Copying Data From a Worksheet

ExcelApplication's CopyPaste method allows you to copy data from an external worksheet. It can also be used to copy data from a worksheet within the same workbook.

CopyPaste is a method of the Worksheet object and should be called from the destination worksheet. CopyPaste uses a CopyPasteProperties obj ect to store information on precisely how data should be copied. Options include: values, formulas, formatting, merged cells, comments, height and width.

Step 1: Setting up your code to copy from an external worksheet:

In the code below, a source worksheet is opened and a destination worksheet is created. The CopyPaste method takes an Area object to represent the area to be copied, so this is created first.

```
 private ExcelApplication xlw;
private Workbook wb, wb2;
private Worksheet wsSrc, wsDest;

private void GenerateWorkbook()
{
 //--- Store the path to the source workbook
 string strPathSrc = @"C:\\WorkDirectory\\MySourceWorkbook.xls";

 //--- Create an Excelwriter Application object
 xlw = new ExcelApplication();

 //--- Use the Application object to create a workbook
 //--- that will contain the destination worksheet
 wb = xlw.Create();

 //--- Specify the destination worksheet
 wsDest = wb.Worksheets[0];
 wsDest.Name = "DestinationSheet";

 //--- Open the workbook that contains the (source) worksheet to be copied
 wb2 = xlw.Open(strPathSrc);

 //--- Specify the worksheet that will be the source of the data
 wsSrc = wb2.Worksheets[0];

 //--- You must first create an area that represents the information
 //--- in the source worksheet that you want to copy.
 Area areaSrc = wsSrc.CreateArea("A1:C3");
```

Step 2: Creating a CopyPasteProperties object

Now that the data to be copied is mapped to an Area object, a CopyProperties object should be created from the destination workbook. The CopyProperties object is used to specify precisely what within the area should be copied. In the code below, only the values will be copied.

```
CopyPasteProperties properties =
wb.CreateCopyPasteProperties(CopyPasteProperties.CopyPasteType.ValuesOnly);
```

The ValuesOnly type is a convenience type. The same result could be achieved by using the None type and adding a second line of code where the CopyValues property was set to true.

```
CopyPasteProperties properties =
wb.CreateCopyPasteProperties(CopyPasteProperties.CopyPasteType.None);
properties.CopyValues = true;
```

Using the None type allows you to set Copy properties individually. The available properties include:

```
[CopyPasteProperties].CopyColumnWidth = [true|false];

 [CopyPasteProperties].CopyComments = [true|false];

 [CopyPasteProperties].CopyFormatting = [true|false];

 [CopyPasteProperties].CopyFormulas = [true|false];

 [CopyPasteProperties].CopyMergedCells = [true|false];

 [CopyPasteProperties].CopyRowHeight = [true|false];

 [CopyPasteProperties].CopyValues = [true|false];
```

Besides None, other CopyPasteType values are available, saving you unnecessary lines of code:

CopyPasteType	Sets the following CopyPasteProperty values to true
None	No CopyPasteProperty values set.
All	CopyComments, CopyFormatting, CopyFormulas, CopyMergedCells and CopyValues **Note**: CopyColumnWidth and CopyRowHeight are set to false to avoid unexpected results.
AllPlusRowHeightAndColumnWidth	CopyComments, CopyFormatting, CopyFormulas, CopyMergedCells, CopyValues, CopyColumnWidth and CopyRowHeight
FormulasAndFormatting	CopyFormulas and CopyFormatting
ValuesAndFormatting	CopyValues and CopyFormatting
ValuesFormulasAndFormatting	CopyValues, CopyFormulas and CopyFormatting
ValuesOnly	CopyValues

Optionally, if you intend to use the CopyPasteType of All, you can save even more lines of code by using CopyPaste method without the properties object parameter. Omission of the properties object parameter will result in the assumption of the value of "All".

Step 3: Calling the CopyPaste method from the Destination Worksheet

To complete the code sample that is being built, only the CopyPaste method now needs to be called. You will pass it the address of the cell that represents the upper left corner of the location to paste and the Area and CopyPasteProperties objects that were created above.

```
//---Call the CopyPaste method from the destination worksheet.
     wsDest.CopyPaste("C3", areaSrc, properties);

     //---Open the spreadsheet in the browser to check the results
     xlw.Save(wb, Page.Response, "ExcelApp.xls", false);
```