ASPOSE
Your File Format Experts

CASE STUDIES

Aspose.Words for Java

Competitive Upgrade Case Study

Seniram, INC.
Computer & Web Services

Seniram, Inc.

Competitive analysis of custom developed Microsoft Word mail-merge using Adobe ColdFusion against Aspose.Words for Java

Adam Webster, Senior Consultant, November 7, 2020

About Seniram, Inc.

Seniram is a team with a diverse set of skills who rally behind the goal of helping your business be successful. Our relationship with clients is personal. Our mission is to make your life at work more enjoyable. One way we accomplish this is to free you from the mundane tasks, which are necessary for business success, but wear away the hours and make a person dread the day before it even begins. Work is necessary, but it does not have to be torture.

Problem

Over the years, we have automated many tasks for customers as our mission statement describes. Custom building applications that interface with other off-the-shelf applications are a necessity. We built one such tool for accomplishing mail-merge tasks with Microsoft Word. Our in-house built solution was advanced for its time, however; it was limited to work only on Microsoft Windows-based servers. While searching for newer, better, faster, more efficient solutions to offer our clients, we began migrating services away from Windows-based services to Linux. While we did not completely abandon Microsoft Windows hosting, those we could migrate to Linux would reduce our customers' recurring costs as well as our expenses. We set up a test case for one of our clients using ColdFusion/Lucee on Linux and created a ColdFusion Component which integrated with Aspose.Words for Java. We found integrating with Aspose.Words for Java extremely easy and the processing time to be much faster than our older solution. We wrote the ColdFusion interface in only a couple of hours. Our test case application will be used by a small business with under 10 users to start.

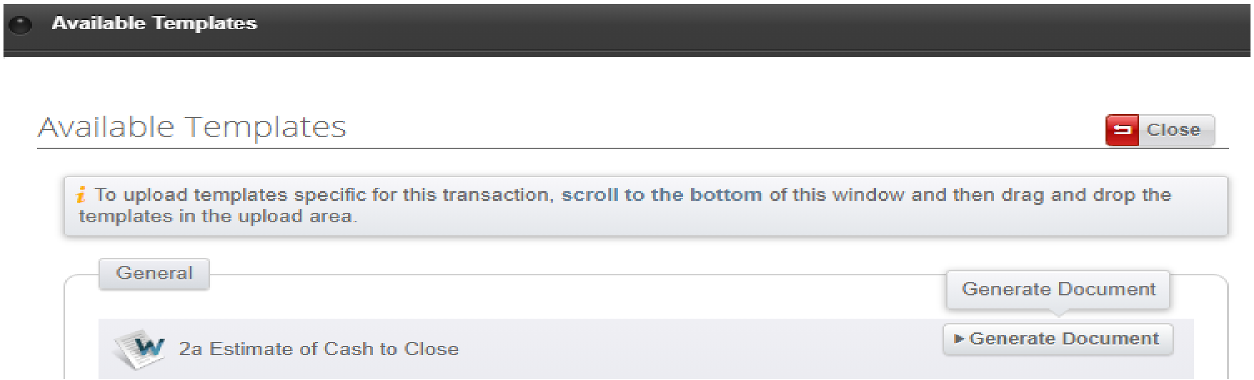
Solution

We researched several options for completing mail-merge operations with Microsoft word documents on a Linux system. Most were clunky and took many hours integrating directly with Java POI and altering the documents was cumbersome. Aspose.Words for Java made the tasks simple and reusable.

We create a ColdFusion component that calls the Aspose.Words Java object. The component accepts certain inputs which query the application database and matches up the database fields with the available mail-merge fields. It then loops through the Word document replacing the fields with their values from the provided data query.

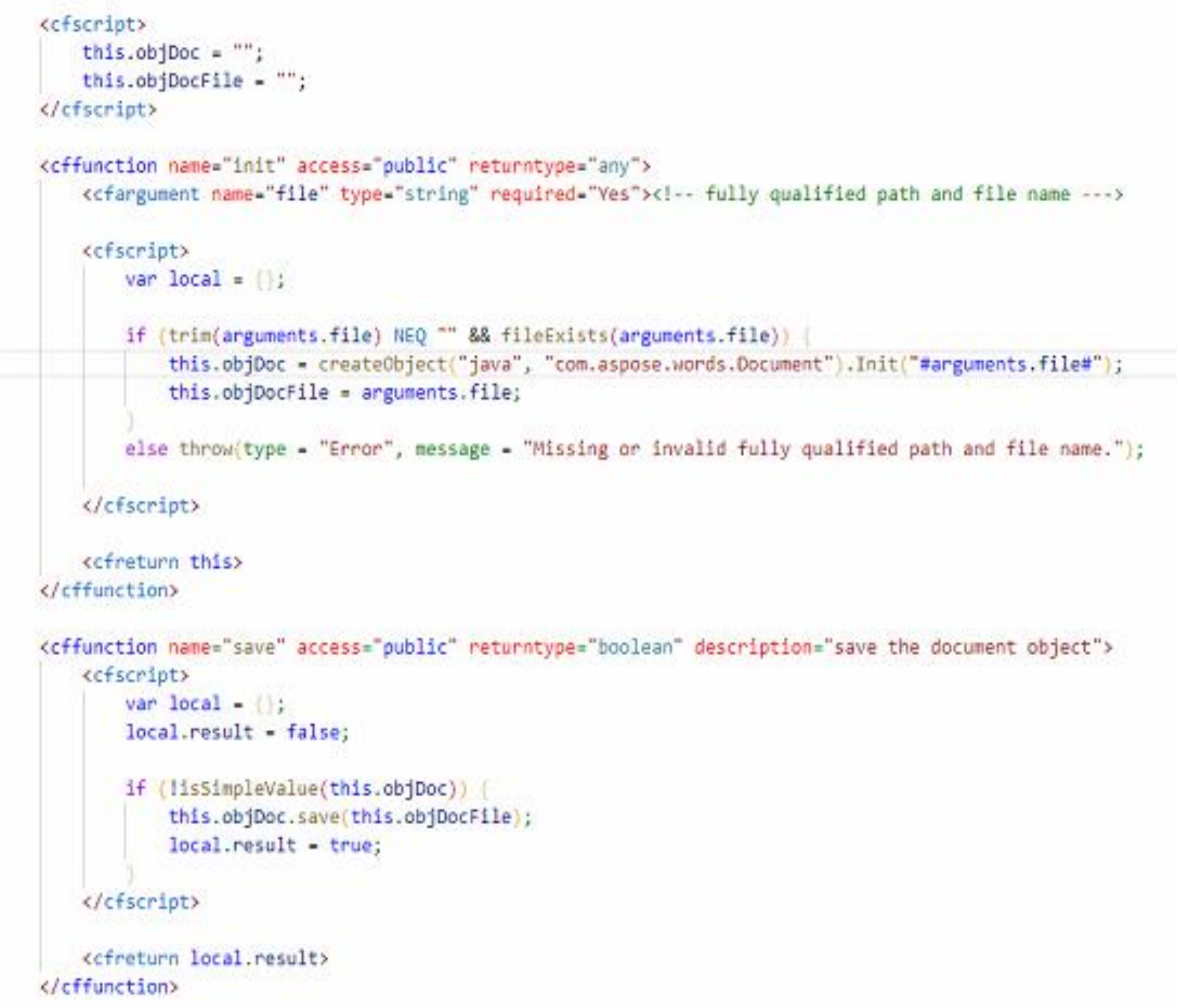
The entire process takes milliseconds and is initiated by the user clicking a button on the web application for the specified Microsoft template and almost instantaneously receives the finished merged document.

Aspose.Words for Java allowed us to provide a compact solution with fewer lines of code that executed much faster and more efficiently than our existing solution. The entire process consumes fewer system resources and can be run by multiple users simultaneously.



i To upload templates specific for this transaction, **scroll to the bottom** of this window and then drag and drop the templates in the upload area.

Figure 1:Template Selection / Generate Document (Mail Merge)



```
<cfscript>
    this.objDoc = "";
    this.objDocFile = "";
</cfscript>

<cffunction name="init" access="public" returntype="any">
    <cfargument name="file" type="string" required="Yes"><!-- fully qualified path and file name --->

    <cfscript>
        var local = {};

        if (trim(arguments.file) NEQ "" && fileExists(arguments.file)) {
            this.objDoc = createObject("java", "com.aspose.words.Document").Init("#arguments.file#");
            this.objDocFile = arguments.file;
        }
        else throw(type = "Error", message = "Missing or invalid fully qualified path and file name.");

    </cfscript>

    <cfreturn this>
</cffunction>

<cffunction name="save" access="public" returntype="boolean" description="save the document object">
    <cfscript>
        var local = {};
        local.result = false;

        if (!isSimpleValue(this.objDoc)) {
            this.objDoc.save(this.objDocFile);
            local.result = true;
        }
    </cfscript>

    <cfreturn local.result>
</cffunction>
```

Figure 2:ColdFusion Component instantiating Aspose.Words for Java object

Experience

We looked at several different options without success. Integrating directly with the Java POI library was promising but development involved constantly searching the POI documentation to achieve any task and this research went deeper and deeper down the rabbit hole as the Java objects were granular. Aspose.Words for Java simplified the tasks providing higher-level access.

Initial implementation took under two hours to simplify our ColdFusion component and integrate calling Aspose.Words for Java. The most difficult part was placing the jar file in the correct folder for Lucee to locate.

The use of Aspose.Words for Java will provide a more efficient solution for our end client and offer additional functionality we will integrate in the future.

Next Steps

We are looking at a PDF manipulation for Smart Forms. We are awaiting sample Smart Form PDFs from this client to determine if Aspose.PDF for Java is necessary.

Summary

We were amazed at how easy Aspose.Words for Java was to implement, and the vast number of features offered through the library.