Introduction
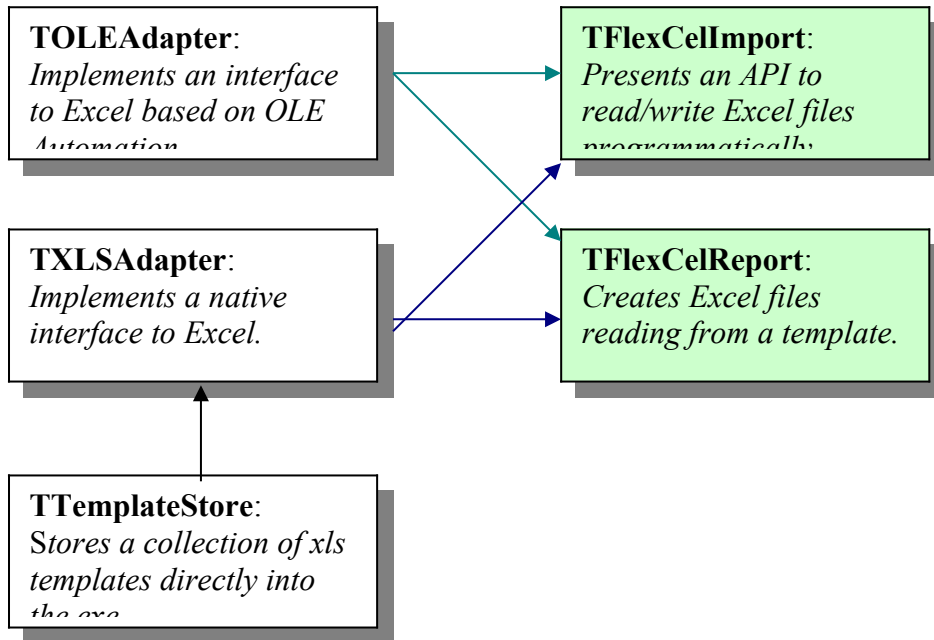
This is a suite of 5 Delphi components targeted to reading/writing Excel files. They are organized in this way:

TOLEAdapter: *Implements an interface to Excel based on OLE Automation.*	⟶	**TFlexCelImport**: *Presents an API to read/write Excel files programmatically.*
TXLSAdapter: *Implements a native interface to Excel.*	⟶	**TFlexCelReport**: *Creates Excel files reading from a template.*
TTemplateStore: *Stores a collection of xls templates directly into the exe.*		

The 2 main components are TFlexCelImport and TFlexCelReport. The others should be connected to them.

Here there is a little 'marketing' description of them. You can find a more 'technical' one in their respective documents.

OLEAdapter, XLSAdapter
These two components present a uniform API for TFlexCelImport and TFlexCelReport, and are not to be used directly. Just drop one of them into your datamodule, and connect the FlexCelImports/Reports to it.

TemplateStore
This one holds a collection of templates into the exe. If you connect it to an XlsAdapter, it will read the files from the store instead of the disk.

FlexCelReport
 This is the main component of the suite. It reads a template Xls file, fills in the blanks with values from memory or the database, and writes them back. Supports images, comments, multiple sheets, and many other things.

FlexCelImport
 The FlexCelImport component allows to read/write values from any ExcelFile, again, native or via OLE.

Choosing between OLE mode and Native Mode

The components included give you two ways to work: native and via OLE automation. I' have tried to make them as similar as possible, but some differences remain, and you will have to decide what to use.

Advantages of Native Mode

- **Speed**: It is faster than OLE, but there isn't a too big difference either. If speed is vital to you, you should choose Native mode. Also, is much better for reading files
- **No need for Excel installed:** This might be an advantage too, and because of this, native will be the only way supported if I ever make a Kylix version.
- **Images**: You can use images in the two modes, but only in native you can populate them from the database.
- **Independence of Excel Version:** As the Excel versions change so much, it might happen that with a particular one (maybe XP Chinese SP1...) Ole automation fails. As here you don't use the Chinese Excel, this is no concern.
- **File is created Closed:** This is an advantage for servers, you never have to launch Excel.
- **Templates can be stored in the Exe:** If you use a TemplateStore, you don't need to distribute any Excel file for your reports to work.

Advantages of OLE Mode
- **Documented API:** Here we are using a documented and simple API (OLE), so the probabilities of a code bug are much less. (Just take a look at how many units there are in Native mode, against just one for OLE)
- **File is created Open:** This might be an advantage for clients, because they can see the report and save it only if they want to. In native mode, you have to prompt the user for the filename, create the file and then open it.
- **Access to multiple file formats:** Native mode only deals with Biff8 (this is Excel 97-2000-XP). With OLE, you can read and write HTML files, CSV, and many other popular file formats.
- **Formulas and Pivot tables can de calculated before saving:** If you are using formulas or pivot tables, Excel can calculate them when it saves the file. In native mode, Excel will calculate when you open the file.

License

This is free software, subject to the Mozilla Public License. If you find it useful, and if you can afford it (come on... it's just $30) you can register it at

http://shareit1.element5.com/product.html?productid=161531

You will make my day a little happier :-) .If you can't afford it, well, don't worry, you can use it the same. (I would like an email anyway, telling me that you use it and how, but that's up to you). The real reason behind me asking you to contact, is that I would like to know how many people if someone is using this, and how many people thinks it's

so good they can pay $30 for it. Supporting a product is a complex work, and I want to be sure it's being useful to someone.

Now, for the cut and paste…

The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF
ANY KIND, either express or implied. See the License for the specific language governing rights and
limitations under the License.

Thanks

There are so many people I would like to thank that I'm afraid I will forget many. But anyway and in no particular order:

- People at OpenOffice (www.OpenOffice.org) for their great work documenting the Excel interface, and for the source code from where I learned so much.
- People at POI (http://jakarta.apache.org/poi/) for their source code to create Excel files in Java.
- People of GlyFx (http://www.glyfx.com/) for some of the icons (the ones that look nice…) on the demo and biffedit apps.
- The Torry team (www.torry.net) and Robert Czerwinski (http://delphi.icm.edu.pl/) for allowing me to share this with you.
- You, and all the people who wrote and supported this effort.

Contact

For any comments, suggestions, etc, please write to agallero@netscape.net, I'll be glad to hear from you. I am creating a (very) basic web page at
http://www.geocities.com/adriangallero/ . You can look there for news.