

Effective Use of Styles

Whenever you change the `Cell.Style` property, that `Cell` gets its own `Style` object. However, if you create a new `Style` object and then call `SetStyle` or `ApplyStyle` on the area (`Area.ApplyStyle(Style)`, `Area.SetStyle(Style)`) or range (`Range.ApplyStyle(Style)`, `Range.SetStyle(Style)`) you want to affect, `ExcelApplication` will not have to create a separate `Style` object for each cell, reducing the amount of memory that is needed.

If you are applying the same style to more than one or two cells, you should say:

```
//Create a global style
GlobalStyle style = wb.CreateStyle();

//Set a cell's style to the new global
style
ws.Cells[0,0].Style = style;

//Set an area's style to the new global
style
ws.CreateArea(1,1,5,5).SetStyle(style);

//Set a range's style to the new global
style
ws.CreateRange("D5:E6").SetStyle(style)
;
```

```
'Create a global style
Dim style As GlobalStyle =
wb.CreateStyle()

'Set a cell's style to the new global
style
ws.Cells(0, 0).Style = style

'Set an area's style to the new global
style
ws.CreateArea(1, 1, 5,
5).SetStyle(style)

'Set a range's style to the new global
style
ws.CreateRange("D5:E6").SetStyle(style)
```

If you want to apply a single style to all the cells in a column, you can get the `ColumnProperties` object for that column and call the `SetStyle` method:

```
//Get the properties for the column you
want to alter
ColumnProperties properties =
ws.GetColumnProperties(0);

//Apply a style to that column
properties.ApplyStyle(style);
```

```
'Get the properties for the column you  
want to alter  
Dim properties As ColumnProperties =  
ws.GetColumnProperties(0)  
  
'Apply a style to that column  
properties.ApplyStyle(style)
```

Using one of these two techniques to apply styles will use less memory than creating a new style for each cell or modifying the Cell.Style property directly.

For more information about using Styles in ExcelWriter, please refer to our tutorial [Styles in ExcelWriter](#).