

# ExcelTemplate API

## *DataBindingProperties*

The `DataBindingProperties` object contains the properties information necessary to bind data to an `ExcelWriter` template.

- `DataBindingProperties.MaxRows` — Sets or returns the maximum number of rows to be bound to the template file when the `ExcelTemplate.BindData` or `ExcelTemplate.BindColumnData` methods are called.
- `DataBindingProperties.Transpose` — Sets or returns whether the data being bound to a template file should be transposed before being bound to the template using the `ExcelTemplate.BindData` method.
- `DataBindingProperties.WorksheetName` — Sets or returns the name of the worksheet to which the data will be bound during an `ExcelTemplate.BindData`, `ExcelTemplate.BindCellData`, `ExcelTemplate.BindColumnData`, or `ExcelTemplate.BindRowData` method call.

## *ExcelTemplate*

The `ExcelTemplate` object represents an `ExcelWriter` template spreadsheet.

- `ExcelTemplate.ALL_ROWS` — Used as `DataBindingProperties.MaxRows`, to indicate that as much of the data source should be imported as possible.
- `ExcelTemplate.BindCellData(Object, String, DataBindingProperties)` — Sets an object as a data source for a single cell in a template.
- `ExcelTemplate.BindColumnData`
- `ExcelTemplate.BindData`
- `ExcelTemplate.BindRowData`
- `ExcelTemplate.ContentType` — Sets or returns the generated spreadsheet's MIME content type.
- `ExcelTemplate.CreateDataBindingProperties()` — Creates a `DataBindingProperties` object for assigning the property values associated with one or more data binding calls.
- `ExcelTemplate.DecryptPassword` — To decrypt an encrypted template, set `DecryptPassword` to the password set using `EncryptPassword` or in Microsoft Excel (Tools -> Options... -> Security).
- `ExcelTemplate.EncryptPassword` — If you set `EncryptPassword`, `ExcelWriter` will encrypt the generated spreadsheet - when `Process` is called - using RC4 encryption.
- `ExcelTemplate.ExcelTemplate()` — Creates a new `ExcelTemplate` object.
- `ExcelTemplate.ExcludeMacros` — Sets whether `ExcelWriter` should remove template macros from the generated spreadsheet or include them.
- `ExcelTemplate.LicenseKey` — Sets or returns the 23-character license key that enables `ExcelWriter` for the current instance.
- `ExcelTemplate.Open`
- `ExcelTemplate.PreserveStrings` — If you set `PreserveStrings` to true, `ExcelWriter` will preserve numeric strings as strings.
- `ExcelTemplate.Process()` — The `Process` method enters data source values in a template's data markers, and creates the output file (the new spreadsheet) in memory. The `Save` method can then save the output file to disk, stream it to the browser, or both.
- `ExcelTemplate.RemoveExtraDataMarkers` — Sets whether `ExcelWriter` should remove data markers in the template that do not bind to data sources in code.
- `ExcelTemplate.Save`
- `ExcelTemplate.SaveToStream()` — Creates a new `System.IO.Stream` object and writes the processed template to it.
- `ExcelTemplate.StretchCellReferencesInCharts` — Sets or returns whether to stretch cell references in charts.
- `ExcelTemplate.Version` — Returns the exact version of `ExcelWriter`.