

# Bhaviks B.I. Babble

... on Microsoft Business Intelligence technology and more!

## Setting Up a SharePoint 2013 BI Demo VM

I just built up a new demo environment for BI with Windows Server 2012, SQL 2012 SP1 and SharePoint 2013. This is primarily to take advantage of the new Power View enhancements that came out with SQL 2012 SP1 but I do step through getting all the other BI services up and running. What follows is a general guide, providing an overview of the main steps and links to other information where necessary. I am focusing on the SQL and SharePoint install here, so I assume you have a Windows Server installation ready and have setup a domain and DNS services. I used Windows Server 2012 for my VM.

### Pre-requisites

1. If you don't have an internet connection from your VM, make sure you have downloaded the Sharepoint 2013 pre-requisites. An [in-depth guide exists on TechNet](http://social.technet.microsoft.com/wiki/contents/articles/14582.install-sharepoint-2013-prerequisites-offline-or-manually-on-windows-server-2012-a-comprehensive-guide.aspx) (<http://social.technet.microsoft.com/wiki/contents/articles/14582.install-sharepoint-2013-prerequisites-offline-or-manually-on-windows-server-2012-a-comprehensive-guide.aspx>) which discusses options on obtaining the pre-requisites, and then [this guide will help you](http://www.dontpapanic.com/blog/?p=241) (<http://www.dontpapanic.com/blog/?p=241>) install them quickly.
2. Then you will also need the following
  - ∞ Windows Server 2012
  - ∞ Microsoft Office 2013 (Excel and PowerPoint)
  - ∞ SharePoint 2013
  - ∞ [SQL Server 2012 Samples](http://msftdbprodsamples.codeplex.com/releases/view/55330) (<http://msftdbprodsamples.codeplex.com/releases/view/55330>). It's a good idea to deploy all these to test all your features are working as expected.
  - ∞ [Visual Studio 2010 Tools for Office Runtime](http://www.microsoft.com/download/en/details.aspx?id=20479) (<http://www.microsoft.com/download/en/details.aspx?id=20479>). The MDS Client for Excel and PowerPivot 2012 will need this. You can get these clients and other goodies in the [SQL Server 2012 Feature Pack](http://www.microsoft.com/download/en/details.aspx?id=29065) (<http://www.microsoft.com/download/en/details.aspx?id=29065>)
  - ∞ [SQL Server 2012 Data Mining Addins](http://www.microsoft.com/download/en/details.aspx?id=29061) (<http://www.microsoft.com/download/en/details.aspx?id=29061>)

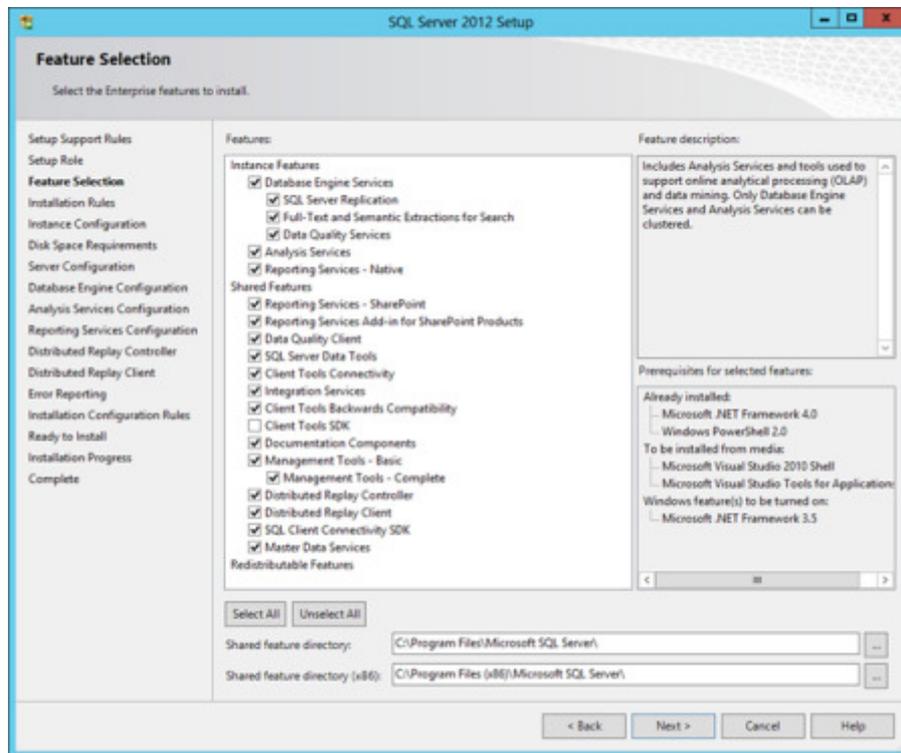
### Installing SQL Server 2012

1. Ensure Windows has the Active Directory Domain Services role installed. **This is important so we can use a proper domain account for services.** Ive seen problems when using built in service accounts. It will suggest you add in the DNS Server role a well. I did so. In my virtual machine I setup 2 network adapters – one with a fixed IP on an internal virtual LAN Segment, the other bridged so it could access my host's internet connection.
2. Install a DEFAULT instance of SQL Server 2012. I used slipstreamed media which has SP1 built in. Things to note:

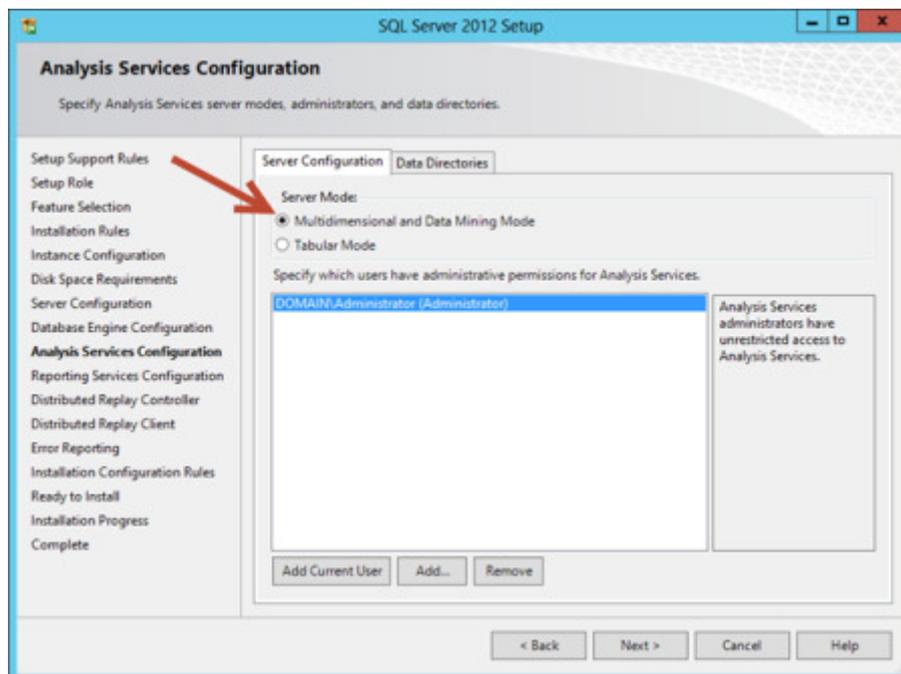
☞ I used the domain Administrator account to run all services except for the Browser, Agent and Full Text Filter.

☞ Remember to add the domain Administrator account as an admin to the relevant services as you configure them in the install Wizard. You don't want to be locked out later

These are the features I selected (pretty much all)..



... Noting that when I configured the services, I picked **Multidimensional and Data Mining Mode**:



1. I then ran the Reporting Services Configuration tool to setup the Native SSRS instance.
2. I wanted a Tabular version of SSAS, so I ran the installer again and chose to install a NAMED instance of SQL Server 2012, which I called TABULAR. This time the only feature I selected was Analysis Services, but when asked to choose the mode I selected Tabular.

## SQL Server 2012 Help

For some reason Microsoft decided not to include local help files on the SQL install media so you need to [get it yourself \(http://www.microsoft.com/en-us/download/details.aspx?id=347\)](http://www.microsoft.com/en-us/download/details.aspx?id=347) and manually install it. It can be a bit tricky so pay attention to the instructions that follow the download:

### Product Documentation for Microsoft SQL Server 2012 for firewall and proxy restricted environments

If your download does not start after 30 seconds, click here: [Start download](#)

#### Instructions

Instructions for getting SQL Server product documentation into secured environments

1. Click the Download button on this page to start the download.
2. Click Save to save the download to a USB drive or other portable media that can be carried into the firewalled environment.
3. Double-click the self-extracting EXE to unpack the contents.
4. Specify a location of your choice to extract the contents to. For example, a network share that is accessible to your destination server.
5. Launch Help Library Manager by going to Start, All Programs, SQL Server 2012, Help & Community, and clicking Manage Help Settings or by launching SQL Server Management Studio, clicking the Help menu, and then selecting Manage Help Settings.
6. In Help Library Manager, click Choose online or local help.
7. Select I want to use local help and click OK.
8. Click Install content from disk and browse to the directory where you unpacked the downloaded EXE.
9. Select the HelpContentSetup.msha file and click Open.
10. Click Next.
11. Click Add next to the documentation you want to install.
12. Click Update.
13. Exit Help Library Manager.

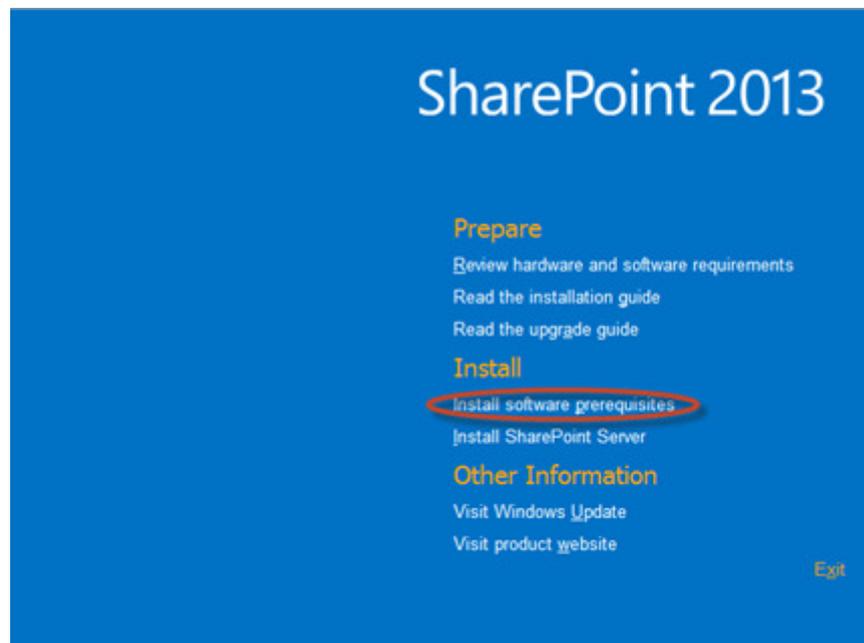
## Developer Tools

At this point I would recommend installing the following:

- ∞ [SQL Server Data tools \(http://msdn.microsoft.com/en-us/data/tools.aspx\)](http://msdn.microsoft.com/en-us/data/tools.aspx) (i.e. SSDT – highly recommended)
- ∞ [SSDT Power Tools \(http://visualstudiogallery.msdn.microsoft.com/9b0228c6-15d1-44de-9279-66dde12bf861\)](http://visualstudiogallery.msdn.microsoft.com/9b0228c6-15d1-44de-9279-66dde12bf861) (the official mechanism to update SSDT)
- ∞ Visual Studio 2010 with SP1 (if you want need complete .Net programmability)

## Installing Sharepoint 2013

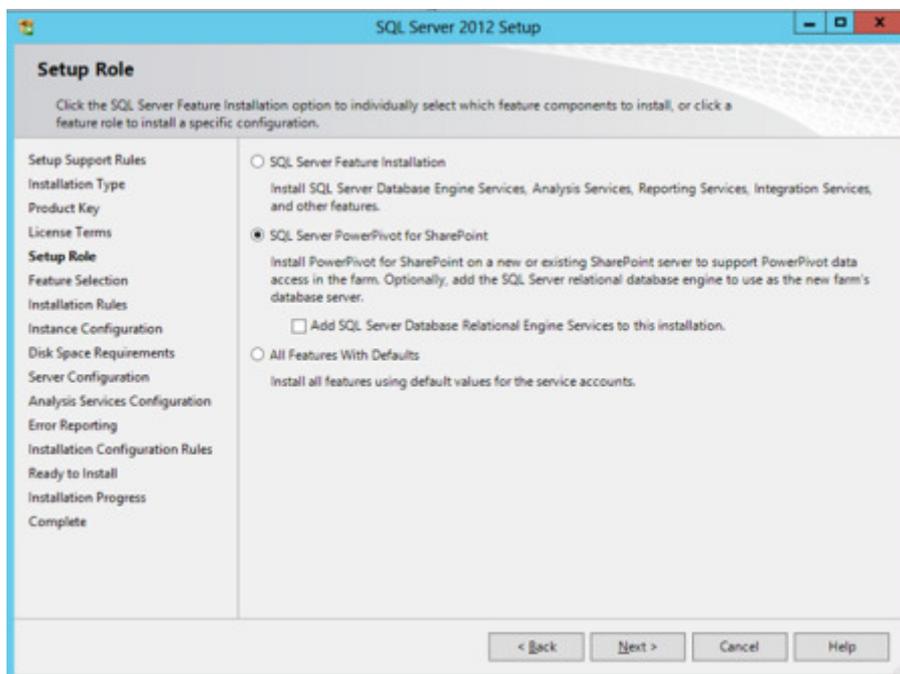
1. Start the SharePoint 2013 installer (autorun from the dvd/mounted ISO, or manually run default.hta). *If you havent already* installed the pre-requisites then do so using the option shown below:



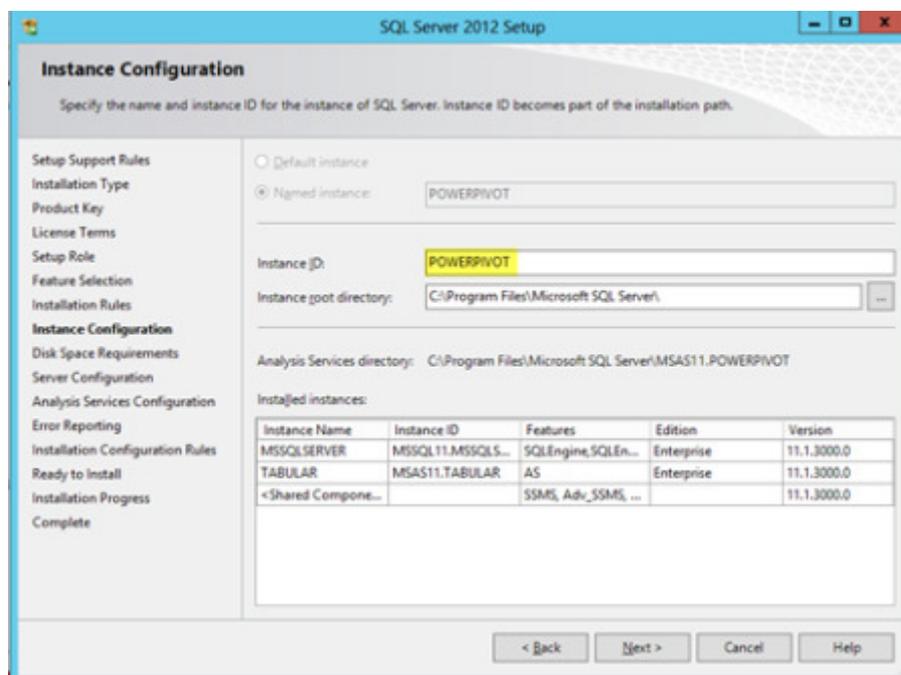
2. After pre-requisites are installed, from the install launcher shown above select the “Install SharePoint Server” option. When the SharePoint install has completed, De-select the option to run the configuration wizard.

### Installing PowerPivot for SharePoint

1. You will need to get back to the SQL Server 2012 installation media and run the SQL installer. Step through the usual installer screens and select the PowerPivot option when you are asked to choose the Role as shown below. Be sure to de-select the option to install the relational database (we will use the default instance we set up earlier to host SharePoint databases)



1. At the Instance Configuration screen of the wizard i entered POWERPIVOT for my Instance ID



- Step through the remainder of the installer, remembering to add yourself as an administrator when prompted. I also chose to run the service using the Administrator account.

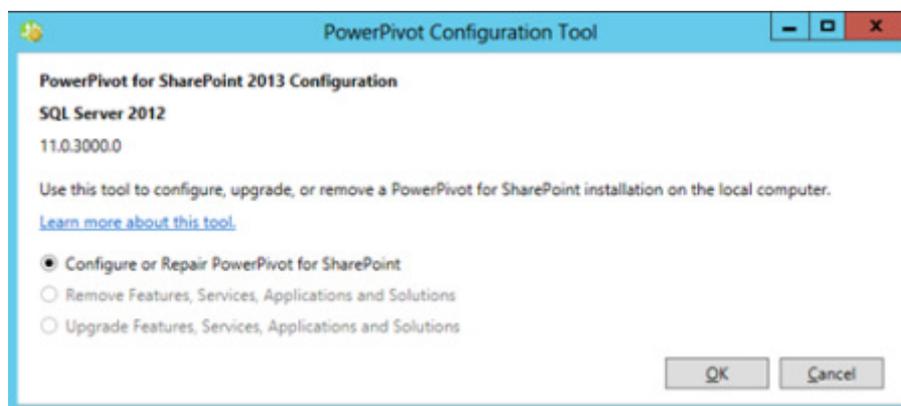
### **Base setup of up the Sharepoint Farm**

Instead of using the SharePoint Products Configuration Wizard to configure our farm, we use the new PowerPivot Configuration tool to do this for us. The beauty of it is it will setup a bare minimum Sharepoint farm. Nice and lean, so it will run quicker which is important for demos. I find the Microsoft supplied VMs HUGE (75Gb) and slow. [Check here](http://bhavikmerchant.wordpress.com/2012/04/14/converting-the-microsoft-integrated-sql-2012-demo-from-hyper-v-to-vmware-workstation/) (<http://bhavikmerchant.wordpress.com/2012/04/14/converting-the-microsoft-integrated-sql-2012-demo-from-hyper-v-to-vmware-workstation/>) for my guide on how to convert one from Hyper-V to VMWare. Just a reminder that [Windows 8 includes client Hyper-V](http://technet.microsoft.com/en-us/library/hh857623.aspx) (<http://technet.microsoft.com/en-us/library/hh857623.aspx>) which is great if you want to stay away from VMWare products for any reason.

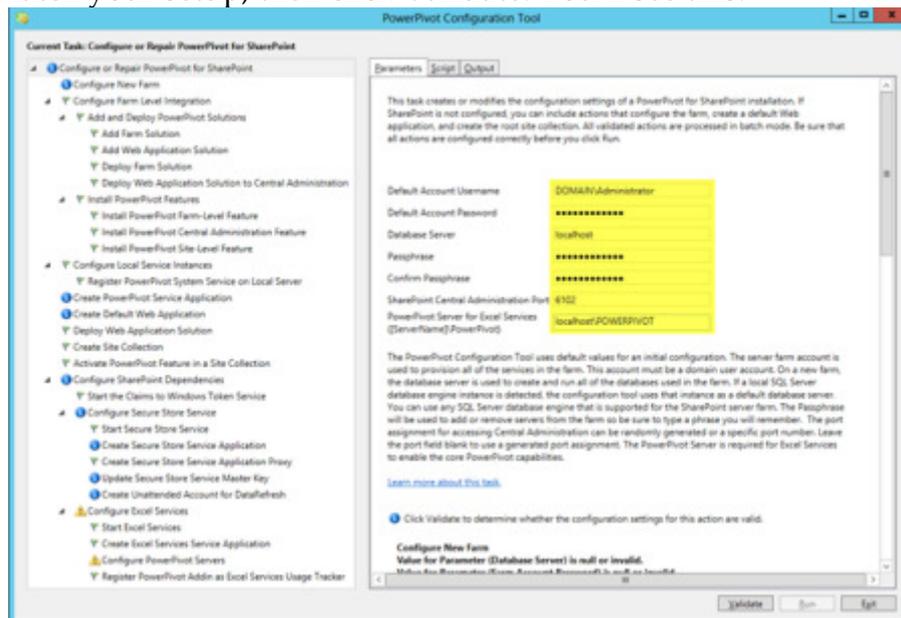
- Start the PowerPivot Configuration Tool from the SQL Server 2012 Program Group:



- Pick the first option as shown – there will be no choice on an unconfigured farm anyway



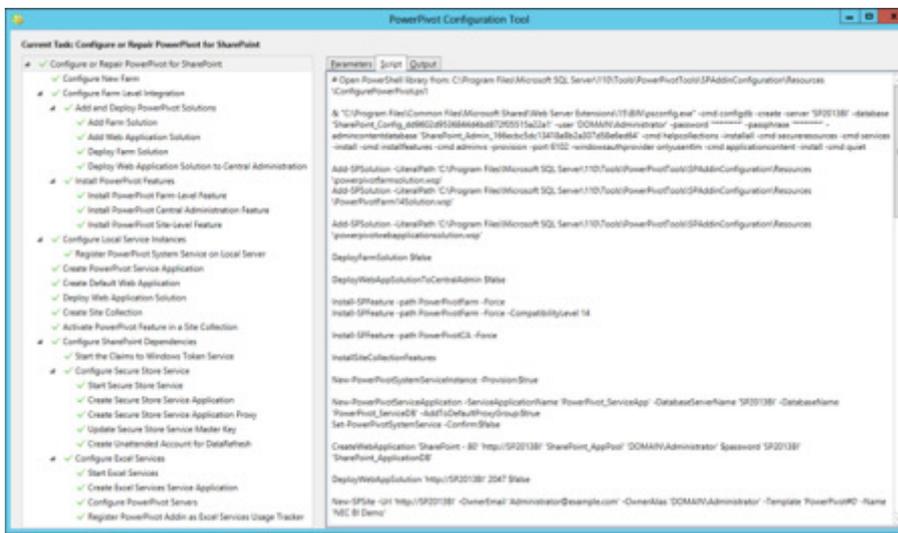
2. The tool will perform some checks and then present you with a list of tasks. Fill in the first screen with settings to match your setup, then click Validate. You'll see this:



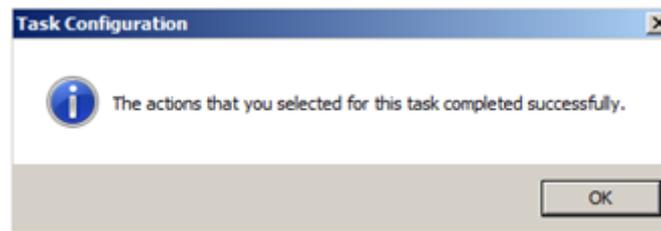
At this point you *could* press Run and have it perform all those activities. However, what's great about this setup compared to the SQL 2008 R2 is that we can actually customise *some* of Service Application and associated database names to remove the dreaded GUIDs that get appended to them. You can also control some of the IIS settings such as Web Application name and App Pool. These are the tasks I suggest click on to customise:

- ∞ Create PowerPivot Service Application
- ∞ Create Default Web Application
- ∞ Deploy Web Application
- ∞ Create Site Collection
- ∞ Create Unattended Account for DataRefresh

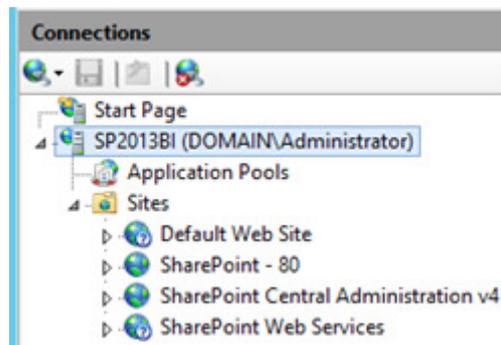
1. When you've finished customising you'll notice it informs you where you've made changes to the tasks. You can click the Script tab to view the PowerShell commands that will be executed. This is great to learn from and also to save away for a future scripted setup. I can see this being useful to tweak for a proper production setup. **Beware: you need to select the topmost task to get the whole script showing up.** This means if you are looking to find the PowerShell commands for just a select task/task group, you can click the appropriate section.



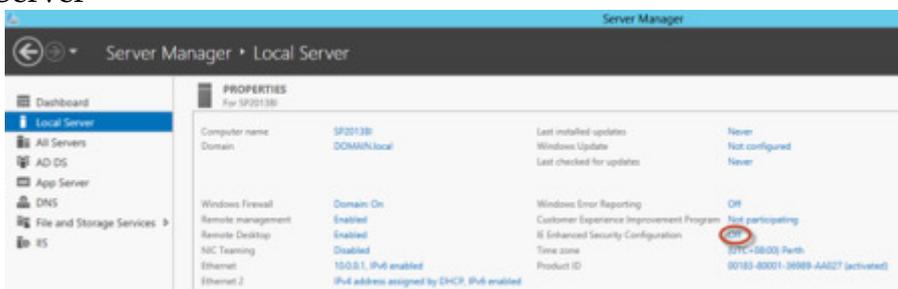
2. Click Validate, and if successful, click Run. This will take a bit of time... If all goes well you will see this:



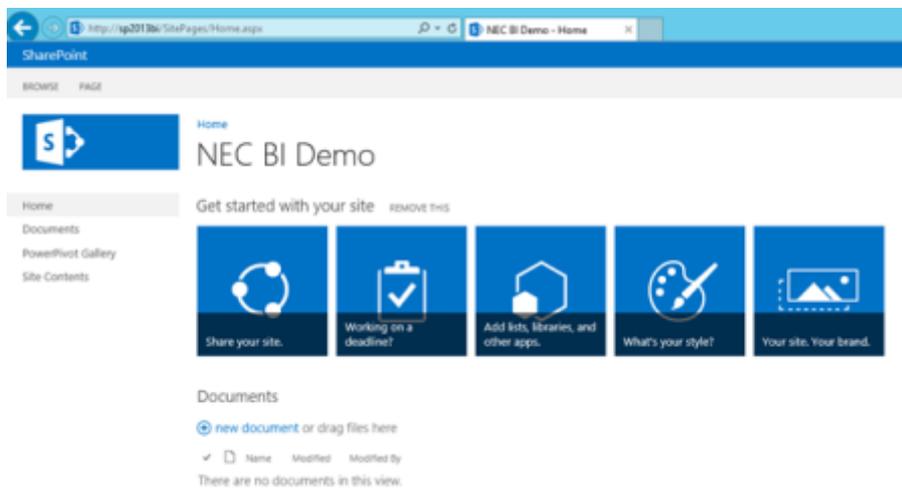
And you can verify in IIS that the sites were setup:



3. Navigate to our new SharePoint site. I usually at this point remember to turn off Internet Explorer Enhanced Security by going to Server Manager and selecting the appropriate option under Local Server



4. Finally I add the Sharepoint Site to the trusted sites in Internet Explorer.  
5. So now our site should work and if you browse to it you should see the vanilla Sharepoint page configured with a PowerPivot Gallery.

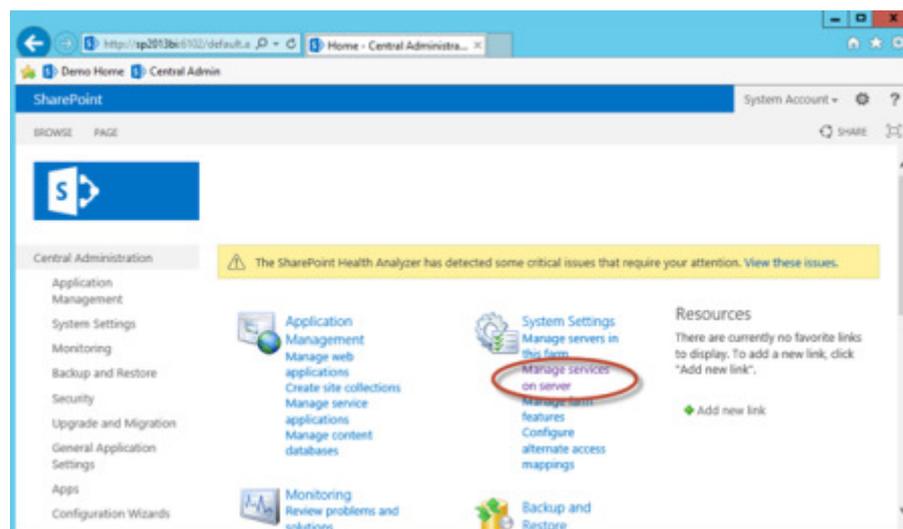


There are two more components left to configure for a basic BI setup. SSRS/Power View and then PerformancePoint.

### Enabling SharePoint Enterprise Features and Services

Before we try to setup these other BI services it's a good idea to turn on other services we might need for demos and also Enterprise features for the Site Collection and Site.

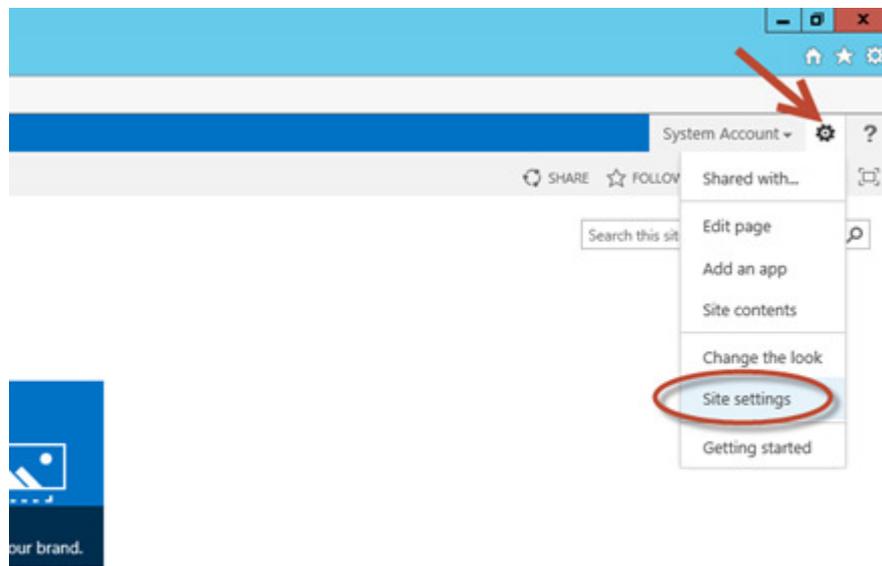
1. First in Central Admin select Manage Services on Server:



In there I enabled:

- ∞ Business Data Connectivity Service
- ∞ PerformancePoint Service
- ∞ Visio Graphics Service

1. Then in SharePoint go to Site Settings which is now accessed from a little “gear” icon in the top right corner of the browser window:



1. Then manage the Site Collection Features:

Site Collection Administration

- [Recycle bin](#)
- [Search Result Sources](#)
- [Search Result Types](#)
- [Search Query Rules](#)
- [Search Schema](#)
- [Search Settings](#)
- [Search Configuration Import](#)
- [Search Configuration Export](#)
- [Site collection features](#)
- [Site hierarchy](#)
- [Site collection audit settings](#)
- [Portal site connection](#)
- [Content Type Policy Templates](#)
- [Site collection app permissions](#)
- [Storage Metrics](#)
- [Content type publishing](#)
- [Help settings](#)
- [SharePoint Designer Settings](#)
- [HTML Field Security](#)
- [Site collection health checks](#)
- [Site collection upgrade](#)

Make sure the following are activated (Many will already be active, but ive stated all for completeness):

- ∞ PerformancePont Services Site Collection Features
- ∞ Power View Integration Feature
- ∞ PowerPivot Feature Integration for Site Collections
- ∞ SharePoint Server Enterprise Site Collection features
- ∞ SharePoint Server Publishing Infrastructure

1. Next we have to set enable some features at the Site level. So go back to Site Settings as in Step 1 above. This time select Manage Site Features:

Site Actions

- [Manage site features](#)
- [Enable search configuration export](#)
- [Site Collection Web Analytics reports](#)
- [Site Web Analytics reports](#)
- [Reset to site definition](#)
- [Delete this site](#)

Make sure the following are active:

- ☞ BICenter Data Connections Feature
- ☞ Mobile Browser View (if you plan to test mobile features)
- ☞ PerformancePont Services Site Features
- ☞ SharePoint Server Enterprise Site features
- ☞ SharePoint Server Publishing

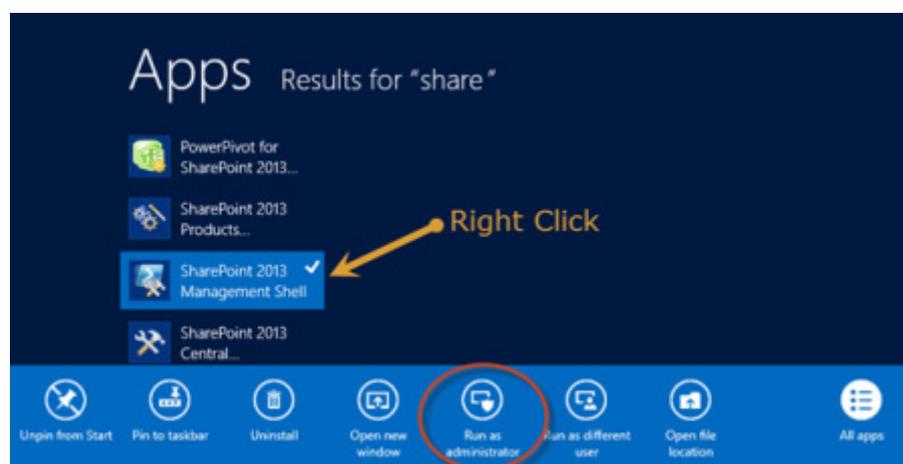
Note we don't turn on the Report Server File Sync because that is only required for SSRS 2008 R2 to synch back to the SSRS catalogue. In SQL 2012 SSRS is now a proper Sharepoint Service Application – something that was done to address performance problems in Sharepoint integrated mode (<http://sqlcat.com/sqlcat/b/technicalnotes/archive/2010/11/03/reporting-services-performance-in-sharepoint-integrated-mode-in-sql-server-2008-r2.aspx>).

### Configuring SSRS Integration and Power View

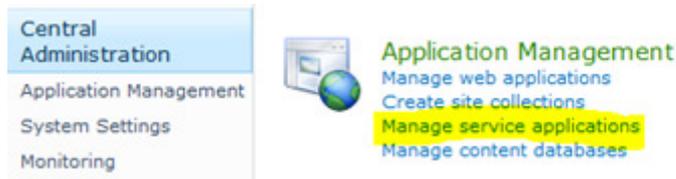
This part tripped me up the first time. I thought I had everything working but SSRS and reports simply wouldn't work! After some research, trial and error I understood all the components and config required to get it to work. The main problem is that with the steps we've taken so far, the SSRS Service itself (new in SQL 2012) and its proxy are not actually setup in Sharepoint yet.

A couple of quick PowerShell commands I found on the MSDN SSRS Integration Configuration Guide (<http://msdn.microsoft.com/en-us/library/gg492276.aspx>) helped here, in the section entitled "Install and Start the Reporting Services SharePoint Service". The note there explains why we don't have SSRS services configured – we installed SQL first so there was no farm for it to configure SSRS on.

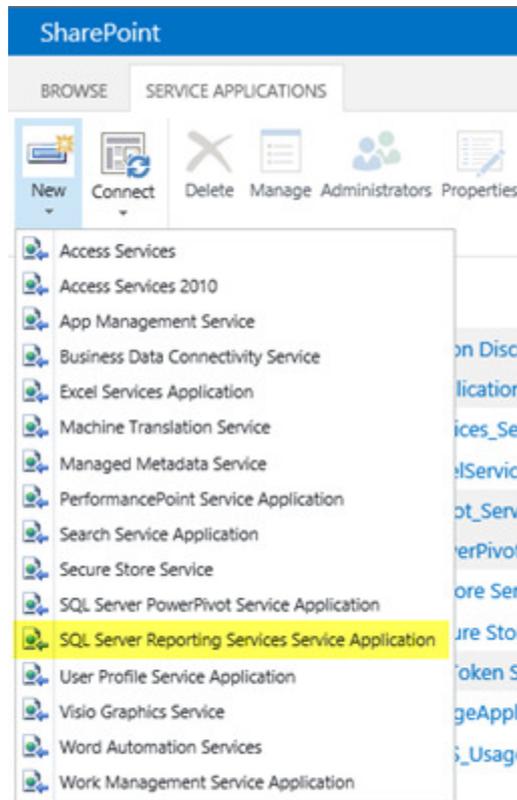
1. Run the Sharepoint PowerShell as an admin:



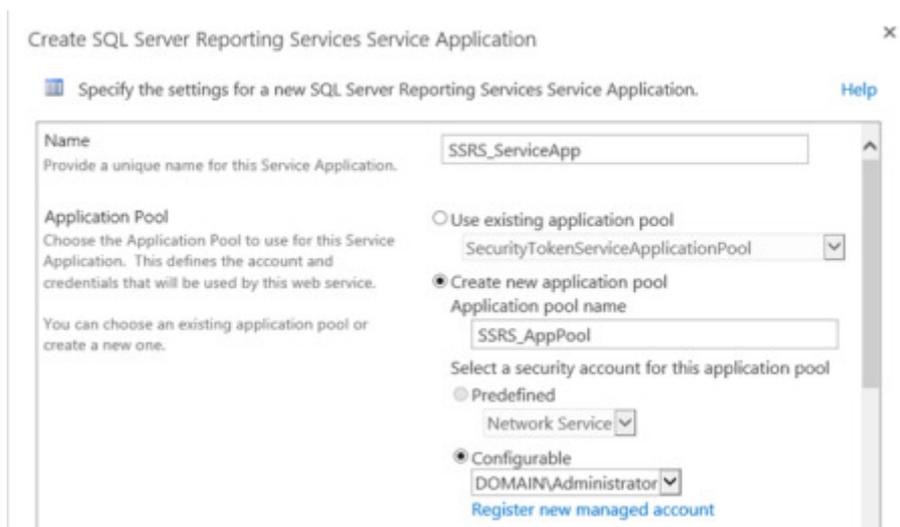
1. Run some PowerShell commands to install the service installed, add the proxy and then start the service:
  1. Install-SPRSService
  2. Install-SPRSServiceProxy
  3. `get-spserviceinstance -all |where {$_.TypeName -like "SQL Server Reporting*"}`
2. Next we can create a SSRS Service Application from Central Admin



Heres where we select the one we want:



The settings ive used are highlighted (first page):

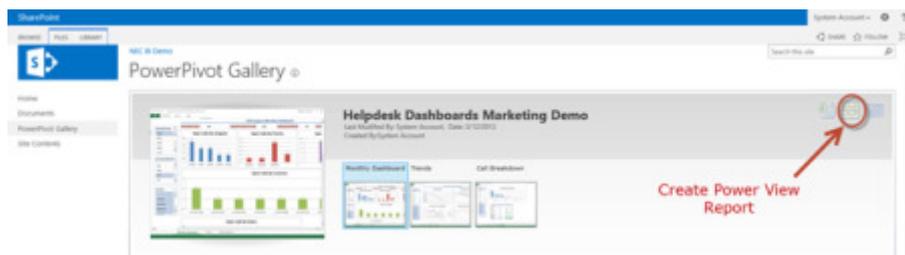


Then scrolling down:

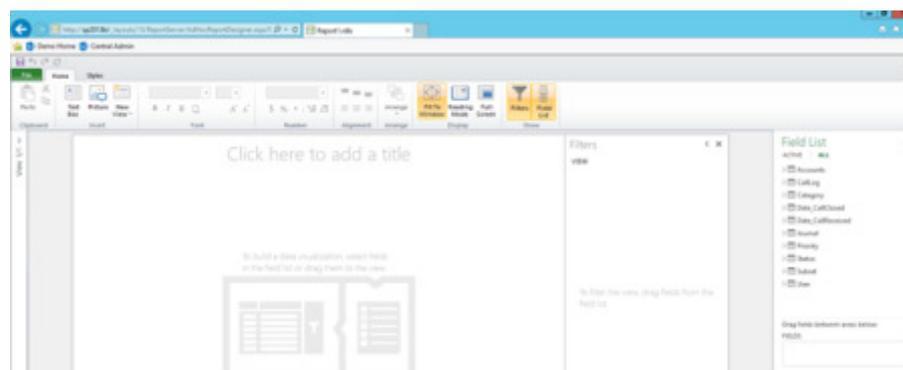
This should get you to the Success screen shown below – Click the highlighted link to ProvisionSubscriptions and Alerts.

This will take us to the page below where the SSRS service account is given access the SQL Agent. Note the credentials below are sent to the SQL server.. so using the “sa” account will work here too. Note that you can create subscription and alerts if the SQL Agent service is not running – so turn it on if you want to demo this feature.

- How to test this? Make sure you have Silverlight installed and then upload a PowerPivot workbook into the PowerPivot Gallery. The Gallery has a link automatically if Silverlight isn't present. You will see a new "Create Power View Report" option in the gallery:



Clicking that will open up the Power View designer:



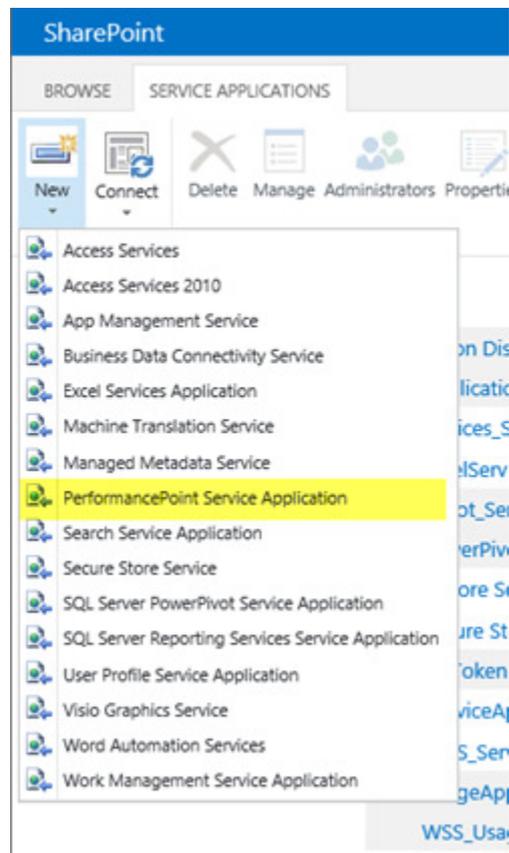
## Configuring PerformancePoint Services

This part is much easier!

- Verify the PerformancePoint Service is running. In Central Admin you need to go to "Manage Services on Server" (shown below), and once in there just Start the PerformancePoint Service if it isn't running.



- Next we need to add the PerformancePoint Service Application. So in Central Admin once again we go to the Manage Service Applications page as we did with the SSRS service.



1. Configure the service similar to how we did SSRS, shown below:

**New PerformancePoint Service Application** ✕

Specify settings for this service application. You can change these settings later from the Manage Service Applications page.

**Name:**  
Specify a name and default status for this service application.

The setting makes this service application available by default for web applications in this farm to use. Do not check this setting if you wish to specify manually which web applications should use this service application.

Use of the default database server and database name is recommended for most cases. Refer to the administrator's guide for advanced scenarios where specifying database information is required.

Use of Windows authentication is strongly recommended. To use SQL authentication, specify the credentials which will be used to connect to the database.

PerformancePoint\_ServiceApp

Add this service application's proxy to the farm's default proxy list.

Database Server: SP2013BI

Database Name: PerformancePoint\_Application\_DB

Database authentication:

Windows authentication (recommended)

SQL authentication

Account:

Password:

and continuing:

**New PerformancePoint Service Application**

**Failover Server**  
You can choose to associate a database with a specific failover server that is used in conjunction with SQL Server database mirroring.

**Failover Database Server**

**Secure Store and Unattended Service Account:**  
The Secure Store Service is used to store the Unattended Service Account used for authenticating to data sources. The Unattended Service Account is set after configuring the PerformancePoint Service application. The setting is located in "Manage service applications" in SharePoint Central Administration under the PerformancePoint Services management page. A running Secure Store Service Application and Proxy are required.

**Application Pool**  
Choose the Application Pool to use for this Service Application. This defines the account and credentials that will be used by this web service. You can choose an existing application pool or create a new one.

Use existing application pool  
SecurityTokenServiceApplicationPool

Create new application pool  
Application pool name: PerformancePoint\_AppPool

Select a security account for this application pool

Predefined  
Network Service

Configurable  
DOMAIN\Administrator  
[Register new managed account](#)

Buttons: Create, Cancel

If all goes well you should see this:

**New PerformancePoint Service Application**

**The PerformancePoint Service application was successfully created.**

- The PerformancePoint Service Application, PerformancePoint\_ServiceApp, and the associated service proxy were successfully created. The settings for this service can be modified through manage service applications in SharePoint Central Administration.
- The service will use application pool, PerformancePoint\_AppPool, to run instances of the service application, PerformancePoint\_ServiceApp.
- PerformancePoint\_AppPool is running under the DOMAIN\Administrator account.

**Additional configuration steps:**

- To access data sources using shared credentials instead of per-user identity, configure the PerformancePoint Unattended Service Account and/or create Target Applications in the Secure Store using SharePoint Central Administration.
- The PerformancePoint web application and site collection features must be enabled in order to use the PerformancePoint capabilities.
- Trusted locations should be configured through manage service applications to prevent use of any site collection or sites, with PPS features enabled, that could allow for unauthorized access to data sources.

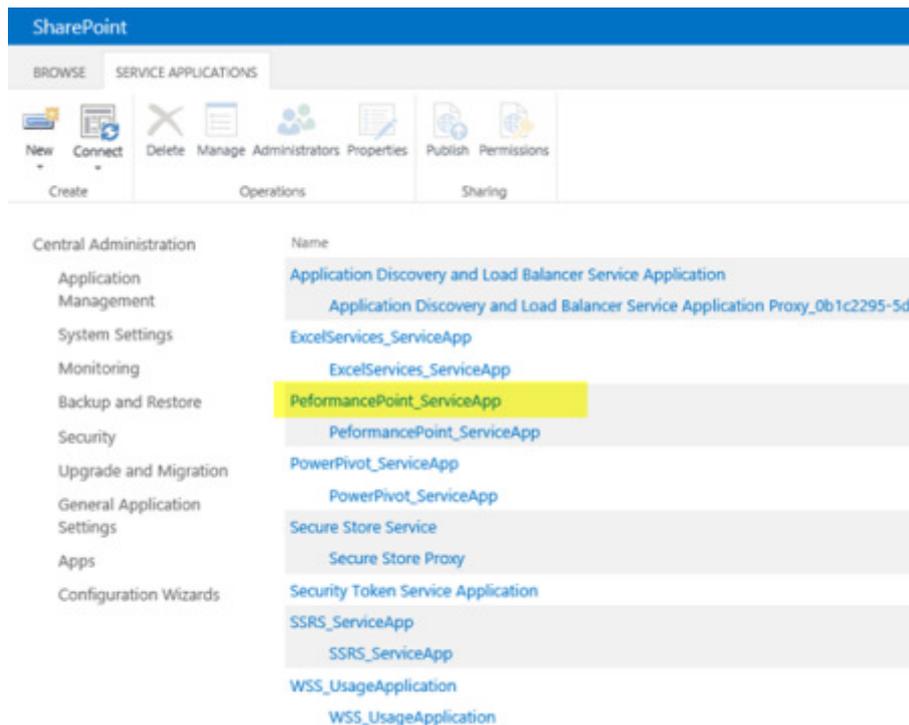
**Warning: Connecting to Microsoft SQL Server Analysis Services:**

- In order for PerformancePoint to connect to Analysis Services data sources (including PowerPivot workbooks), you may need to install the PowerPivot for SharePoint installation package (2013 or later). This package must be installed on every farm server that runs the PerformancePoint service.

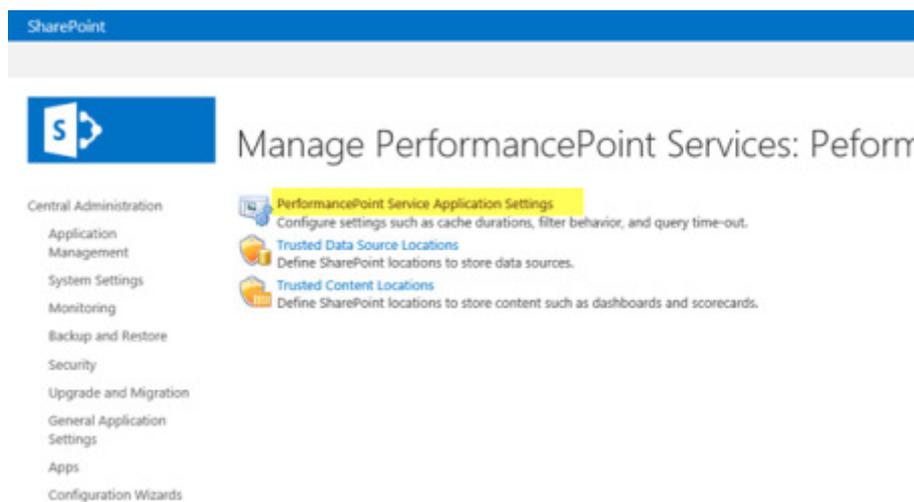
Buttons: OK, Cancel

- Next we need to configure the Unattended Service Account – its not absolutely necessary but a good idea to get done now. It all depends on how you plan to access your data sources –

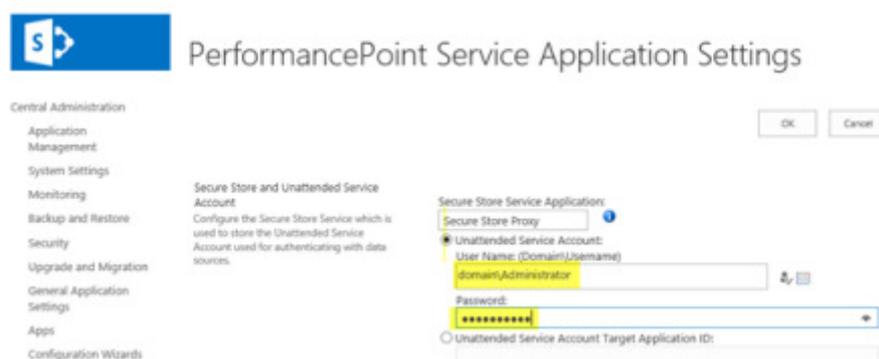
Unattended gives you the highest level of caching in PPS and the data source. So go back Manage Service Applications in Central Admin, and click on the newly provisioned service:



Once there, you need to change the app settings:



1. We only need to set the account, the other settings on the page can be left at their defaults



1. Its worth noting that there is a new option available in SharePoint 2013 for PPS which is to use the EffectiveUserName property in the connection string to SSAS:

**Analysis Services EffectiveUserName**  
The Analysis Services EffectiveUserName capability is an alternative to Windows delegation for allowing individual users to securely access Analysis Services data.

Use the EffectiveUserName connection string property instead of Windows delegation.

This setting applies to all Analysis Services data sources configured for per-user authentication. When this setting is enabled, all connections to Analysis Services data for individual users will be made using the EffectiveUserName connection string property instead of Windows delegation.



Using EffectiveUserName requires that the PerformancePoint Services managed account is an administrator in Analysis Services. EffectiveUserName is not supported for http(s) or PowerPivot.

## Client Tools

Since we haven't used the Business Intelligence Centre site template there will be no shortcut button to get to Dashboard designer. This can be reached via the following URL, which you can add to the site menu, a page or even customise via Sharepoint Designer (<http://www.sharepointoscar.com/blog/2011/10/25/adding-performancepoint-dashboard-designer-shortcut-to-a-page/>):

[http://<siteURL \(http://<siteURL\) >/\\_layouts/ppswbparts/designerredirect.aspx](http://<siteURL (http://<siteURL) >/_layouts/ppswbparts/designerredirect.aspx)

Just a few things left now to make this complete. You will need to install:

1. Excel 2013 to demonstrate the now built PowerPivot and Power View
2. Excel 2010 and then the items below (I have not found any addins compatible with Excel 2013 yet!)
  1. Data Mining Addins
  2. Master Data Services Addin
3. Visio 2013
4. PowerPoint 2013 (to show export capability from Power View)

## Samples

Now install the SQL Server samples (<http://msftdbprodsamples.codeplex.com/releases/view/55330>) if you need some data to play around with.

About these ads (<http://en.wordpress.com/about-these-ads/>)



This entry was posted on Saturday, December 8th, 2012 at 5:37 AM and is filed under PerformancePoint, Power View, SharePoint 2013, SQL 2012. You can follow any responses to this entry through the RSS 2.0 feed. You can leave a response, or trackback from your own site.

## One Response to *Setting Up a SharePoint 2013 BI Demo VM*

**Brad Saide says:**

March 19, 2013 at 9:44 AM

The tools for 2013 were released on the 9th of January and can all be found here...

<http://www.microsoft.com/en-au/download/details.aspx?id=35580>

**Reply**

Theme: Contempt by Vault9.

Blog at WordPress.com.