

A Guide to Selecting Between a Template Designer and a File Library



White Paper

Introduction

This guide will walk you through the differences between using a template designer and a file library for your document generation (docgen) needs. Before you start looking at products, you first need to decide between the two. Your choice determines which products you are looking at.

And it's rare, exceedingly rare, where this is a difficult decision. This should be an easy choice based on your needs. It's also very important to get right, because it determines who can design templates and how they have to go about it.

Definitions

A docgen system built around a template designer will use Microsoft Office, or a similar product, for the template design. You then are placing tags (often called fields) in the template to control the output. This provides an easy intuitive designer, but it also limits you to what the editor (for example Word) is capable of.

A docgen system built around a file library is a library of programming calls where a programmer can then design a template based on building the output file from their code. In this case you write code, run it, and see the generated output. In this approach, you can do anything the output format is capable of.

Template Designers



The Good

First, a non-programmer can design the template – themselves. Some designers may require more training than others but when the designer is fundamentally Word, most any business user can design templates.

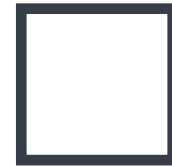
Second, you see immediately what you are going to get. It may not be an exact match as control tags take up some space and when iterative data populates it grows tables, etc. But you have a really good idea of what you are going to get.



The Bad

First, you are limited by the capabilities of the editor. For example, if you want text in the footer based on the content of the page you are on – Word can't do that. PDF output can, but because Word cannot, you can't get this.

Second, your document is laid out based on how Word (or whatever file format you use) lays out the content. If you want something at a very specific position on the page, that can be challenging.



The Ugly

The output is not going to match what you see in the editor. Office does not layout a page as it specifies in the OpenXML (DOCX, PPTX, XLSX) spec. It's close, but the differences can be frustrating at times.

File System



The Good

First, you get exactly what you want. If you want lines of text exactly 12.23 points apart, you will get text exactly 12.23 points apart (not for OpenXML, because again, Office doesn't implement it exactly).

Second, your functionality is limited only by the output format. If you want the last row of a table on a page to say "continued next page..." in PDF output, you can do that (again, you can't for DOCX output, even with the file system approach, because Word does not have this concept).

Third, you can design your output based on metadata within your data. If your data is setting colors, fonts, table column widths, and more, then you can use that information to create the document.



The Bad

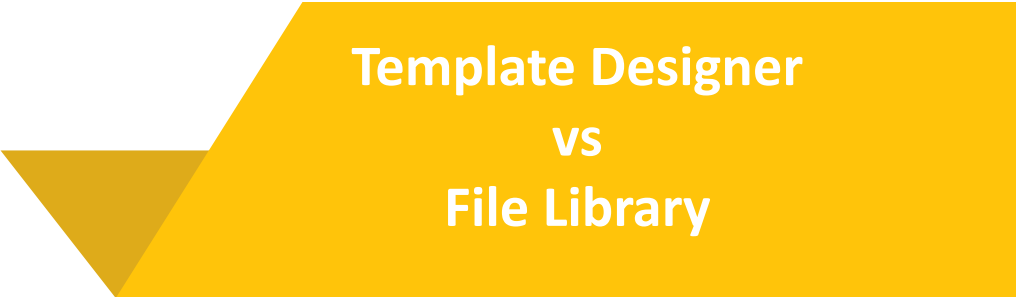
First, to see the generated output, you need to run the program. This takes time because you have to compile and run. Sometimes "run" means start up your web app click your way in and generate a document. Not only does this take time, but it breaks the flow of designing a template as you test.

Second, only programmers can design templates. Yes, a business user can tell the programmer what they want, but the actual design is limited to programmers.



The Ugly

It can be really hard to go back a month later and understand what was done, and why. So revising a template after some time has passed and be slow and painful. (XSLT-FO is referred to by many, including this author, as a write-once language.)



**Template Designer
vs
File Library**

Selecting Your Approach

The fact that you've read this far means you do have a development team handling the template design and you're good with that. So your question comes down to the following:

1. Is there functionality you must have that you can only get with a file system approach?
2. Do you have the output layout & formatting in your data and that data drives the design of the final document?
3. Are you good with saying business users will never be able to design templates?

If your answer to the above three is no, no, & no, then you want a template designer based approach. If your answer is yes to either of the first two, then you want a file library approach.



Template Designer Products

Docmosis

Generate documents and reports based on templates. Output in PDF/Doc/ODT from Java, PHP, C#, Ruby and more.

Ecrion

Ecrion makes customer communications management software for companies who want to establish genuine connections across multiple engagement channels.

Formstack

Use Formstack document generation software to merge data into custom-built documents. Document automation tools save hours of time and money.

HotDocs

No matter your industry or company size, HotDocs from AbacusNext has a solution to help speed up your document creation workflow.

Windward Studios

The Global Leader in Document Generation Solutions. Revolutionize your docgen. Windward provides seamless integration in your CRM or custom apps.

XPertDoc

Our specialty is document generation and document automation. Our mission is to enable organizations to digitally transform their document processes.

File Library Products

Aspose

The Aspose APIs can create, edit, export, and convert over 100 file formats.

iText

The itext library is the most powerful library for creating PDF output. They are PDF only.

OpenXML SDK

The OpenXML SDK is an open source library that can read & write DOCX, PPTX, & XLSX files, providing a LINQ based API. It is .Net (C#, VB.NET, etc.) only.

XSLT-FO

This is a library that uses XSLT and can create output files of pretty much any format. It runs on any system that runs Java (i.e. anywhere). (Understanding existing XSLT scripts can be challenging. If you use this – write lots of comments in your scripts.)

Conclusion

Was this helpful, too much, too little, or something else? Please give us your feedback.

This paper was written by David Thielen, the founder and CEO at Windward.