ASPOSE

Your File Format Experts

CASE STUDIES

Aspose.Total for .NET Case Study

fermator

AUTOMATIC DOORS FOR LIFTS

Fermator Group

Using Aspose.Total to build reports, spreadsheets and labels

Andreu Garcia, Head of Software Engineering, October 2016

About Fermator

Fermator Group is the largest mono-product manufacturer of automatic doors for lifts in the world. Fermator doors are installed worldwide for light, medium and heavy traffic duty lifts, for domestic use, offices, airports, hotels and public buildings. With headquarters in Spain and production plants located in France, Italy, Poland, Greece, India, China and Brazil, the Fermator Group responds effectively to the demands of all markets requiring automatic doors for lifts.

Since the first manufacturing plant started operations, the company decided to have its own ERP system, in order to respond quickly to a market in constant evolution. This system has been in use for more than twenty years and now, although the strategy is still valid, the group has changed a lot (new companies, new markets, new processes, new products…) and requires a new ERP.

My team is responsible for developing a new ERP system, based on .NET technology and following agile methods. The scope of this ERP covers all the business processes included in the value chain of the company (sales, planning, supply chain, manufacturing, assembly, shipment and invoicing), and other cross-activities (like quality, human resources and IT management, among others).

Problem

We faced four main types of requirements related to reporting:

- Reports generation – Our system will have to generate simple documents (like invoices or packing lists, with a master & detail structure), but also complex ones (like assembly orders, with some schemas generated dynamically when the report is built).
- Documents generation – Some times, the information in the system is used as a starting point to elaborate an editable document. This is the case for quotes (sales people get a first version of the commercial quote from the system and complete it with particular conditions for each negotiation). Another example is the “model specification” document, a vision of all the configurable characteristics available for each door type and model.

- Spreadsheets generation – Another need to be solved is the spreadsheets generation. It may be a simple export of the visualized data in a grid, or complex reports in a spreadsheet format to cross information from several sources.
- Labels generation – Our ERP supports also all our manufacturing processes. Many times, these are based on identification labels, which are read by workers with hand-held scanners. This implies bar codes generation (when the space in the label or the document is not limited) or bi-dimensional codes (we use QR codes) when the label has reduced dimensions.

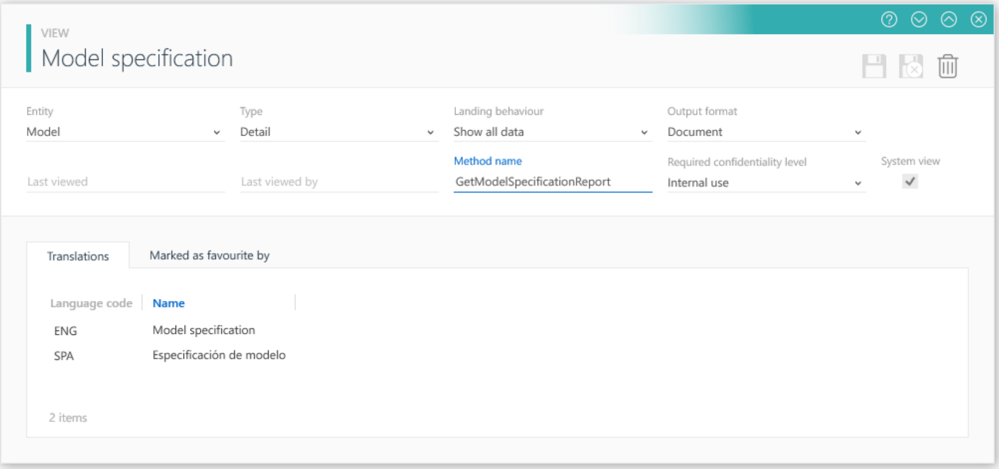
As a summary, reports are used almost by anyone in the company; documents, by the sales and engineering departments; spreadsheets, by the management and direction levels; and labels, by everyone in the shop floor.

Solution

We tested Aspose.Words for .NET and other tools, and Aspose.Words really convinced us. We were able to define a number of Microsoft Word templates, which, nested together, make a stack of documents to generate a really professional set of reports.

This implementation (the templates stack) is useful to decouple header, footer, table of contents and other blocks of our reports. Thus, we can reuse them as building blocks and maintain these parts easily with one single update.

All the system reports have a parameter, where we define the method to be invoked to generate each report.



VIEW

Model specification

Entity	Type	Landing behaviour	Output format	
Model	Detail	Show all data	Document	
Last viewed	Last viewed by	Method name	Required confidentiality level	System view
		GetModelSpecificationReport	Internal use	✓

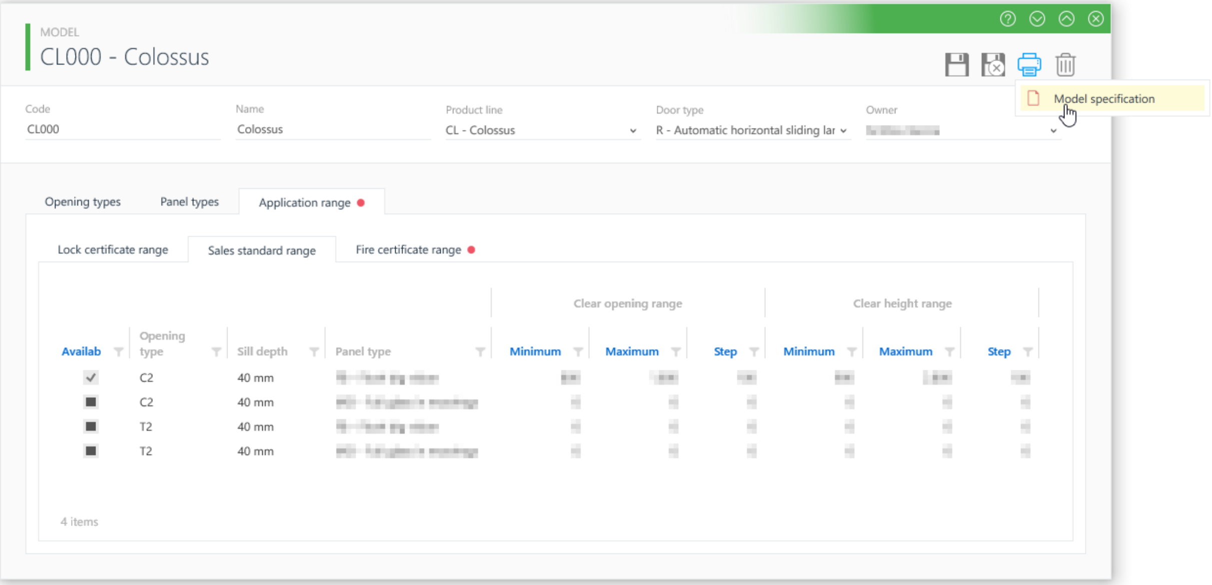
Translations | Marked as favourite by

Language code	Name
ENG	Model specification
SPA	Especificación de modelo

2 items

Capture 1: Reports parameterization.

From many places within the application, the user can click on a printer button, getting a list of available reports for that given context. When the user selects on one of these reports, the parametrized method is invoked and the report generated using the Aspose libraries.



MODEL

CL000 - Colossus

Model specification

Code	Name	Product line	Door type	Owner
CL000	Colossus	CL - Colossus	R - Automatic horizontal sliding lar	[redacted]

Opening types | Panel types | Application range ●

Lock certificate range | Sales standard range | Fire certificate range ●

				Clear opening range			Clear height range		
Availab	Opening type	Sill depth	Panel type	Minimum	Maximum	Step	Minimum	Maximum	Step
✓	C2	40 mm	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
■	C2	40 mm	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
■	T2	40 mm	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
■	T2	40 mm	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

4 items

Capture 2: Generating the document.

Then, the user will see an application window with the report in PDF format. From here, he will be able to send it directly to his default printer or to save it as PDF, DOCX or ODT. This part is not developed yet, but we have done a concept test and it will work.

Experience

Finding a solution: We have strong knowledge in our team about reporting with MS Access and with MS SQL Server Reporting Services. As we are in a process to change from Access to .NET, we thought Access was not an option. On the other hand, Reporting Services is very good to make charts and user defined reports, but we needed quick response and a wide variety of output formats.

The sales support and product support teams from Aspose helped us in this phase of the process so it was really simple to try and check the product features.

Implementation: We installed a fully functional license in a couple of hours. There was a learning curve to generate some reports. This was probably the toughest stage. We dedicated one person for a week to investigate the potential of the tool.

Once we were able to generate three or four types of reports, the knowledge was shared with all the team members and they became productive in a couple of days. We are really surprised by the amount of examples we can find on the technical web pages, GitHub solution and forums from Aspose.

Outcome: At the end, we have what we were looking for. A powerful tool, highly customizable, easy to learn (if you have got minimal knowledge about MS Word and .NET framework) and very quick. We will be able to generate reports faster than with MS Reporting Services and, what is most important, with almost no limitation. If you can do it with MS Word, you will be able to do it with Aspose.Words for .NET.

Next Steps

We have only tried Aspose.Words for .NET, but, as we described in the requirements section, we will have to generate reports in spreadsheet formats and barcodes. We have no doubt Apose.Total for .NET will fit our requirements, so we are betting directly for the whole package.

Summary

As a summary, we can only say one thing: excellent product and excellent service. Sometimes, you can find a good product with a lack of customer care. This is not the case for Aspose. Examples are available and technical support, diligent.

You can even setup a reporting team with different professional levels and skills (designers and programmers) to build your reports by phases, getting user approval after investing just the time you require to write an MS Office document, and being sure you will be able to get the same output when it comes to programming.

We definitely recommend Aspose for .NET product family. It is powerful, fast, and easy to learn.