

Area

Description

An `Area` object represents a rectangular set of cells.

C#

```
[DefaultMember("Item")]  
public sealed class Area
```

vb.net

```
<DefaultMember("Item")> _  
Public NotInheritable Class Area
```

Remarks

To create an `Area` object, call [Worksheet.CreateArea](#)

Examples

C#

```
ExcelApplication xla = new ExcelApplication();  
    Workbook wb = xla.Create();  
    Worksheet ws = wb.Worksheets[0];  
    Area a = ws.CreateArea(4, 4, 15, 6);
```

vb.net

```
Dim xla As New ExcelApplication()  
    Dim wb As Workbook = xla.Create()  
    Dim ws As Worksheet = wb(0)  
    Dim a As Area = ws.CreateArea(4, 4, 15, 6)
```

Properties

Name	Description
AllColumnWidths	Sets the width - in points (1/72 of an inch) - of all columns in the <code>Area</code> . When converted to character widths, this must be a value that is between 0 - 255.

AllColumnWidthsInChars	Sets or returns the width of all columns in this Area as a number of '0' characters in the font of the 'Normal' style. This must be a value between 0 - 255.
AllRowsHeight	Sets the height - in points (1/72 of an inch) - of all rows in this Area. This must be a value that is between 0 - 409.
ArrayFormula	Sets or returns an array entered formula. An array entered formula is a formula that can have multiple output values. To set an array entered formula, surround the formula string with a '{' and '}'.
BorderAround	Returns a Border object that represents a border around the area. Use the <code>Border</code> object returned to set the appearance of the border.
ColumnCount	Sets or returns the number of columns in the area.
Dimensions	Sets or returns the dimensions of the area in the format "A1:B7".
FirstColumn	Sets or returns the 0-based number of the first column within the rectangular area of cells.
FirstRow	Sets or returns the 0-based number of the first row within the rectangular area of cells.
PopulatedCells	Returns a rectangular area of cells from the first (top, left) populated cell to the last (bottom, right) populated cell in the Area. The Area returned will include cells that contain no data but have been formatted.
RowCount	Sets or returns the number of rows in the area.
WorksheetIndex	Returns the 0-based index of the worksheet that contains the specified area, or -1 if the worksheet that contained the area doesn't exist anymore.

Indexers

Name	Description
<code>Item(Int32, Int32)</code>	Returns a cell within the specified area of cells.

Methods

Name	Description
<code>ApplyStyle(Style)</code>	Applies a style to the area. When a style is applied - rather than set (see Area.SetStyle(Style)) - only the differences between the new style and style properties previously assigned to the area (through the ExcelWriter API or in Microsoft Excel) will take effect. For example, if the area has a background color and the new style applied does not contain a background color, the area's color will not be affected. However, if the new style includes background color, it will replace the existing background color of the area.
<code>AutoFitHeight()</code>	Sets the height of each of the rows within the area to the height of the highest cell value in that row within the bounds of the Area.
<code>AutoFitWidth()</code>	Sets the width of each of the columns within the area to the width of the widest cell value in that column within the bounds of the Area.
<code>CalculateFormulas()</code>	Calculates any Calculation Engine Supported Formulas in the area. Any formulas in the area that are not supported by the Calculation Engine will be set as 0.
<code>ClearContent()</code>	Clears the content of all cells in the area. Calling this method is equivalent to setting an area's cell values to null.

CreateHyperlink(String)	Creates a new hyperlink in the area pointing to the specified href. This hyperlink is automatically added to the worksheet. All cell values in the area will be converted to hyperlinks.
GetColumnWidth(Int32)	Returns the width - in points (1/72 of an inch) - of the specified column. When converted to character widths, this must be a value that is between 0 - 255.
GetColumnWidthInChars(Int32)	Sets or returns the width of the column as a number of '0' characters in the font of the 'Normal' style. This must be a value between 0 - 255.
GetRowHeight(Int32)	Returns the height - in points (1/72 of an inch) - of the specified row. When converted to character widths, this must be a value that is between 0 - 409.
GroupColumns(Boolean)	Groups or outlines the columns contained within the area.
GroupRows(Boolean)	Groups or outlines the rows contained within the area.
ImportData(Object())	Imports data from a two-dimensional array of objects to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(Object>(), String(), DataImportProperties)	Imports data from a two-dimensional array of objects to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(System.Data.DataTable)	Imports data from an ADO.NET DataTable to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(System.Data.DataTable, DataImportProperties)	Imports data from an ADO.NET DataTable to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(System.Data.DataView)	Imports data from an ADO.NET DataView to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(System.Data.DataView, DataImportProperties)	Imports data from an ADO.NET DataView to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(Object(.))	Imports data from a rectangular array of objects to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(Object(., String(), DataImportProperties)	Imports data from a rectangular array of objects to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(System.Data.IDataReader)	Imports data from an IDataReader to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(System.Data.IDataReader, DataImportProperties)	Imports data from an IDataReader to the specified Area . The new data will <u>overwrite</u> values and formulas in the target worksheet cells, but existing formatting will be preserved.
MergeCells()	Merges all cells in the area into a single cell.
RemoveConditionalFormat()	This method removes any ConditionalFormat objects from the Area.
SetColumnWidth(Int32, Double)	Sets the width - in points (1/72 of an inch) - of the specified column. When converted to character widths, this must be a value that is between 0 - 255.
SetColumnWidthInChars(Int32, Double)	Sets or returns the width of the column as a number of '0' characters in the font of the 'Normal' style. This must be a value between the 0 - 255.

SetConditionalFormat(ConditionalFormat)	This method copies the specified ConditionalFormat object and associates it with this Area. If any ConditionalFormat objects already exist within the Area, they will be removed and replaced by the specified one.
SetDataValidation(DataValidation)	Assigns a data validation rule to all cells in the Area. If the DataValidation object uses a local reference or area in a formula (i.e. "=A5:B7"), then the area will adjust those local references accordingly based on the area's location. The original DataValidation object will not be affected.
SetRowHeight(Int32, Double)	Sets the height - in points (1/72 of an inch) - of the specified row. When converted to character widths, this must be a value that is between 0 - 409.
SetStyle(Style)	Sets or returns a style for the area. When a style is set - rather than applied (see Area.ApplyStyle) - the area to which the style is assigned acquires all of that style's properties, including font properties and number formatting. Setting a style overwrites any pre-existing formatting on the area.
ToString()	Returns a string representation of the Area object.
UngroupColumns()	Ungroups the columns within the area.
UngroupRows()	Ungroups the rows within the area.

Extension Methods

Overload	Description
ImportData(Microsoft.SharePoint.SPList, DataImportProperties)	Imports data from a SharePoint List to the specified Area. The new data will overwrite values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(Microsoft.SharePoint.SPList)	Imports data from a SharePoint List to the specified Area. The new data will overwrite values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(Microsoft.SharePoint.SPView, Microsoft.SharePoint.SPList, DataImportProperties)	Imports data from a SharePoint View to the specified Area. The new data will overwrite values and formulas in the target worksheet cells, but existing formatting will be preserved.
ImportData(Microsoft.SharePoint.SPView, Microsoft.SharePoint.SPList)	Imports data from a SharePoint View to the specified Area. The new data will overwrite values and formulas in the target worksheet cells, but existing formatting will be preserved.