Chartsheet

Description

A `Chartsheet` object represents a worksheet that contains only a chart.

C#

```
public sealed class Chartsheet : Worksheet
```

vb.net

```
Public NotInheritable Class Chartsheet
  Inherits Worksheet
```

Remarks

To create a `Chartsheet`, use Worksheets.createChartsheet() or Chart.MoveChart(). To get an existing `Chartsheet`, get a Worksheet using Workbook.Worksheets[i] and check if it is of type `Chartsheet`.

Anchors behave slightly differently on Chartsheets than on regular worksheets. When creating an anchor, the row and column properties are ignored. The OffsetX and OffsetY properties specify a percentage value, with 0 corresponding to the left or top edge, and 100 corresponding to the bottom or right edge of the Chartsheet. This allows you to add a shape or picture to an arbitrary position on the Chartsheet.

Similarly, the Width and Height properties on Shapes, Pictures, and Groups specify a percentage of the total width or height of the chartsheet. So a Shape with a width of 25 would occupy 25% of the Chartsheet.

You can have as many Shape, Pictures, and Groups as you wish on the chartsheet. However, you may only have one chart on the chartsheet, which is specified with the Chart property.

Some Worksheet properties are not valid on a Chartsheet.

The following table summarizes the behavior of Worksheet properties on Chartsheets. If you try to access or set a property on a Chartsheet that is not valid, an InvalidOperationException will be thrown. Some properties will behave as AutoProperties, but will have no effect in the saved file. If the property is supported but there are differences in behavior from Worksheets, the differences are described in the Notes column.

Property or method is not valid on Chartsheets, and will throw an InvalidOperationException
Property or method is fully supported on Chartsheets
Property behaves as an autoproperty, but will have no effect on the resulting file.

Property	Valid on Chartsheet	Notes
Cells	⚠	Returns a Cells object, but using any methods or properties on the object will throw an InvalidOperationException.
Charts	✗	
Comments	✗	
FirstShownColumn	⚠	
FirstShownRow	⚠	
FreezePanes	✗	
GridlinesColor	⚠	

Hyperlinks	✗	
IsProtected	✓	
IsSelected	✓	
Name	✓	
NamedRanges	⚠	Returns an empty enumeration
PageSetup	✓	Returns a ChartPageSetup object
Pictures	✓	
PopulatedCells	✗	
Position	✓	
ProtectPasswordHash	✓	
ShapeGroups	✓	
Shapes	✓	
ShowFormulas	⚠	
ShowGridlines	⚠	
ShowRowColHeaders	⚠	
ShowZeroValues	⚠	
StandardHeight	✗	
StandardWidth	✗	
StandardWidthInChars	✗	
SummaryColumns	✗	
SummaryRows	✗	
TabColor	✓	
Workbook	✓	
ViewState	⚠	Returns SheetViewState.Normal. Setting it to a different value with throw an InvalidOperationException.
Visibility	✓	
ZoomPercentage	✓	Defaults to 120%

Method	Valid on Chartsheet	Notes
Item(Int32, Int32)	✗	
Item(String)	✗	
CopyPaste	✗	
CreateAnchor	✓	The row and column properties will be ignored. The offset properties should specify a percentage of the entire chartsheet.
CreateArea	✗	

CreateAreaOfColumns	✗	
CreateAreaOfRows	✗	
CreateNamedRange	✗	
CreateRange	✗	
DeleteColumn	✗	
DeleteColumns	✗	
DeleteRow	✗	
DeleteRows	✗	
GetColumnProperties	✗	
GetNamedObject	✗	
GetNamedRange	✗	
GetRowProperties	✗	
GroupColumns	✗	
GroupRows	✗	
ImportData	✗	
InsertColumn	✗	
InsertColumns	✗	
InsertHorizontalPageBreak	✗	
InsertRow	✗	
InsertRows	✗	
InsertVerticalPageBreak	✗	
Protect	✓	
Select	✓	
UngroupColumns	✗	
UngroupRows	✗	
Unprotect	✓	

Examples

C#

```
//--- Create a Chartsheet
          ExcelApplication xla = new ExcelApplication();
          Workbook wb = xla.Create();
          Worksheet ws = wb.Worksheets[0];
          Chartsheet cs = wb.Worksheets.CreateChartsheet
               (ChartType.Pie.Pie3D, "Chart");

          //--- Get the first Chartsheet from a Workbook
          ExcelApplication xla = new ExcelApplication();
          Workbook wb = xla.Open("C:\\MySpreadsheet.xls");
          bool found = false;
          for(int i = 0; i < wb.Worksheets.Count; i++)
          {
               if (found == false)
               {
                    Worksheet ws = wb.Worksheets[i];

                    if(ws is Chartsheet)
                    {
                         Chartsheet cs = (Chartsheet)ws;
                         found = true;
                    }
               }
          }
```

vb.net

```
'--- Create a Chartsheet
          Dim xla As New ExcelApplication()
          Dim wb As Workbook = xla.Create()
          Dim ws As Worksheet = wb.Worksheets(0)
          Dim cs As Chartsheet = wb.Worksheets.CreateChartsheet _
               (ChartType.Pie.Pie3D, "Chart")

          '--- Get the first Chartsheet from a Workbook
          Dim xla As New ExcelApplication()
          Dim wb As Workbook = xla.Open("C:\MySpreadsheet.xls")
          Dim found As Boolean = False
          Dim i As Integer
          For i = 0 To wb.Worksheets.Count - 1
               If found = False Then
                    Dim ws As Worksheet = wb.Worksheets(i)

                    If ws Is Chartsheet Then
                         Dim cs As Chartsheet = CType(ws, Chartsheet)
                         found = True
                    End If
               End If
          Next
```

Properties

Name	Description
Chart	Returns a Chart object representing the chart in the chart sheet.