will allow the user to determine which columns allow editing. The value is True by default when a grid is created or when new columns are added.

ShowButtons Property

Returns or sets the ShowButtons property for an object

Syntax:

***object*.ShowButtons [= boolean]**

The ShowButtons property syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
boolean	**A boolean expression that evaluates to determine whether the Add/Delete buttons are displayed**

Remarks

AddLookup Method

Add a lookup list item for a specific column in a grid

Syntax:

***object*.AddLookup(col as integer, value as string)**

The method syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
col	**An integer expression that evalutes to a column number for the grid. Columns are zero based.**
value	**A string expression that evaluates to an item in the list for a column**

Example:

```
axgrid.addlookup 1,"item1"          'add lookup list item for column 1
axgrid.addlookup 1,"item2"          'add another lookup list item for col 1
axgrid.addlookup 3,"item3"          'add lookup list item for column 3
```

AutoSetup Method

Setup grid according to predetermined arguments

Syntax:

***object*.AutoSetup(NRows As Variant, NCols As Variant, NFixedRows As Variant, NFixedCols As Variant, theMSFlexGridFormatString As Variant)**

The AutoSetup method syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
nRows	**An integer expression to determine the number of rows**
nCols	**An integer expression to determine the number of columns**
nFixedRows	**An integer expression to determine the number of fixed rows**
nFixedCols	**An integer expression to determine the number of fixed columns**
FormatString	

Example:

```
axGrid.AutoSetup 2, 3, 1, 0, "Field Name              |Data Type              |Description
"
```

Remarks:

axGrid parses the FormatString at design time and interprets it to get the following information: number of rows and columns, text for row and column headings, column width, and column alignment. The FormatString property is made up of segments separated by pipe characters (|). The text between pipes defines a column, and it may contain the special alignment characters <, ^, or >, to align the entire column to the left, center, or right. The text is assigned to row zero, and its width defines the width of each column. The FormatString may also contain a semi-colon (";"), which causes the remainder of the string to be interpreted as row heading and width information. The text is assigned to column zero, and the longest string defines the width of column zero.

axGrid will create additional rows and columns to accommodate all fields defined by the FormatString, but it will not delete rows or columns if only a few fields are specified. If you want, you can do this by setting the Rows and Cols properties.

ClearAllLookups Method

Clear all lookup lists for grid

Syntax:

***object*.ClearAllLookups**

The ClearAllLookups method syntax has these parts:

Part	**Description**
object	**An object expression that evaluates to a axGrid object**

Example:

ClearLookup Method

Clear all lookup list items for a specific column in a grid

Syntax:

***object*.ClearLookup(Col As Integer)**

The ClearLookup method syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
col	**An integer expression to determine which column to use to delete the lookup list items**

Example:

```
axgrid.ClearLookup 1        'remove all lookup list items for col 1
```

ColHasLookup Method

This method is used to determine whether or not a column has lookup list items and returns a True/False boolean expression

Syntax:

***object*.ColHasLookup(Col As Integer)**

The ColHasLookup method syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
col	**An integer expression to determine which column to evaluate**

Example:

```
if ColHasLookup(1) then
  msgbox "yes, has lookup"
endif
```

RemoveLookup Method

Remove a specific lookup list item for a column

Syntax:

***object*.RemoveLookup(Col As Integer, Value As String)**

The method syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
col	**An integer expression to determine which column to evaluate**
value	**A string expression used to search through the lookup list items for the specified column**

Example:

```
axgrid.removelookup(1, "text1")    'remove "text1" from lookup list for col 1
```

RowEmpty Method

Used to determine if a row is empty, meaning it contains no text in any of the columns for that row. Returns a boolean expression

Syntax:

***object*.RowEmpty(ByVal Row As Integer)**

The method syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
row	**An integer expression that determines which row to evaluate**

Example:

```
if axgrid.rowempty(1) then
  msgbox "row 1 is empty"
endif
```

ShowAboutBox Method

Displays the about box for an object

Syntax:

***object*.ShowAboutBox**

The method syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**

Example:

AfterAddRow Event

Occurs after a new row has been added to the grid

Syntax:

***object*.AfterAddRow(row as integer)**

The event syntax has these parts:

Part	**Description**
object	**An object expression that evaluates to a axGrid object**
row	**An integer expression that evaluates to the row number that was just created**

Remarks:

AfterDeleteRow Event

Occurs after deleting a row in a grid

Syntax:

***object*.AfterDeleteRow(row as integer)**

The AfterDeleteRow event syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
row	**An integer expression that evaluates to the row number that was just deleted**

Remarks:

AfterEdit Event

Occurs after editing a cell

Syntax:

***object*.AfterEdit(Row As Integer, Col As Integer, NewValue As String)**

The event syntax has these parts:

Part	**Description**
object	**An object expression that evaluates to a axGrid object**
row	**An integer expression that evaluates to the current row**
col	**An integer expression that evaluates to the current col**
newvalue	**A string expression that evaluates to the text currently in the specified row and column**

Remarks:

BeforeAddRow Event

Occurs before adding a new row to a grid

Syntax:

***object*.BeforeAddRow(Cancel As Boolean)**

The event syntax has these parts:

Part	**Description**
object	**An object expression that evaluates to a axGrid object**
cancel	**A boolean expression that determines if the row addition will occur**

Remarks:

BeforeDeleteRow Event

Occurs before the deletion of a row in a grid

Syntax:

***object*.BeforeDeleteRow(Row As Integer, Cancel As Boolean)**

The event syntax has these parts:

Part	Description
object	**An object expression that evaluates to a axGrid object**
row	**An integer expression that evaluates to the current row**
cancel	**A boolean expression that is used to determine if the row deletion will occur**

Remarks:

BeforeEdit Event

Occurs before editing a cell when the user starts to type characters

Syntax:

***object*.BeforeEdit(Row As Integer, Col As Integer, Cancel As Boolean)**

The BeforeEdit event syntax has these parts:

Part	Description
object	An object expression that evaluates to a axGrid object
row	An integer expression that evaluates to the current row
col	An integer expression that evaluates to the current col
cancel	A boolean expression that is used to determine whether the editing will be allowed

Remarks:

GetColWidth Method

Returns the width of a specific column

Syntax:

***object*.GetColWidth(col as integer)**

The method syntax has these parts:

Part	Description
object	An object expression that evaluates to a control object
col	an integer expression that evaluates to a column number

Example:

AllowSelection Property

Returns or sets a value to determine whether or not the user can select more than column or row

Syntax:

***object*.AllowSelection [= boolean]**

The property syntax has these parts:

Part	**Description**
object	**An object expression that evaluates to a control object**
boolean	**A boolean expression that evaluates to true or false**

Remarks

AllowUserResizing Property

Returns or sets a value to determine whether the user can resize the columns

Syntax:

***object*.AllowUserResizing [= boolean]**

The property syntax has these parts:

Part	**Description**
object	**An object expression that evaluates to a control object**
boolean	**A boolean expression that evaluates to true or false**

Remarks

BackColorBkg Property

Returns or sets the color of the background behind the grid

Syntax:

***object*.BackColorBkg [= long]**

The property syntax has these parts:

Part	**Description**
object	**An object expression that evaluates to a control object**
long	**A long expression that evaluates to a color value**

Remarks

BackColorFixed Property

Returns or sets the background color of the fixed cells (row and column headers)

Syntax:

***object*.BackColorFixed [= long]**

The property syntax has these parts:

Part	Description
object	**An object expression that evaluates to a control object**
long	**A long expression that evaluates to a color value**

Remarks

BorderStyle Property

Returns or sets the borderstyle for the control

Syntax:

***object*.BorderStyle [= integer]**

The property syntax has these parts:

Part	Description
object	An object expression that evaluates to a control object
integer	An integer expression that evaluates to one of the items in the list below

Settings:

Contant	Value	Description
No Border	0	No border
Single	1	Single line border
Thin Raised	2	Thin raised border
Thick Raised	3	Thick raised border
Thin Inset	4	Thin inset border
Thick Inset	5	Thick inset border
Etched	6	Etched single line border
Bump	7	Raised single line border

Col Property

Returns or sets the current column number

Syntax:

***object*.Col [=integer]**

The property syntax has these parts:

Part	Description
object	**An object expression that evaluates to a control object**
integer	**An integer expression that evaluates to a column number**

Remarks

ColAlign Property

Returns or sets the borderstyle for the control

Syntax:

object.BorderStyle [= integer]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
integer	***An integer expression that evaluates to one of the items in the list below***

Settings:

Contant	***Value***	***Description***
No Border	***0***	***No border***
Single	***1***	***Single line border***
Thin Raised	***2***	***Thin raised border***
Thick Raised	***3***	***Thick raised border***
Thin Inset	***4***	***Thin inset border***
Thick Inset	***5***	***Thick inset border***
Etched	***6***	***Etched single line border***
Bump	***7***	***Raised single line border***

ColHeader Property

Returns or sets a value to determine whether or not column headings are displayed

Syntax:

object.ColHeader [= boolean]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
boolean	***A boolean expression that evaluates to true or false***

Remarks

ColMask Property

Returns or sets a value for a column to determine what type of edit mask to use

Syntax:

object.ColMask(col as integer) [= integer]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
col	***An integer expression that evaluates to a specific column***
integer	***An integer expression that evaluates to one of the items in the list below***

Settings:

Contant	***Value***	***Description***
	0	***No mask***
	1	***Uppercase***
	2	***Numeric only***
	3	***Date only***
	4	***Checkbox***

Cols Property

Returns or sets a value to determine the total number of columns

Syntax:

object.Cols [= integer]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
integer	***An integer expression***

Remarks

ColWidth Property

Returns or sets the width of the specific column

Syntax:

object.ColWidth(col as integer) [= integer]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
col	***An integer expression that evaluates to a specified column***
integer	***An integer expression that evaluates to a width for the specified column***

Remarks

FixedStyle

BorderStyle Property

Returns or sets   the borderstyle for the control

Syntax:

object.BorderStyle [= integer]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
integer	***An integer expression that evaluates to one of the items in the list below***

Settings:

Contant	***Value***	***Description***
Flat	***0***	***Flat border***
3D	***1***	***3D border***

FontFixed Property

Returns or sets   the font used for the column and row headers

Syntax:

object.FontFixed [= font]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
font	***A object expression that evaluates to font object***

Remarks

ForeColorFixed Property

Returns or sets the forecolor used for the fixed cells (column and row headers)

Syntax:

object.ForeColorFixed [= long]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
long	***A long expression that evaluates to a color value***

Remarks

FormatString Property

Sets a format string that sets up a the control's column widths, alignments, and fixed row and column text.

Syntax:

object.FormatString [= string]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
string	***A string expression for formatting text in rows and columns, as described in Remarks.***

Remarks

axGrid parses the FormatString at design time and interprets it to get the following information: number of rows and columns, text   for row and column headings, column width, and column alignment.
The FormatString property is made up of segments separated by pipe characters (|). The text between pipes defines a column, and it may contain the special alignment characters <, ^, or >, to align the entire column to the left, center, or right. The text is assigned to row zero, and its width defines the width of each column.

The FormatString may also contain a semi-colon (";"), which causes the remainder of the string to be interpreted as row heading and width information. The text is assigned to column zero, and the longest string defines the width of column zero.   axGrid will create additional rows and columns to accommodate all fields defined by the FormatString, but it will not delete rows or columns if only a few fields are specified. If you want, you can do this by setting the Rows and Cols properties.

GridLineColor Property

Returns or sets the color used to draw the lines between the cells of the grid

Syntax:

object.GridLineColor [= long]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
long	***A long expression that evaluates to a color value***

Remarks

GridSolid Property

Returns or sets a value to determine if the grid lines are solid or dashed

Syntax:

object.GridSolid [= boolean]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
boolean	***A boolean expression that evaluates to true or false***

Remarks

LeftCol Property

Returns or sets the left-most visible column (other than a fixed column) in the grid control.

Syntax:
object.LeftCol [= value]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
value	***An integer expression specifying the left most column***

Remarks

ListBoxRows Property

Returns or sets the default number of items to display in the dropdown listbox when a column has a dropdown

Syntax:

object.ListBoxRows [= value]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
value	***A integer expression that evaluates to the default number of items displayed in a dropdown listbox***

Remarks

RowHeader Property

Returns or sets a value to determine of row headers are displayed

Syntax:

object.RowHeader [= value]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
value	***A boolean expression that evaluates to true or false***

Remarks

RowHeight Property

Returns or sets the height of the specified row, in twips.

Syntax:

object.RowHeight(number) [= value]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
number	***Integer. The number of the row in the grid control***
value	***Single. A numeric expression specifying the height of the row in twips.***

Remarks

Rows Property

Returns or sets the total number of rows in the grid

Syntax:

object.Rows [= value]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
value	***Integer: The total number of rows in the grid***

Remarks

SelectionMode Property

Returns or sets the selection mode for the control

Syntax:

object.SelectionMode [= value]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
value	***Integer: An expression that evaluates to one of the items in the list below***

Settings:

Contant	***Value***	***Description***
SelectionFree	***0***	***Free. Allows selections to be made normally, spreadsheet-style.***
SelectionByRow	***1***	***By Row. Forces selections to span entire rows, as in a multi-column list-box or record-based display.***
SelectionByCol	***2***	***By Column. Forces selections to span entire columns, as if selecting ranges for a chart or fields for sorting.***

ShowGrid Property

Returns or sets a value to determine if grid is displayed

Syntax:
object.ShowGrid [= value]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
value	***Boolean: An expression that evaluate to true or false***

Remarks

Text Property

Returns or sets the text contents of a cell or range of cells.

Syntax:
object.Text [= string]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
string	***A string expression that evaluates to the text content of a cell***

Remarks
When retrieving, the Text property always retrieves the contents of the current cell, defined by the Row and Col properties.

TextMatrix Property

Returns or sets the text contents of an arbitrary cell.

Syntax:

```
object.TextMatrix(row,col)          'set value
object.TextMatrix(row,col,value)    'retrieve value
```

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
row,col	***Numeric expressions specifying which cell to read or write.***
string	***A string expression containing the contents of an arbitrary cell.***

Remarks

This property allows you to set or retrieve the contents of a cell without changing the Row and Col properties.

TopRow Property

Returns or sets the uppermost visible row (other than a fixed row) in the grid.

Syntax:

object.TopRow [= value]

The property syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
value	***Long. A numeric expression specifying the uppermost row in the grid***

Remarks

Remove Method

Remove the specified row from the grid

Syntax:

object.Remove(index)

The method syntax has these parts:

Part	***Description***
object	***An object expression that evaluates to a control object***
index	***Integer: An expression that evaluates to a specified row in the grid***

Example: