

Workbook

Description

A [Workbook](#) object represents an Excel workbook. To return a [Workbook](#) object: Call [ExcelApplication.Create\(\)](#) to create a new Excel workbook. Call [ExcelApplication.Open\(\)](#) to open an existing Excel workbook.

C#

```
[DefaultMember("Item")]
public sealed class Workbook : System.Collections.Generic.IEnumerable<Worksheet>
```

vb.net

```
<DefaultMember("Item")> _
Public NotInheritable Class Workbook
    Implements System.Collections.Generic.IEnumerable(Of Worksheet)
```

Examples

C#

```
//--- Create a new Workbook
    ExcelApplication xla = new ExcelApplication();
    Workbook wb = xla.Create();

    //--- Open an existing Workbook
    ExcelApplication xla = new ExcelApplication();
    Workbook wb = xla.Open(@"C:\Reports\Report.xls");
```

vb.net

```
'--- Create a new Workbook
    Dim xla As New ExcelApplication()
    Dim wb As Workbook = xla.Create()

    '--- Open an existing Workbook
    Dim xla As New ExcelApplication()
    Dim wb As Workbook = xla.Open("C:\Reports\Report.xls")
```

Properties

Name	Description
------	-------------

DisplayTabs	Shows or hides the worksheet tabs at the bottom of the workbook window.
DocumentProperties	Returns a DocumentProperties object that represents properties of the specified workbook, such as name, author, and summary. In Microsoft Excel, these properties are displayed in the file's Properties sheet. To view the Properties sheet in Excel, open Excel's File menu, and select Properties.
EncryptPassword	Sets or returns the password used to encrypt the spreadsheet. Standard "Office 97/2000 Compatible" encryption is used to encrypt the spreadsheet. This is the default encryption used by Excel 97 and later versions.
ExcludeMacros	Sets or returns whether ExcelWriter should remove macros from the generated workbook or leave them in when saved.
FirstDisplayedTab	Sets or returns the first worksheet tab to display in the tab bar at the bottom of the workbook window.
Height	Sets or returns the height of the workbook's window
HorizontalPosition	Sets or returns the horizontal (x) position of the workbook window's upper left corner when displayed in Excel.
IsProtected	Returns whether or not the worksheet is protected.
NamedRanges	Returns the named ranges of the Workbook as an array of NamedObject objects as an IEnumerableable collection. Though the named ranges are not returned in any particular order, they are iterable.
NumberFormat	Returns the NumberFormat used for formatting of numbers and dates.
Palette	Returns a Palette object that represents the specified workbook's color palette.
PasswordToModify	This sets the workbook so that it can be opened in read-only mode with the option to open in read-write mode, if the user supplies a matching password when opening the spreadsheet.
ProtectPasswordHash	Sets or returns the password hash that is used to protect the workbook.
ReadOnly	Determines whether the workbook can only be opened in read-only mode.
ReadOnlyRecommended	Sets or returns whether Excel will recommend - when the workbook is opened - that it be viewed in read-only mode.
ShowHorizontalScrollBar	When true, the horizontal scroll bar will be shown in the generated workbook. When false, the horizontal scroll bar will be hidden.
ShowVerticalScrollBar	Shows or hides the vertical scroll bar in the generated workbook.
TabRatio	Sets or returns the ratio between the workbook's tab bar and the horizontal scroll bar.
UseRCFormulaNotation	Indicates or sets the type of references used in this workbook.
VerticalPosition	Sets or returns the vertical (y) position of the workbook window's upper left corner when displayed in Excel.
Width	Sets or returns the width of the workbook's window.
Worksheets	Returns a Worksheets collection which provides access to all worksheets in the specified workbook, including charts displayed as sheets.

Indexers

Name	Description
------	-------------

<code>Item(Int32)</code>	Returns the worksheet specified by a 0-based index. This property is an indexer for the Workbook class.
<code>Item(String)</code>	Returns the worksheet with the specified name. This property is an indexer for the Workbook class.

Methods

Name	Description
<code>CreateConditionalFormat()</code>	Creates a new ConditionalFormat object that can be set on Cells , Areas and Ranges .
<code>CreateCopyPasteProperties(CopyPasteType)</code>	Creates a CopyPasteProperties object with the specified data type properties set.
<code>CreateDataImportProperties()</code>	Creates a DataImportProperties object that contains a set of properties that can be used when importing data to a Worksheet or Area .
<code>CreateDataValidation(ValidationType, ComparisonType, Object)</code>	Creates a DataValidation object which can be set on a Range , Area , or Cell .
<code>CreateDataValidation(ValidationType, ComparisonType, Object, Object)</code>	Creates a DataValidation object which can be set on a Range , Area , or Cell .
<code>CreateDataValidation(ValidationType, ComparisonType, String)</code>	Creates a DataValidation object which can be set on a Range , Area , or Cell .
<code>CreateDataValidation(ValidationType, ComparisonType, String, String)</code>	Creates a DataValidation object which can be set on a Range , Area , or Cell .
<code>CreateDataValidation(ValidationType, Object)</code>	Creates a DataValidation object which can be set on a Range , Area , or Cell .
<code>CreateDataValidation(ValidationType, String)</code>	Creates a DataValidation object which can be set on a Range , Area , or Cell .
<code>CreateNamedRange(Area(), String)</code>	Creates a Range of cells from an array of Areas . A Range is a collection of Area s; an Area area is a rectangular collection of cells.
<code>CreateNamedRange(String, String)</code>	Creates a named Range of cells from a specified formula. A Range is a collection of Area s; an Area area is a rectangular collection of cells.
<code>CreateNamedStyle(String)</code>	Creates a named global style, which can be bound to cells, areas, ranges, rows, and columns. Any later changes to this style will affect all cells to which the style was assigned.
<code>CreateRange(String)</code>	Creates a Range of cells from a specified formula. A Range is a collection of Area s; an Area area is a rectangular collection of cells.
<code>CreateRange(Area())</code>	Creates a Range of cells from an array of Areas . A Range is a collection of Area s; an Area area is a rectangular collection of cells.
<code>CreateStyle()</code>	Creates a global style, which can be bound to cells, areas, ranges, rows, and columns. Any later changes to this style will affect all cells to which the style was bound.
<code>GetEnumerator()</code>	Returns an IEnumerator for the Worksheets in a Workbook
<code>GetNamedObject(String)</code>	Returns the NamedObject object that represents the name of a specified array, number, picture, or range. If the named object does not exist, the method returns null.
<code>GetNamedRange(String)</code>	Returns the specified named Range . If the named range does not exist, the method returns null.
<code>GetNamedStyle(String)</code>	Returns a NamedStyle if it exists. If the style does not exist, null is returned.

<code>GetTextHeight(String, Font, Double)</code>	Returns the minimum height required to fit the given text string using the given font.
<code>GetTextWidth(String, Font, Double)</code>	Returns the minimum width required to fit the given text string using the given font.
<code>Protect(String)</code>	Protects the workbook structure using the supplied password. End users will not be able to add or remove worksheets unless they enter the password.
<code>RemoveFormulas()</code>	The RemoveFormulas method removes Excel formulas from the entire workbook, while leaving their values in cells. This method does NOT remove Conditional Formatting or Data Validation.
<code>Unprotect()</code>	Removes protection from the workbook.