ExcelDropDown

Generated by Doxygen 1.8.5

Contents

1 Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

    ExcelMousewheelSupport

        ExcelMouseWheelSupport.bas **adds mouse wheel scrolling support to Microsoft Excel ActiveX Controls (e.g** **1**

2 Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

    ExcelDropDown

        ExcelDropDown.cls **provides a versatile drop down functionality for Microsoft Excel worksheets using VBA** **3**

3 Namespace Documentation

3.1 ExcelMousewheelSupport Namespace Reference

ExcelMouseWheelSupport.bas adds mouse wheel scrolling support to Microsoft Excel ActiveX Controls (e.g.

Functions

- Sub StartMouseWheelHook (ByVal Obj As Object)

  Starts the hook to make the specified object scrollable with mouse wheel.
- Sub StopMouseWheelHook (ByVal Obj As Object)

  Stops the hook if the hook was started with the specified object.

Variables

- Const SCROLL_SPEED = 2

  Scroll speed (number of rows to scroll at once).
- Const MAINWINDOW_CLASSNAME = "XLMAIN"

  Class name of the Excel main window. May changes in further Excel versions.

3.1.1 Detailed Description

ExcelMouseWheelSupport.bas adds mouse wheel scrolling support to Microsoft Excel ActiveX Controls (e.g. the ComboBox). This is achieved by low level mouse event capturing via message hook.

Version

1.0.1

Author

MarcoWue (GitHub)

Copyright

Copyright (c) 2013 by MarcoWue.
This work is made available under the terms of the Creative Commons Attribution 3.0 Unported License (CC BY 3.0).

Dependencies

None.

Usage

Note

ExcelMouseWheelSupport needs activated macros to work!

Call StartMouseWheelHook() when the specified control gets the focus. This will start the mouse event capturing. Call StopMouseWheelHook() after the control has lost focus.

Note

After you started the hook, be sure to call StopMouseWheelHook() whenever the Worksheet, Workbook or Window is being deactivated.

Credits

Thanks to Jaafar Tribak on whose work this code is based on. See here.

Version History

- 1.0.1 (2013-09-18) *MarcoWue*

  – Ported to Doxygen code documentation.

- 1.0.0 (2013-09-03) *MarcoWue*

  – Initial release.

3.1.2 Function Documentation

3.1.2.1 Sub ExcelMousewheelSupport::StartMouseWheelHook (ByVal Obj As *Object*)

Starts the hook to make the specified object scrollable with mouse wheel.

Parameters

in	*Obj*	The object to be made scrollable.

3.1.2.2 Sub ExcelMousewheelSupport::StopMouseWheelHook (ByVal Obj As *Object*)

Stops the hook if the hook was started with the specified object.

Parameters

in	*Obj*	The object on which the hook was started.

3.1.3 Variable Documentation

3.1.3.1 Const SCROLL_SPEED = 2

Scroll speed (number of rows to scroll at once).

This constant can be changed to adjust the scrolling behaviour.

4 Class Documentation

4.1 ExcelDropDown Class Reference

4.1.1 Detailed Description

ExcelDropDown.cls provides a versatile drop down functionality for Microsoft Excel worksheets using VBA.

Version

1.0.1

Author

MarcoWue (`GitHub`)

Copyright

Copyright (c) 2013 by MarcoWue.
This work is made available under the terms of the Creative Commons Attribution 3.0 Unported License (`CC BY 3.0`).

Dependencies

- ExcelMouseWheelSupport.bas

- Microsoft Forms 2.0 Object Library (Menu *Tools* > *References*)

Usage

Note

ExcelDropDown needs activated macros to work!

Do the following to use ExcelDropDown in your Excel workbook. These steps are only required once per workbook.

1. Define data validation for the desired cells (*Data* > *Data Validation*).

    - Choose *List* as validation criteria.
    - Choose a data source. You can also specify a name via "=MyName" (e.g. in order to use data on other worksheets). To define and and manage names, have a look at *Formula* > *Name Manager* in the Excel main window.
    - Uncheck *In-cell dropdown*.
    - You presumably want to deactivate *Input Message* and *Error Alert*.

2. Activate the *Developer Tab* in the Excel Settings (How-To).

3. Press *Alt+F11* to run the VBA editor.

    - Add a reference (Menu *Tools* > *References*) to "Microsoft Forms 2.0 Object Library".
    - Import *ExcelDropDown.cls* and *ExcelMouseWheelSupport.bas* into the VBA project of your Excel workbook (Menu *File* > *Import File...*)
    - Put the following code into *ThisWorkbook* (replace occurrences of *Table1* with desired table name)

```
Dim WithEvents Table1DropDown As New ExcelDropDown

Private Sub Workbook_Open()
    Set Table1DropDown = New ExcelDropDown

    '' Set desired options here
    Table1DropDown.ListScrollable = False

    '' At last set the target worksheet
    Table1DropDown.Worksheet = Worksheets("Table1")
End Sub
```

    - If you want to support multiple sheets in your workbook, create a separate ExcelDropDown object for each sheet like shown above.
    - Save and reopen the workbook.

Version History

- 1.0.1 (2013-09-18) *MarcoWue*

    - Ported to Doxygen code documentation.

- 1.0.0 (2013-09-03) *MarcoWue*

    - Initial release.

Public Member Functions

- Event ValidateDropDown (ByVal Target As Range, ByRef Cancel As Boolean)

  Is raised to validate if drop down should be shown in a specific cell.
- Event BeforeDropDown (ByVal Target As Range, ByRef Cancel As Boolean)

  Is raised immediately before drop down is shown.
- Event AfterDropDown (ByVal Target As Range)

  Is raised immediately after drop down is shown.
- Event ExitDropDown (ByVal Target As Range)

  Is raised immediately after drop down is hidden.
- Property Get Worksheet () As Worksheet

  The worksheet ExcelDropDown is assigned to.
- Property Get ComboBoxControl () As ComboBox

  *The ComboBox control ExcelDropDown uses (**read-only**).*
- Property Get CurrentCell () As Range

  *The cell the drop down is currently shown in (**read-only**).*
- Property Get AutoShow () As Boolean

  Determines whether the drop down should be shown automatically.
- Property Get AutoDropDown () As Boolean

  Determines whether the drop down should expanded automatically.
- Property Get ButtonInsideCell () As Boolean

  Determines the drop down button position.
- Property Get ListFixed () As Boolean

  Determines the drop down behaviour.
- Property Get ListRowCount () As Long

  Determines the length of the shown drop down list.
- Property Get ListScrollable () As Boolean

  Determines mouse wheel scrolling support of the drop down list.
- Property Get ItemsFastProcessing () As Boolean

  Determines if advanced list item processing is active.
- Property Get ItemsSort () As Boolean

  Determines whether list items should be sorted alphabetically.
- Property Get ItemsRemoveDuplicate () As Boolean

  Determines whether duplicate list items should not be shown.
- Property Get ItemsDuplicateCompare () As VbCompareMethod

  Determines whether ItemsRemoveDuplicate should be case sensitive.
- Property Get ItemsRemoveBlank () As Boolean

  Determines whether empty list items should not be shown.
- Property Get ItemsAddFirstBlank () As Boolean

  Determines whether a first blank item should be added to the drop down list.
- Property Get ItemsTrim () As Boolean

  Determines whether leading and trailing spaces should be removed from items.
- Function Show (ByVal Target As Range, Optional ByVal NoAutoDropDown As Boolean=False) As Boolean

  Shows the drop down at the specified target cell.
- Sub Hide ()

  Hides the currently shown drop down.

4.1.2 Member Function Documentation

4.1.2.1 Event ValidateDropDown (ByVal Target As *Range,* ByRef Cancel As *Boolean*)

Is raised to validate if drop down should be shown in a specific cell.

Parameters

in	*Target*	The new target cell where the drop down is going to be shown.
out	*Cancel*	Set to `True` to prevent the drop down from being shown.

4.1.2.2 Event BeforeDropDown (ByVal Target As *Range,* ByRef Cancel As *Boolean*)

Is raised immediately before drop down is shown.

If desired you can modify ComboBoxControl appearance or content here.

Parameters

in	*Target*	The new target cell where the drop down is going to be shown.
out	*Cancel*	Set to `True` to prevent the drop down from being shown.

4.1.2.3 Event AfterDropDown (ByVal Target As *Range*)

Is raised immediately after drop down is shown.

Parameters

in	*Target*	The cell where the drop down is currently shown at.

4.1.2.4 Event ExitDropDown (ByVal Target As *Range*)

Is raised immediately after drop down is hidden.

Parameters

in	*Target*	The cell where the drop down was shown at.

4.1.2.5 Property Get Worksheet ( )

The worksheet ExcelDropDown is assigned to.

Setting this property causes initialization of the ComboBox control.

To handle multiple worksheets, create a separate ExcelDropDown object for each desired sheet.

4.1.2.6 Property Get ComboBoxControl ( )

The ComboBox control ExcelDropDown uses (**read-only**).

You may want to change its appearance or content utilizing the BeforeDropDown event.

Is `Nothing` before property Worksheet is set.

4.1.2.7 Property Get CurrentCell ( )

The cell the drop down is currently shown in (**read-only**).

Is `Nothing` when no drop down is displayed at the moment.

4.1.2.8 Property Get AutoShow ( )

Determines whether the drop down should be shown automatically.

If `True`, the drop down box is shown automatically whenever an appropriate cell is selected in the worksheet.

4.1.2.9 Property Get AutoDropDown ( )

Determines whether the drop down should expanded automatically.

If `True`, it is is expanded automatically every time the drop down is shown.

4.1.2.10 Property Get ButtonInsideCell ( )

Determines the drop down button position.

If `True`, the drop down button is shown inside the target cell.

If `False`, it is shown besides of the cell.

4.1.2.11 Property Get ListFixed ( )

Determines the drop down behaviour.

If `True`, the user can only choose between the shown list items and is not able to enter text into the drop down field.

4.1.2.12 Property Get ListRowCount ( )

Determines the length of the shown drop down list.

Is expressed as number of rows.

4.1.2.13 Property Get ListScrollable ( )

Determines mouse wheel scrolling support of the drop down list.

If `True`, mouse wheel support is enabled by utilizing a so called hook callback function which captures mouse activity on a low level basis.

Note

> If you are encountering strange behaviour set this option to `False`.

4.1.2.14 Property Get ItemsFastProcessing ( )

Determines if advanced list item processing is active.

If `True`, the drop down list items are displayed solely by faster internal Excel routines utilizing the ListFillRange property of the ComboBox control.

If `False`, additional processing of the items is done which may takes some time with larger data amounts.

Note

> Please set to `False` if you want to customize list items in the BeforeDropDown event or if you have set one of the following properties to `True`:
> - ItemsSort
> - ItemsRemoveDuplicate
> - ItemsRemoveBlank
> - ItemsAddFirstBlank

4.1.2.15 Property Get ItemsAddFirstBlank ( )

Determines whether a first blank item should be added to the drop down list.

If `True`, a first blank item will be added to the drop down list regardless of the ItemsRemoveBlank property.

4.1.2.16 Function Show (ByVal Target As *Range,* Optional ByVal NoAutoDropDown As *Boolean* = `False` **)**

Shows the drop down at the specified target cell.

This method usually has not to be called if AutoShow is set to `True`.

Call this method to show the drop down manually. Please note that the target cell must always have data validation enabled with a valid 'List' criteria.

Parameters

in	*Target*	The cell the drop down should be shown at.
in	*NoAutoDrop-Down*	If `True`, the AutoDropDown property will be ignored

Returns

> `True` if the drop down could be shown successfully, `False` otherwise.

4.1.2.17 Sub Hide (   )

Hides the currently shown drop down.

This method usually has not to be called by the user as the drop down is always hidden automatically e.g. when the cell selection is changing.

Index