

# SharePointIntegration.Save(ExcelTemplate, Microsoft.SharePoint.SPDocumentLibrary, String, Boolean)

Introduced in build 7.1.0.1855

## Description

Generates an Excel binary or OOXML file and saves it to a SharePoint Document Library.

### C#

```
public static void Save(this ExcelTemplate template,
Microsoft.SharePoint.SPDocumentLibrary docLib, string fileName, bool overwrite)
```

### vb.net

```
Public Shared Sub Save(ByVal template As ExcelTemplate, ByVal docLib As
Microsoft.SharePoint.SPDocumentLibrary, ByVal fileName As String, ByVal overwrite As
Boolean)
```

## Parameters

### ***template***

The current ExcelTemplate object which is being saved.

### ***docLib***

SharePoint Document Library that contains the Excel spreadsheet being opened

### ***fileName***

Name of the output file. ExcelWriter will save the file to the document library using this name. If a file with the same name exists, it will be overwritten by the new Excel file if the overwrite parameter is set to True.

### ***overwrite***

Set to True to overwrite an Excel file if one already exists by that name in the Document Library

## Exceptions

### ***ArgumentNullException***

Save will throw this exception if null (C#) or Nothing (VB.NET) is passed to the method.

### ***ArgumentException***

## Remarks

ExcelWriter allows you to save in both the Excel 97-03 BIFF8 format (.xls) or the new Office Open XML (.xlsx) format. The template file must be of

the expected output format. Hence, if you wish to output .xls files, you must start with a .xls template, and if you wish to output .xlsx files, you must start with a .xlsx template. ExcelWriter does not support the creation or editing of .xlsx files with the [ExcelApplication](#) object.

You can call `Save` more than once for a single instance of [ExcelTemplate](#). This allows you to save more than one copy of a generated file, and/or both save the file on the server and stream it to the client.



This is an extension method for the `ExcelTemplate` object to be used for saving spreadsheets to SharePoint Document Libraries. To use this method, you must add a reference to `SoftArtisans.OfficeWriter.ExcelWriter.SharePointIntegration.dll`

## Examples

### C#

```
//--- Retrieve current AllowUnsafeUpdates value
bool currentAllowUnsafeUpdates = documentLibrary.ParentWeb.AllowUnsafeUpdates;

//--- You must allow unsafe updates in order for the file to be saved in the
//--- Document Library from a web application
documentLibrary.ParentWeb.AllowUnsafeUpdates = true;
xlt.Save(documentLibrary, "populated.xlsx", true);

//--- Update the document library
documentLibrary.Update();

//--- Return the AllowUnsafeUpdates value to its original state
documentLibrary.ParentWeb.AllowUnsafeUpdates = currentAllowUnsafeUpdates;
```

### vb.net

```
'--- Retrieve current AllowUnsafeUpdates value
Dim currentAllowUnsafeUpdates As Boolean =
documentLibrary.ParentWeb.AllowUnsafeUpdates

'--- You must allow unsafe updates in order for the file to be saved in the
'--- Document Library from a web application
documentLibrary.ParentWeb.AllowUnsafeUpdates = True
xlt.Save(documentLibrary, "populated.xlsx", True)

'--- Update the document library
documentLibrary.Update()

'--- Return the AllowUnsafeUpdates value to its original state
documentLibrary.ParentWeb.AllowUnsafeUpdates = currentAllowUnsafeUpdates
```