

# Automating Flex Grid Component

## Introduction to VSFlexGrid

Before I talk about how to automate application designed on VS Flex Grid, let me explain a bit about VS Flex Grid itself.

**VSFlexGrid 8.0** is part of ComponentOne Studio Enterprise, the largest and most complete component toolset for developing all layers of Windows, Web, and Mobile applications.

The **VSFlexGrid** control is a powerful, full-featured grid. It provides new ways to display, edit, format, organize, summarize, and print tabular data.

The **VSFlexGrid** control was designed to be used either in **unbound** mode, where the grid "owns" the data, or in **data-bound** mode, where the grid is used to view and edit data that belongs to a database.

The **VSFlexGrid** control has two properties that determine its dimensions: **Rows** and **Cols**. When used in bound mode, these properties are set automatically based on how much data is available on the data source. In unbound mode, you can set them to arbitrary values.

## Tips for Automation

Automating with QTP on ActiveX flex grids is a bit tricky. If you just record the action you perform on this grid, ( like selecting a row or editing value in a cell) QTP can't recognize the cells or rows since the Flex Grid is treated as a single object and not as a table with rows and columns. When you perform a normal recording with QTP, the script is created with X and Y co-ordinates. All cells and rows in the flex grid are recognized by their co-ordinates. If scripts are recorded with X and Y co-ordinates, it would fail on execution when screen resolutions are modified. Also, it would be difficult to pass the X and Y co-ordinates to perform an action on cells.

Following are the ways in which you can perform various actions on the flex grid, without having to specify the co-ordinates.

- Editing data in a cell of the flex grid:

```
Set obj=Browser("").Page("").Frame("").ActiveX("ComponentOne FlexGrid").Object
obj.Select 2,2 - This selects the cell located at row 2 and column 2.
Browser("").Page("").Frame("").ActiveX("ComponentOne FlexGrid").Type <value>
```

- To open a look up or pop up window located in the cell of the flex grid. The window can be opened by clicking on the button located inside the cell. The cell with button would look like:



```
Set obj=Browser("").Page("").Frame("").ActiveX("ComponentOne FlexGrid").Object
obj.Select <row>,<column>
Browser("").Page("").Frame("").ActiveX("ComponentOne FlexGrid").Type micReturn
```

- To get row count from flex grid:

```
Set obj=Browser("").Page("").Frame("").ActiveX("ComponentOne FlexGrid").Object
RowCount= obj.GetROProperty("Rows")
```

- Selecting data from list box of a cell in the flex grid:

The following function can be used to select values from drop down or from context menus.

```
Function contextMenuSelect(downBy,rightBy)
If downBy<>"" Then
    Set WshShell = CreateObject("WScript.Shell")
    WshShell.sendKeys "{PGUP}"
    For i = 1 To downBy
        WshShell.sendKeys "{DOWN}"
        wait(0.5)
    Next
    If rightBy>0 Then
        WshShell.sendKeys "{RIGHT}"
        wait(0.5)
    End If
    If rightBy>1 Then
        For j=1 to rightBy
            WshShell.sendKeys "{DOWN}"
            wait(0.5)
        Next
    End If
    WshShell.sendKeys "{ENTER}"
    Set WshShell = nothing
End If
End Function
```

```
obj.Select <row>,<column>      -Select the desired cell
Browser("").Page("").Frame("").ActiveX("ComponentOne FlexGrid").Type micReturn -
Open the drop down
contextMenuSelect <downby>,<rightby> -Pass the location of the item in the drop down to select
```

- To fetch data from a field in the flex grid:

```
set objTable=Browser("").Window("").Page("").Frame("").ActiveX("").Object
status=objTable.TextMatrix(1,2)
```

- To fetch co-ordinates of a cell based on value/string in another cell, the following function can be used:

**object.GetTextLocation** (*TextToFind*, *Left*, *Top*, *Right*, *Bottom*, [*MatchWholeWordOnly*])

Where,

L=	Left co-ordinate of the search area within the screen.
T=	Top co-ordinate of the search area within the screen.
R=	Right co-ordinate of the search area within the screen.
B=	Bottom co-ordinate of the search area within the screen.
<i>TextToFind</i>	= Value to find.
<i>MatchWholeWordOnly</i>	= Optional, Boolean value. If True, the method searches for occurrences that are whole words only and not part of a larger word. If False, the method does not restrict the results to occurrences that are whole words only.

**Default value = True**

The method returns the exact co-ordinates of the specified text to the supplied arguments (l,t,r,b).

The function can be used as below, to fetch the co-ordinates of a cell. After executing this function, the L (left) and T (Top) arguments would hold the X and Y co-ordinates of the cell.

```
L=-1
```

```
T=-1
```

```
B=-1
```

```
R=-1
```

```
getValue=Browser("").Page("").Frame("").ActiveX("").GetTextLocation("<pass text>",l, t, r, b,  
True)
```