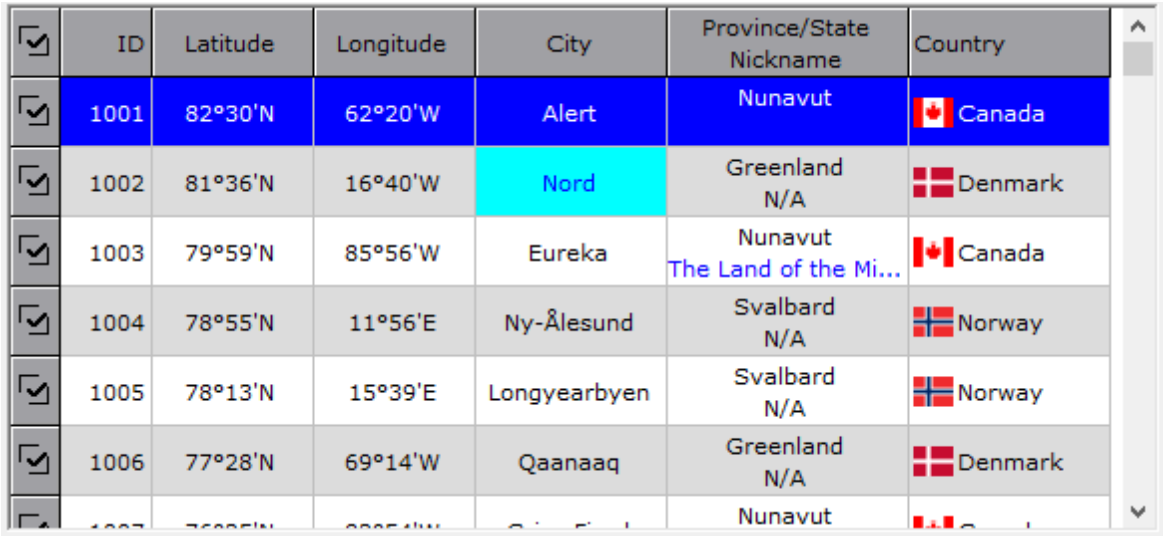
Table of Contents

1 Introduction

MegaGrid ActiveX is a versatile, feature-rich, and easy-to-use grid control, and it allows developers to virtually control every aspect of populating data in the grid.



ID	Latitude	Longitude	City	Province/State Nickname	Country
1001	82°30'N	62°20'W	Alert	Nunavut	Canada
1002	81°36'N	16°40'W	Nord	Greenland N/A	Denmark
1003	79°59'N	85°56'W	Eureka	Nunavut The Land of the Mi...	Canada
1004	78°55'N	11°56'E	Ny-Ålesund	Svalbard N/A	Norway
1005	78°13'N	15°39'E	Longyearbyen	Svalbard N/A	Norway
1006	77°28'N	69°14'W	Qaanaaq	Greenland N/A	Denmark
				Nunavut	

It is workable in any Windows application compatible with the ActiveX OLE technology such as Word, Access, Excel, Visual Basic, Visual C++, Visual FoxPro, Delphi, C++ Builder, PowerBuilder, dBase, VB.Net, C#.Net, and Internet Explorer.

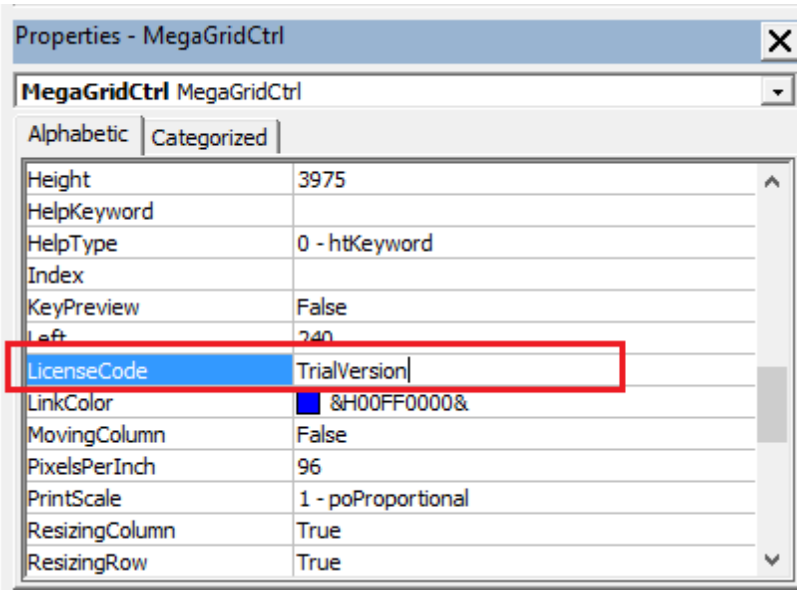
A few quick facts are listed below:

1. A non-databound grid that can be used to populate any structured, semi-structured, or unstructured data - either Small or Big.
2. Industry's 1st grid control with multiple trendy checkbox styles - 19 in total.
3. 8 different alignment options per cell for combined text(s) and image.
4. Up to 4 lines of texts per cell.
5. Configurable settings including background color, text color, font, row heights, column widths, sorting columns, resizing rows and columns, moving columns, shading alternate rows, linked texts, and etc.

2 Installation

2.1 Trial Version

1. UnZip MegaGrid.zip and run the setup.exe file to install MegaGrid ActiveX.
2. The trial version's control randomly pops up a message box saying "You are using the trial version copy of MegaGrid ActiveX".
3. A screenshot of the properties page (Visual Basic 6.0) is listed below:

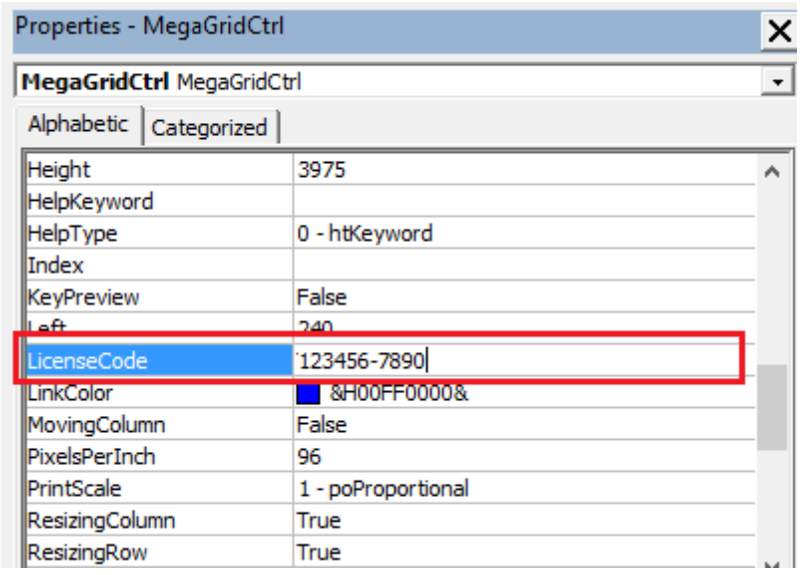


2.2 Full Version

1. Uninstall the trial version copy of MegaGrid ActiveX if applicable.

2. UnZip the full version package's .zip file and run the setup.exe file to install the full version ocx.

3. Enter a 10-digit license code with the format NNNNNN-NNNN into the **LicenseCode** property of the grid control.



3 How to Distribute It

If you want to redistribute the MegaGrid ActiveX as part of your application, please follow the instructions below:

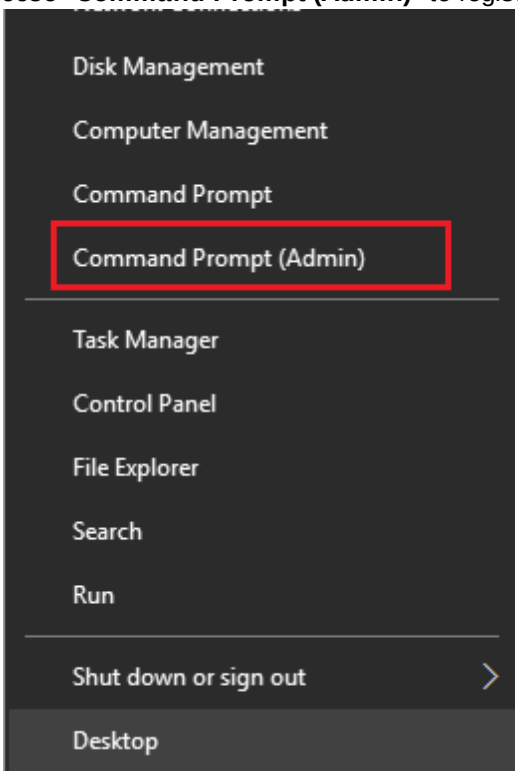
1) For 32-bit version Windows OS, simply put **MegaGrid.ocx** into the windows 32-bit system folder (e.g.

"c:\windows\system32" or "c:\winnt\system32") on the target machine and run "regsvr32 MegaGrid.ocx" to register it.

2) For 64-bit version Windows OS, simply put **MegaGrid.ocx** into the SysWOW64 folder (e.g. "c:\windows\SysWOW64") on the target machine, and run the following commands to register it:
   - cd \windows\SysWOW64
   - regsvr32 MegaGrid.ocx

3) For Windows Vista or above (excluding Windows 10), you need to use an elevated Command Prompt to run *regsvr32.exe* command, so click "**Start"** > "**All Programs**" > "**Accessories**", right-click "**Command Prompt**", and then click "**Run as administrator**".

4) For Windows 10, please choose "**Command Prompt (Admin)**" to register it.



4 Reference Guide

4.1 Properties

4.1.1 AllowCheckBox

Gets or sets a boolean flag indicating whether the check boxes will be displayed or not.

Data Type: Boolean

Remarks:

The default value is TRUE.

4.1.2 AllowZebras

Gets or sets a boolean flag indicating whether to shade alternate rows or not.

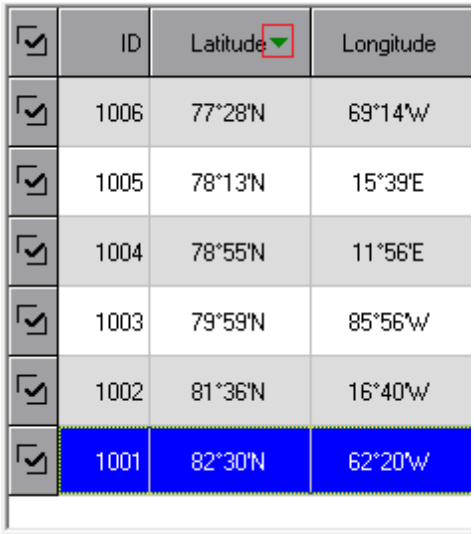
Data Type: Boolean

Remarks:

The default value is TRUE.

4.1.3 ArrowColor

Gets or sets the color of the sorting icon.



Data Type: OLE_COLOR

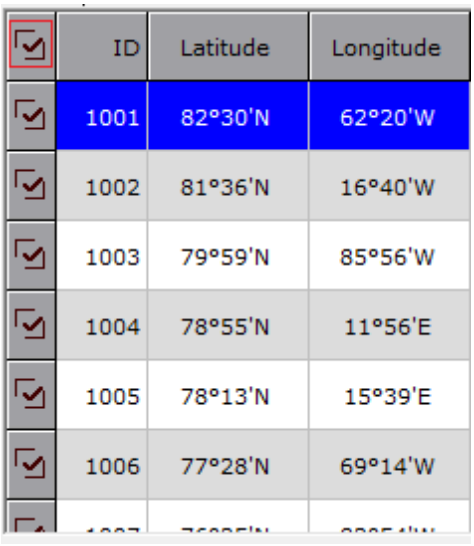
Remarks:

The default value is green, and OLE_COLOR is treated internally as a 32-bit integer with the formula below:

OLE_COLOR = red_value + (green_value * 256) + (blue_value * 256 * 256)

4.1.4 CheckBoxColor

Gets or sets the color of the checkbox.



	ID	Latitude	Longitude
	1001	82°30'N	62°20'W
	1002	81°36'N	16°40'W
	1003	79°59'N	85°56'W
	1004	78°55'N	11°56'E
	1005	78°13'N	15°39'E
	1006	77°28'N	69°14'W

Data Type: OLE_COLOR

Remarks:

The default value is black.

4.1.5 CheckBoxSize

Gets or sets the size of the checkbox.

Data Type: Integer

Remarks:

The default value is 14.

4.1.6 CheckBoxType

Gets or sets the type of the checkbox.

Data Type: An enumeration type with 19 values listed below:

Value	Screenshot
0 - Athens	

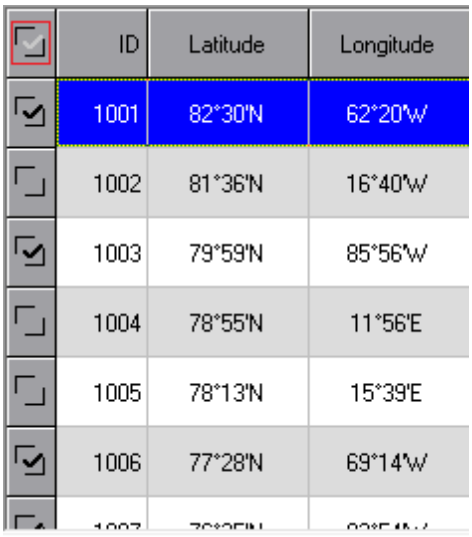
1 - Auckland	
2 - Barcelona	
3 - Budapest	
4 - Calgary	
5 - Cape_Town	
6 - Frankfurt	
7 - Honolulu	
8 - Lisbon	
9 - London	
10 - Moscow	
11 - NYC	
12 - Paris	
13 - Prague	
14 - Rio	
15 - Rome	
16 - Shanghai	
17 - Sydney	
18 - Tokyo	

Remarks:

The default value is Calgary with the value of 4.

4.1.7 CheckHalfColor

Gets or sets the check line's color of the top checkbox on the header row - whenever some checkboxes are checked and some are not checked for the data rows.



	ID	Latitude	Longitude
☑	1001	82°30'N	62°20'W
☐	1002	81°36'N	16°40'W
☑	1003	79°59'N	85°56'W
☐	1004	78°55'N	11°56'E
☐	1005	78°13'N	15°39'E
☑	1006	77°28'N	69°14'W

Data Type: OLE_COLOR

Remarks:

The default value is a light gray color with RGB (200, 200, 200).

4.1.8 CheckThickLine

Gets or sets a boolean flag indicating whether the check line of check box is thick or not.

Data Type: Boolean

Remarks:

The default value is TRUE.

4.1.9 BackColor

Gets or sets the background color of the grid control.

Data Type: OLE_COLOR

Remarks:

The default value is white.

4.1.10 ColumnsCount

Gets or sets the count of columns in the grid control.

Data Type: Integer

Remarks:

The default value is 5.

4.1.11 DefaultCheckAll

Gets or sets a boolean flag indicating whether to toggle on all check boxes or not.

Data Type: Boolean

Remarks:

The default value is TRUE.

4.1.12 DefaultColumnWidth

Gets or sets the width (in pixels) of columns in the grid.

Data Type: Integer

Remarks:

The default value is 80, and you can use ConfigureColumn() API to set up the width of an individual column to overwrite this property value.

4.1.13 DefaultRowHeight

Gets or sets the height (in pixels) of rows in the grid.

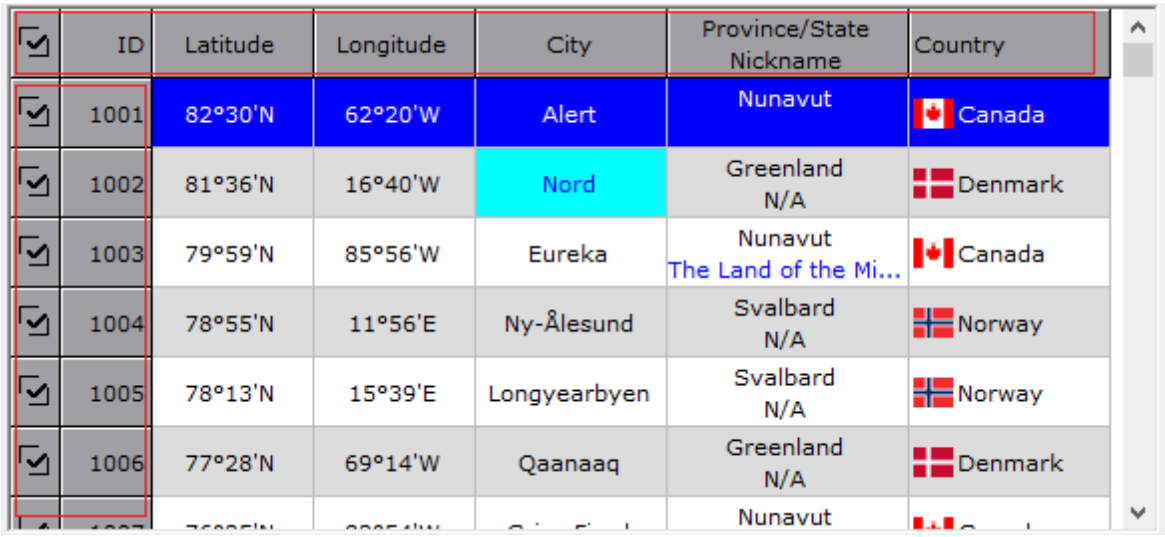
Data Type: Integer

Remarks:

The default value is 24, and you can use ConfigureRow() API to set up the height of an individual row to overwrite this property value.

4.1.14 FixedColor

Gets or sets the background color of the non-scrolling cells - including header's cells and the fixed cells of data rows.



ID	Latitude	Longitude	City	Province/State Nickname	Country
1001	82°30'N	62°20'W	Alert	Nunavut	Canada
1002	81°36'N	16°40'W	Nord	Greenland N/A	Denmark
1003	79°59'N	85°56'W	Eureka	Nunavut The Land of the Mi...	Canada
1004	78°55'N	11°56'E	Ny-Ålesund	Svalbard N/A	Norway
1005	78°13'N	15°39'E	Longyearbyen	Svalbard N/A	Norway
1006	77°28'N	69°14'W	Qaanaaq	Greenland N/A	Denmark
				Nunavut	

Data Type: OLE_COLOR

Remarks:

The default value is a gray color.

4.1.15 FixedColumns

Gets or sets the count of non-scrolling columns.

Data Type: Integer

Remarks:

The default value is 0, and a valid value must be between 0 and the count of columns in the grid.

4.1.16 FixedRows

Gets or sets the count of non-scrolling rows including the header row.

Data Type: Integer

Remarks:

The default value is 1, and a valid value must be between 1 and the count of rows in the grid.

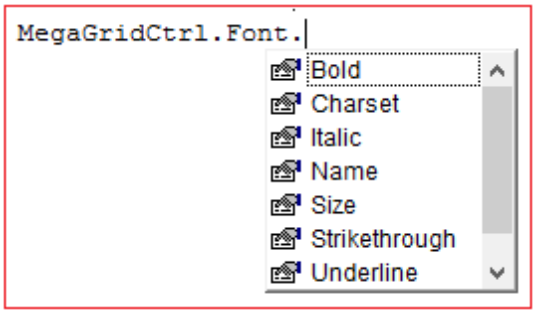
4.1.17 Font

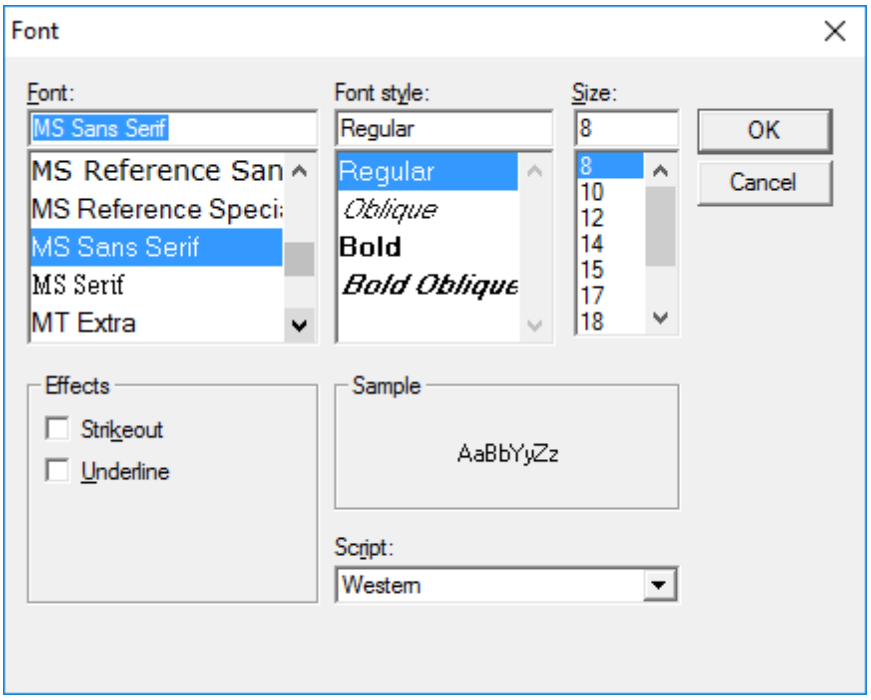
Gets or sets the attributes of text in the grid control - including the character set, font name, size, and style attributes.

Data Type: Font

Remarks:

The default value is a regular font with the name of "MS Sans Serif" and the size of 8. Two screenshots (Visual Basic) are listed below:





4.1.18 ForeColor

Gets or sets the text color of the grid control.

Data Type: OLE_COLOR

Remarks:

The default value is black.

4.1.19 GridLineWidth

Gets or sets the width (in pixels) of grid border line.

Data Type: Integer

Remarks:

The default value is 1.

4.1.20 LicenseCode

Gets or sets the license code.

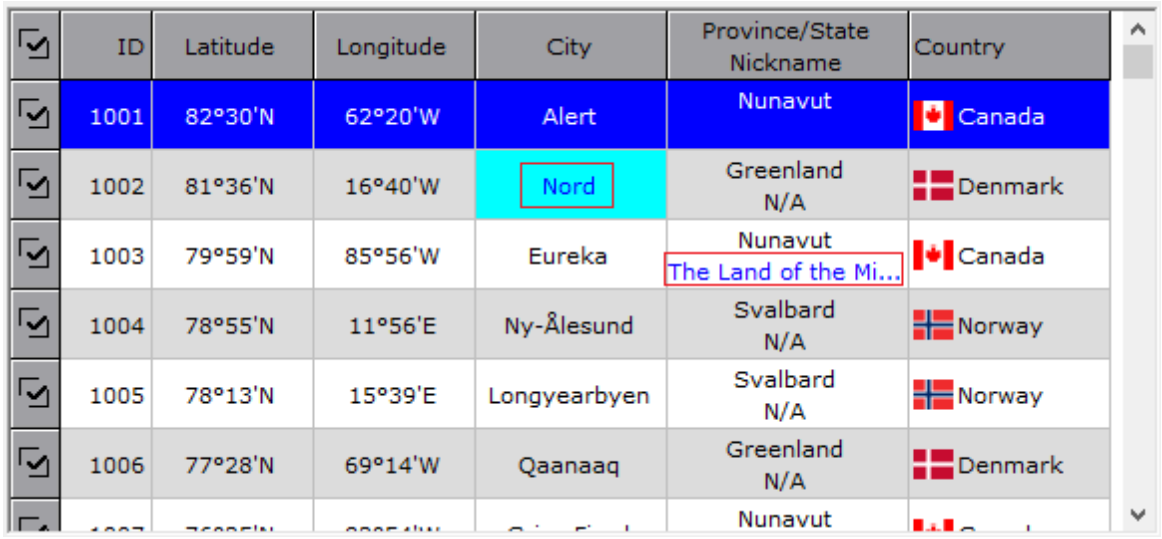
Data Type: String

Remarks:

The value is "TrialVersion" for the trial version's control, and please enter a 10-digit code into this property for the retail version's control.

4.1.21 LinkColor

Gets or sets the color of the linked text.



Data Type: OLE_COLOR

Remarks:

The default value is blue.

4.1.22 MovingColumn

Gets or sets a boolean flag indicating whether the columns can be moved around or not.

Data Type: Boolean

Remarks:

The default value is FALSE.

4.1.23 ResizingColumn

Gets or sets a boolean flag indicating whether the columns can be resized or not.

Data Type: Boolean

Remarks:

The default value is TRUE.

4.1.24 ResizingRow

Gets or sets a boolean flag indicating whether the rows can be resized or not.

Data Type: Boolean

Remarks:

The default value is TRUE.

4.1.25 RowsCount

Gets or sets the count of rows - including the header row - in the grid control.

Data Type: Integer

Remarks:

The default value is 10.

4.1.26 SelectedBackColor

Gets or sets the background color of the selected row in the grid control.

Data Type: OLE_COLOR

Remarks:

The default value is blue.

4.1.27 SelectedTextColor

Gets or sets the text color of the selected row in the grid control.

Data Type: OLE_COLOR

Remarks:

The default value is white.

4.1.28 SelectFirstRowAfterSorting

Gets or sets a boolean flag indicating whether the first row should always be selected after sorting a column.

Data Type: Boolean

Remarks:

The default value is TRUE.

4.1.29 TruncateText

Gets or sets a boolean flag indicating whether to break it off and append the "..." or not if the width of a text is beyond the width of the cell containing it.

Data Type: Boolean

Remarks:

The default value is TRUE.

4.1.30 UnderlinedLink

Gets or sets a boolean flag indicating whether to underline all linked text or not.

Data Type: Boolean

Remarks:

The default value is FALSE.

4.1.31 ZebraColor

Gets or sets the background color of alternate rows.

Data Type: OLE_COLOR

Remarks:

The default value is a light gray color with RGB (220, 220, 220).

4.2 Methods

4.2.1 CheckAllRows

Toggles on/off the checkboxes of all rows.

```
void CheckAllRows(bool IsChecked);
```

Parameters:

IsChecked

Indicates whether the checkboxes of all rows should be toggled on/off.

4.2.2 CheckRow

Toggles on/off the checkbox of an individual data row.

```
void CheckRow(int ID, bool IsChecked, bool IDIsIndex);
```

Parameters:

ID

Identifies a row, and it could be either index or tag.

IsChecked

Indicates whether the checkbox should be checked or not.

IDIsIndex

Indicates whether the first parameter ID is a row index or tag.

4.2.3 ConfigureCellImage

Specifies the attributes of an image that is rendered onto an individual cell.

```
bool ConfigureCellImage(int     RowIndex,
                        int     ColumnIndex,
                        int     ImageType,
                        LPCWSTR ImagePath,
                        int     PaintWidth,
                        int     PaintHeight,
                        int     Alignment);
```

Parameters:

RowIndex

Specifies the row's index of the cell onto which an image will be rendered, and the value must not be less than 1 since the header row does not allow to embed image(s).

ColumnIndex

Specifies the column's index of the cell onto which an image will be rendered, and it is a zero-based index, so the value must be between 0 and the count of columns minus 1.

ImageType

Specifies the image's type, and all valid values are listed below:

Value	Comment

0	BMP Image
1	ICO Image
2	WMF Image
3	EMF Image

ImagePath

Specifies the absolute path of an image.

PaintWidth

Specifies the width (in pixels) of the rectangle into which an image is rendered.

PaintHeight

Specifies the height (in pixels) of the rectangle into which an image is rendered.

Alignment

Specifies the alignment's type, and all valid values are listed below:

Value	Comment
3	If multiple lines (up to 4) of texts are specified via ConfigureCellText, all texts are left-aligned and the image is rendered next to them. Otherwise the image is left-aligned.
4	The image is left-aligned and all texts of cell are drawn next to it.
5	The image is center-aligned and no text is drawn.
6	If multiple lines (up to 4) of texts are specified via ConfigureCellText, all texts are right-aligned and the image is rendered next to them. Otherwise the image is right-aligned.
7	The image is right-aligned and all texts of cell are drawn next to it.

Return Value:

If the operation is successful, it returns TRUE. Otherwise it returns FALSE, for example, the image is not in a valid format.

4.2.4 ConfigureCellText

Specifies multiple lines (up to 4) of texts that are drawn onto an individual cell.

```
bool ConfigureCellText(int      RowIndex,
                       int      ColumnIndex,
                       LPCWSTR  Line_1,
                       bool     IsLink_1,
                       LPCWSTR  Line_2,
```

```
                bool        IsLink_2,
                LPCWSTR     Line_3,
                bool        IsLink_3,
                LPCWSTR     Line_4,
                bool        IsLink_4,
                long        BackColor,
                long        ForeColor,
                int         Alignment);
```

Parameters:

RowIndex

Specifies the row's index of the cell onto which an image will be rendered, and the value must not be less than 1 since the header row does not allow to embed image(s).

ColumnIndex

Specifies the column's index of the cell onto which an image will be rendered, and it is a zero-based index, so the value must be between 0 and the count of columns minus 1.

Line_1

Specifies the 1st line of Unicode text that is drawn onto the cell, and an empty string can be assigned here.

IsLink_1

Specifies whether the 1st line of Unicode text is a linked text or not.

Line_2

Specifies the 2nd line of Unicode text that is drawn onto the cell, and an empty string can be assigned here.

IsLink_2

Specifies whether the 2nd line of Unicode text is a linked text or not.

Line_3

Specifies the 3rd line of Unicode text that is drawn onto the cell, and an empty string can be assigned here.

IsLink_3

Specifies whether the 3rd line of Unicode text is a linked text or not.

Line_4

Specifies the 4th line of Unicode text that is drawn onto the cell, and an empty string can be assigned here.

IsLink_4

Specifies whether the 4th line of Unicode text is a linked text or not.

BackColor

Specifies the cell background's color value in OLE_COLOR format. If it is -1, the background color will be determined by the Color property of the control. For example, 0x0000FF (C/C++) or &H0000FF (Visual Basic) can be passed to indicate the red color.

ForeColor

Specifies the cell text's color value in OLE_COLOR format. If it is -1, the text color will be determined by the ForeColor property of the control.

Alignment

Specifies the alignment's type, and all valid values are listed below:

Value	Comment
0	If ConfigureCellImage() is not called, all texts are left-aligned. Otherwise the alignment will be determined by that API.
1	If ConfigureCellImage() is not called, all texts are center-aligned. Otherwise the alignment will be determined by that API.
2	If ConfigureCellImage() is not called, all texts are right-aligned. Otherwise the alignment will be determined by that API.

Return Value:

If the operation is successful, it returns TRUE. Otherwise it returns FALSE, for example, the index value of row and/or column is out of bound.

4.2.5 ConfigureColumn

Specifies the attributes of an individual header in the grid.

```
bool ConfigureColumn(int     ColumnIndex,
                     int     Width,
                     LPCWSTR Title_1,
                     LPCWSTR Title_2,
                     LPCWSTR Title_3,
                     LPCWSTR Title_4,
                     long    BackColor,
                     long    ForeColor,
                     int     Alignment,
                     bool    AllowSorting,
                     int     DataType);
```

Parameters:

ColumnIndex

Specifies the column's index of the header onto which an image will be rendered, and it is a zero-based index, so the value must be between 0 and the count of columns minus 1.

Width

Specifies the width of the header. If the value is -1, the width of header will be determined by the value of DefaultColumnWidth property; otherwise the value will overwrite the latter.

Title_1

Specifies the 1st line of Unicode text that is drawn onto the column header, and an empty string can be assigned here.

Title_2

Specifies the 2nd line of Unicode text that is drawn onto the column header, and an empty string can be assigned here.

Title_3

Specifies the 3rd line of Unicode text that is drawn onto the column header, and an empty string can be assigned here.

Title_4

Specifies the 4th line of Unicode text that is drawn onto the column header, and an empty string can be assigned here.

BackColor

Specifies the header background's color value in OLE_COLOR format. If it is -1, the background color will be determined by either Color or FixedColor property, which depends on whether the current header's cell is a fixed cell or not. For example, 0x0000FF (C/C++) or &H0000FF (Visual Basic) can be passed to indicate the red color.

ForeColor

Specifies the header text's color value in OLE_COLOR format. If it is -1, the text color will be determined by the ForeColor property of the control.

Alignment

Specifies the alignment's type, and all valid values are listed below:

Value	Comment
0	All texts are left-aligned.
1	All texts are center-aligned.
2	All texts are right-aligned.

AllowSorting

Specifies whether the sorting operation is allowed to be against the 1st line of text of all data cells belonging to this column or not.

DataType

Specifies the data type of the 1st line of text of all data cells belonging to this column for the sorting purpose, and all valid values are listed below. If AllowSorting is set to FALSE, the value of this parameter is irrelevant.

Value	Comment
0	String
1	Integer
2	Double
3	Date & Time

Return Value:

If the operation is successful, it returns TRUE. Otherwise it returns FALSE, for example, the index values of row and/or column are out of bound.

4.2.6 ConfigureRow

Specifies the attributes of an individual data row in the grid.

```
bool ConfigureRow(int  RowIndex,
                  int  Tag,
                  bool  IsChecked,
                  int  Height);
```

Parameters:

RowIndex

Specifies the row's index, and the value must be 0 and the count of rows minus 1. If the value is 0, this method is used to only define the height of the header row and the parameters (*Tag* and *IsChecked*) are irrelevant.

Tag

Specifies the tag value of the row, which is used to uniquely identify the current row. After sorting a column, all data rows are reshuffled, so the tag value is used to identify which is which.

IsChecked

Specifies whether the checkbox of the row is toggled on or not. If the AllowCheckBox property is set to false, the value of this parameter is irrelevant.

Height

Specifies the height of the row, If the value is -1, the height of the row will be determined by the value of DefaultRowHeight property; otherwise the value will overwrite the latter.

Return Value:

If the operation is successful, it returns TRUE. Otherwise it returns FALSE, for example, the index value of row and/or column is out of bound.

4.2.7 GetCellColumnIndex

Gets the column index of the cell that contains the point with the screen coordinates (X, Y).

```
long GetCellColumnIndex(long X, long Y);
```

Parameters:

X

Specifies the x coordinate.

Y

Specifies the y coordinate.

Return Value:

If the operation is successful, it returns a valid column index - between 0 and the count of columns minus 1. Otherwise it returns -1.

Notes:

In a mouse event handler**,** the mouse coordinates (X, Y) are known, so this method is used to get the corresponding column index.

4.2.8 GetCellRowIndex

Gets the row index of the cell that contains the point with the screen coordinates (X, Y).

```
long GetCellRowIndex(long X, long Y);
```

Parameters:

X

Specifies the x coordinate.

Y

Specifies the y coordinate.

Return Value:

If the operation is successful, it returns a valid row index - between 0 and the count of rows minus 1. Otherwise it returns -1.

Notes:

In a mouse event handler**,** the mouse coordinates (X, Y) are known, so this method is used to get the corresponding row index.

4.2.9 GetCellLeft

Gets the left value (in pixels) of a cell rectangle in the grid.

```
long GetCellLeft(long RowIndex, long ColumnIndex);
```

Parameters:

RowIndex
Specifies the row index of the cell.

ColumnIndex
Specifies the column index of the cell.

Return Value:

It returns the left value of the cell boundaries.

4.2.10 GetCellTop

Gets the top value (in pixels) of a cell rectangle in the grid.

```
long GetCellTop(long RowIndex, long ColumnIndex);
```

Parameters:

RowIndex
Specifies the row index of the cell.

ColumnIndex
Specifies the column index of the cell.

Return Value:

It returns the top value of the cell boundaries.

4.2.11 GetCellRight

Gets the right value (in pixels) of a cell rectangle in the grid.

```
long GetCellRight(long RowIndex, long ColumnIndex);
```

Parameters:

RowIndex
Specifies the row index of the cell.

ColumnIndex
Specifies the column index of the cell.

Return Value:

It returns the right value of the cell boundaries.

4.2.12 GetCellBottom

Gets the bottom value (in pixels) of a cell rectangle in the grid.

```
long GetCellBottom(long RowIndex, long ColumnIndex);
```

Parameters:

RowIndex
Specifies the row index of the cell.

ColumnIndex
Specifies the column index of the cell.

Return Value:

It returns the left value of the cell boundaries.

4.2.13 GetCheckedRowsCount

Gets the count of all checked rows in the grid - excluding the header row.

```
long GetCheckedRowsCount();
```

Return Value:

It returns the count of all checked rows.

4.2.14 GetCheckedRowTagAt

Gets the tag value of an individual checked row in the grid.

```
long GetCheckedRowTagAt(long Index);
```

Parameters:

Index

Specifies the index of an individual checked row, and a valid value must be between 0 and the count of all checked rows minus 1.

Return Value:

It returns the tag value of an individual checked row.

4.2.15 GetSelectedRowIndex

Gets the row index of the selected row in the grid.

```
long GetSelectedRowIndex();
```

Return Value:

It returns the row index of the selected row.

4.2.16 GetSelectedRowTag

Gets the tag value of the selected row in the grid.

```
long GetSelectedRowTag();
```

Return Value:

It returns the tag value of the selected row.

4.2.17 GetVisibleColumnsCount

Gets the count of all visible columns - excluding fixed columns - in the grid.

```
long GetVisibleColumnsCount();
```

Return Value:

It returns the count of all visible columns.

4.2.18 GetVisibleColumnsStart

Gets the column index of first visible column - not a fixed column - in the grid.

```
long GetVisibleColumnsStart();
```

Return Value:

It returns the column index of first visible column.

4.2.19 GetVisibleRowsCount

Gets the count of all visible rows - excluding fixed rows - in the grid.

```
long GetVisibleRowsCount();
```

Return Value:

It returns the count of all visible rows.

4.2.20 GetVisibleRowsStart

Gets the row index of first visible row - not a fixed row - in the grid.

```
long GetVisibleRowsStart();
```

Return Value:

It returns the row index of first visible row.

4.2.21 Refresh

Repaints the grid.

```
void Refresh();
```

4.2.22 SelectRow

Selects an individual row identified with the row index or tag value.

```
void SelectRow(long ID, bool IDIsIndex);
```

Parameters:

ID

Specifies the row index or tag value.

IDIsIndex

Indicates whether ID is a row index or tag value.

4.2.23 SetVisibleRow

Scrolls an individual row - identified with the row index or tag value - to the visible region of the grid.

```
void SetVisibleRow(long ID, bool IDIsIndex);
```

Parameters:

ID
Specifies the row index or tag value.

IDIsIndex
Indicates whether ID is a row index or tag value.

4.3 Events

4.3.1 OnClickLink

Is fired when a linked text in the grid gets clicked.

```
void OnClickLink(long RowIndex, long ColumnIndex, long RowTag, LPCWSTR LinkText);
```

Parameters:

RowIndex
Specifies the row index of the cell containing the linked text that gets clicked.

ColumnIndex
Specifies the column index of the cell containing the linked text that gets clicked.

RowTag
Specifies the row tag's value of the cell containing the linked text that gets clicked.

LinkText
Specifies the linked text that gets clicked.

4.3.2 OnClickRow

Is fired when a row in the grid gets clicked.

```
void OnClickRow(long RowIndex, long Tag);
```

Parameters:

RowIndex
Specifies the index of the row that gets clicked.

Tag
Specifies the tag's value of the row that gets clicked.

4.3.3 OnColumnMoved

Is fired when a column in the grid is being moved around.

```
void OnColumnMoved(long FromColumn, long ToColumn);
```

Parameters:

FromColumn
Specifies the index of the original column.

ToColumn
Specifies the index of the destination column.

4.3.4 OnKeyDown

Is fired when a key gets pressed while the grid has the focus.

```
void OnKeyDown(long KeyCode, long ShiftState);
```

Parameters:

KeyCode
Specifies the key on the keyboard.

ShiftState
Indicates whether the Shift, Alt, or Ctrl keys are combined with the keystroke.

Condition	Comment
(ShiftState & 1) is non-zero	Shift key gets pressed.
(ShiftState & 2) is non-zero	Ctrl key gets pressed.
(ShiftState & 4) is non-zero	Alt key gets pressed.

4.3.5 OnKeyUp

Is fired when a key gets released while the grid has the focus.

```
void OnKeyUp(long KeyCode, long ShiftState);
```

Parameters:

KeyCode
Specifies the key on the keyboard.

ShiftState

Indicates whether the Shift, Alt, or Ctrl keys are combined with the keystroke.

Condition	Comment
(ShiftState & 1) is non-zero	Shift key gets released.
(ShiftState & 2) is non-zero	Ctrl key gets released.
(ShiftState & 4) is non-zero	Alt key gets released.

4.3.6 OnMouseDown

Is fired when a mouse button gets pressed.

```
void OnMouseDown(long X, long Y, long ButtonFlag, long ShiftState);
```

Parameters:

X

Specifies the x coordinate (in pixels) of the mouse pointer.

Y

Specifies the y coordinate (in pixels) of the mouse pointer.

ButtonFlag

Specifies which button gets pressed (1 for the left button, 2 for the middle button, and 3 for the right button).

ShiftState

Indicates whether the Shift, Alt, or Ctrl keys are combined with the mouse button.

Condition	Comment
(ShiftState & 1) is non-zero	Shift key gets pressed.
(ShiftState & 2) is non-zero	Ctrl key gets pressed.
(ShiftState & 4) is non-zero	Alt key gets pressed.

4.3.7 OnMouseUp

Is fired when a mouse button gets released.

```
void OnMouseUp(long X, long Y, long ButtonFlag, long ShiftState);
```

Parameters:

X

Specifies the x coordinate (in pixels) of the mouse pointer.

Y

Specifies the y coordinate (in pixels) of the mouse pointer.

ButtonFlag

Specifies which button gets released (1 for the left button, 2 for the middle button, and 3 for the right button).

ShiftState

Indicates whether the Shift, Alt, or Ctrl keys are combined with the mouse button.

Condition	Comment
(ShiftState & 1) is non-zero	Shift key gets pressed.
(ShiftState & 2) is non-zero	Ctrl key gets pressed.
(ShiftState & 4) is non-zero	Alt key gets pressed.

4.3.8 OnMouseMove

Is fired when a user moves the mouse pointer.

```
void OnMouseMove(long X, long Y, long ShiftState);
```

Parameters:

X

Specifies the x coordinate (in pixels) of the mouse pointer.

Y

Specifies the y coordinate (in pixels) of the mouse pointer.

ShiftState

Indicates whether the Shift, Alt, or Ctrl keys are combined with the mouse button.

Condition	Comment
(ShiftState & 1) is non-zero	Shift key gets pressed.
(ShiftState & 2) is non-zero	Ctrl key gets pressed.
(ShiftState & 4) is non-zero	Alt key gets pressed.

4.3.9 OnMouseWheelDown

Is fired when the mouse wheel gets rotated downward.

```
void OnMouseWheelDown(long X, long Y, long ShiftState);
```

Parameters:

X

Specifies the x coordinate (in pixels) of the mouse pointer.

Y

Specifies the y coordinate (in pixels) of the mouse pointer.

ShiftState
Indicates whether the Shift, Alt, or Ctrl keys are combined with the mouse wheel.

Condition	Comment
(ShiftState & 1) is non-zero	Shift key gets pressed.
(ShiftState & 2) is non-zero	Ctrl key gets pressed.
(ShiftState & 4) is non-zero	Alt key gets pressed.

4.3.10 OnMouseWheelUp

Is fired when the mouse wheel gets rotated upward.

```
void OnMouseWheelUp(long X, long Y, long ShiftState);
```

Parameters:

X
Specifies the x coordinate (in pixels) of the mouse pointer.

Y
Specifies the y coordinate (in pixels) of the mouse pointer.

ShiftState
Indicates whether the Shift, Alt, or Ctrl keys are combined with the mouse wheel.

Condition	Comment
(ShiftState & 1) is non-zero	Shift key gets pressed.
(ShiftState & 2) is non-zero	Ctrl key gets pressed.
(ShiftState & 4) is non-zero	Alt key gets pressed.

4.3.11 OnPaintCell

Is fired when a cell in the grid finishes the internal drawing and the host application can start to customize drawing against it.

```
bool OnPaintCell(long    HDCVal,
                 long    RowIndex,
                 long    ColumnIndex,
                 long    Left,
                 long    Top,
                 long    Right,
                 long    Bottom,
                 long    CellState);
```

Parameters:

HDCVal
Specifies a handle to the device context.

RowIndex
Specifies the row index of the cell.

ColumnIndex
Specifies the column index of the cell.

Left
Specifies the x coordinate (in pixels) of the left side of the cell rectangle.

Top
Specifies the y coordinate (in pixels) of the top side of the cell rectangle.

Right
Specifies the x coordinate (in pixels) of the right side of the cell rectangle.

Bottom
Specifies the y coordinate (in pixels) of the bottom side of the cell rectangle.

CellState
Specifies the cell state.

Condition	Comment
(CellState & 1) is non-zero	The cell is non-scrolling.
(ShiftState & 2) is non-zero	The cell is selected.

4.3.12 OnSelectRow

Is fired when a row in the grid gets selected.

```
void OnSelectRow(long RowIndex, long Tag);
```

Parameters:

RowIndex
Specifies the index of the selected row.

Tag
Specifies the tag value of the selected row.