# Developer guidelines

The [OpenXML specification](https://www.ecma-international.org/publications/standards/Ecma-376.htm) is a large and complicated beast. In order for ClosedXML, the wrapper around OpenXML, to support all the features, we rely on community contributions. Before opening an issue to request a new feature, we'd like to urge you to try to implement it yourself and log a pull request.

Here are some tips.

\* Before starting a large pull request, log an issue and outline the problem and a broad outline of your solution. The maintainers will discuss the issue with you and possibly propose some alternative approaches to align with the ClosedXML development conventions.

\* Please submit pull requests that are based on the `develop` branch.

\* Where possible, pull requests should include unit tests that cover as many uses cases as possible.

\* We recommend Visual Studio 2019 or higher as the development environment. If you do use Visual Studio, please install these extensions:

\* [CodeMaid](https://marketplace.visualstudio.com/items?itemName=SteveCadwallader.CodeMaid), for ensuring consistent code syntax, whitespace convention, etc.

\* If you use a version of Visual Studio lower than 2017, you should install [editorconfig](<https://marketplace.visualstudio.com/items?itemName=EditorConfigTeam.EditorConfig>) support. Read more about [EditorConfig](http://www.editorconfig.org).

\* We use 4 spaces for code indentation. This is the default in Visual Studio. Don't leave any trailing white space at the end of lines or files.

## Working with Excel file internals

Excel files (`.xlsx` and `.xlsm`) are zip packages. You can easily verify this by renaming the extension any Excel file to `.zip` and opening the file in your favourite `.zip` file editor.

Internally, the file contains files (also known as parts) that represent different entities in the Excel framework, for example `workbook.xml` and `table1.xml`. The [OpenXML specification](https://www.ecma-international.org/publications/standards/Ecma-376.htm) documents all these parts and their contents.

Making changes to the ClosedXML code may change the input or output of the package parts. For example if you add support for a currently unsupported element, you will have to ensure that you read the appropriate package part into the ClosedXML model and also support writing of the package parts to the file.

### Comparing the internals of Excel files

A ClosedXML developer will often want to compare the internals of 2 similar Excel files. For example if you want to compare the output of a specific package part before and after your code changes. The long, difficult way would be to extract the package parts of the 2 files and manually compare the relevant parts. To ease this, we recommend this tooling stack:

- [Total Commander](https://www.ghisler.com/download.htm)

- [WinMerge](http://winmerge.org/downloads) version `2.14.0`, because subsequent versions for [some reason](https://bitbucket.org/winmerge/winmerge/issues/152/displayxmlfiles-plugin-not-included-with) excludes the required `DisplayXMLFiles.dll` plugin.

- Set Total Commander [to use WinMerge](https://superuser.com/questions/238039/can-i-replace-internal-diff-in-total-commander-with-a-custom-tool) as its compare tool.

- In WinMerge, enable `Plugins > Automatic Prediffer`

Now, to compare 2 similar, but not exact Excel files:

- In Total Commander, navigate to the 1st file in the left-hand pane and the 2nd file in the right-hand pane.

- Press `Ctrl+PageDown` to "enter" the package. You should see, among others, a `[Content\_Types].xml` file in both panes.

- You can now compare all package parts by selecting `Commands > Synchronise Dirs...`. Press `Compare`. This will do a full, recursive comparison. You can filter out parts that are identical.

- You can select an item that differs and press `Ctrl+F3` to open the two parts in WinMerge and see the exact comparison of the part's contents. The XML files should automatically reformat/reindent to ease the comparison instead of showing the entire XML contents on a single line. This is the reason for requiring the `DisplayXMLFiles.dll` plugin.

- In Total Commander, you can also navigate to specific files in the left-hand and right-hand panes and select `File > Compare by Content...`. This will open WinMerge directly.

- Note that since WinMerge reformats the XML, it does so in a temporary file. If you make changes to the contents of any of the 2 panes in WinMerge and save the file, it will not be saved back into the Excel file.

## Code conventions

ClosedXML has a fairly large codebase and we therefore want to keep code revisions as clean and tidy as possible. It is therefore important not to introduce unnecessary whitespace changes in your commits.

To ensure you follow the coding conventions, please do the following steps before you commit your code:

- In Visual Studio, run `CodeMaid > Cleanup Active Document` or `Ctrl+M, Space` on each file that you have altered. This will ensure the correct whitespace consistency.

- Some files, not all, have a header in the first line: `// Keep this file CodeMaid organised and cleaned`. For these files, also run `CodeMaid > Reorganize Active Document` or `Ctrl+M, Z`. This will reorder properties and methods alphabetically into a predetermined order. For example, public properties and methods will be organized before private properties and methods. Not all files require this yet. Please take note of the headers.