ASPOSE
Your File Format Experts

CASE STUDIES

Aspose.Cells Case Study

Gjensidige Insurance Denmark

Using Aspose.Cells to write data from SAS to Excel.

Bjarke Felbo, 24th of August 2012

About Gjensidige Insurance

Gjensidige Insurance is a multinational insurance company, which primarily is located in Scandinavia. The team of Analysis Denmark consists of 12 members, who examine and manipulate huge amounts of data. This data is turned into easily understandable graphs and sheets using Excel sheets or other similar tools. Many leaders throughout the company then examine this material and use it to better lead the company in the right direction.

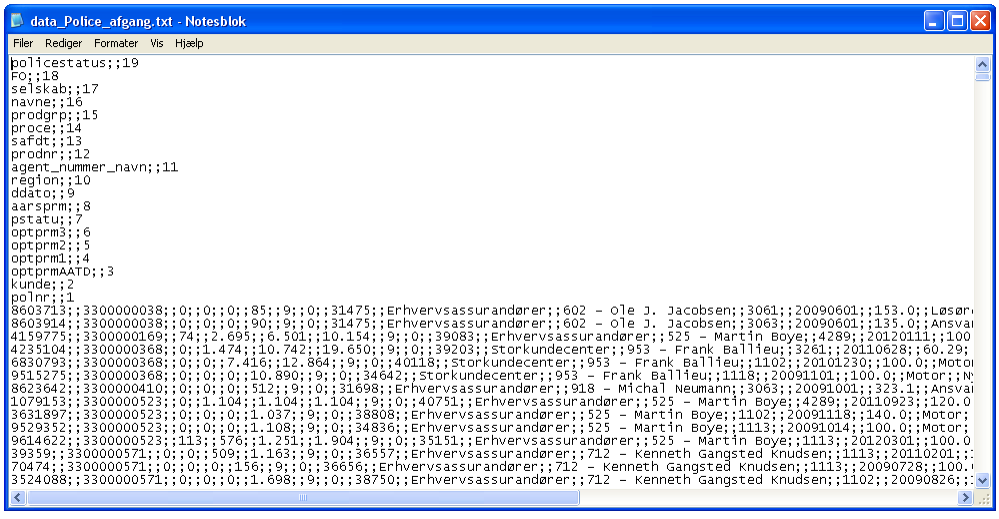
Problem

We are using a SAS server for processing our SAS applications and we needed it to output our data to Excel in a more advanced way than the SAS software allowed us to do. Furthermore, we needed it to be able to do so efficiently and without errors. We were a bit puzzled by which solution would be the best, but a simple C# application and a flat data file (.txt) seemed to work out quite well.

Solution

We chose Aspose.Cells as it seemed as a professional product that simply worked without any issues. The input is SAS data that is written to a flat .txt file by SAS, which is then read by our C# .NET application.

Below is an example of such a text file:



This data is then inserted into an Excel template using our .NET application and Aspose.Cells. The issue concerning the right Excel solution for us was whether it could support our highly advanced Excel sheets with plenty of macros etc.

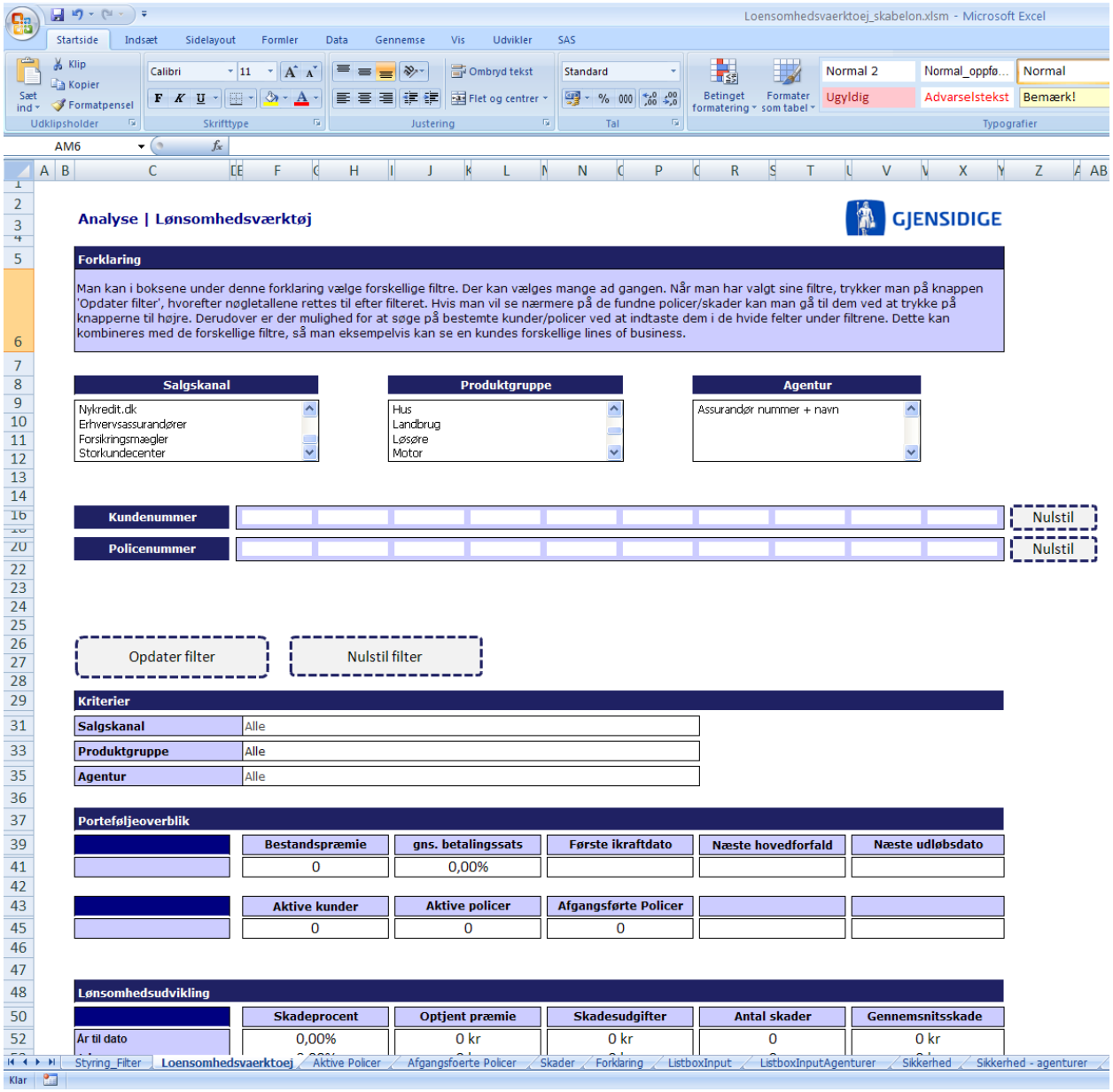


Figure 1: An example of one of our filtering solutions in Excel that we needed Aspose.Cells to handle.

We also needed it to not cause any issues with our SAS applications if any errors were encountered. Another important criterion was the ability to handle Excel sheets with a high amount of data without using a great deal of the server's memory. Instead of having the usual approach of opening an entire Excel workbook, insert data into all worksheets and then save it, we chose to insert data into only one worksheet at a time. While this made performance worse, it helped prevent errors. After all, stability is the most important criteria for us.

Experience

Finding a solution: When we first contacted Aspose we had been using Bytescout Spreadsheet SDK for 10 months. However, its performance was not good enough for us seeing as the output of a large Excel file could almost cause our server to halt. When using Bytescout Spreadsheet SDK with one of our larger outputs with more than 100.000 rows and 20 columns the memory usage would climb until the program crashed at around 1,5-2 gigabytes of allocated memory. After the switch to Aspose.Cells the memory for writing the same amount of data would climb steadily to around 700 megabytes at which point it would stop.

Implementation: The transition from Bytescout Spreadsheet SDK to Aspose.Cells was extremely easy and took less than a day. There were no substantial challenges with using Aspose.Cells. We were worried concerning our large amounts of data, macros and formatting, but Aspose.Cells handled all of those without any issues at all.

Outcome: The outcome is that we are now able to let our server handle all of our needs concerning the output of data from SAS to Excel and we have yet to encounter any errors while using Aspose.Cells.

Next Steps

We do not currently have any plans to take our solution further. However, we are thinking of ways to use Aspose's products in order to directly output our data into other kinds of files such as PDF documents.

Summary

Aspose.Cells easily handled even large Excel sheets that would end up and did so extremely fast. Additionally, it handled all of our macros without any issues at all. We would therefore definitely recommend Aspose seeing as their product has handled all of our needs seamlessly without any real effort from our side.