

ICONICS, Inc.
100 Foxborough Blvd.
Foxborough, MA 02035
Phone: (508) 543-8600
Fax: (508) 543-1503
[Worldwide Support Centers](#)
www.iconics.com



© **ICONICS, Inc. All rights reserved.**
Specifications are subject to change without notice.
GENESIS, GENESIS32, Pocket GENESIS, BizViz
and their respective modules, OPC-To-The-Core, and
Visualize Your Enterprise are trademarks of ICONICS,
Inc. Other product and company names mentioned
herein may be trademarks of their respective owners.

Introduction

The **Recipe ActiveX** is included in the GENESIS32 installation, and it can be inserted into any container that has the capability to embed ActiveX objects, including GraphWorX32, Visual Basic forms, Microsoft Word, and Microsoft Excel.

The Recipe ActiveX allows you to visualize and edit recipe item values within GraphWorX32.

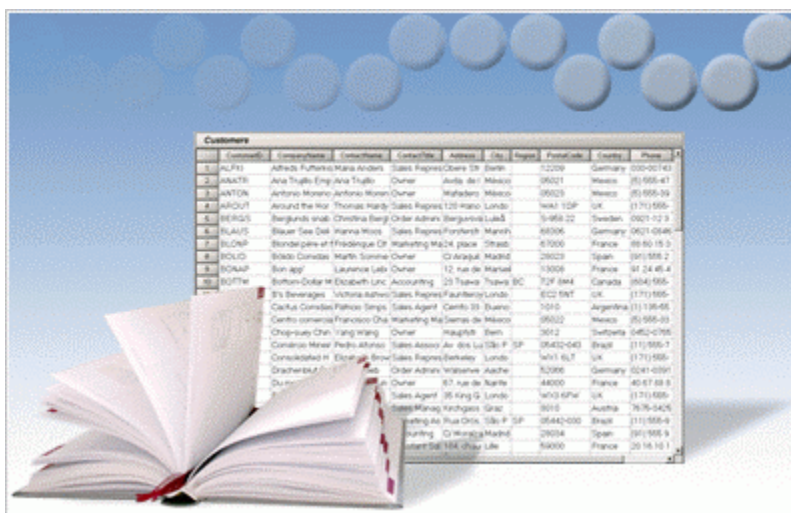
Inserting the ActiveX Into a GraphWorX32 Display

To insert the Recipe ActiveX, choose **Insert Object** from the **Edit** menu in GraphWorX32, or click the **Recipe ActiveX** button on the **ActiveX** toolbar, as shown in the figure below.



Recipe ActiveX Button on the ActiveX Toolbar

The **Recipe ActiveX** will appear in the display, as shown in the figure below.



Recipe ActiveX

Recipe Management and Visualization

The Recipe ActiveX is just one of several applications in the ICONICS Recipe Management System, which consists of the following components:

The Front End

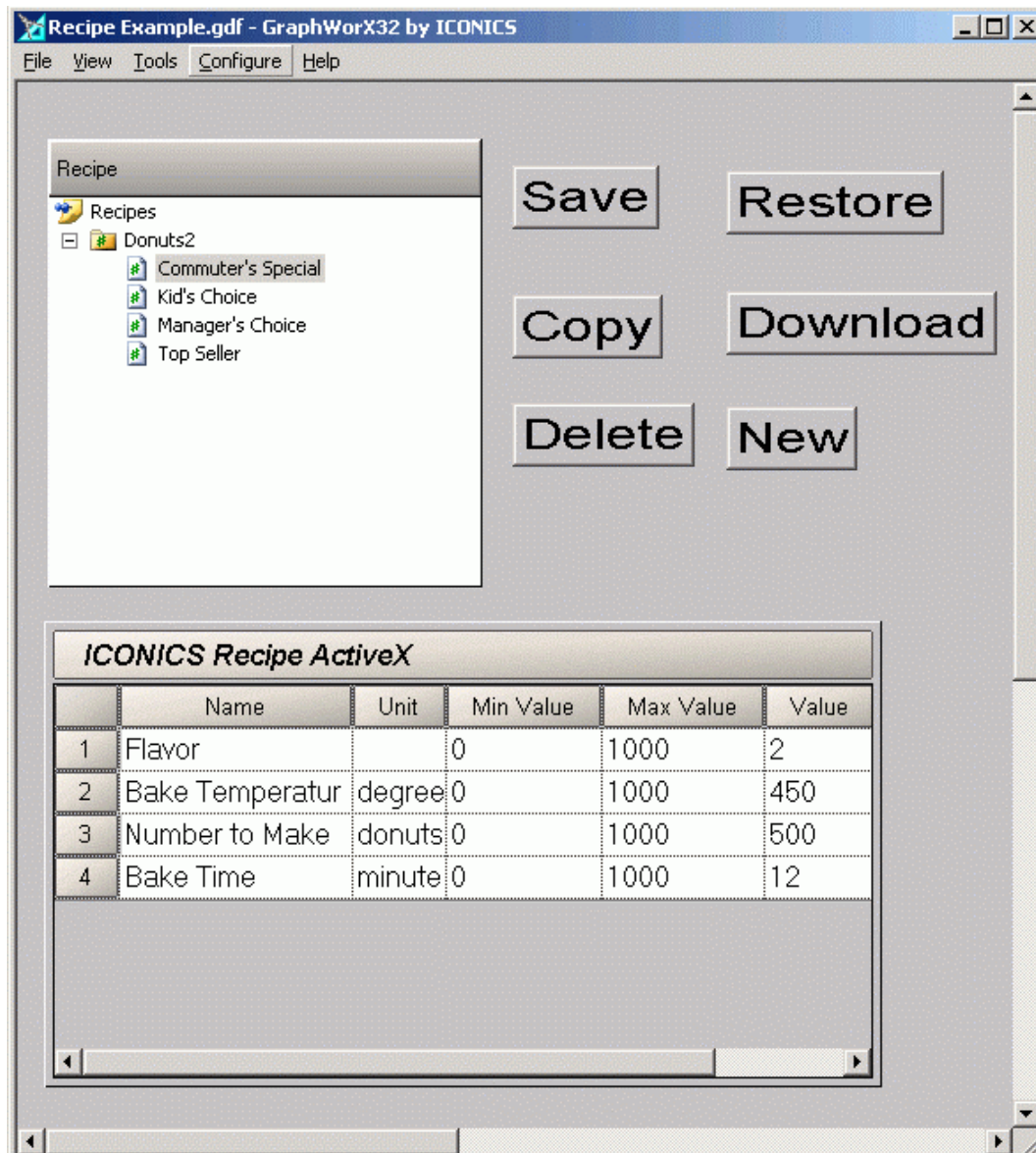
- **Tree Explorer ActiveX:** Consists of a tree control that allows you to browse for and visualize recipes within GraphWorX32. For more information, please see the Tree Explorer ActiveX help documentation.
- **Recipe ActiveX:** Consists of a data table that allows you to visualize and edit recipe item values within GraphWorX32.
- **Recipe Pick Action:** Enables you to create buttons in your GraphWorX32 display that allow you to apply changes made to recipe items via the Tree Explorer ActiveX and the Recipe ActiveX. Changes that you make to the recipe values can be applied to the Recipe Server, the active recipe configuration, and the connected OPC server.

The Back End

- **Unified Data Manager:** Use the Unified Data Manager configurator to create and activate your recipe configuration.
- **OPC Servers:** Recipe parameter values are associated with OPC tags, which enable the connected OPC servers to communicate with the Recipe Server and to relay recipe values to the PLC device in the end-user application.
- **Recipe Server:** Communicates and relays recipe data between the Unified Data Manager configuration database, the Tree Explorer ActiveX, the Recipe ActiveX, and the OPC server to which the recipes are connected.

The Tree Explorer ActiveX, Recipe ActiveX, and associated recipe pick action buttons can be inserted into a GraphWorX32 display, as shown in the figure below, so that recipe data can be viewed directly from the display.

Note: For complete information about recipe configuration and visualization, please see the Recipe Management System section in the GraphWorX32 help documentation.



Tree Explorer ActiveX and Recipe ActiveX in Runtime

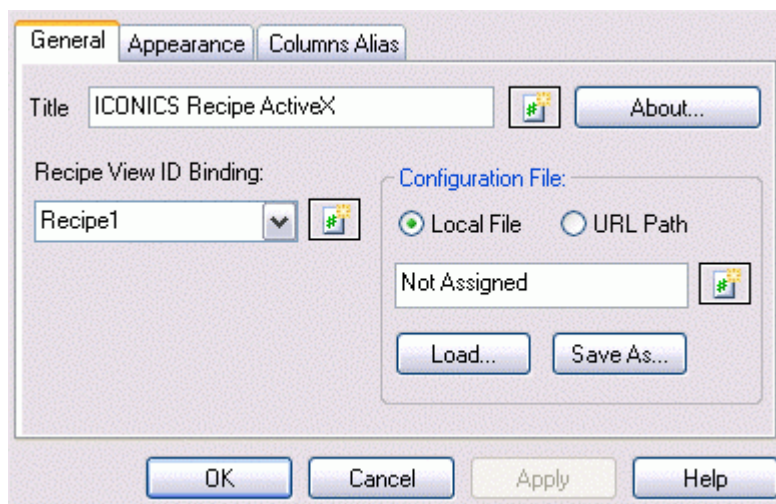
Recipe ActiveX Properties

Double-clicking the Recipe ActiveX opens the **Recipe ActiveX Properties** dialog box, which contains the following property pages:

- **General**
- **Appearance**
- **Columns Alias**

General

The **General** tab of the **Recipe ActiveX Properties** dialog box, shown below, allows you to link the Tree Explorer ActiveX configuration to the Recipe ActiveX.



Recipe ActiveX Properties: General Tab

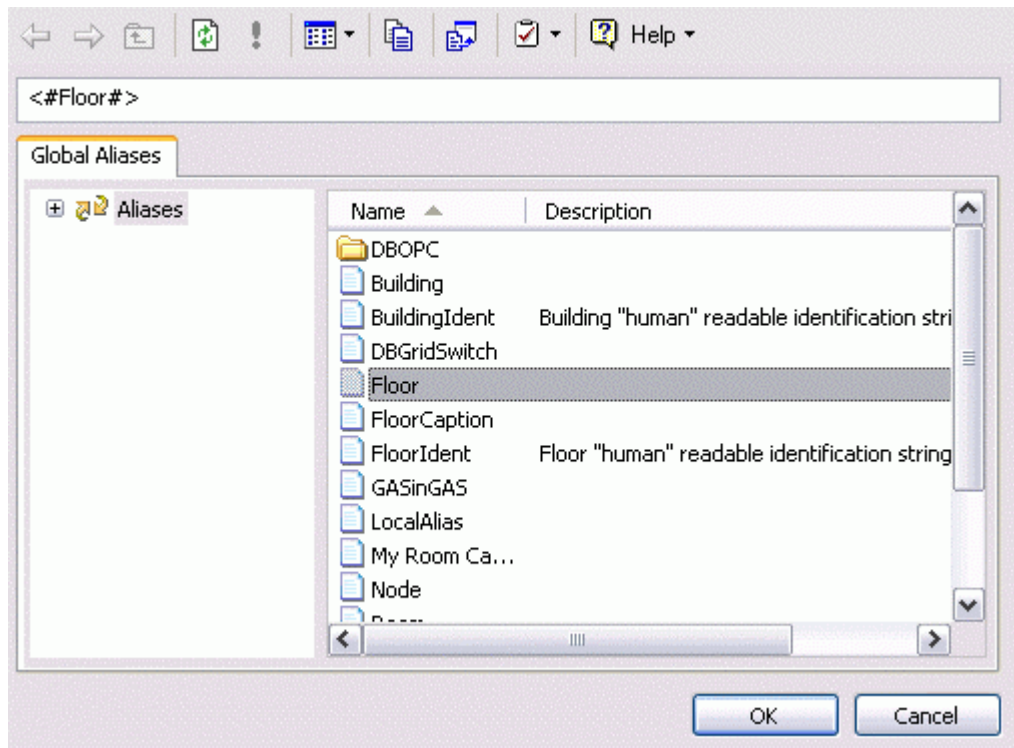
Configure the following **General** tab settings:

- **Title:** Enter a title for the ActiveX.
- **Recipe View Binding:** The **Recipe View ID Binding** field lists all recipe view IDs that you configured in the Tree Explorer ActiveX. Choose the recipe view ID that you want to use from the drop-down list. In the example shown in the figure above, the "Recipe1" view ID is selected. This view ID will be used to link the Tree Explorer ActiveX configuration to the Recipe ActiveX.
- **About:** Clicking the **About** button displays both technical support information and the About Box, which contains registration and serial number information as well as system resource information, such as memory and disk space available.
- **Configuration File:** By default, information is stored as part of the container application's document file. You can also store information as part of a separate (.rcp) file. To save the .rcp file, under **Configuration File** click **Save As** to open the **Save Configuration Info** dialog box. Enter a name for the .rcp file in the **File Name** field, and then click **Save**.

You can also open an existing configuration file by clicking the **Load** button on the **General** tab and browsing for the file. The ActiveX will attempt to load the configuration file. If **URL Path** is selected, the ActiveX will use the specified network URL path upon loading the file.

Note: The ActiveX cannot be saved to a URL path. To create a URL file, save a configuration to a local file and copy the file to the desired network location. When the **URL Path** option is selected, you can also enable the **Use Local Settings** option, which, if checked, stores runtime changes to a local copy of the remote configuration file. It will use this local file every time the HTML page that hosts the Viewer is loaded inside the Web browser.

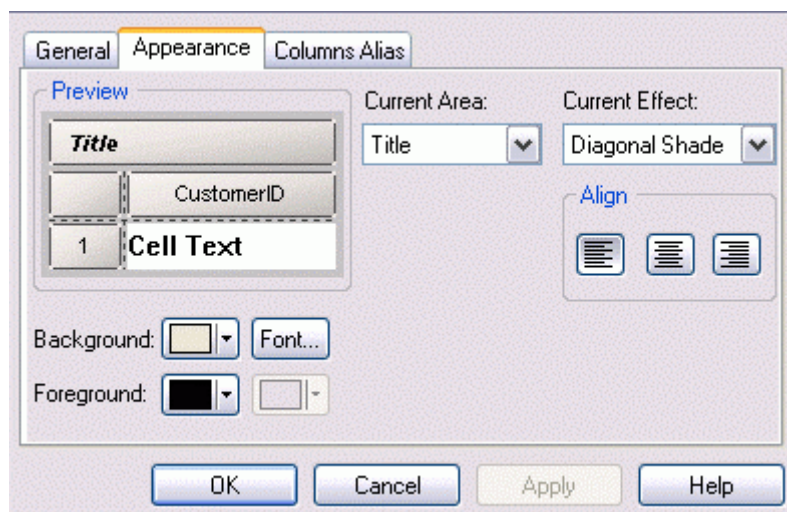
You can also select aliases to use for the configuration file. Clicking the **#** button and selecting **Insert Global Alias** from the pop-up menu opens the Global Aliases tab of the Unified Data Browser, as shown in the figure below. Select a global alias from the Unified Browser, which includes all global aliases in the global alias database. This eliminates the need to manually type in the alias name. All global aliases that are configured in the Global Alias Engine Configurator are conveniently available to choose from inside the browser. The tree control of the Global Alias Engine Configurator is mimicked in the tree control of the Global Alias Browser. Select a global alias by double-clicking the alias name (e.g. "Floor" in the figure below). The alias name appears at the top of the browser, which automatically adds the **<#** and **#>** delimiters to the alias name. Click the **OK** button.



Selecting an Alias From the Global Alias Browser

Appearance

The **Appearance** tab of the **Recipe ActiveX Properties** dialog box, shown below, determines how the recipe grid looks in runtime mode. The **Preview** window allows you to see your changes in configuration mode.



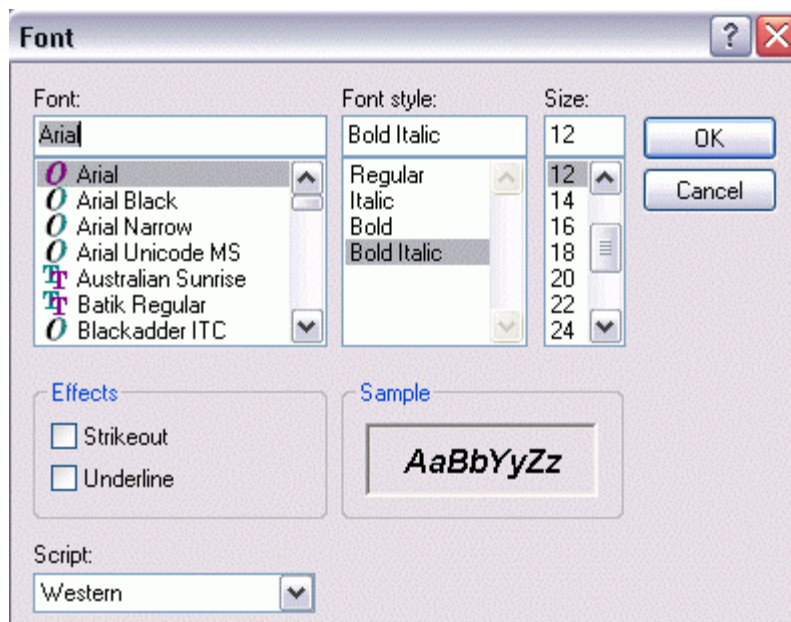
Recipe ActiveX Properties: Appearance Tab

Under **Current Area**, select a recipe grid item to configure (Title, Headers, Cell Text, or Background) and then configure the appearance of the item.

The colors section allows you to choose colors for the background of the display, the background color of the text, the background color of the header and the foreground text color. Select colors for the **Background** and **Foreground** by clicking the color palettes. For the background, you can also choose the color of the horizontal and vertical grid lines in the recipe grid. A **Color** box appears, allowing you to choose a preconfigured color or to create your own color using the **Other** button.

Font

For **Title**, **Cell Text** and **Headers**, click the **Font** button to specify the font size, style, and text effects (i.e. underline and strikethrough) for the recipe grid, as shown in the figure below.



Selecting a Font Style for the Viewer

Visual Effects and Alignment

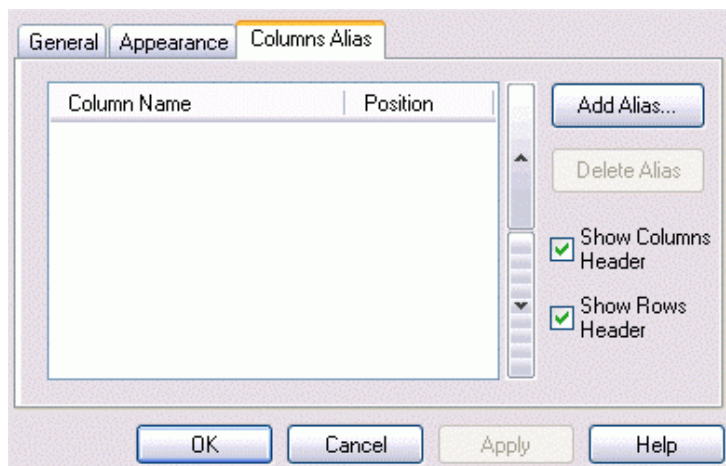
For **Title**, **Cell Text** and **Headers**, the **Align** buttons determine the justification (left, center or right) of the text.

For **Title** and **Headers**, you can apply various stylistic graphical effects (e.g. bumps, gradients, and shading) by selecting from the drop-down list under Current Effect.

For the background, you can select **Line Style** (e.g. solid, dashed, dotted, etc.) for both **Horizontal** and **Vertical** grid lines that appear in the recipe grid.

Columns Alias

The **Columns Alias** tab of the **Recipe ActiveX Properties** dialog box, shown below, allows you to create column names and associate each name with a column that is displayed in the recipe grid.

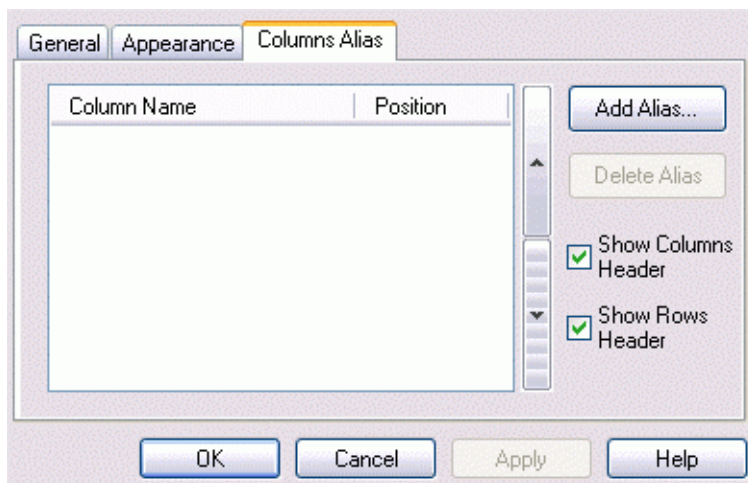


Recipe ActiveX Properties: Columns Alias Tab

Creating Column Aliases

To create a new column alias:

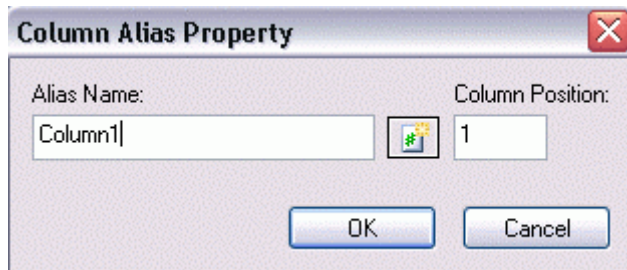
1. On the **Columns** tab, click the **Add Alias** button, as shown in the figure below.



Adding a Column Alias

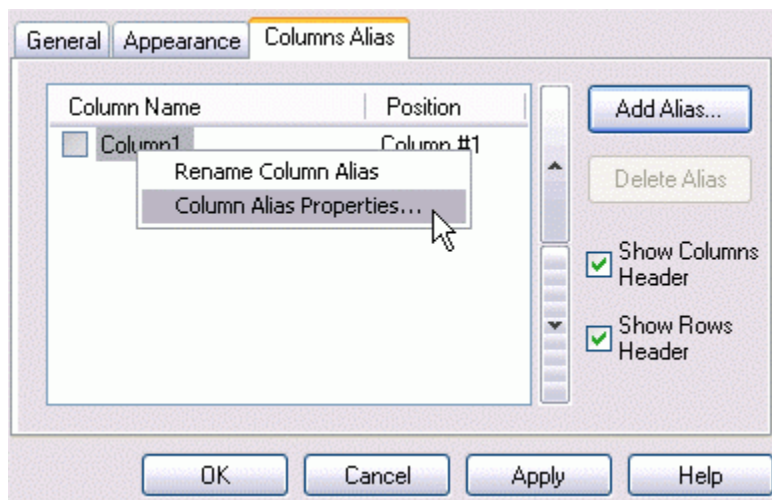
2. This opens the **Column Alias Property** dialog box, as shown in the figure below. In the **Alias Name** field, specify a column name. Then specify a **Column Position** number, which corresponds to the position of the column where the alias will be applied in the Grid display. Click **OK**.
Note: With column aliasing, you can define aliases for the columns that you need. Column positions do not need to be sequential. For example, you could define two aliases: one for column position 3 and another for column position 13.

Note: You can also click the # button to select a global alias from the global alias browser.



Column Alias Property

3. The new column alias and position number appear in the under the **Column Name** list, as shown in the figure below. You have the following additional options:
 - **Add Alias:** You can add additional column aliases to the list. When multiple column aliases are selected, click the up and down arrow buttons to change the order in which the columns are displayed in the Recipe ActiveX Grid. The **Position** number corresponds to the position of each data item in the Grid display.
 - **Delete Alias:** Deletes the selected column alias from the list.
 - **Rename Column Alias:** Enables you to change the name of the selected column alias.
 - **Column Alias Properties:** Opens the **Column Alias Property** dialog box for the selected column alias.



Editing Column Alias Properties

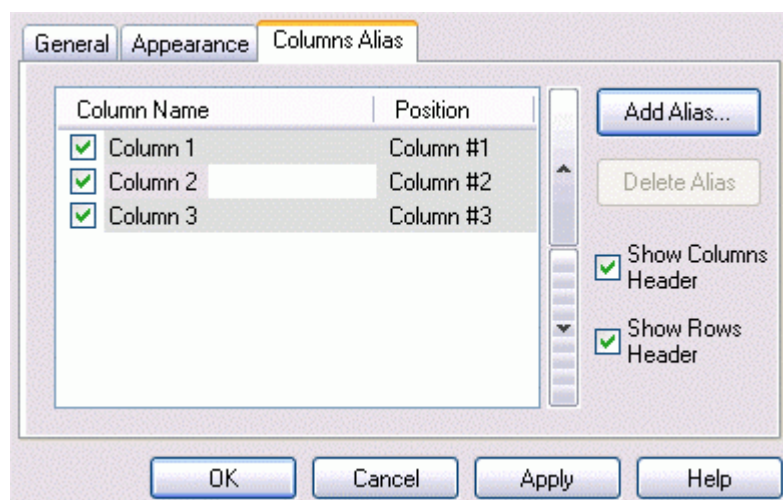
4. When you enter the display into runtime mode, you will see the associated data items displayed in the recipe grid with the substituted column aliases, as shown in the figure below. You can change the column width by moving the column dividers.

ICONICS Recipe ActiveX				
	Column1	Unit	Min Value	Max V
1	Flavor		0	1000
2	Bake Temper	degrees	0	1000
3	Number to Ma	donuts	0	1000
4	Bake Time	minutes	0	1000

Column Aliases Displayed in Recipe Grid

Hiding Columns

Suppose that you have configured a database connection that reads a data set with multiple columns, and you have created a column alias for each column of data that is read from the database, as shown in the figure below. You can hide each columns from the display by simply unchecking the check box next to the column.



Multiple Column Aliases

When you start runtime mode, the unchecked column will be hidden.

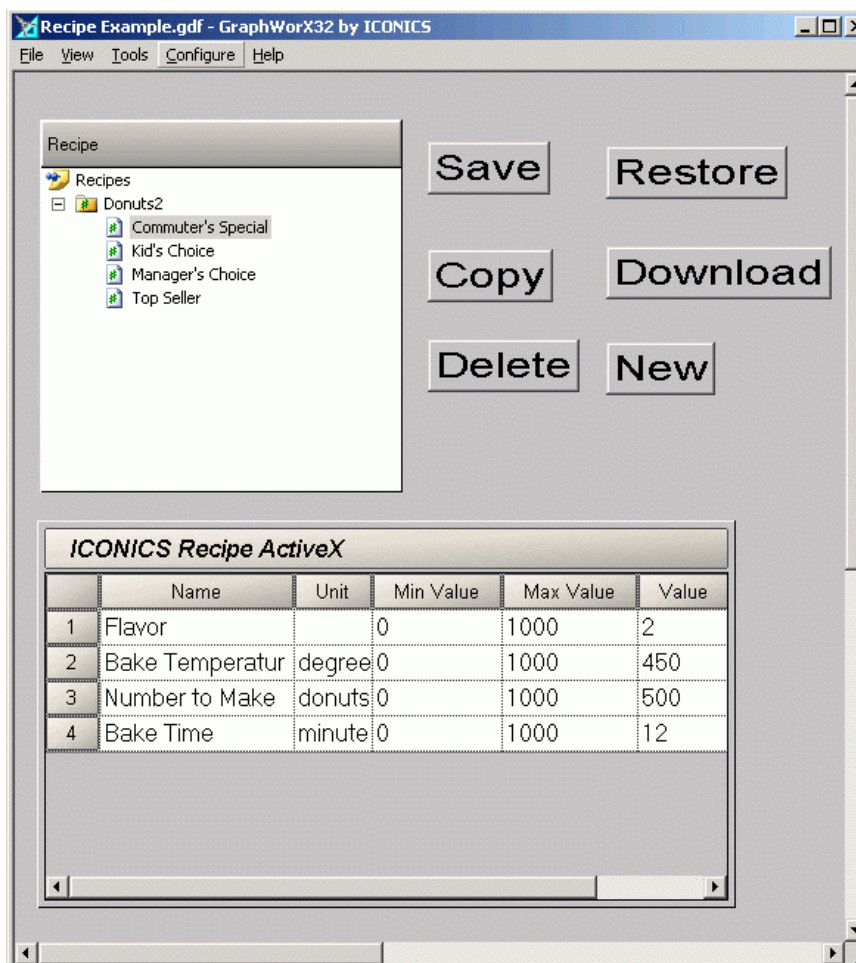
Note: You can also hide the column headers and row headers. To hide the column names in the Grid, uncheck the **Show Columns Header** check box on the **Columns Alias** tab. To hide the row names (or numbers) in the Grid, uncheck the **Show Rows Header** check box on the **Columns Alias** tab.

Viewing Recipe Data in Runtime

Once you have completed your Recipe ActiveX configuration in GraphWorX32 (in conjunction with the Tree Explorer ActiveX and recipe pick action buttons), you are ready to view the recipe item values in runtime mode.

Note: For complete information about recipe configuration and visualization, please see the Recipe Management System section in the GraphWorX32 help documentation.

1. Click the **Runtime** menu in GraphWorX32 to start runtime mode. As you can see in the figure below, the Tree Explorer ActiveX displays the recipes and recipe items from the active recipe configuration in the Unified Data Manager. In the example shown in the figure below, the "Donuts2" recipe is selected, and its four recipe items are displayed in the tree control: **Commuter's Special**, **Kid's Choice**, **Manager's Choice**, and **Top Seller**.
2. When you click on a recipe item in the Tree Explorer ActiveX, the parameters (ingredients) are displayed in the Recipe ActiveX, which consists of a data table with rows and columns. Each row lists a parameter, and for each parameter there are multiple data columns for units, minimum value, maximum value, value, associated OPC tag, etc. In this example recipe configuration, the parameters are **Flavor**, **Bake Temperature**, **Number to Make**, and **Bake Time**.



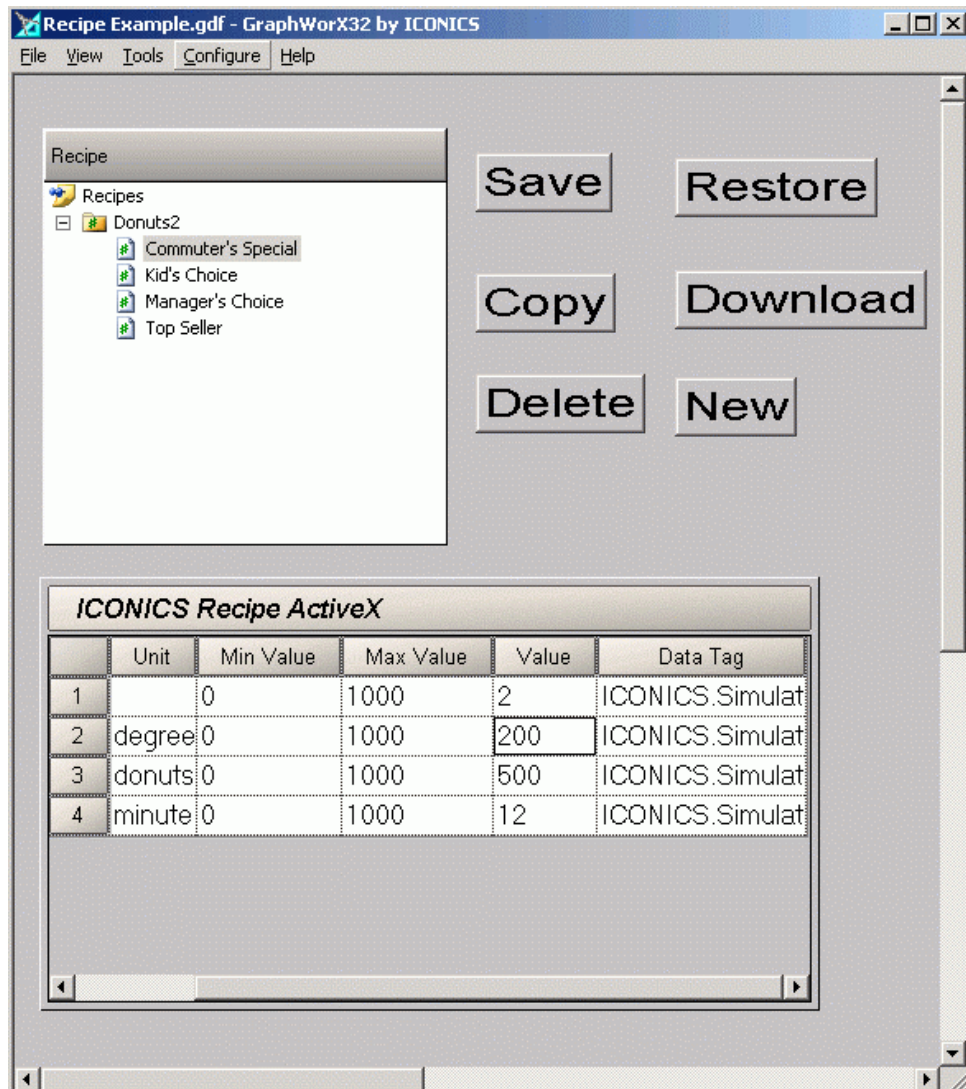
Tree Explorer and Recipe ActiveX in Runtime

3. In runtime mode, the Recipe ActiveX allows you to change the values of recipe parameters (ingredients) in real time. To change a value, you can simply click inside the data table in the **Value** column and type in the new value, as shown in the figure below. When you change the values of recipe parameters in the Recipe ActiveX, you need a way to update the values to the Recipe Server, which then applies the changes to the active recipe configuration. Each recipe parameter is associated with an OPC tag, so you also need a way to download the updated values the OPC server. The way to do this is by using the **Recipe Pick Action** in GraphWorX32, which allows you to save new values to the recipe configuration and download recipe values to the OPC server, which then relays the new values to the PLC device in the end-user application.

	Max Value	Value	Data Tag
1	1000	2	ICONICS.Simulat
2	1000	400	ICONICS.Simulat
3	1000	500	ICONICS.Simulat
4	1000	12	ICONICS.Simulat

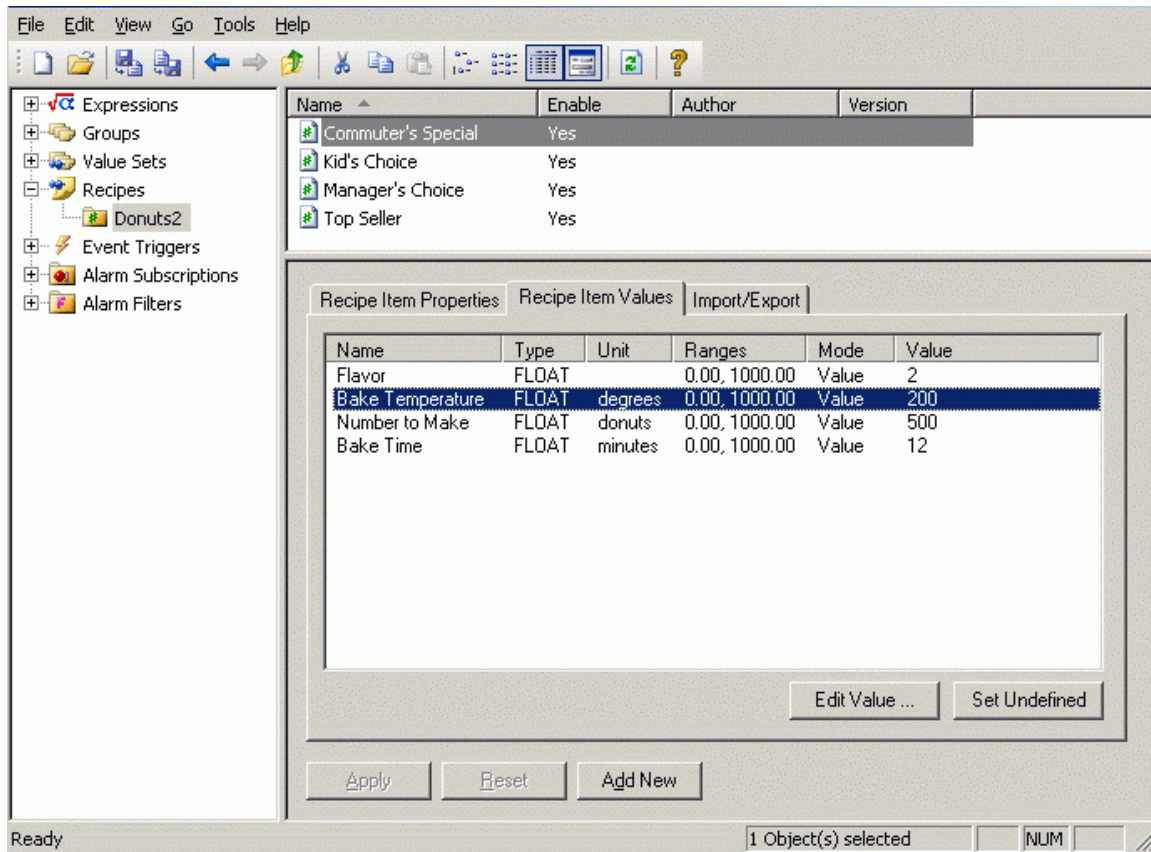
Changing Recipe Values in Real Time

4. When you change the value of a recipe parameter in the Recipe ActiveX (e.g. the **Bake Temperature** is changed from 400 degrees to 200 degrees in the figure below), you must click the **Save** button to commit the value changes to the Recipe Server.



Recipe Value Changed

5. The Recipe Server then updates the value change in the active recipe configuration. To see the recipe configuration change in the Unified Data Manager, select **Global Refresh** from **View** menu in the Unified Data Manager configurator, and then go to the **Recipe Item Values** tab, as shown in the figure below. As you can see in the example below, the value for the **Bake Temperature** has changed from 400 degrees to 200 degrees.

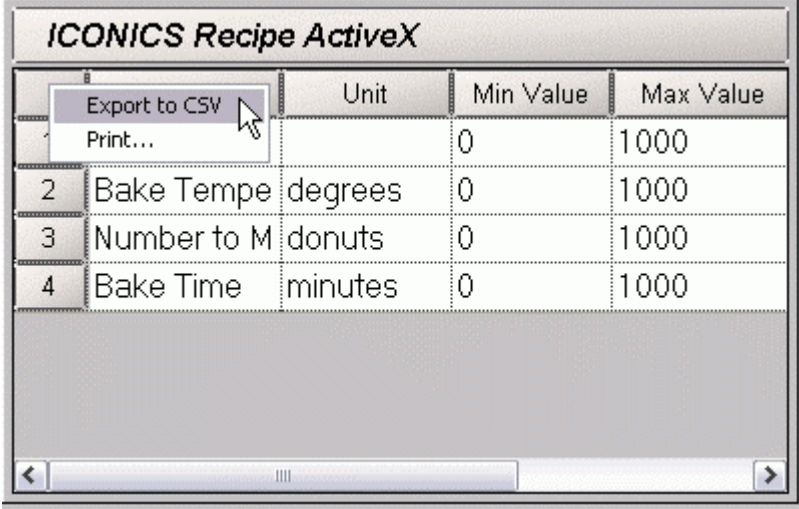


Value Change Updated in Active Recipe Configuration

- Each recipe parameter is associated with an OPC tag, so you also need a way to download the updated values to the connected OPC servers. After you save the recipe value changes, click the **Download** button in the GraphWorX32 to download the values from the Recipe Server to the OPC servers. The OPC servers resolve the values of the recipe and then relay the information to the PLC device to implement the changes in the end-user product.

Exporting and Printing Data

The Recipe ActiveX allows you to export or print data during runtime mode. During runtime mode, right-click on the recipe grid and select **Export to CSV** or **Print**, as shown in the figure below. If you select **Export to CSV**, you can specify the name of the CSV file and save the data. To view the exported data, open the CSV file in Microsoft Excel.



The screenshot shows a window titled "ICONICS Recipe ActiveX". Inside, there is a table with four columns: an index column, a description column, a "Unit" column, a "Min Value" column, and a "Max Value" column. The table contains three rows of data. A context menu is open over the first row, showing "Export to CSV" and "Print..." options. A mouse cursor is pointing at "Export to CSV". Below the table is a large grey rectangular area and a horizontal scrollbar.

		Unit	Min Value	Max Value
1			0	1000
2	Bake Tempe	degrees	0	1000
3	Number to M	donuts	0	1000
4	Bake Time	minutes	0	1000

Exporting and Printing Data

Recipe ActiveX OLE Automation Reference

This reference describes the OLE Automation features available in the Recipe ActiveX. The Recipe ActiveX provides a COM interface that allows automation interfaces run from within the ActiveX container to manipulate the Recipe ActiveX control as it is running. The interface is available to all programming languages that support COM, including Visual Basic (VB), Visual Basic for Applications (VBA), and Microsoft Visual C++.

To access the Automation interface from VB and VBA, the Recipe ActiveX must be made available by selecting **RecipeLib** in the Object Browser.

Control Properties

AutoSelectRow

Type: Boolean

Description

If true, when the user clicks on a cell the whole row will be highlighted.

BackColor

Type: OLE_COLOR

Description

Sets/gets the background color of the column.

Example

```
Dim column As OGridColumnWrapper
Set column = ThisDisplay.Recipe1.GetColumn(1)
column.ModifyBackColor = True
column.BackColor = RGB(0, 255, 0)
```

The above example will change the background color for column 1 to green.

BorderType

Type: String

Description

Sets/gets control's border appearance. Can be one of the following:

"0" - none

"1" - flat border

"2" - 3D border

Example

Sets 3D look for the control's border.

```
ThisDocument.Recipe1.BorderType="2"
```

CellHAlign**Type: Short****Description**

Sets/gets the default cell alignment.

"0" – Left Alignment

"1" – Center Alignment

"2" – Right Alignment

Example

ThisDocument.Recipe1.CellHAlign="2"

ColumnHAlign**Type: Short****Description**

Sets/gets the column header alignment.

"0" – Left Alignment

"1" – Center Alignment

"2" – Right Alignment

Example

ThisDocument.Recipe1. ColumnHAlign ="1"

DisplayGridTitle**Type: Boolean****Description**

Sets/gets Boolean value that controls the visibility of the grid title bar.

Example

ThisDocument.Recipe1. DisplayGridTitle = TRUE

FileName**Type: BSTR****Description**

Sets/gets the name of the Recipe ActiveX configuration file.

Example

ThisDocument.Recipe1.FileName="C:\RecipeConfig.rcp"

Font**Type:** StdFont**Description**

Returns a reference to the control's default font.

Example

To set control's default font to Arial, 14 pt., use the following statements:

With ThisDocument.Recipe1.Font

.Name="Arial"

.Size=14

End With

ForeColor**Type:** OLE_COLOR**Description**

Sets/gets foreground color (cell's font color) property of the Recipe ActiveX.

Example

ThisDocument.Recipe1. ForeColor=RGB(0,0,0)

GridBackColor**Type:** OLE_COLOR**Description**

Sets/gets color value used to paint the gaps between rightmost column and right edge of the grid, last row and bottom edge of the grid, and row headers column (the 'back area' of the grid).

Example

ThisDocument.Recipe1. GridBackColor =RGB(0,127,0)

GridDefRowHeight**Type:** Integer**Description**

Sets/gets the default row height for data grid.

Example

ThisDocument.Recipe1. GridDefRowHeight=16

GridHeaderShade**Type:** Short**Description**

Sets/gets the cell effect for the grid's title bar header.

“0” = Soft Bump
“1” = Diagonal Shade
“2” = Horizontal Shade
“3” = Vertical Shade
“4” = Horizontal Bump
“5” = Vertical Bump
“6” = Metal

GridHorizLinesColor and GridVertLinesColor

Type: OLE_COLOR

Description

Sets/gets color for horizontal or vertical grid lines.

Example

ThisDocument.Recipe1. GridHorizLinesColor =RGB(127,127,127)

ThisDocument.Recipe1. GridVertLinesColor = RGB(127,127,127)

GridHorizLineStyle and GridVertLineStyle

Type: Integer

Description

Sets/gets style for horizontal or vertical grid lines. Can be one of the following:

- 0 - no line
- 1 - solid line
- 2 - dot line
- 3 - dash line
- 4 - dashdot line
- 5 - dashdotdot line

Example

ThisDocument.Recipe1. GridHorizLineStyle =0

ThisDocument.Recipe1. GridVertLineStyle = 2

GridHorizLinesWidth and GridVertLinesWidth

Type: Integer

Description

Sets/gets the widths for the horizontal or vertical grid lines when corresponding grid line style (property GridHorizLineStyle or GridVertLineStyle) is set to 1 - solid line. For all other line styles, these properties must be set to 1.

Example

Sets both horizontal and vertical grid lines to solid line style, 2 pixels wide.

```
ThisDocument.Recipe1. GridHorizLineStyle =1  
ThisDocument.Recipe1. GridVertLineStyle = 1  
ThisDocument.Recipe1. GridHorizLinesWidth =2  
ThisDocument.Recipe1. GridVertLinesWidth = 2
```

GridShade**Type: Short****Description**

Sets/gets the cell effect for the grid column's headers.

"0" = Soft Bump

"1" = Diagonal Shade

"2" = Horizontal Shade

"3" = Vertical Shade

"4" = Horizontal Bump

"5" = Vertical Bump

"6" = Metal

GridTitleBackColor**Type: OLE_COLOR****Description**

Sets/gets the background color for the grid's title bar header.

Example

Sets title bar background color to dark blue.

```
ThisDocument.Recipe1. GridTitleBackColor =GRB(0,0,127)
```

GridTitleFont**Type: StdFont****Description**

Returns a reference to the default font for the grid's title bar header.

Example

To set control's default font to Arial, 14 pt., use the following statements:

With ThisDocument.Recipe1.Font

.Name="Arial"

.Size=14

End With

```
ThisDocument.Recipe1. GridTitleFontColor =GRB(0,0,127)
```

GridTitleHAlign**Type: Short****Description**

Sets/gets the grid's title bar header alignment.

"0" – Left Alignment

"1" – Center Alignment

"2" – Right Alignment

Example

ThisDocument.Recipe1. GridTitleHAlign ="1"

GridTitleTextColor**Type: OLE_COLOR****Description**

Sets/gets the text color for the grid's title bar header.

Example

Sets grid's title bar text color to yellow.

ThisDocument.Recipe1. GridTitleBackColor =GRB(255,255,0)

HeaderColor**Type: OLE_COLOR****Description**

Sets/gets the background color for grid's column headers.

Example

Sets color of grid's column header to dark green.

ThisDocument.Recipe1. GridTitleBackColor =GRB(0,127,0)

HeaderFont**Type: StdFont****Description**

Returns a reference to the default font for the grid's column headers.

HeaderTxtColor**Type: OLE_COLOR****Description**

Sets/gets the text color for grid's column headers.

Example

Sets grid's title bar text color to yellow.

ThisDocument.Recipe1. GridTitleBackColor =RGB(255,255,0)

HideGridPopupMenu

Type: Boolean

Description

Enables/disables Recipe ActiveX runtime popup menu.

InternalGridEdges

Type: Short

Description

Sets the amount of space between the ActiveX borders and the grid borders.

RecipeClientID As String

Type: BSTR

Description

Sets/gets the Recipe Identification string used for binding to the tree explorer tag.

Example

ThisDocument.Recipe1.RecipeClientID="Recipe1"

RecipeTitle As String

Type: BSTR

Description

Sets/gets the title shown in the window title bar of the Database Mining ActiveX control during runtime mode.

Example

ThisDocument.Recipe1.RecipeTitle="Grid Title"

ShowColsHeader and ShowRowsHeader

Type: Boolean

Description

Show/hide the rows/columns headers.

URLPathName

Type: BSTR

Description

Sets/gets the URL path for a configuration file over the Web. Must set URLPathUsed to "True" for this to take effect.

Example

ThisDocument.Recipe1.URLPathName=
"http://www.iconics.com/recipecfg.rcp"

URLPathUsed

Type: Boolean

Description

Enables/disables the use of a URL path.

Example

ThisDocument.Recipe1.URLPathUsed=True

UseLocalSettings;

Type: Boolean

Description

When the URL path is enabled, it allows you to use or not to use the local settings configuration file.

Example

ThisDocument.Recipe1.UseLocalSettings=True

Control Methods

AboutBox

Shows **About** dialog box.

Example

ThisDocument.Recipe1>AboutBox

GetColumnsNumber As Integer

Returns the number of columns currently present in the grid.

Return Value

On success, returns the number of columns; if it fails or the grid doesn't have any column, returns 0.

GetGridCellValue(ColNum As Long, RowNum As Long)

Returns the value of the cell that has coordinates (ColNum,RowNum).

Return Value

On success, returns the value of the cell; if the cell coordinates are not valid or if the grid is not visible on the screen, it will fail returning an empty string.

GetRowsNumber As Integer

Returns the number of rows currently present in the grid.

Return Value

On success, returns the number of rows; if it fails or the grid does not have any columns, returns 0.

GetSelectedColIndex As Integer

Returns the coordinates of the currently selected column.

Return Value

On success, returns the coordinates of the currently selected column; if it fails, returns 0.

GetSelectedRange(VARIANT* TopCrd, VARIANT* LeftCrd, VARIANT* BottomCrd, VARIANT* RightCrd)

Returns the top, left, right and bottom coordinates of the currently selected area in the grid.

Return Value

On success, returns the coordinates of the currently selected area in the grid; if it fails there is no selected area, and returns all the coordinates with a negative value.

GetSelectedRowIndex As Integer

Returns the coordinates of the currently selected row.

Return Value

On success, returns the coordinates of the currently selected row; if it fails, returns 0.

LoadConfigFile() As Boolean

It loads the configuration file specified in newVal.

Example

```
Dim ret As Boolean
```

```
ret = Recipe1.LoadConfigFile("C:\My Configs\cfg.rcp")
```

Function LoadConfigURL() As Boolean

Loads the configuration file specified in the URL string newVal.

Example

```
Dim ret As Boolean
```

```
ret = Recipe1.LoadConfigURL("http://localhost/cfg.rcp")
```

Function ReplaceHost() As Long

It works over data source tags, recipe tags and path name attributes and replaces node name **substring** within URL path only. Returns 0 on success and HRESULT when something fails.

Example

'similar to ReplaceTag

'VBA example, works only in configure mode

'replaces host name in whole display

Dim Status As Long

```
Status = ThisDisplay.ReplaceHost("Host1", "Host2")
```

'if Status <> 0 then there was no replacement performed or an error occurs

```
If Status <> 0 Then
```

```
    MsgBox "No replacements"
```

```
Else
```

```
    MsgBox "Tags replaced"
```

```
End If
```

Function ReplaceHostEx() As Long

It works over data source tags, recipe tags and path name attributes and replaces node name **substring** within URL path only, and supports case-sensitivity, wildcard strings and MatchWholeWord flag. Returns 0 on success and HRESULT when something fails.

Example

'similar to ReplaceTag

'VBA example, works only in configure mode

'replaces host name in whole display, regarding "case" and "whole words" options

Function SaveConfigFile() As Boolean

It saves your current Recipe ActiveX configuration to the file specified in newVal.

Example

```
Dim ret As Boolean
```

```
ret = Recipe1.SaveConfigFile("C:\My Configs\cfg.rcp")
```

Sub UnselectAllRows()

Unselects the currently selected area if present.

Example:

```
ThisDisplay.Recipe1.UnselectAllRows
```

Control Events

Event OnGridLButtonClicked(nCol As Integer, nRow As Long)

Description

This event is called when the user clicks on any grid's cell giving the cell's coordinates (nRow,nCol).

Event OnGridLButtonDbIClk(nCol As Integer, nRow As Long)

Description

This event is called when the user double clicks on any grid's cell giving the cell's coordinates (nRow,nCol).

Event OnGridRButtonClicked(nCol As Integer, nRow As Long)

Description

This event is called when the user right clicks on any grid's cell giving the cell's coordinates (nRow,nCol).

Note: this event will be fired up only if HideGridPopupMenu property is set to false

Event OnGridRButtonDbIClk(nCol As Integer, nRow As Long)

Description

This event is called when the user right double clicks on any grid's cell giving the cell's coordinates (nRow,nCol).

Note: this event will be fired up only if HideGridPopupMenu property is set to false

Event OnGridSelectionChanged(bIsDragging As Integer, bKeyPressed As Integer)

Description

This event is called when the user start to drag the mouse inside the grid's cell area, or a range of cells, rows or columns has been selected, or if the current selected range has been modified.

bIsDragging will be set to true if the user is still dragging and when he will release the mouse this event will be called once more with bIsDragging set to false.

bKeyPressed specifies if the user has selected the cells by pressing SHIFT and an arrow key.

Event OnUpdate()

Description

This event is called every time the grid's values are refreshed