



Brian Sentance

Excel Services and Beyond - A deployment dream for quants?

The closest quants used to get to the deployment of their pricing libraries within other systems was writing spreadsheet add-ins. **Brian Sentance** of Xenomorph Software investigates whether the new server-side version of Excel from Microsoft can address the issue of rapid deployment and systems integration of complex financial products

Not the yawn of a new era?

Sit down. Put on your best white coat and developer beard. Pour yourself a stiff drink. The new version of Microsoft Excel, version 12, actually might be really useful. Yep, straight up, there might be a reason (a real reason!) to upgrade to the latest incarnation of Excel.

Whether you like spreadsheets or not, it is hard to deny that Excel has become the “end-user computing” glue for financial markets. Excel brings together pricing models, real-time data, market histories, terms and conditions in such an intuitive, productive and flexible manner that it is the mission-critical system for a whole range of new and not so new financial products and processes. Due primarily to the fact that they are not taken seriously as a “corporate” systems development tool, spreadsheets are also a nightmare to control and support, and are a massive source of operational risk to many financial institutions, both large and small.

So what have Microsoft done to wake me up about Microsoft Excel in version 12? As usual, they have done lots of things. However, head and shoulders above the raft of new features is the

ability to deploy spreadsheets to a scalable, secure, Internet-accessible server environment. This offers quants (and a lot of other people) a deployment route for spreadsheets and pricing models that has the potential to make a real business impact in the management and processing of complex derivatives and structured products.

Server-Side Excel – There can be only one!

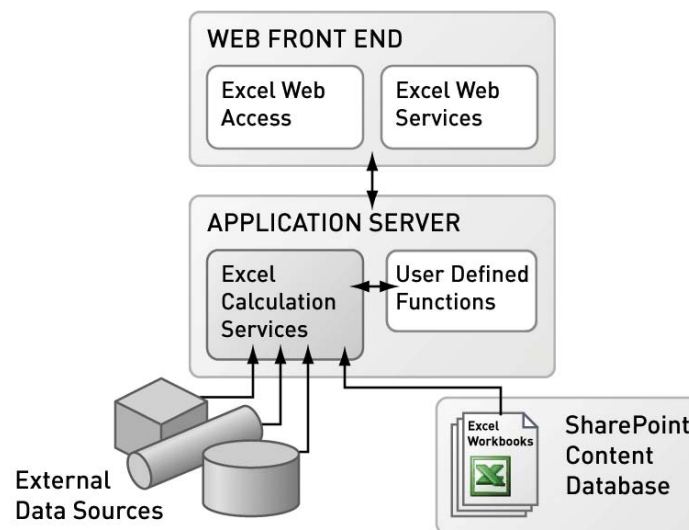
Excuse the quote above from the movie *Highlander*, but it is my personal preference over “One Version of the Truth” which Microsoft is using when talking about their new Excel Services technology. Excel Services is a server technology that is included in Microsoft Office SharePoint Server 2007 (more of which below).

The goals of Excel Services are to provide:

- Centralized, server-based deployment of workbooks
- Scalable multi-user, multi-workbook calculation.
- Interactive, connected, thin-client Excel user interface.
- Web Service API for programmatic access to server Excel
- Security and control over data inputs, outputs and display options.

This is illustrated in Figure 1 below:
As part of Office SharePoint Server 2007, Excel Services also takes advantage of the other capabilities of Microsoft Windows SharePoint Services, including enterprise content

Figure 1 – Excel Services overview



management features such as checking out, checking in, auditing, and versioning capabilities. Sounds a little like source code control for spreadsheet

files and unsurprisingly to a large extent that's what it is.

Sharing a "corporate" spreadsheet

Sheets are shared within Excel Services by publishing the workbook to a trusted location. The trusted location can be a SharePoint repository or as simple as a network share. At publishing time an author can specify is:

- The sheets and range of data you like to be visible.
- The input parameters you'd like a user to be able to override.

Other than those options, "Publish" is simply a glorified "Save As". It's really that simple.

The example we are going to publish is shown below in Figure 2 and concerns the pricing of a simple Vodafone option.

Going back to Figure 1, you can see that there are two ways in which I could access this sheet within Excel Services. The first is a web-based user interface called Excel Web Access, the second being programmatic access via Excel Web Services.

Thin-client Excel

With the Web-based UI, Excel Web Access, users can have both interactive access and read-only access to workbooks in a browser by means of Dynamic HTML. Figure 3 below shows how the Vodafone option sheet is rendered when accessed over the web.

It is important to note that this is not (repeat not!) simply opening up an Excel sheet inside a browser with Excel installed on your desktop. The web-based access to Excel Services does need Excel to be installed and no "tubby" ActiveX controls are installed into your browser environment. So there is one copy of the sheet ("One Version of the Truth"), centrally deployed

Figure 2 - Vodafone option in Excel 2007

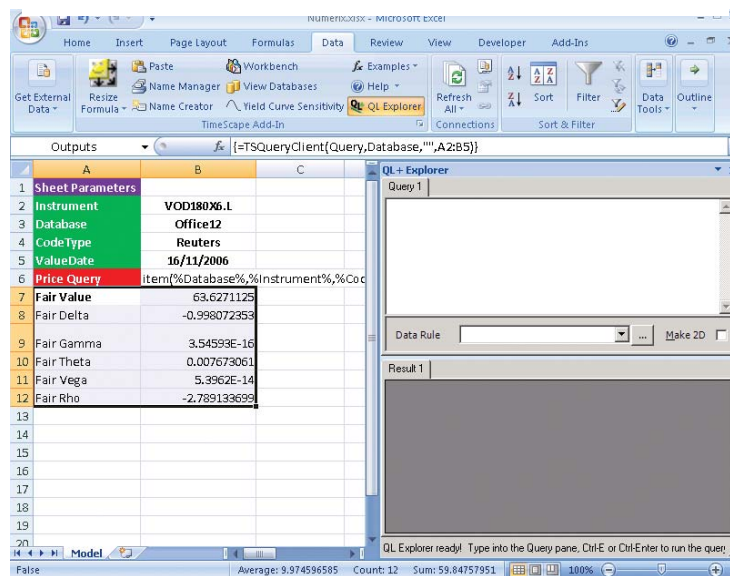
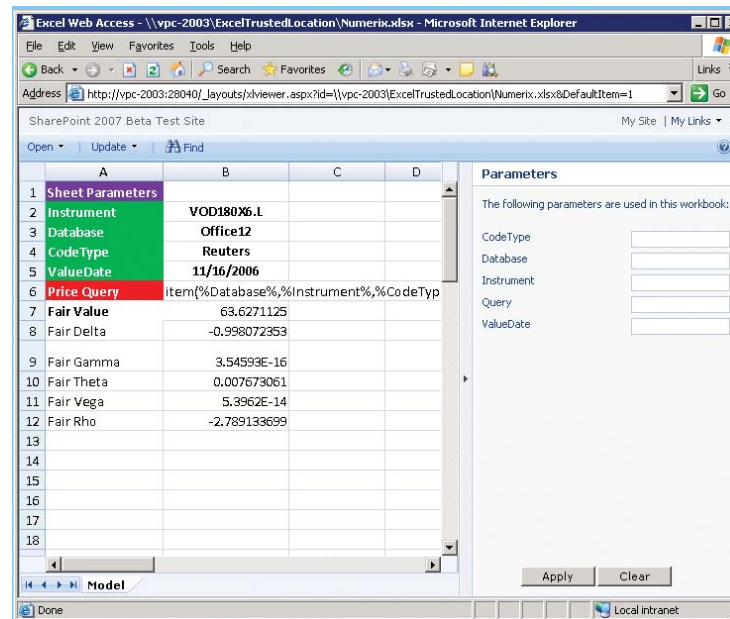


Figure 3 - Excel web access



and potentially accessible anywhere by using just a browser.

Excel for IT development staff?

I am sure that there are significant psychological barriers to this, but we may get to a stage where systems development staff want to use Microsoft

Excel. The Web services interface to Excel Services enables external server applications to directly set cell values, refresh data sources and calculate formula results, enabling them to use their business logic without needing a developer to reengineer it in code. In both Excel Web Services and Excel Web Access, only the data that is designated by the workbook author is available; formulas and other business logic that underlie that data can remain protected. Which brings us to security.

Security

When SharePoint 2007 was architected security was high on the agenda. A SharePoint site may need to be accessed over the Internet, and users with different roles might need different levels of access permissioning. Security for Excel Services comprises the following areas:

- Authoring workbooks.
- Using a workbook.
- Trusted data connection libraries.
- Trusted data providers.
- "Managed Code" (aka .NET) User Defined Functions (UDFs).

I think the most important thing of note is that user defined functions (for more background see references [1] and [2]) have to be implemented as managed code (aka .NET 2.0 and above compliant), and that both old and new Excel 2007 XLL addins are not directly supported in Excel Services. More of this later!

Multi-Use of centralized workbooks

Firstly, how does Excel Services cope with multiple users working against one sheet? For this let's take a look at a workbooks lifecycle within Excel Services as shown in Figure 4 below.

Excel Services deals with multi-users working against the same workbook by making an in-memory copy of the workbook. Each in-memory

copy of a workbook represents the state of a separate user session. When a web service call or Excel Web Access makes a request to a workbook, the session will be referenced via a session ID. So in the above example

- A web service call is made to Excel Services to open a workbook.
- Excel Services loads the workbook if necessary, then makes a copy of the workbook and returns a session ID.
- All set range, get range and calculation requests are made with reference to this session ID.
- Once the workbook has been finished with the workbook needs to be closed (if it is not explicitly closed, it is timed out after a configurable amount of inactive time).

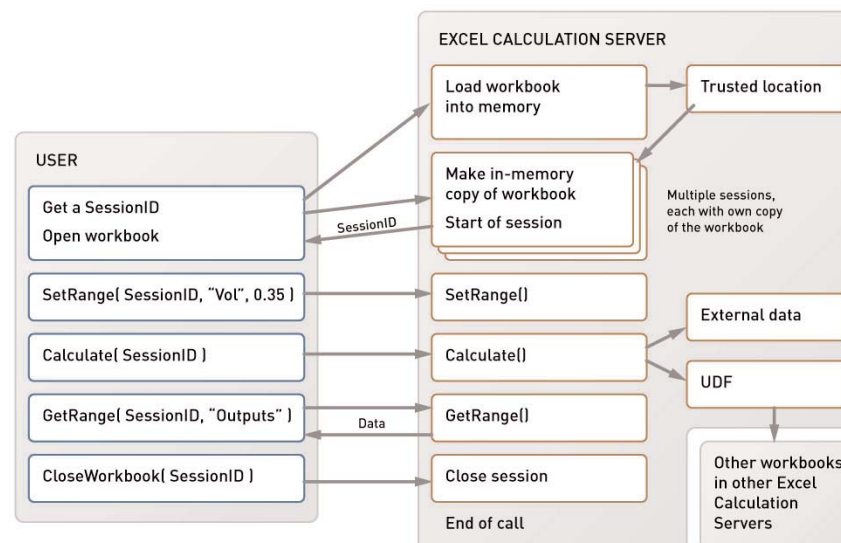
At no time does the user need to directly make copies of the workbook to a file store. This is one of the main features that Microsoft are pushing in Sarbanes Oxley compliance for general spreadsheet usage (mainly in accounting), but obviously it is a great feature for financial markets too.

Caching and scalability

One of Excel Services main strength in dealing with a large number of users accessing a small number of workbooks is the ability to cache both function and workbook results. If the parameters sent to a sheet or user defined function are the same as a previous call, the results are returned without performing the calculation.

Additionally, in order to deal with large numbers of calculations from large numbers of users, Excel Services can scale out within the SharePoint layer and calculations can also be distributed across Microsoft's Compute Cluster Server. On this last point, then the out-of-the-box integration with Microsoft's clustering technology is done via a public API, so it can be connected to other clustering/grid technology. However my guess would be that Microsoft will

Figure 4 – Workbook Life cycle within Excel Services



be using Excel Services as a great way of pushing its clustering technology to a wider audience.

Excel 2007 Limitations

The story sounds pretty good so far, so where's the bad news? Well there is some, as this is the first version of the product from Microsoft and as such it does have some limitations that are important. Key points include:

- No direct support for XLLs (either old kind or Excel 2007 XLLs)
- No VBA code (and this is unlikely to happen in the future)
- No external workbook references (but UDFs can be used to work around this)
- No real-time data functionality (so no DDE or RTD)
- Excel service only reads a couple of new types of Excel worksheet format, XLSX and XLSB.
- No 3D graphs can be viewed via web viewer.

Given the number of derivative and fixed income pricing libraries implemented as XLLs, the omission of direct support for them is a pain. The hurdle is more inconvenient than insurmountable however, and my guess would be that there will both in-house and vendor based solutions developed to automatically wrap an old XLL

so that it can be used inside Excel Services. For a good article using XLLs in Excel 2007 client and using/migrating to the new XLL interface then take a look at reference [3].

The lack of VBA support is equally painful, but maybe more technically understandable and justifiable than the exclusion of XLLs. I would think that we might be presented with some form of .NET based scripting language for Excel Services and Excel Client before too long.

Summary

I think that quants should take a good look at Excel Services. In my view the technology has the potential

to both improve and change business processes around instrument pricing and model deployment. It is also a great opportunity for quants to be much more visible within the organization and potentially to do so much more than just create and deliver great pricing libraries.

For Xenomorph's take on Excel Services see: <http://www.xenomorph.com/wilmott>

My thanks to:

Martin Luker-Brown, *Senior Consultant, Xenomorph*
Chris Budgen, *Chief Technology Architect, Xenomorph*
Danny Khen, *Excel Development Team, Microsoft*

REFERENCES/LINKS

- [1] User-defined functions in Excel Services:
<http://msdn2.microsoft.com/en-us/library/ms493934.aspx>
- [2] User-defined functions for both Excel client and server:
<http://blogs.msdn.com/excel/archive/2006/09/22/764030.aspx>
- [3] Upgrading XLLs to Excel 2007:
<http://msdn2.microsoft.com/en-us/library/aa730920.aspx>
- [4] Excel Services entry point on MSDN:
<http://msdn2.microsoft.com/en-us/library/ms519581.aspx>