Usage:

1. Load config(example:Project.config) 🡪 AddModuleWords(example already added) 🡪 Load raw/formatted excel 🡪 DeleteEmptyRows 🡪 NormalizeNames 🡪 Update/Overwrite config.
2. Others are merely some basic excel input/output, data views and autocomplete words input functions.
3. There are a lot of text processing going on here, so if you met an unhandled exception, don’t panic, it’s only some bad format data, you may need to reopen the application after formatted the data. Most exceptions are null or out of index.
4. Software structure: the main UI—datagrid is extremely hard to manipulate in code, especially for the “new place holder”, since it is a row that has not been generate yet. So in order to make a smooth interoperate with excel and xml, I have to go for some third-party tools, these tools are real amazing, especially NPOI, AutoCompleteBox. Besides that, I also had to look up a lot from internet, especially StackOverflow and MSDN. If you check up my code, the datagrid helper functions are directly from StackOverflow, and the code for datagrid event handler are true expert-level, I draw those with a lot of digging and use them under my intention. Pay attention for dataGrid\_RowEditEnding and dataGrid\_PreparingCellForEdit, as they clearly show you how WPF UI framework are framed and how to hack the mechanism.
5. Of course, things are only go easier with .net original tech, such as Linq to XML, command binding, TextInfo, FileWatcher and so on.
6. Look what beyond: We can implement a true MVVM application, and add some other convenient commands such as redo undo and bock selection, and we can even host a WCF service to exchange data.