

oleGridSort from Timeamber Limited

(for Microsoft® Visual Basic® 4.0)

An OLE automation server that provides full sorting capability, to any instance of the Microsoft® Visual Basic® 4.0 MS Grid control for any column (with an optional secondary column), ascending or descending, under program control.

The OLE automation server is supplied in 3 forms:-

- An out-of-process server for 16-bit environments (Windows® 3.x).
- An out-of-process server for 32-bit environments (Windows® 95/NT).
- An in-process server for 32-bit environments (Windows® 95/NT).

New/Modified properties & methods

Using oleGridSort

Terms and Conditions

Licensing

Distribution

Disclaimer

Copyright © Timeamber Limited 1996

New/Modified properties & methods

The following properties and methods are new, or modified to function differently to the underlying MS Grid control:-

Properties

Clip (modified)
IgnoreFixedCols (new)
IsInitialized (new)
MinSortLen (new)
NewGrid (new)
NewIndex (new)
SecondSortCol (new)
SortCol (new)
SortOrder (new)
Text (modified)

Methods

AddItem (modified)
LoadData (new)
RemoveItem (modified)
SaveData (new)

Clip property (modified)

Syntax

[*form.*]oleGridSort_instance.Clip[= *string*]

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>string</i>	An expression that evaluates to a string. If <i>string</i> contains text for more than one column, each column must be delimited by the vbTab character. If <i>string</i> contains text for more than one row, each row must be delimited by the vbCr character.

Remarks

Functionality is similar to the same property of the underlying MS Grid control with the following additions:-

- ✦ If the **Clip** property modifies the contents of the SortCol column or the SecondSortCol column (if

SecondSortCol <> SG_NOSECONDSORT), and the SortOrder property <> SG_SORT_UNSORTED, a grid sort will be invoked.

IgnoreFixedCols property (new)

Sets or returns a boolean flag to indicate whether the grid sort omits or includes fixed columns.

Syntax

[*form*.]oleGridSort_*instance*.IgnoreFixedCols[= *boolean*]

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>boolean</i>	An expression that evaluates to True or False. If True, any FixedCols are omitted from the grid sort. If False (default), all columns are included.

Remarks

If this property is True, and the values of SortCol or SecondSortCol are subsequently set to a value < **FixedCols**, an error occurs.

IsInitialized property (new)

Returns True if the underlying MS Grid has been successfully sub-classed by the NewGrid property, otherwise it returns False.

Syntax

[form.]oleGridSort_instance.IsInitialized

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.

Remarks

This is the only property that may be used before setting the **NewGrid** property.

MinSortLen property (new)

Sets or returns the minimum number of characters for the SortCol part of the sort key when the SecondSortCol property is used.

Syntax

[*form*.]oleGridSort_*instance*.MinSortLen[= *numchars*]

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>numchars</i>	Any positive integer expression in the range: 1 to 128 (see Remarks below).

Remarks

This property is ignored if **SecondSortCol** is SG_NOSECONDSORT, or SortOrder is SG_SORT_UNSORTED.

If the number of characters in **SortCol** is less than **MinSortLen** for any row, the sort key string is padded with (**MinSortLen - Len(SortCol)**) spaces before the **SecondSortCol** string for that row is appended to it.

The maximum length of the sort key string is 255 characters. This includes **SortCol**, any **MinSortLen** padding, and **SecondSortCol**.

If not specifically set, this property defaults to 10.

If the value of this property is set outside the allowed range, no error is generated, but **SecondSortCol** is set to SG_NOSECONDSORT.

NewGrid property (new)

Sub-classes the MS Grid control to which all properties/methods of this instance of oleGridSort will refer.

Syntax

Set [*form*].*oleGridSort_instance*.**NewGrid** = *grid_control*

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>grid_control</i>	An object expression that evaluates to a MS Grid control object.

Remarks

Once this property is set, all properties and methods of the sub-classed MS Grid should be referenced using this class and not the MS Grid control.

Important: This property must be set before any other property or method of the oleGridSort class may be used, with the exception of the IsInitialized property.

NewIndex property (new)

Returns the row number of the last row inserted by the *oleGridSort_instance*.AddItem method.

Syntax

[*form*.]*oleGridSort_instance*.**NewIndex**

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.

Remarks

This property is reset to SG_NULLINDEX when a grid sort is invoked, or the RemoveItem method is used.

SecondSortCol property (new)

Sets or returns the column number of the underlying MS Grid that will provide the secondary part of the sort key.

Syntax

[*form*.]*oleGridSort_instance*.**SecondSortCol**[= *column_number*]

The **SecondSortCol** property has these parts:

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>column_number</i>	A numeric expression that specifies a column number in the range 0 to <i>oleGridSort_instance</i> .Cols - 1.

Remarks

If the SortOrder property is not SG_SORT_UNSORTED and the previous value of **SecondSortCol** <> *column_number*, a grid sort will be invoked.

To disable secondary sorting, set this property to SG_NOSECONDSORT.

If this property is set to the same column as SortCol, it will be reset to SG_NOSECONDSORT.

If *column_number* is outside the allowed range, an error is generated.

SortCol property (new)

Sets or returns the column number of the underlying MS Grid that will provide the primary sort key.

Syntax

[*form*.]*oleGridSort_instance*.SortCol[= *column_number*]

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>column_number</i>	A numeric expression that specifies a column number in the range 0 to <i>oleGridSort_instance</i> .Cols - 1.

Remarks

If *column_number* is outside the allowed range, an error is generated.

If the SortOrder property is not SG_SORT_UNSORTED and the previous value of **SortCol** <> *column_number*, a grid sort will be invoked.

SortOrder property (new)

Sets the sort order (ascending, descending, or unsorted) for the underlying MS Grid, or returns the current sort order value.

Syntax

[*form*.]*oleGridSort_instance*.SortOrder[= *sort_order*]

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>sort_order</i>	A numeric expression that specifies the new sort order value.

Remarks

sort_order must be one of: SG_SORT_UNSORTED, SG_SORT_ASCENDING or SG_SORT_DESCENDING otherwise an error is generated.

If *sort_order* is not SG_SORT_UNSORTED and the previous value of **SortOrder** <> *sort_order*, a grid sort will be invoked.

Text property (modified)

Syntax

[*form*.]*oleGridSort_instance*.**Text**[= *string*]

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

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>string</i>	An expression that evaluates to a string.

Remarks

Functionality is similar to the same property of the underlying MS Grid control with the following additions:-

- ✦ If the **Text** property modifies the contents of a cell in the SortCol column or the SecondSortCol column (if **SecondSortCol** <> SG_NOSECONDSORT), and the SortOrder property <> SG_SORT_UNSORTED, a grid sort will be invoked.

AddItem method (modified)

Syntax

[*form*.]oleGridSort_instance.AddItem delimited_string[, rownum]

The **AddItem** method syntax has these parts:

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>delimited_string</i>	An expression that evaluates to a string. If <i>delimited_string</i> contains text for more than one column, each column must be delimited by the vbTab character.
<i>rownum</i>	A numeric expression that specifies a row number in the range <i>oleGridSort_instance</i> .FixedRows to <i>oleGridSort_instance</i> .Rows - 1

Remarks

The functionality is similar to the same method of the underlying MS Grid control with the following differences:-

- ✦ *rownum* is ignored if the SortOrder property is not SG_SORT_UNSORTED.
- ✦ If the **SortOrder** value is not SG_SORT_UNSORTED, **AddItem** inserts *delimited_string* into the appropriate row determined by the value of **SortOrder**, and the value of the string comprising SortCol[+_SecondSortCol].
- ✦ **AddItem** sets the NewIndex property.

LoadData method (new)

Loads data into the MS Grid from a file previously created using the SaveData method, re-setting the number of rows and columns read from the file header.

Syntax

[*form*.]*oleGridSort_instance*.**LoadData** *pathname*[, *keep_fixedcols*]

The **LoadData** method syntax has these parts:

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>pathname</i>	A string expression that specifies a file containing data to be loaded into the MS Grid.
<i>keep_fixedcols</i>	A boolean expression. If True, the left-most column to be loaded will be <i>oleGridSort_instance</i> .FixedCols. If omitted or False, the left-most column to be loaded will be column 0.

Remarks

The current data contents of the MS Grid will be overwritten by the loaded data.

The data is loaded from row number: *oleGridSort_instance*.FixedRows.

If the SortOrder property is not SG_SORT_UNSORTED, a grid sort will be invoked.

SaveData method (new)

Saves the data within the MS Grid to a file.

Syntax

[*form*.]*oleGridSort_instance*.**SaveData** *pathname*[, *ignore_fixedcols*]

The **SaveData** method syntax has these parts:

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>pathname</i>	A string expression that specifies a file to receive the data contents of the MS Grid.
<i>ignore_fixedcols</i>	A boolean expression. If True, the left-most column of data saved will be <i>oleGridSort_instance</i> .FixedCols. If omitted or False, the left-most column will be column 0.

Remarks

The contents of any **FixedRows** are not saved.

The created file contains an 8 byte header which comprises the number of rows (2 bytes), columns (2 bytes), and the length of the data (4 bytes).

RemoveItem method (modified)

Syntax

[*form.*]oleGridSort_*instance*.RemoveItem *index*

The **RemoveItem** method syntax has these parts:

Part	Description
<i>form</i>	The form on which this (Public) instance of the oleGridSort class is declared.
<i>oleGridSort_instance</i>	The instance name of the oleGridSort class.
<i>index</i>	A numeric expression that specifies a row number in the range <i>oleGridSort_instance</i> .FixedRows to <i>oleGridSort_instance</i> .Rows - 1

Remarks

Functionality is similar to the same property of the underlying MS Grid control with the following additions:-

- ✦ The NewIndex property is reset to SG_NULLINDEX.

Const SG_SORT_UNSORTED = 0

Disables sorting of the underlying MS Grid control.

Const SG_SORT_ASCENDING = 1

The underlying MS Grid will sort in ascending order.

Const SG_SORT_DESCENDING = 2

The underlying MS Grid will sort in descending order.

Const SG_NULLINDEX = -1

Indicates that the NewIndex property value is invalid.

The NewIndex property will have this value for any of the following reasons:-

- The AddItem method has not been used.
- The grid sort has been invoked since the last AddItem.
- The RemoveItem method has been used since the last AddItem.

Const SG_NOSECONDSORT = -1

Disables the SecondSortCol property.

Using oleGridSort

Important: The **oleGridSort** instance wraps all properties and methods of the 'sub-classed' MS Grid control, and holds some of these internally for improved performance. Therefore ALL references to the MS Grid's properties and methods should be made using the properties/methods of the specific instance of this server.

[Declaring instances of oleGridSort](#)

[Sub-classing a Grid control](#)

[Using In-Process or Out-Of-Process OLE Automation servers](#)

[Hints and Tips](#)

Declaring instances of oleGridSort

Method #1 - a single instance (early binding)

Use following syntax, in the **Declarations** section of the **Form** on which the **MS Grid** control is placed:-

```
Public instance_name As New [SG.]oleGridSort
```

or in the **Declarations** section of a **Module**:-

```
Public instance_name As New [SG.]oleGridSort
```

where:

instance_name is any valid name, e.g: **MySortedGrid**

Method #2 - a single instance (later binding)

Use the following syntax, in the **Declarations** section of the **Form** on which the **MS Grid** control is placed:-

```
Public instance_name As [SG.]oleGridSort
```

or in the **Declarations** section of a **Module**:-

```
Public instance_name As [SG.]oleGridSort
```

Then within a suitable procedure, such as the Form_Load() event, use the following syntax:-

```
Set instance_name = New [SG.]oleGridSort
```

where:

instance_name is any valid name, e.g: **MySortedGrid**

Method #3 - a single instance (late binding) not recommended!

Use the following syntax, in the **Declarations** section of the **Form** on which the **MS Grid** control is placed:-

```
Public instance_name As [SG.]oleGridSort
```

or in the **Declarations** section of a **Module**:-

```
Public instance_name As [SG.]oleGridSort
```

Then within a suitable procedure, such as the Form_Load() event, use the following syntax:-

```
Set instance_name = CreateObject("[SG.]oleGridSort")
```

where:

instance_name is any valid name, e.g: **MySortedGrid**

Method #4 - an array of instances

Use the following syntax in the **Declarations** section of a **Module**:-

Public *instance_name*([*instance_count*]) **As** [SG.]oleGridSort

or, in the **Declarations** section of the **Form** on which the **MS Grid** control is placed:-

Private *instance_name*([*instance_count*]) **As** [SG.]oleGridSort

where:

instance_name is any valid name, e.g: **MySortedGrid**

instance_count is the specification of the number of array elements in any of the valid formats. If this value is omitted, the array is dynamic.

e.g: **Public** MySortedGrid() **As** SG.oleGridSort

Method #5 - an array of instances

Use the following syntax in the **Declarations** section of a **Module**:-

Public *instance_name*(*instance_count*) **As New** [SG.]oleGridSort

or, in the **Declarations** section of the **Form** on which the **MS Grid** control is placed:-

Private *instance_name*(*instance_count*) **As New** [SG.]oleGridSort

where:

instance_name is any valid name, e.g: **MySortedGrid**

instance_count is the specification of the number of array elements in any of the valid formats.

e.g: **Public** MySortedGrid(2) **As New** SG.oleGridSort

Sub-classing the Grid control

Method #1 - a single instance

In the MS Grid's parent form's Form_Load() event procedure:-

```
Private Sub Form_Load()  
    Set MySortedGrid.NewGrid = Grid1  
  
End Sub
```

Method #2 - an array of instances

This method is more involved because of the restriction of not being able to pass control arrays to procedures.

In the Form_Load() event procedure for the MS Grid's parent form, use similar code to the following:-

```
Private Sub Form_Load()  
Dim grdTemp(1) As Grid    'this example assumes a two-element array  
Dim iCount As Integer  
  
For iCount = 0 To 1  
    Set grdTemp(iCount) = Grid1(iCount) 'cannot pass control  
                                        'arrays directly!  
  
    'If the class instance array is dynamic  
    ReDim Preserve MySortedGrid(iCount)  
  
    'If later binding is used  
    Set MySortedGrid(iCount) = New SG.oleGridSort  
  
    Set MySortedGrid(iCount).NewGrid = grdTemp(iCount)  
  
    Set grdTemp(iCount) = Nothing  
Next  
  
End Sub
```

Using In-Process or Out-Of-Process OLE Automation servers

There can be a considerable difference in performance overhead between using this class in an In-Process server and an Out-of-Process server.

The In-Process server occupies the same address space as the application program, and as such, suffers little or no overhead. The Out-of-Process server on the other hand, occupies a different address space and so for each transfer of data to/from OLE client to OLE server, there are a number of memory translations that take place. This is referred to as *marshalling*.

The internal functionality of the sort mechanism of **oleGridSort** has been designed with these considerations in mind which, in the 16-bit environment (which can only use out-of-process servers) creates a limitation that restricts this class to a maximum data size within the grid of 64K bytes (e.g. 1024 rows x 8 cols x 8 characters per col).

The reason for this is that when an instance of **oleGridSort** invokes the sort function, it first copies the entire grid data into its data area as a single string, which is then parsed into an array for the sorting process. It must do that otherwise the performance penalty of iteratively copying grid rows to the array is prohibitively high. The maximum size for a string in the 16-bit version of VB is 64Kb.

The maximum string length in the 32-bit version of VB 4.0 is approximately 2Gb, and therefore no practical limitation exists in this case.

Hints and Tips

- ◆ Use the Object Browser in VB 4.0's design environment to view all the properties and methods of **oleGridSort**, and to get context help for those properties and methods that are new or have been modified to function differently to the MS Grid control.
- ◆ If more than one property that can invoke the grid sort is modified, set the SortOrder property to SG_SORT_UNSORTED first. This will prevent cascading sorts.

Example:

```
With MyGrid
    .SortOrder = SG_SORT_UNSORTED
    .SortCol = 2
    .SecondSortCol = 3
    .SortOrder = SG_SORT_ASCENDING
End With
```

'Sort constants

```
Public Const SG_SORT_UNSORTED = 0
Public Const SG_SORT_ASCENDING = 1
Public Const SG_SORT_DESCENDING = 2
Public Const SG_NOSECONDSORT = -1
```

'Error constants

```
Public Const SG_ERR_INVALIDOBJ = &H800407CF
Public Const SG_ERR_NODATA = &H800407CE
Public Const SG_ERR_NOMEMORY = &H800407CD
Public Const SG_ERR_KEYTOOLONG = &H800407CC
Public Const SG_ERR_INVALIDCOL = &H800407CB
Public Const SG_ERR_INVALIDSORT = &H800407CA
Public Const SG_ERR_LOADLIBFAILED = &H800407C9
```

'Miscellaneous constants

```
Public Const SG_NULLINDEX = -1
```

Terms and Conditions

oleGridSort is a shareware product and is provided by **Timeamber Limited** at no charge for evaluation. Feel free to share it with friends and colleagues for evaluation, but please do not pass it on altered in any way. The essence of user-supported software is to provide personal computer developers and users with quality software without high prices, yet to provide incentive for programmers to continue to develop new products.

oleGridSort may not be distributed unlicensed as part of an application.

Please encourage other developers to register their copies if they find it useful. All licensed users receive a copy of the latest version of the **oleGridSort** product, free upgrades, and free support for the life of the product.

If you continue to use **oleGridSort** after 30 days, you must purchase a Development Licence (see README.TXT for details) from **Timeamber Limited**.

Corporate/Site license arrangements can be made by contacting **Timeamber Limited**.

oleGridSort is free from run-time royalties.

Licencing

If you continue using **oleGridSort** after a 30 day evaluation period, you must purchase a development licence. To do this, print out and fill in the Development Licence Purchase form provided and send it with your remittance (see README.TXT for details) to **Timeamber Limited**.

You will receive the latest version of **oleGridSort** and a licence (.LIC) file.

Support

Support is provided to licenced users in the following ways.

- Any updates of **oleGridSort** will be sent to you as and when available.
- Timely response to queries/problems via email or phone.

All licenced users will be notified of major updates pending.

Distribution

If **oleGridSort** is unlicensed, none of the files may be distributed as part of an end user application! They may only be used to evaluate the product.

If the end user application is to run only under Windows 3.xx, the following files should be included in the distribution set:-

- SG16.EXE
- SGSE16.DLL
- LFUTIL16.DLL
- SG.HLP

If the end user application is 16-bit, and is to run only under Windows 95/NT, the following files should be included in the distribution set:

- SG32.EXE
- SGSE32.DLL
- LFUTIL32.DLL.
- SG.HLP

If the end user application is 32-bit, the following files should be included in the distribution set:-

- SG32.EXE (out-of-proc server) and/or SG32.DLL (in-proc server)
- SGSE32.DLL
- LFUTIL32.DLL.
- SG.HLP

If the run-time environment may be either or both, include all the above files.

No other files included with this product may be distributed as part of the end user application!

See README.TXT for any last minute variations of the above details.

Disclaimer

Users of **oleGridSort** must accept this disclaimer of warranty:

"**oleGridSort** is supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of **oleGridSort**."

Timeamber Limited

34 Park View
Kiveton Park
Sheffield S31 8SE
England

Managing Director and **oleGridSort** author: **Ian Carter**
Company Number: 3148184

Tel/Fax (auto): (+44)(0)1909-773689
Mobile: 0468-520441

Compuserve: 100435,3040
Email: 100435.3040@compuserve.com

