Netwoven Inc.

Business Intelligence White Paper - II

Excel Services in SharePoint 2010

Vivek Karnataki
Principal Consultant
Netwoven, Inc.

info@netwoven.com

Contents

Introduction

This paper is the second paper in Netwoven's Business Intelligence series. It provides information about Microsoft's Excel Services for Business Intelligence.

Microsoft Office Excel is one of the most popular tools used in organizations today to perform a wide variety of tasks from managing project schedules to performing complex financial analysis. Over the past few years, a lot more features have been added to Excel making it a really useful Business Intelligence tool. Since Excel has now become such a vital instrument in almost every department in an organization, it has become even more imperative to ensure better control over the management and distribution of Excel spreadsheets. In addition to that, data contained within Excel spreadsheets is frequently required to be consumed by other applications such as a dashboard or a reporting application.

Excel Services addresses many of these common challenges encountered by organizations and thereby extends the reach of Excel spreadsheets and the knowledge contained within so that it can be used for the benefit of the entire organization. Excel Services is a server-side technology built in to Microsoft SharePoint Server 2010 that provides a secure, controlled and configurable way of hosting and distributing Excel spreadsheets. With the help of Excel Services, you can permit multiple users to safely and concurrently access data within a spreadsheet; incorporate a complete workbook or a part of it into a dashboard or a portal; perform intensive workbook calculation on the server instead of client machines or build custom applications that can leverage the information contained within Excel spreadsheets.

Excel Services Overview

Broadly, Excel Services is a shared service hosted in SharePoint Server 2010 that can be leveraged to display and share Excel workbooks from a document library in SharePoint Server 2010. There are four primary means of interacting with the Excel workbooks using Excel Services

- **Excel Web Access Web Part**: Allows you to securely share Excel workbooks hosted on SharePoint Server 2010
- **Excel Web Services**: Permits custom applications to programmatically access, calculate and extract the data within Excel workbooks
- **Excel Services REST API**: Gives you the ability to access Excel workbook parts or elements directly through a URL
- **ECMAScript Object Model**: Allows you to customize, automate and build mashups using Excel Web Access Web Part

What's new in Excel Services 2010

Excel Services was first introduced along with Microsoft Office SharePoint Server 2007 and in Excel Services 2010; several new features have been added for IT users, both the knowledge worker and the administrator as well.

General Improvements

Better user experience for Excel users: in SharePoint 2007, although the rendering of the Excel workbook in the Excel Web Access (EWA) web part was almost similar to that of the Excel client version, there were some significant differences, such as difficulties in scrolling, the inability to resize rows and columns and the inability to copy data from the EWA web part and paste into other applications. Excel Services 2010 improves upon some of these short-comings and makes the user experience more like what they are used to in the Excel client. Excel Services 2010 has better scrolling experience giving it a more Excel client like feeling, that enables you to scroll around the entire range quickly and easily. The EWA web part is now AJAX based, thereby making the periodic external data refresh seamless. We can now easily send multiple values from other web parts to the EWA web part. This simplifies the creation of feature rich dashboard and mashups. In addition to these improvements there are host of other improvements such as the ability to type into the EWA web part grid, ability to select multiple cells, rows or columns and the ability to copy from the web part and paste into another applications.

Better integration with SharePoint: Excel Services has strong integration with SharePoint Server for security, content management, version control, compliance and service administration. Excel Services 2010 also supports all the BI capabilities within SharePoint Server 2010 such as PerformancePoint Services and PowerPivot.

Better tools for application development: In addition to the introduction of the ECMAScript or JavaScript Object Model and the REST API, which has been discussed above, Excel Services 2010 improves upon the existing application development options as well. There are new additions to the Excel Web Services API that enable custom applications to . There are significant improvements to the EWA Web Part that makes the experience of working with Excel workbooks within the web part very close to that of an Excel client. The new improvements to the UDF now enable users to

Multi-user Collaboration: Multi-user collaboration permits multiple users to concurrently work and make changes to any workbook. This feature is supported by using an algorithm in Excel Calculation Services (ECS). Users can simultaneously view and edit workbooks and these edits are processed in the order in which they are received by the ECS.

Slicer feature: The Slicer feature is a new type of data filter in Microsoft Excel 2010, which is interactive and flexible in design and layout. This filter gives the authors the ability to easily write OLAP data models and build rich interactive reports around them.

Support for new Excel 2010 features: Most of the new features introduced in Excel 2010 are supported in Excel Services 2010 such as viewing Sparklines and Slicers, viewing PivotTable named sets and PowerPivot files and the new Excel 2010 functions. In addition to this, Excel Services also improves upon

the handling of unsupported features of Excel 2010. In the earlier version, if a workbook contained unsupported features, Excel Services would not open the file at all. Excel Services 2010 now supports a lot of the features of Excel 2010 but also permits opening the workbook if it contains certain unsupported features.

Additional Programmability Options

REST API: The REST API is a client-server protocol that defines entities, such as Ranges and Charts, within an Excel workbook and permits other applications to access those entities using a URL. The API permits access to the entities using the stateless HTTP protocol and also provides a method for users to set values in the workbook.

JavaScript or ECMAScript Object Model: The ECMAScript model enables syndication, mashups, and automation of Excel Services and the extension of Excel Services by third parties. It mirrors the Excel Services Web Services API functionality; however, it is not a proxy for this API.

Administration

Delegation of Services Permissions: The SharePoint Server 2010 central administrator can now delegate permissions to administer Excel Services (and other shared applications) to other users.

Unattended Service Account: The unattended service account is an encrypted and secure account with low privileges. The Secure Store Service (SSS) stores the credentials of the unattended service account and they are stored or cached as needed per session or connection. By impersonating the unattended service account, the SharePoint Server 2010 databases and other external data sources that can be accessed from Excel Services are protected.

Trusted Locations: Trusted locations are file paths to SharePoint sites, SharePoint document libraries, UNC paths or HTTP web sites where the administrator has explicitly allowed workbooks to be accessed from. Trusted locations are provided by default and no administrator action is typically needed. Only the workbooks located in trusted locations are loaded by Excel Services 2010.

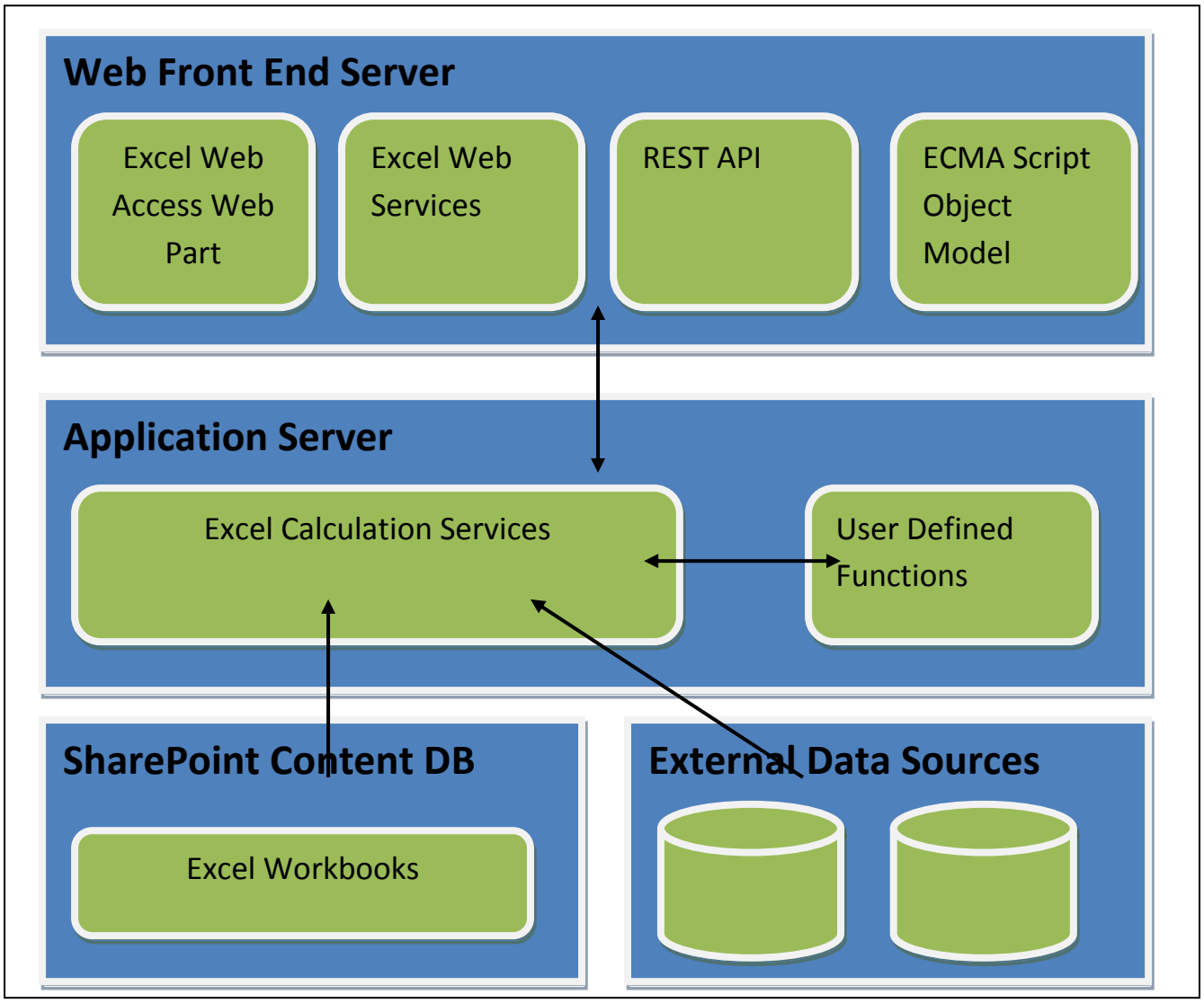
Support for Windows Powershell: Windows Powershell provides simple command line scripting interface to perform tasks for administration and deployment. Powershell scripts, called command-lets (cmdlets), are reusable and hence can be used for repetitive tasks such as performing maintenance and administration. Excel Services 2010 supports Windows Powershell and has specific cmdlets for performing tasks such as getting the Excel Services application or getting a trusted location object and modifying their settings.

Excel Services Architecture

At its core, Excel Services is built using ASP.NET and Microsoft SharePoint Foundation technologies. There are 5 key components of Excel Services

- Excel Web Access (EWA) Web Part
- Excel Web Services
- REST API
- ECMAScript
- Excel Calculation Services (ECS)

Architecturally, since Excel Service is built upon SharePoint, these components can be grouped according to the SharePoint Servers that these components typically get deployed upon. The Web Front End (WFE) Server hosts the EWA web part, the Excel Web Services, the REST API and the ECMAScript Object Model whereas the Excel Calculation Services runs on the SharePoint Application Server.



Excel Services Components

This section discusses the components of Excel Services in detail.

Excel Web Access (EWA) Web Part

As the name implies, EWA web part is a SharePoint web part that renders Excel workbooks on a web page and permits multiple users to simultaneously interact with those workbooks. The EWA web part essentially creates a the HTML markup for the workbook when it's added onto a web part page and closely mimics the functionality provided by the Excel client.

To use the web part to view workbooks, you need to add the EWA web part to a SharePoint page and then modify the web part properties to add the location of the workbook that you want to display, enter the named range within the workbook and then set the rows and columns that you want to display.

Using the EWA web part, knowledge workers can easily share workbooks across the organization and yet secure parts of the workbook if needed. They can build Business Intelligence dashboards, portals or mashups by connecting the EWA web parts to other web parts. Since the web part grid supports AJAX, it's possible to have data displayed in the EWA web part refreshed periodically without interrupting the users. The EWA web part supports both the IFilterValues interface and the IWebPartParameters interface, which permits sending multiple values to the EWA web part.

The EWA web part can be easily extended programmatically as well. Most of the classes used to extend the functionality of the EWA web part reside in the Microsoft.Office.Excel.Server.WebUI namespace. Some of the properties that can be modified programmatically are

- Configure the EWA web part
- Apply theme or styling using CSS to the web part and the web page
- Change the properties of the EWA web part

Excel Web Services

Excel Web Services is an API that provides programmatic access to the Excel Web Service. These APIs can be used to harness the power of Excel Calculation Services (ECS) on the Server. Using the web services, you can incorporate the server-side workbook logic into an application, automate the updating of Excel workbooks and create application specific user interfaces around server side Excel Calculation.

Two of the biggest scenarios for the usage of Excel Web Services are

- Using the logic in Excel spreadsheets in a server application: This permits the logic of a calculation to reside in the workbook on the server. Any custom application that needs to use the logic does not have to recode the logic all over again in a programming language. Instead, the developer can just call the Excel Web Service to incorporate the logic residing in the spreadsheet. Another variation on this would be providing custom UI to Excel-based server applications which use Excel Web Services to interact with a server-side calculation session.

- Automating spreadsheet updates on servers: This works especially well in combination with the new Open XML file format, which greatly simplifies the task of programmatically creating an Excel file from scratch or using a template. Once the new file has been created, it often needs to be calculated – for example, if there are external data feeds that need to be updated. It is straightforward for developers to write code to use Excel Web Services to do all of this, then retrieve an up-to-date copy of the calculated file and save it back to the server, or deliver it to any other destination.

Each web application or a site has its own instance of the web service. You must access the API from the correct site to ensure that the site context is used for authentication and authorization. For example, the URL of the API for the web application at http://SharePointServer is the following:
`http://SharePointServer/_vti_bin/ExcelService.asmx`
And the URL of the API for the site at http://SharePointServer/site is the following:
`http://SharePointServer/site/_vti_bin/ExcelService.asmx`

It's very straightforward to use the Excel Web Service in custom application, however, each time Excel Services opens a workbook, a new session is created. Each created session has an associated session ID that is unique. The session ID becomes a key for the Excel Services web service methods to perform operations on the caller's session, and maintain state for the life of the session. To perform any operation on the workbook using an API method, a valid session ID is required, which implies that the session is still active.

Another approach of using the Web Service is by linking the code directly to the web service assembly and call methods in-process instead of using SOAP. For this approach to work, the web service calling code must have SharePoint Foundation site context, which means the code, runs within SharePoint Foundation. For this, you need to add a reference to the Microsoft.Office.Excel.Server.WebServices assembly.

Excel Web Services (when accessed through SOAP or direct linking) exposes a relatively rich set of error codes when things go wrong. Different mechanisms may consume these errors in different ways but if you are using Visual Studio to access the API, the errors will be presented in the form of a SoapException thrown from the call. Even when using direct linking, users will still see SoapExceptions being thrown — the same as when using a generated proxy.

REST API

Representational State Transfer or REST services are based on the premises that an addressing schema is used to locate networked resources and a methodology is used for returning the representations of these resources. Excel Services 2010 support REST and the REST APIs provides a framework for easy discovery and access to data and objects defined within an Excel workbook. Simply put using the REST APIs you can access the elements defined in an Excel workbook using a URL.

The first part of the URL is a representation of the workbook location. The second part is the path to the requested element inside the workbook. Overall, there is another "marker" which is our entry point (an aspx page)

For example, if you have a workbook called USSales.xlsx which is located in a document library called "Sales Forecast". To display the sales made in the Western Region of the US, you create a chart called "WesternUSSales". You can get direct access to this chart via a simple URL:

```
http://server/_vti_bin/ExcelRest.aspx/Sales%20Forecast/USSales.xlsx/Model/Charts('WesternUSSales')
```

Each Excel Services REST URL is built out of three parts:

Marker path: "`http://server/_vti_bin/ExcelRest.aspx`". To access REST you always need to preface it with this.

File path: "`/Sales%20Forecast/USSales.xlsx`". This is the file that contains the element you are interested in.

Element path: "`/Model/Charts('WesternUSSales')`". This is the path inside the workbook to the element you request.

The result of this URL is a PNG stream which would be the chart that is named WesternUSSales. When the workbook updates, the REST path will return the new chart – which is live and up-to-date. Placing this URL inside an img tag on a page will place the image on the page:

```
<img src =
"http://server/_vti_bin/ExcelRest.aspx/Sales%20Forecast/USSales.xlsx/Model/Charts('WesternUSSales')">
```

The REST API has a built in discovery mechanism that allows the developers to explore the content of the Excel workbook programmatically or manually by supplying ATOM feeds that contain information the elements that reside a specific workbook.

The anchor for starting discovery is of the format - the marker path, the file path to the workbook and then append '/Model' to the URL. In the above example, this would mean:

```
http://server/_vti_bin/ExcelRest.aspx/Sales%20Forecast/USSales.xlsx/Model
```

Four resource collections are currently supported and they will show up when the above URL is used for discovery. They are – Ranges, Charts, Tables, PivotTables. Following any of these links will result in another ATOM feed that will display the elements defined in those collections for the workbook.

The REST API can be used to return data in different formats such as HTML, ATOM feeds, Image and a Workbook. To return the data in the required format, you will have to append the URL with the querystring parameters ?format=html, ?format=atom, ?format=image and ?format=workbook respectively.

Overall, REST APIs in Excel Services enable the users to

- Discover the elements that are defined in a workbook such as Charts, Ranges, Tables and PivotTables
- Retrieve the items in the workbook using an Image, HTML, ATOM feed or Excel workbook format
- Set values in the workbook and recalculate the workbook before retrieving elements

Excel Calculation Services (ECS) and User Defined Functions (UDFs)

The Excel Calculation Services (ECS) is responsible to load workbooks, perform calculations, call custom code (User-defined Functions) and refresh external data. It also maintains a session for the duration of interaction with the same workbook by a user. Underneath, ECS a web service and runs as part of the IIS on each application server machine. The WFE components (that is, the EWA and the API) communicate with the ECS via Web services. You can access the web service by using the following format of the URL

```
http://ECSMachineURL/TheSSPName/ExcelCalculationServer/ExcelService.asmx
```

User-defined functions (UDFs) give you the ability to use formulas in cells to call custom methods written in managed code and deployed to SharePoint Server 2010. UDFs extend the calculation and data import capabilities of Excel. Once a UDF is created and deployed to the SharePoint Server, end users can use UDFs just as they would use any in-built Excel function.

Briefly, the following steps need to be followed to build a UDF assembly

- Reference the Excel Services UDF assembly, Microsoft.Office.Excel.Server.Udf in your project.
- Mark the custom UDF class with the UdfClass attribute.
- Mark the methods within the UdfClass marked class with the UdfMethod attribute.
- Deploy the built UDF assembly in to a local directory, GAC or a network share. The identity of the assembly can be exposed using the full path or the strong name of the assembly.
- In the **Enable Assembly** section in Central Administration, select the **Assembly enabled** check box to enable Excel Calculation Services to call the assembly.

ECMAScript / JavaScript Object Model

Excel Services 2010 now support ECMAScript / JavaScript browser side object model that permits developers to closely interact with the Excel Web Access (EWA) web part. With JavaScript Object Model (JSOM), developers can now build a whole new range of compelling solutions by detecting and reacting to a user's interactions with a EWA web part and programmatically interacting with one or multiple EWA web parts.

The JSOM can be used by inserting the JavaScript code on the page that contains the EWA web parts. Code can be added to the web part page directly by editing the .aspx page or by using a Content Editor web part to add the JavaScript code.

By using the JSOM, a developer can add additional functionality to pages that contain EWA web parts by accessing items such as sheets, ranges, tables, PivotTables, and charts. It is possible to set and retrieve values from individual cells or from Excel style A1 ranges or named ranges. The JSOM also provides for events that are raised when the user changes the active selection or active cell or when the user starts editing a cell. The JSOM can also be used to scroll to a different region and to switch the displayed sheet or named item. The JSOM can be used by itself to provide a means for automating Excel Web Access, or to provide part-to-part communication between different EWA web parts. It can also be used with other APIs such as the new Visio Services JavaScript Object Model, Bing Maps SDK and many others to build custom solutions/mash-ups.

About the Author

Vivek Karnataki is a Principal Consultant with Netwoven and specializes in Enterprise Content Management custom development using SharePoint 2007 and 2010, Business Intelligence using SQL Server and PerformancePoint Services. He also specializes in advanced .NET application development using .NET 4.0.

About Netwoven

Netwoven is a professional services firm specialization in the design and implementation of Enterprise Content Management and Business Intelligence solutions. This paper is part of the Business Intelligence series of papers that Netwoven will be publishing over time. For additional information, please contact Netwoven at info@netwoven.com.