

Aspose White Paper

The Pitfalls of Microsoft Office Automation

About Aspose

Aspose is a leading vendor of .NET, Java and SharePoint development components, and rendering extensions for platforms such as Microsoft SQL Server Reporting Services and JasperReports. Aspose's core focus is to offer the most complete and powerful set of file management products on the market. Aspose products support some of the most popular file formats in business, including: Word documents, Excel spreadsheets, PowerPoint presentations, PDF documents, Microsoft Visio diagrams and Microsoft Project files. We also offer OCR and image manipulation tools.

Aspose Pty Ltd, Suite 163, 79 Longueville Road, Lane Cove, NSW, 2066, Australia.

[Aspose has a paperless policy.](#)

Contents

About Aspose.....	2
Introduction	4
Why Avoid Microsoft Office Automation?	5
Security	5
Stability.....	6
Scalability/Speed.....	7
Aspose.Cells for .NET vs Microsoft Excel Automation	8
Aspose.Cells for .NET vs Microsoft Excel Automation - A Customer Perspective	8
Aspose.Slides vs Microsoft PowerPoint Automation	8
Price	9
Functionality	10
Summary.....	11
References.....	12

Introduction

Microsoft Office files dominate the business world. Microsoft Excel spreadsheets are the default spreadsheet format in finance, and Microsoft Word word processing is considered important enough to be taught in schools.

When developing business solutions and applications, developers who need to work with text or numbers have to consider these formats. But how can developers work with them?

- Learn the in and outs of file formats and program solutions from scratch.
This approach is time-consuming, expensive and a distraction from most companies' core business.
- Use Microsoft Office Automation.
Microsoft Office provides ways of using their products both client and server-side to create and manipulate documents. Microsoft does not intend for these applications to be used in this fashion, so there are issues with security, stability, scalability and speed, as well as price and functionality.
- Use a third-party component or API.
Third-party solutions allow developers to focus on their core business and work with Microsoft Office files without depending on Office Automation or working with low level file formats. If the solution is secure, feature-rich, stable and reasonably priced, integrating one into your applications can save you time and money. Careful research is needed before committing to a third-party solution to make sure that provides the features you need.

Option 1, developing your own solution from scratch, is untenable for most organizations. It's an approach that soaks up all the time and resource thrown at it and doesn't give fast or accurate results. Microsoft Office file formats are sophisticated and support a large number of features that users expect applications to support.

Options 2 or 3 are better. This paper compares the two, explains the pitfalls of Microsoft Office Automation and uses Aspose's well-established APIs to illustrate issues and solutions.

Why Avoid Microsoft Office Automation?

"All current versions of Microsoft Office were designed, tested, and configured to run as end-user products on a client workstation. They assume an interactive desktop and user profile. They do not provide the level of reentrancy or security that is necessary to meet the needs of server-side components that are designed to run unattended."

Microsoft, Article 257757

Microsoft has made server-side automation available, but this is not the intended usage of Microsoft Office applications, and therefore it's not the most secure or performant way to run them or automate your own applications. This is the core issue with Microsoft Office Automation. It is also the reason organizations look for alternatives to Office Automation.

The main issues with Microsoft Office Automation can be categorized as security, stability, scalability and speed, price, and functionality.

Security

"Office applications were never intended for server-side use. Therefore, Office applications do not take into consideration the security problems that distributed components face. Office does not authenticate incoming requests. Office also does not protect you from unintentionally running macros, or from starting another server that might run macros, from your server-side code."

Microsoft, Article 257757

End-user applications run in a simple, one-user context. Microsoft Office applications are designed to interact with one user at a time. That does not apply when the application runs on a server-- many users expect concurrent access, particularly in web applications.

Aspose products are developed to be equally secure whatever environment they operate in. Aspose's APIs and Cloud products allow you to fully control you code. Apps run in the same context as the application they are part of, under that user. Aspose's Cloud APIs authenticate every incoming request.

To avoid the threat posed by un-safe macros, Aspose products do not automatically run macros when they load a file.

Stability

"[...]Office's implementation of MSI capabilities is counterproductive in a server-side environment. Furthermore, the stability of Office in general cannot be assured when Office is run server-side because it has not been designed or tested for this type of use. Using Office as a service component on a network server may reduce the stability of that computer, and therefore may reduce the stability of your whole network."

Microsoft, Article 257757

Microsoft Office applications are designed for desktop use where they are used by one user at a time. Running an end-user application in a server environment mean that it is used in a way that it is simply not designed for. Firstly, it is a single-threaded application running where a multi-threaded application is needed. Requests have to be tightly controlled to avoid the application locking, or leaving rouge processes in its wake. End-user applications interact with users and, used server-side, can freeze waiting for user-input through dialogs that the user can't see. It's simply not stable.

Web applications, by their very nature multi-threaded server-side applications, experience the same issues.

APIs are part of your application and work in whatever environment you need it to. Aspose's APIs are packaged as a simple DLL or JAR and can be packaged with your application, whether it will run server- or client-side. At no point does APIs require user input.

APIs provide your applications with an extended feature set so that you can parse, manipulate and save Microsoft Office files. The end user does not have to have a license installed locally to modify data or export Microsoft Office files, this is because Microsoft Office is involved at no point in this process.

Stability is paramount to reputation of good software. Aspose APIs are thoroughly tested throughout the software development lifecycle and are extensively used in internal projects to ensure that they are robust and stable.

Scalability/Speed

“Server-side components need to be highly reentrant, multi-threaded COM components that have minimum overhead and high throughput for multiple clients. Office applications are in almost all respects the exact opposite. Office applications are non-reentrant, STA-based Automation servers that are designed to provide diverse but resource-intensive functionality for a single client. The applications offer little scalability as a server-side solution. Additionally, the applications have fixed limits to important elements, such as memory.”

Microsoft, Article 257757

Scalability and speed are central to software projects. Developers need to know that the system they are building will meet the demands of its users. If a company grows, can a business solution be rolled out to new offices and servers without too much trouble? If an online application becomes very popular, will it be badly affected by usage spikes?

"Aspose was a clear winner in performance. Time taken to generate thousands of rows with considerable columns was less than a second."

The Digital Group

Aspose has collected evidence that working with Aspose tools is faster than depending on Microsoft Office Automation.

Aspose.Cells for .NET vs Microsoft Excel Automation

Aspose ran a set of tests with Aspose.Cells and Microsoft Excel Automation. We created files with 2000 rows and 50 columns data. The files were about 2MB for XLS format and 0.5MB for XLSX format.

Table 1: XLS and XLSX files with 2000 rows, 50 columns of data

Product	Time	
	XLS	XLSX
Aspose.Cells	1.06s	1.64s
Excel Automation	299s	297s

Aspose.Cells for .NET vs Microsoft Excel Automation - A Customer Perspective

One of Aspose's customers, The Digital Group (www.thedigitalgroup.com), did their own testing after implementing Aspose.Cells. They tested three scenarios:

Table 2: Three performance tests comparing Aspose.Cells to Excel Automation

Product	Number of records	Time
Aspose.Cells	2500	0.73s ¹
Excel Automation	2500	120s
Aspose.Cells	1050	0.72s
Excel Automation	1050	55s
Aspose.Cells	5014	0.99s
Excel Automation	5014	319s

Aspose.Slides vs Microsoft PowerPoint Automation

To benchmark Aspose.Slides vs Microsoft Office Automation, we developed a simple console application that ran a comparison. The application ran on an Intel Core i5 desktop machine, using Aspose.Slides for .NET 7.8.0 and Microsoft Office 2013.

¹ Rounded from seven decimals to two.

Table 3: Full text scan of 56 presentations (PPT and PPTX)

Product	Time
Aspose.Slides	26s
PowerPoint Automation	90s

Table 4: Converting 24 files from PPT to PPTX

Product	Time
Aspose.Slides	11s
PowerPoint Automation	22s

Price

The cost of Microsoft Office Automation is not just implementation and testing time, but also the cost of licensing. A solution depending on Automation needs a Microsoft Office license on the server, and one on each client machine. For a limited roll-out, this might not add much cost, but for a large roll-out, or for a solution that is expected to grow in terms of users and servers, it becomes an issue.

When selling a solution, the company that developed it has to consider the cost of Microsoft licenses as part of the cost of the solution. If the client company already uses Microsoft Office, this is not much of an issue, but if they don't, it can add considerable cost to the final solution.

Often, a solution's output is a Microsoft Office file but not one that the end user needs to open. With Microsoft Automation, even simple conversions demand that the software is installed, whether the end user will ever use it.

Using a third-party API can still add cost but they tend to have different license models.

Some third-party APIs and components are licensed per server or machine it is installed on.

That may affect cost and have unexpected side effects if the company, or service, needs to expand.

"It wasn't just functionality and performance that convinced Ipsos that Aspose's tools were the solution they were looking for. The licensing model was also more cost effective than other options."

Ipsos

Aspose's licensing model is based on developers and locations of use (.NET, Java and Android) or calls (Cloud). This way, cost is predictable: OEM licenses allow free distribution and is a good option for web applications and desktop software. In this instance, the cost of the license is part of the development cost.

Functionality

Microsoft Office Automation locks developers into the functionality available in the Microsoft Office applications. The upside of this is that it supports most or all functionality. The drawback is that it doesn't support extended feature sets and cannot easily be extended.

Many third-party APIs support a very particular set of functionality, for example conversion, or image manipulation, or table creation, and no more.

Aspose's APIs support most Microsoft Office features. When new features are released, Aspose's development team works to implement them quickly.

Aspose also goes beyond Microsoft Office functionality. For example, Aspose.Slides for .NET and Java, APIs that manage presentation files, allow developers to secure individual items on a slide, as well as entire slides and presentations, thus locking them for editing. This handy feature allows companies to control how their presentations look, and ensure the integrity of important presentations. Another example is Aspose.Cells, an API for working with spreadsheets. It allows developers to export a System.DataView or DataTable directly into a binary Microsoft Excel file.

Aspose APIs offer a large number of conversion and output options, many of which are not native to Microsoft Office applications. For example, Aspose.Words, designed to work with Microsoft Word files, support export to Open Office file formats and EPUB in addition to the standard file formats that Microsoft Word offers.

Aspose's APIs are extensible and allow developers to modify or add functionality. They can also contact Aspose's development team for support and to request new features. APIs are updated regularly and customer-requested features added. For Enterprise-level

"It integrated smoothly with the existing development environment, did not require Windows Automation and provided powerful document controls."

Oracle

organisations, Aspose offers a bespoke development service – Sponsored Support. Companies can sponsor a feature, or set of features, to ensure speedy development.

Summary

Microsoft Office applications are simply not meant to be used server-side or to perform business critical automation tasks. Third-party APIs, like Aspose's .NET, Java, Cloud and Android solutions, overcome this issue by providing programmable features that are 100% independent from Microsoft Office.

Using purpose-built APIs answers the main issues with Microsoft Office Automation:

- **Security:** Microsoft Office applications are not designed to cater for the security needs of client, server-side and distributed applications. Aspose's APIs are designed for client and server-side operation and are safe.
- **Stability:** Microsoft Office Automation is inherently unstable on both client and server-side since they are end-user applications. Aspose's APIs are performant, stable and robust.
- **Scalability and speed:** Because they are not designed to be reentrant or multithreaded, Microsoft Office applications easily become slow when used through Microsoft Office Automation. APIs are part of your application and don't suffer this issue.
- **Price:** Microsoft Office Automation demands a licensed version of Microsoft Office on the server and on every machine that runs the application that uses it, whether or not the end user will use Microsoft applications to open or manipulate the file. Aspose's licensing model is easy to understand and cost effective in comparison.
- **Functionality:** Microsoft Office offers a wide range of great functionality that Microsoft Office Automation can access. Aspose's APIs offer the same functionality and a number of added features useful to developers and end users.

References

The Digital Group, September 2013, "Aspose.Cells for .NET Comparison to Interop Case Study". URL: <http://www.aspose.com/community/files/66/corporate/case-studies/entry495331.aspx>

Microsoft, July 12, 2013: "Article 257757: Considerations for server-side Automation of Office", revision 19.3. URL: <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q257757>

America



USA



+1 888 277 6734



Aspose.US

sales@aspose.com

Europe



UK



+44 141 416 1112



AsposeScotland

sales.europe@aspose.com

Oceania



Australia



+61 2 8003 5926



Aspose.AsiaPacific

sales.asiapacific@aspose.com