

CASE STUDIES

Aspose.Total for .NET case study



Kilgray Translation Technologies

Using Aspose.Total for .NET to address file format issues in a server environment in team translation workflows

István Lengyel, Chief Operating Officer, 13 January 2012

About Kilgray Translation Technologies

Kilgray Translation Technologies is the fastest growing provider of computer-assisted human translation tools. The company was founded in 2005 in Hungary, and today it has offices in 6 countries. Kilgray has revolutionized the way teams work together on translation, allowing several people to work simultaneously on the same set of documents, ensuring maximum translation quality in the shortest turnaround time. Kilgray's technology is the most popular server technology among translation companies and increasingly popular among enterprise customers, especially among games developers.

Problem

Translation tools need to allow translators to translate multiple document types in a single editor. Microsoft Office formats are still very popular, but Kilgray's flagship product, memoQ also needs to support a myriad of other formats, from Adobe InDesign through DITA to SQL scripts. Kilgray created filters for RTF, DOCX, PPTX, XLSX, and has filters for XLS and PPT using Office Interop, but never created binary filters for DOC, XLS, PPT. The import of these files normally happens through the client software, which automates the locally installed instance of Microsoft Office. However, users can also use the extensive web services API to integrate their business management tools with memoQ server. In this scenario, Microsoft Office's licensing prevented Kilgray from being able to use Office Interop to handle the XLS, PPT and DOC formats, and prevented the creation of the real-time preview for translation which is done by Office through saving to HTML. As over 40 companies have integrated memoQ with their internal systems, there was a pressure on the company to come up with a solution to import these file formats without requiring a human to convert them to the Office 2007/2010 OpenXML formats.

Solution

memoQ is a .NET application, so Aspose.Total for .NET appeared to be a good solution. Our software processes Microsoft Word, Excel and PowerPoint files, and buying all components together appeared like a good idea. We have made it optional to replace Microsoft Office in the process, and now memoQ can directly import DOC and XLS files and create the preview, without using Office Interop. This speeds up the import process, especially in the case of Microsoft Excel, and allows computers without Microsoft Office installed to process these binary formats. The user does not see any difference, except for the fact that memoQ works on all computers, also on computers without Microsoft Office installed. This development decreases the number of support requests due to problems with Microsoft Office installations, and allows users not to license Microsoft Office – this is, for example, an extra financial burden for Mac users who use memoQ under virtualization and who need to buy a license for Microsoft Office for Windows even if they have a license for Microsoft Office for Mac.

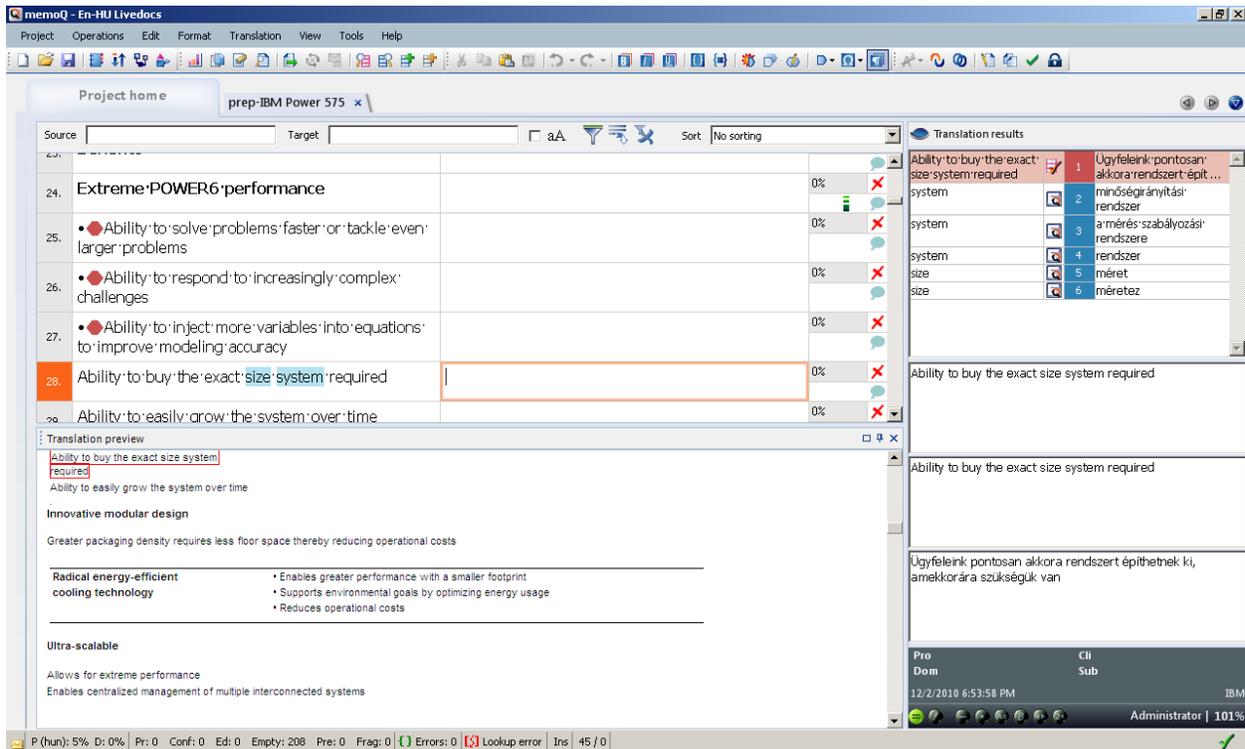


Figure 1: The translation editor and the real-time preview in memoQ.

Experience

Finding a solution: Aspose components were recommended to us by two independent experts in translation technology who have already integrated Aspose components with memoQ. We first heard about Aspose at our user conference. We initially had some concerns about testing Aspose with all the applicable file formats, but the implementation was easier than we thought. This way our product lifecycle manager, who is not a permanent member of the development team, could simply integrate the components – and when he presented it to the management, this came as a surprise. We used a free trial for 30 days, and requested some extra time before the purchase to test the product with Excel as well. We also turned to Aspose support with some initial questions and issues, and support was very helpful. At the moment, the Aspose.Cells and Aspose.Words components are reliable enough to replace Microsoft Office, but we did not find Aspose.Slides to be on par with the other two. Support has promised an improvement, and as soon as this component also becomes reliable, we're going to integrate it.

Implementation: The implementation was surprisingly easy – it required less than a week to integrate the Aspose components into memoQ. The prototype created was almost the final deliverable, and it did not take up much effort from the development team.

Outcome: With the Aspose components in place, users of the memoQ web service API and people without Microsoft Office can also get a preview and import DOC and XLS. This will allow these users to rely completely on memoQ technology and build portals that allow automated quotation and project

processing. It will also allow us to cut some dependencies on Microsoft Office and increase user satisfaction.

Next Steps

The first step was to cut dependency on Microsoft Office. There is however another benefit that we can have through Aspose's products, and that is the import and export of PDF files for translation. PDF files are a nightmare for translation due to the fact that you need to get them into a format that you can process, such as DOCX. Unfortunately many translation customers are not ready to send anything but PDF, and in this case there is a necessary preparation phase. Some of our competitors have integrated an OCR technology that works with mixed results. We originally integrated a text-only PDF filter, which works quite reliably but loses the formatting and the output is TXT. With two components from Aspose, we plan to convert a PDF to DOCX, and then convert the translated DOCX back to PDF. We do not expect perfection here, but we expect this to level us up against the competition. We also believe that translating Microsoft Project files may be interesting in the future, and Aspose.Tasks will help there. It is also possible that we can make use of Aspose components to handle Microsoft Visio diagrams in translation workflows.

Summary

Aspose.Total for .NET allowed us to address the problem of managing three critical filetypes through the API at once. Although currently we only trust Aspose.Words and Aspose.Cells, we believe that Aspose.Slides will soon become powerful enough to address our needs. It was pleasant to work with the support and sales team at Aspose, and we look forward to integrating more of the Aspose file type components.