Meerkat Deployment

Table of Contents

[1. Pre-requisites 3](#_Toc386118764)

[1.1 Required Roles and Services 3](#_Toc386118765)

[1.2 Required Software 3](#_Toc386118766)

[1.3 Required Accounts 3](#_Toc386118767)

[2. Preparing the Environment 4](#_Toc386118768)

[2.1 Active Directory 4](#_Toc386118769)

[a) SQL Service Account 4](#_Toc386118770)

[b) SharePoint Farm Account 4](#_Toc386118771)

[c) SharePoint Installation/Admin Account 4](#_Toc386118772)

[d) Web Application Account 4](#_Toc386118773)

[e) Reporting Service Account 4](#_Toc386118774)

[2.2 SQL Server Installation and Setup 4](#_Toc386118775)

[2.3 SharePoint Configuration 4](#_Toc386118776)

[3. Meerkat Installation 6](#_Toc386118777)

[3.1 Meerkat DB 6](#_Toc386118778)

[a) SQL 6](#_Toc386118779)

[b) Visual Studio 6](#_Toc386118780)

[3.2 Mega Menu 7](#_Toc386118781)

[a) Web Deploy 7](#_Toc386118782)

[b) MVC2 9](#_Toc386118783)

[c) IIS 10](#_Toc386118784)

[d) Azure End Points 12](#_Toc386118785)

[3.3 News Aggregator 12](#_Toc386118786)

[a) News Aggregator Configuration 15](#_Toc386118787)

[b) Sample snippet 17](#_Toc386118788)

[3.4 Lightswitch Apps 18](#_Toc386118789)

[a) Meerkat Admin 18](#_Toc386118790)

[b) Meerkat Capture 18](#_Toc386118791)

[c) Configure LightSwitch Apps 18](#_Toc386118792)

[d) Deploy the LightSwitch apps 19](#_Toc386118793)

[3.5 Meerkat SharePoint Site 20](#_Toc386118794)

[a) Site Import 20](#_Toc386118795)

[b) MegaDropDown 20](#_Toc386118796)

[c) Light Switch 20](#_Toc386118797)

[d) SharePoint Designer – master page edits 21](#_Toc386118798)

[e) Filter Web part 22](#_Toc386118799)

[3.6 Reports Deployment 23](#_Toc386118800)

[a) SQL Server Reporting Services Deploy 23](#_Toc386118801)

# Pre-requisites

This section covers the various hardware and software prerequisites which must be in the environment before you begin a Meerkat install.

## 1.1 Required Roles and Services

The following roles must be set up in the environment before you can begin installation.

* IIS Role – This role must be enabled in the server which will host SharePoint. Make sure it’s enabled. Since we’re using an Azure Virtual machine based on a SharePoint gallery image, it should already be enabled since the machine already has the SharePoint software preinstalled.
* Active Directory and Domain Naming Services. (AD & DNS) – In a production environment, best practice is to host this on a different server other than the one hosting SharePoint.

## 1.2 Required Software

The following software are required before you can start install

* SharePoint 2013 – Latest patch (Current is December 2013)
* SQL Server 2012 SP1 – Latest patch is CU8
* MVC2 for ASP.Net
* Microsoft Web Deploy
* GitHub
* GitExtensions
* Meerkat installation bits from Github at <https://github.com/AphelionSoftware/Meerkat.git>

## 1.3 Required Accounts

In keeping with the Microsoft best practices it is recommended to have a Service account per item of functionality provided

SQL

* SQL Service

SharePoint

* Farm account
* Installation account
* Web application account
* Reporting Services account

# Preparing the Environment

This section covers the steps needed to prepare the environment for a Meerkat installation. This assumes that in a perfect production environment, SQL, Active Directory and SharePoint are on different servers

## 2.1 Active Directory

In a Meerkat production environment, users will be managed from Active Directory and therefore if there is no active directory setup, it’s wise to set up one before you proceed. If this is your first time setting up Active Directory, follow this link; <https://social.technet.microsoft.com/wiki/contents/articles/12370.step-by-step-guide-for-setting-up-a-windows-server-2012-domain-controller.aspx>

Once set up, the following accounts need to be created:

### SQL Service Account

This account is used to set up SQL for SharePoint and is therefore granted the highest rights on the database. During SQL installation, it must be made the identity of both the MSSSQLSERVER and SQLSERVERAGENT services. It may not necessarily require admin rights on any server or even the domain. The rights it needs will be assigned during SQL Server setup. Call this account

* DomainName\Meerkat\_SQL

### SharePoint Farm Account

This account is essentially the owner of the SharePoint environment and has admin rights over the entire farm whether services, sites or databases. Due to its need for control over multiple areas of the environment, the account should be granted local admin rights on every server in the farm during installation. All other rights will be granted by SQL during set up. Call this account

* DomainName\Meerkat\_SPFarm

### SharePoint Installation/Admin Account

This is the account used to install SharePoint and carry out specific admin functions on it thereafter. It is the owner of all the software bits for SharePoint and therefore any modification to certain elements of the software require this account and would fail if you use a different account. The account obviously needs local admin rights on the SharePoint server but must also be granted SecurityAdmin and dbcreator server roles on SQL server. All other permissions are granted once you run the configuration wizard. Call this account

* DomainName\Meerkat\_SPInstall

### Web Application Account

A simple domain user account will suffice for this account since it doesn’t require any special privileges. It will be the owner of the web application and can also be set up as the owner of the service applications as well. However, before doing so, it must be registered as a managed service account in SharePoint first. Call this account

* DomainName\Meerkat\_SPApp

### Reporting Service Account

This is the account that will be used to manage the reporting services service application in SharePoint. It also doesn’t need any special permissions and should therefore a normal domain user account. However, just like the web application account, it must be registered as a managed service account. Call this account

* DomainName\Meerkat\_SSRS

## 2.2 SQL Server Installation and Setup

An instance of SQL Server 2012 needs to be set up before you continue. The easiest way to set it up is using a script, which performs an unattended install of SQL Server and configures the Database engine services. A sample script that installs Database engine services is provided here:

<http://blogs.technet.com/b/ilikesql_by_dandyman/archive/2013/03/10/how-to-automate-the-installation-of-sql-server-ssdt-office-2013-and-visual-studio-2012.aspx>

However, after this, you need to run the installer again to install PowerPivot and reporting services in integrated mode. Make sure to add the RS add in from the installer as well.

## 2.3 SharePoint Configuration

This involves the final configuration of SharePoint server. SharePoint will already be installed on the SharePoint virtual machine from the gallery image so all you need is to run the configuration to create the databases in SQL.

Before running configuration, please make sure to check that the database engine you will be using is set to a maximum degree of parallelism of 1. Please refer to the PowerPoint story board called “Aphelion.Meerkat.Max Degree of Parallelism.v001” for a detailed description of how this is done.

Once all checks out, run the SharePoint configuration wizard using the PowerShell “PSConfig.ps1” script and then create a new web application on port 80 with a host header entry of “Meerkat”, ensuring that a descriptive database name is chosen e.g. “SharePoint\_Content\_Meerkat”. In the options section, create a new Application Pool for it, running under its own account and leave the authentication as NTLM so that we can use Active Directory for authentication. All other settings for the web application can be left as defaults.

Set up the service applications using the “Services.ps1” script. However, after this you may need to run the different PowerShell cmdlets for installing Reporting Services and Power Pivot services. The following services applications must be set up:

* SQL Reporting Services
* State Service
* Usage Service
* Secure Store Service
* Security Token Service
* Managed Metadata Service
* Reporting Services
* Power Pivot Services

**NOTE:** Both PSConfig.ps1 and Services.ps1 are in the Meerkat Github repo under the “SharePoint Site” folder.

# Meerkat Installation

This section is about the installation of the different bits of Meerkat, which are as follows:

1. Meerkat DB
2. Mega Menu
3. News Aggregator
4. Meerkat Capture
5. Meerkat Admin
6. Meerkat SharePoint Site

## Meerkat DB

### SQL

The Kenyan shape files database is also required for this section. It’s referenced in the SQL script that is run below so you need to restore the DB to SQL before you begin. The backup can be found in the Meerkat Github repository under the SQL Solution folder. It’s called “KenyaShapes.bak”

The Meerkat database is the database that hosts all data that Meerkat generates and it’s also the Database used for reporting. It’s as essential to the solution as the Config database is to SharePoint. To set this up, you’ll need to log in to the database server and get the SQLScripts folder from the Github repo for Meerkat. Once done, navigate to the SQL Solution folder and run the Post Deployment Script.

### Visual Studio

You can also do this from within Visual Studio, using the solution found under SQL Solution in the Meerkat repository in Github. Once open in Visual Studio, right click on the project and select Publish. You may need to allow remote connections and create a SQL login to allow for publishing.

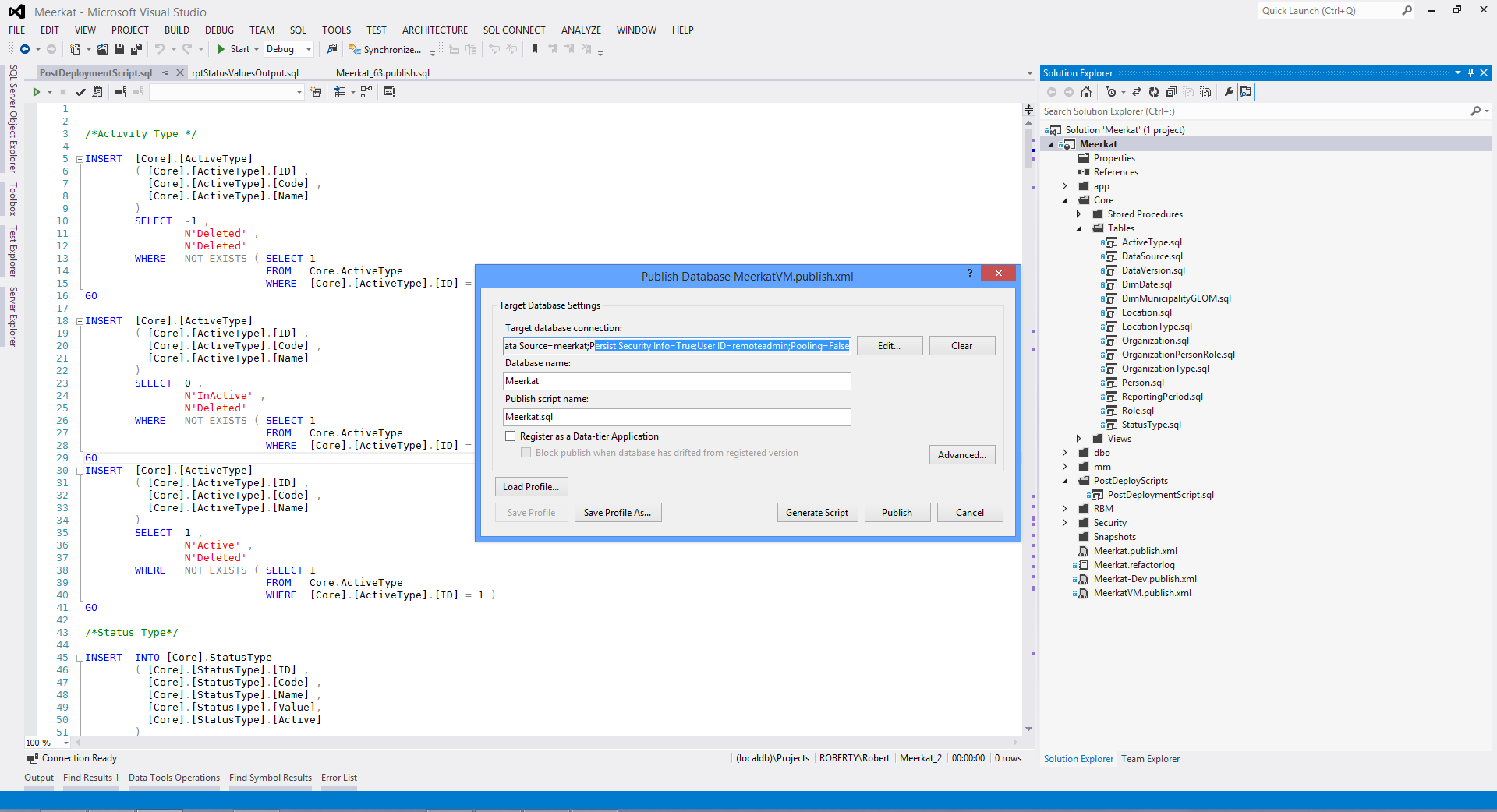


Figure 8: Example of publishing from Visual Studio.

If you are updating an existing database, you want to make sure in the Advanced Publish Settings you select “Always re-create database” and “Back up database before deployment” – this will however kill all the data.

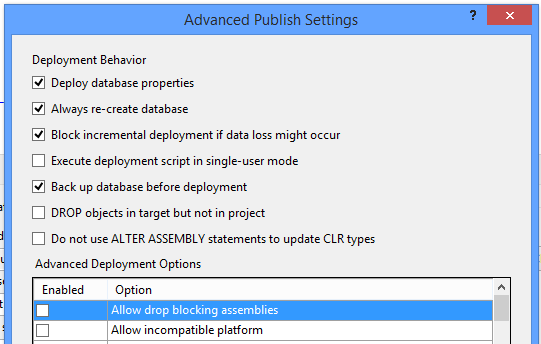


Figure 9: Example of advanced publish settings.

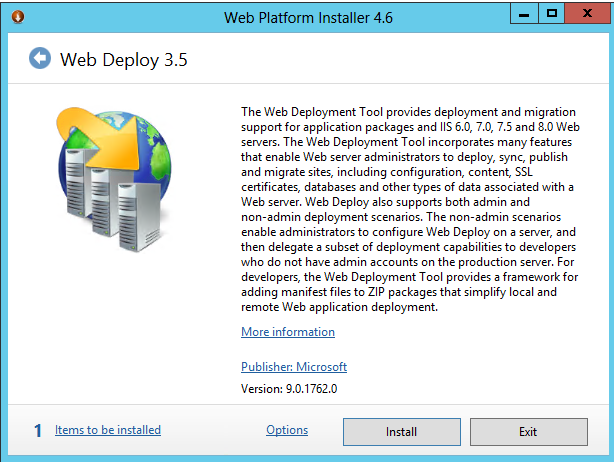
## Mega Menu

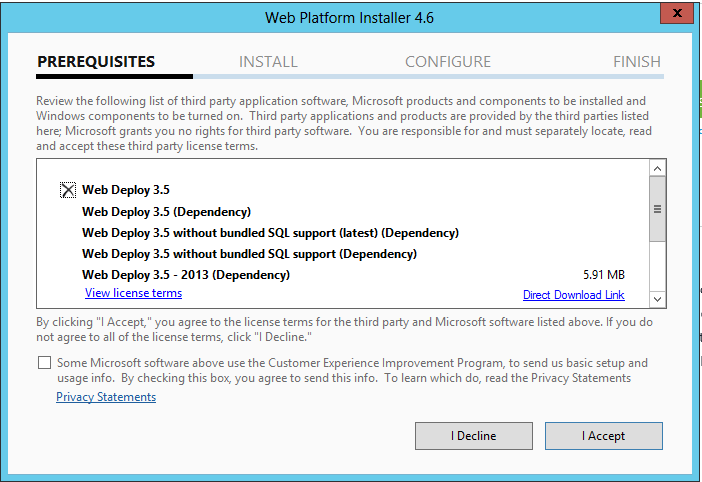
The Mega Menu is set up to enable easy navigation between the sites in the solution. It is configured from different points.

### Web Deploy

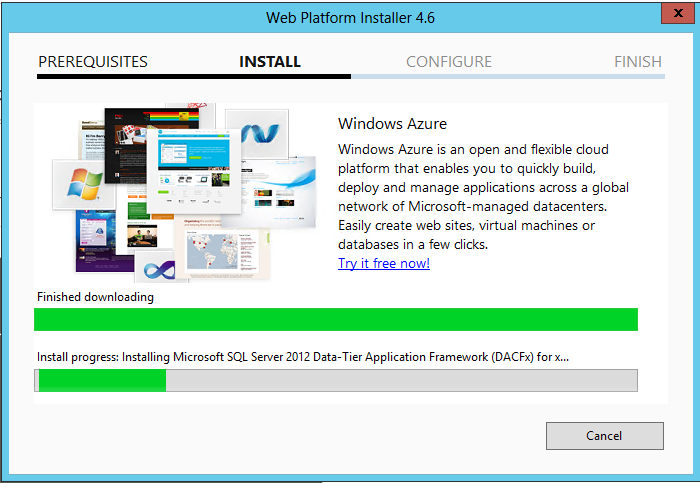
Microsoft web deploy is used to easily deploy packages to IIS. Download the application from <http://www.iis.net/downloads/microsoft/web-deploy> and run the application to set it up.

Click ‘Install’ to install the Web Deploy package

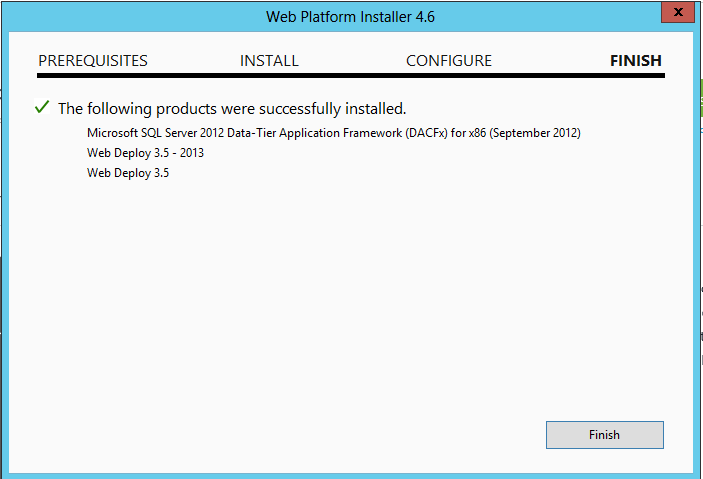




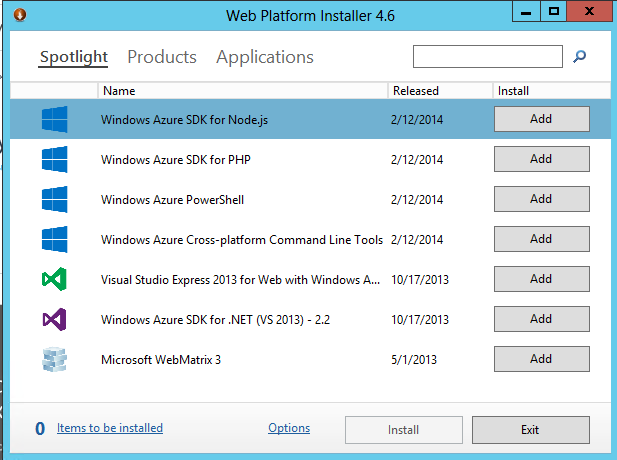
Allow the installation to complete.



Completion screen will be displayed.



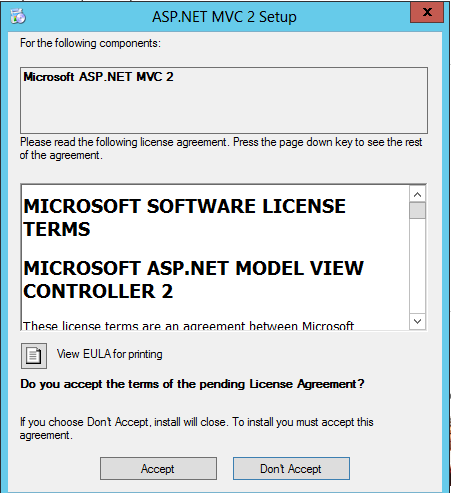
Click Exit on the additional items list.



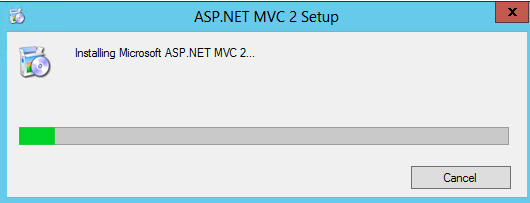
### MVC2

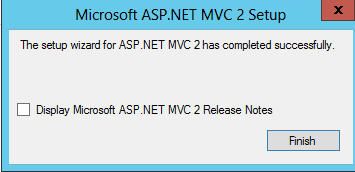
MVC2 is required for the MegaMenu and it can be downloaded from the Microsoft site <http://www.microsoft.com/en-us/download/details.aspx?id=22079>

Once downloaded from the site, run the installer and when prompted, click accept to continue.



Allow to install, and click finish.

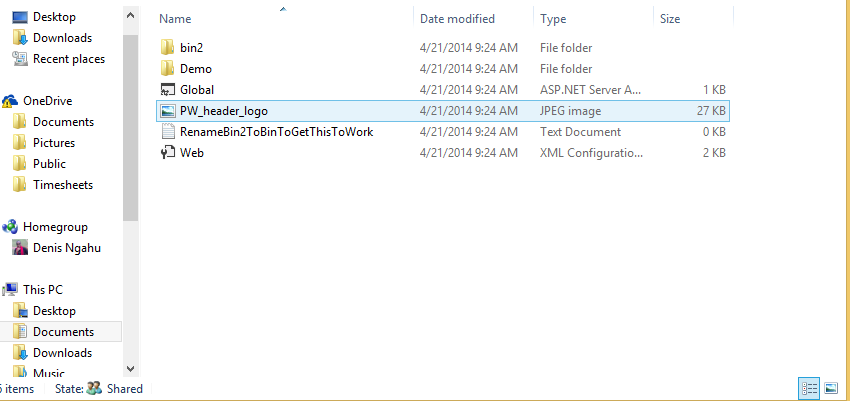




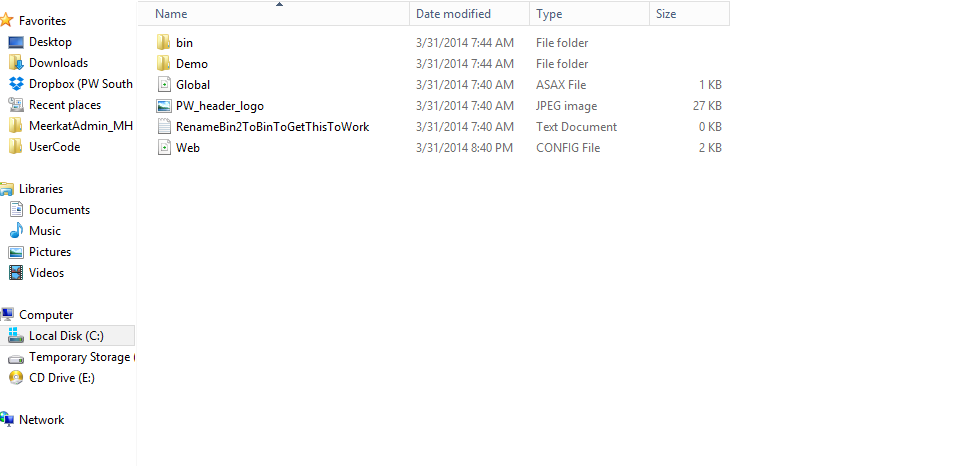
### IIS

In Github there is a folder called Megamenu. In it there is another folder with the same name. Copy it to the following path **C:\inetpub\wwwroot\MegaMenu.**

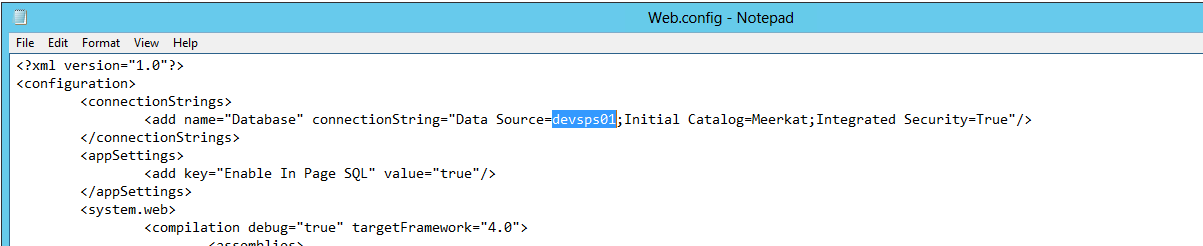
Once you copy the files, you’ll notice that the top most folder is called bin2:



Change it to bin



Also, open the Web.config file and edit it to point to the correct SQL server instance by changing the value of the Data Source as shown below.



The rest of the changes need to be made in IIS. This can be done by opening Internet information Services manager and following the “Aphelion.Meerkat.Create and Configure IIS Application.v001” storyboard. The following settings need to be changed though:

* Application pool Name – MegaMenu
* Site Name – MegaMenu
* Identity – Use a domain account of your choice in your environment that has access to the Meerkat DB.
* Physical path – C:\inetpub\wwwroot\MegaMenu.
* Host Name – Name of your server.
* Port for site binding – Port 8082

### Azure End Points

For the MegaMenu to work, the endpoints must be configured properly to point to the port that the IIS site will be bound to. To proceed with this configuration, login to the Windows Azure portal at <https://Manage.windowsazure.com>/@mgstaceyyahoo.onmicrosoft.com then navigate to the Virtual Machines section. Follow the “Aphelion.Meerkat.Add an Azure Endpoint.v001” storyboard guide to complete the steps required for this section. The specific settings will be as follows:

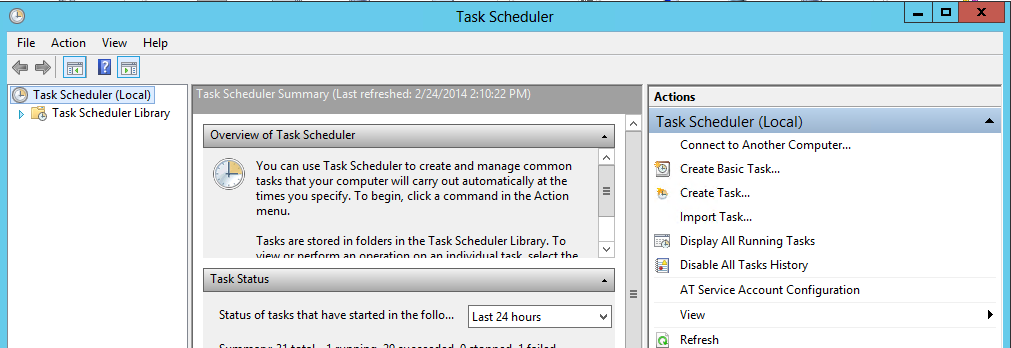
* Virtual machine Name – Name of your virtual machine.
* Name of endpoint – MegaMenu
* Private & public ports – Port 8082

## News Aggregator

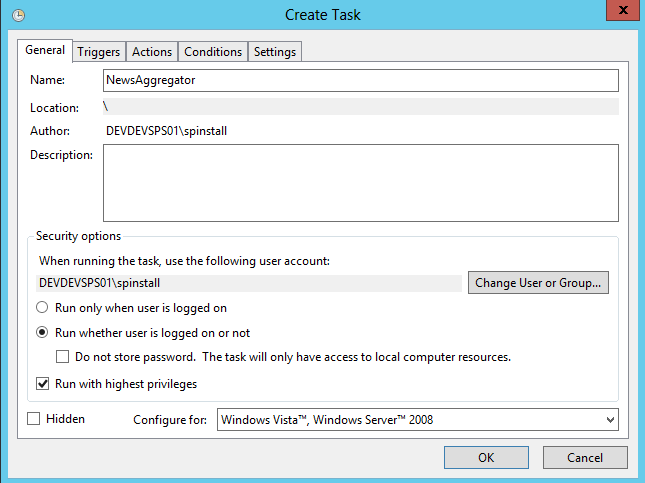
On Git there is a folder called NewsAgg. Found under \GitHub\Meerkat\NewsAgg

Within this folder there is an executable called ETL.Runner.Console, this needs to be setup to run on a schedule.  
A Windows task schedule to run once daily is fine.

Open Windows Task Scheduler, and **Create Task**

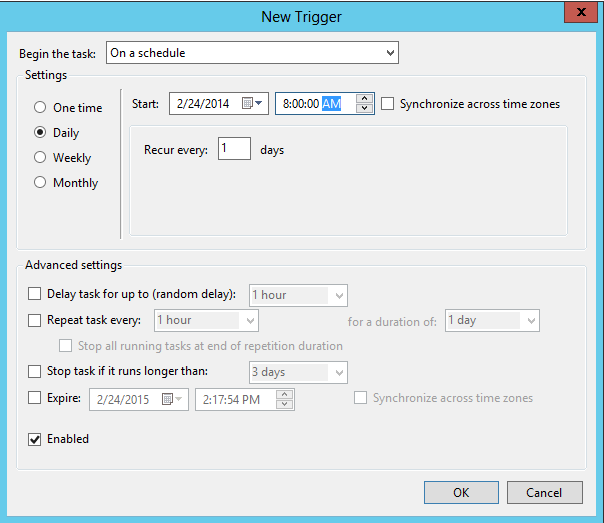


Fill in the information for the task, Set the task to **Run whether user is logged on or not** also select the **Run with highest privileges**

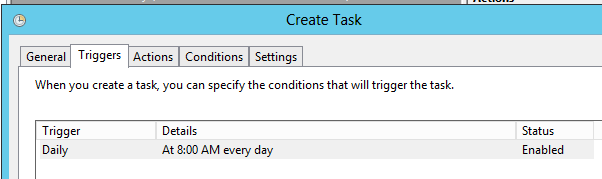


The account being used should have permissions to SharePoint and to be able to create files on the file system.

On the **Triggers** tab, create a new trigger  
Set the task to run daily and a pre-set time.

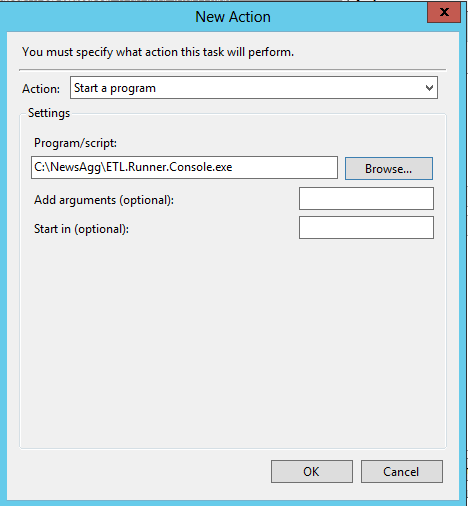


Ensure the trigger is enabled



Under the **Actions** tab, create a new action.

In this action select the ETL.Runner.Console.exe application.

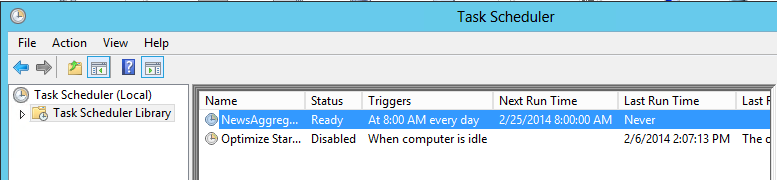


Click OK when complete.

You will need to enter the credentials for the account the News Aggregator runs.

Remember the account also needs to be able to create content on SharePoint.

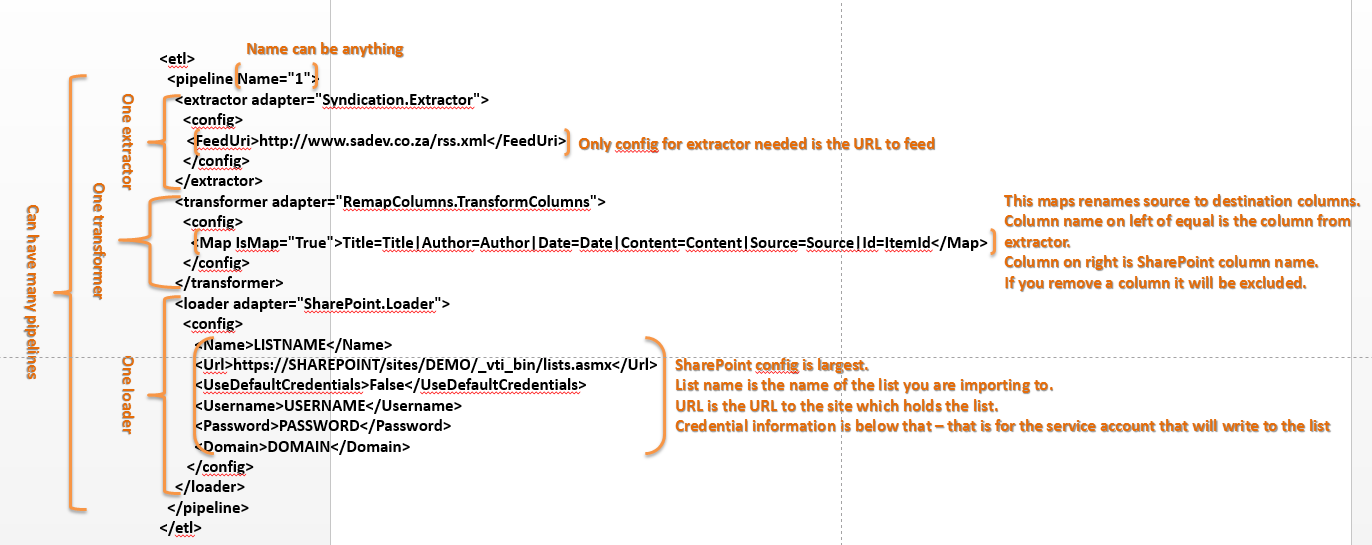
The finished task is created below



### News Aggregator Configuration

There is a config file called ETL.Config, this file is a sequence of repeating nodes telling what information to download and where to go to.

Edit this as required for each news section per outcome



### Sample snippet

<pipeline Name="OutcomeHome">  
 <extractor adapter="Syndication.Extractor">  
 <config> <FeedUri><![CDATA[https://news.google.com/news/feeds?hl=en&gl=za&as\_epq&as\_oq&as\_eq&as\_scoring=r&as\_drrb=q&as\_qdr=a&as\_nsrc&as\_nloc=Kenya&as\_author&as\_occt=any&q=Millennium+development+goals+location:kenya&um=1&ie=UTF-8&output=rss]]></FeedUri>  
 </config>  
 </extractor>  
 <transformer adapter="UniqueRows.TransformAdapter">  
 <config>  
 <UniqueColumnName>UniqueId</UniqueColumnName>  
 <Filename>C:\NewsAgg\syndicationRun0.sqlite</Filename>  
 </config>  
 </transformer>

<transformer adapter="RemapColumns.TransformColumns">  
 <config>  
 <Map IsMap="True">Title=Title|Author=Author|Date=Date|Content=Content|Source=Source|Id=ItemId|UniqueId=UniqueId</Map>  
 </config>  
 </transformer>  
 <loader adapter="SharePoint.Loader">  
 <config>  
 <Name>ExternalNews</Name>

<Url>http://meerkat01/\_vti\_bin/lists.asmx</Url>

<UseDefaultCredentials>True</UseDefaultCredentials>

<Username>UserName</Username>

<Password>Password</Password>

<Domain>Domain</Domain>

<ApplyGoogleImgSrcHackTo>Content</ApplyGoogleImgSrcHackTo>

</config>

</loader>

</pipeline>

## Lightswitch Apps

### Meerkat Admin

Meerkat Admin is a LightSwitch app that is used to collect data and store it in the Meerkat Database. It’s deployed in the same way as the MegaMenu where you create an application pool and site in IIS then add azure endpoints. To do this, follow the “Aphelion.Meerkat.Create and Configure IIS Application.v001” and “Aphelion.Meerkat.Add an Azure Endpoint.v001” storyboard guides accompanying this documentation. However, the following will need to change:

IIS:

* Application pool Name – MeerkatAdmin
* Site Name – MeerkatAdmin
* Identity – Use a domain account of your choice in your environment that has access to the Meerkat DB.
* Physical path – C:\inetpub\wwwroot\ MeerkatAdmin.
* Host Name – Name of your web server.
* Port for site binding – Port 300

Azure Endpoint:

* Virtual machine Name – Name of your virtual machine.
* Name of endpoint – MeerkatAdmin
* Private & public ports – Port 300

### Meerkat Capture

Meerkat Capture is a LightSwitch app that is used to collect data and store it in the Meerkat Database. It’s deployed in the same way as Meerkat Admin where you create an application pool and site in IIS then add azure endpoints. To do this, follow the “Aphelion.Meerkat.Create and Configure IIS Application.v001” and “Aphelion.Meerkat.Add an Azure Endpoint.v001” storyboard guides accompanying this documentation. However, the following will need to change:

IIS:

* Application pool Name – MeerkatCapture
* Site Name – MeerkatCapture
* Identity – Use a domain account of your choice in your environment that has access to the Meerkat DB.
* Physical path – C:\inetpub\wwwroot\ MeerkatCapture.
* Host Name – Name of your web server.
* Port for site binding – Port 330

Azure Endpoint:

* Virtual machine Name – Name of your virtual machine.
* Name of endpoint – MeerkatCapture
* Private & public ports – Port 330

### Configure LightSwitch Apps

From the GIT repo, in the folder of **\Meerkat\Development\Deploy** you will need to access the compiled LS projects for Meerkat Admin and Meerkat Capture to get the deployable files as listed below.

For Meerkat Admin you need to edit: MeerkatAdmin\_2\_1.SetParameters.xml

The first line is the website name and the second line is the connection string for the database.

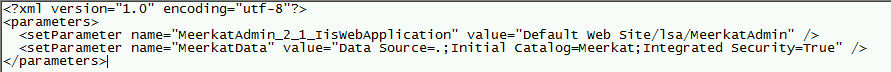
The Value **MeerkatAdmin** is the Name of the web site that we created previously in IIS. This will be adjusted together with the value for the Data Source since we have the DB resting in a different server. However, if the DB is on the same server, the value for Data Source need not change. 

Figure 4: Configuration file before changes.

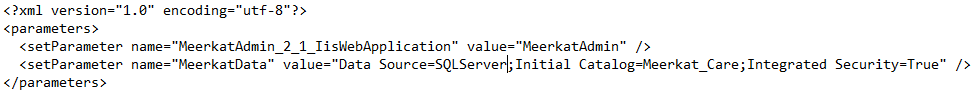


Figure 5: Configuration file after changes, note the website name is change.

For Meerkat Capture you need to edit the following file, which has the same options as the admin config: Meerkat\_Capture.SetParameters.xml

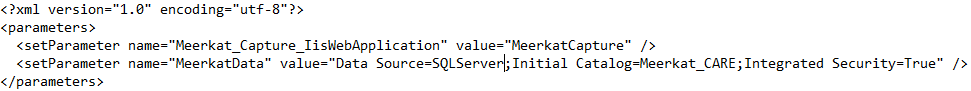


Figure 6: Capture configuration file, after changes.

### Deploy the LightSwitch apps

Deployment of the LightSwitch apps is straightforward, run the batch file in the folder with the /Y switch.

The Command prompt window needs to be run As Administrator.

For the admin client that is: .\MeerkatAdmin\_2\_1.deploy.cmd /Y

For the capture client, the command is: .\Meerkat\_Capture.deploy.cmd /Y

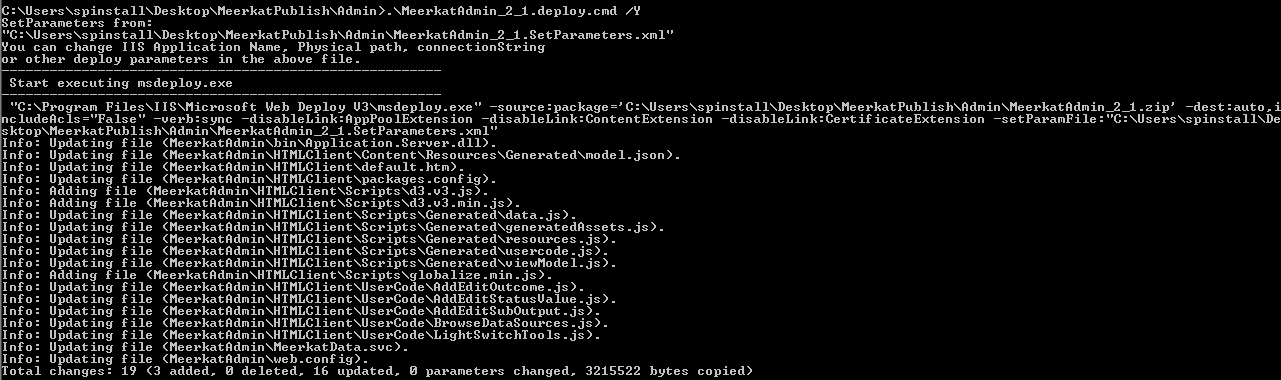


Figure 7: Example of the admin client being run on the server.

## Meerkat SharePoint Site

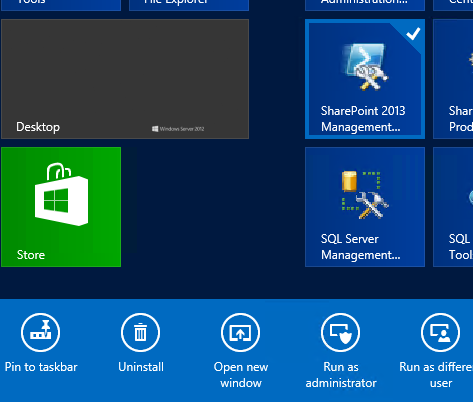
The following steps need to be followed when deploying the SharePoint site for Meerkat.

**NB:** These steps need to be taken on the top level site collection in the order in which they are listed here due to dependencies they cause.

### Filter Web part

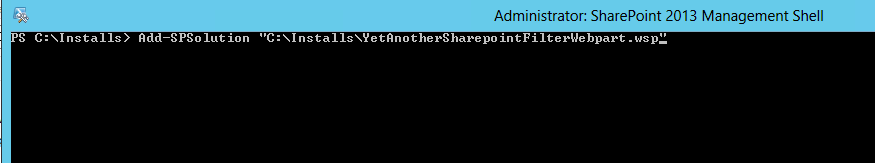
The following steps are used to deploy the Filter WebPart. However, it’s important to note that this WILL cause an IISReset and will therefore be Service Affecting.

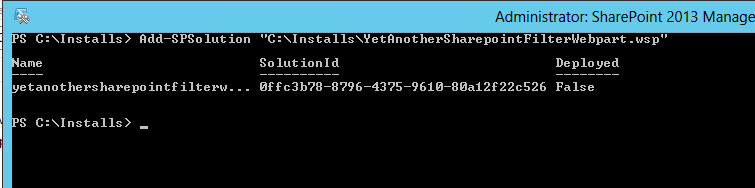
1. Copy the files to a folder – in the example, c:\installs
2. Open SharePoint PowerShell, right click to get the Run as administrator option.



When Powershell is open, enter the following command

Add-SPSolution "C:\Installs\YetAnotherSharepointFilterWebpart.wsp"

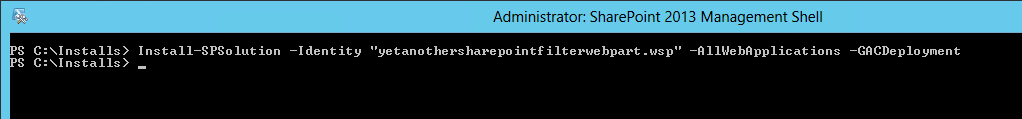




After which we need to install the Solution

Type in the following

Install-SPSolution -Identity "yetanothersharepointfilterwebpart.wsp" -AllWebApplications –GACDeployment



### Site Import

The files needed can be found in the Github repo under the SharePoint site folder.If you haven’t created a top level site collection in your web application, please do so and choose the Team Site template. This should be done under the web application url. As for the file to import, there should be a file called “Meerkat\_05\_template.cmp”. Import this into the site using the instructions below:

1. Open up SharePoint PowerShell as administrator and run the command below to import the site  
   Import-SPWeb –Identity <<http://URLName>> –Path <File\_Location>

e.g

Import-SPWeb –Identity <http://meerkat> –Path D:\meerkat.cmp

1. Open Central Admin, edit the Site collection Administrator list to include your user account

This site collection will have the Root site and 8 Outcome Sites

### MegaDropDown

In each Outcome site there is a file to called **MegaDropDown\_Build.js.** This file is located in the Style Library.The Content of each file is similar but the Outcome information needs to be changed.  
The URL is the address of the IIS web site for the MegaMenu, which we created above in IIS and the 3 lines beneath the URL are to be set per Outcome. Just edit the Outcome Number for each outcome.

$(document).ready(function (e) {

megaDropdown.config.DataSource = "SQL";

megaDropdown.config.CachingEnabled = false;

megaDropdown.config.Logging = false;

megaDropdown.config.SQLRootUrl = "**http://ServerName:8082**";

megaDropdown.config.SQLCategoryTable = "**mm.Outcome1MenuCategory**";

megaDropdown.config.SQLGroupTable = "**mm.Outcome1MenuGroup**";

megaDropdown.config.SQLLinkTable = "**mm.Outcome1MenuLink**";

megaDropdown.config.SQLJSONP = true;

megaDropdown.printConfig("#config");

megaDropdown.build("megadropdown");

});

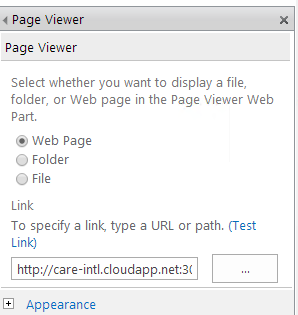
### Light Switch

To configure this, you will need the URLs of the two LightSwitch application setup for Admin and Capture that you set up in the beginning.

1. The SharePoint Page viewer web parts will need to be updated
2. Navigate to the pages located here  
   <http://sitename/Capture>

<http://sitename/Admin>

1. Click Page then Edit the page. It’s the only web part on the page so just click on it and edit the web part properties of the web page viewer web part to point to the correct servername and Light Switch pages as deployed earlier.



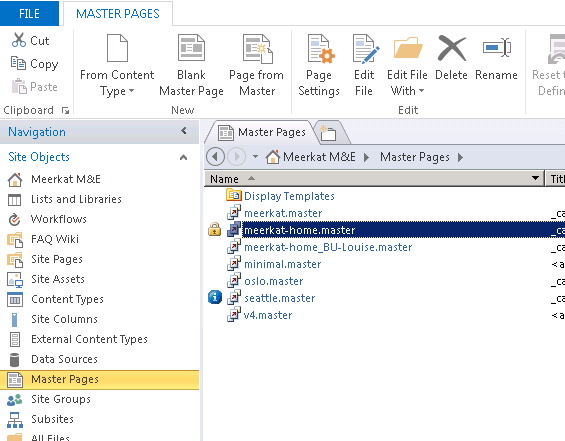
1. Once you’re done, click save and close to save the changes and stop editing.

### SharePoint Designer – master page edits

Using SharePoint Designer open up the master page to make edits if required for the Menu items.

You can download it from <http://www.microsoft.com/en-za/download/details.aspx?id=35491>

There are two master pages, meerkat-home.master and Meerkat.master. Meerkat-home.master is for the Root site, and the Meerkat.master is for the sub sites.



The menu items can be searched for and edited as shown below.

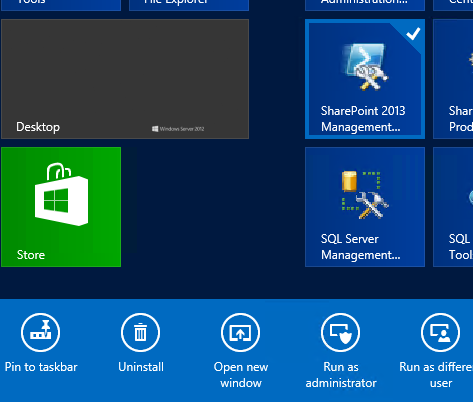




### Filter Web part

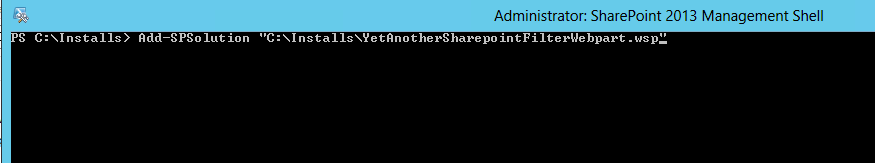
The following steps are used to deploy the Filter WebPart. However, it’s important to note that this WILL cause an IISReset and will therefore be Service Affecting.

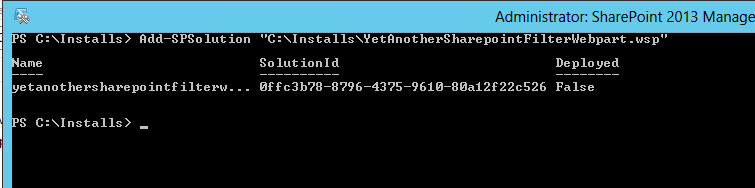
1. Copy the files to a folder – in the example, c:\installs
2. Open SharePoint PowerShell, right click to get the Run as administrator option.



When Powershell is open, enter the following command

Add-SPSolution "C:\Installs\YetAnotherSharepointFilterWebpart.wsp"

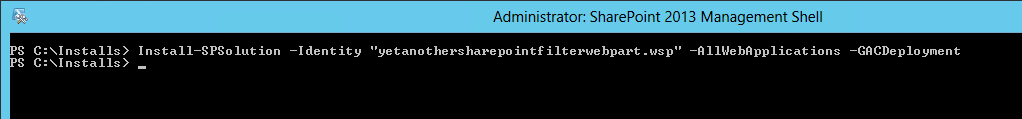




After which we need to install the Solution

Type in the following

Install-SPSolution -Identity "yetanothersharepointfilterwebpart.wsp" -AllWebApplications –GACDeployment



## Reports Deployment

The Reports are located on <http://URL/reports>

There are 3 libraries that contain reports, and a Data Source Library.  
Access the Data source library and edit the Meerkat Data source, ensuring that it points to the correct SQL instance and to the Meerkat Database.  
The Data source requires a user name to connect under, the account we have used is called MNEReports.

This account is a domain user, so you will need to create that user, and a password.  
The user will require access to the Meerkat Database, give the user DB\_DataReader access.

The Reports are housed in 3 libraries depending on their area of use.

* Admin Reports
* Status Reports
* Value Reports.

### SQL Server Reporting Services Deploy

To deploy the reports, you need to open the three SSRS projects using SQL server Data tools. These are all in the Github repository under the following paths:

* Reports\Admin Reports\MeerkatReports\AdminReports.rptproj
* Reports\Status Reports\MeerkatReports\StatusReports.rptproj
* Reports\Value Reports\MeerkatReports\ValueReports.rptproj

For each project you need to first open the project property page and change the target URL to point to the correct server and paths. You should not change the query string, just the servername portion of the URL.

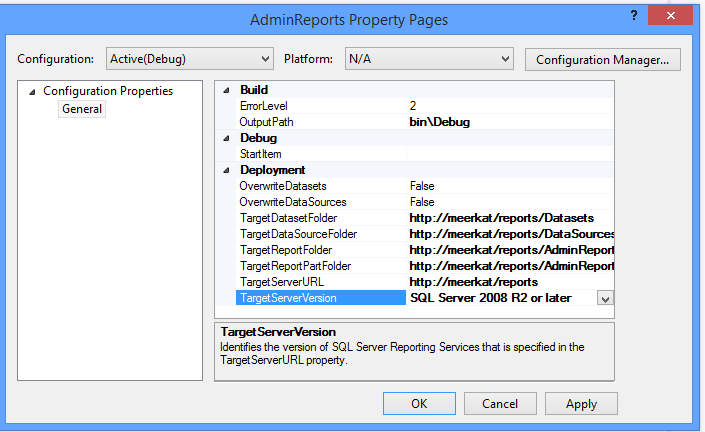


Figure 10: Example of the SRS project property page.

The next step is to open the shared data source in the project and make sure it is pointing to the correct SQL connection string.

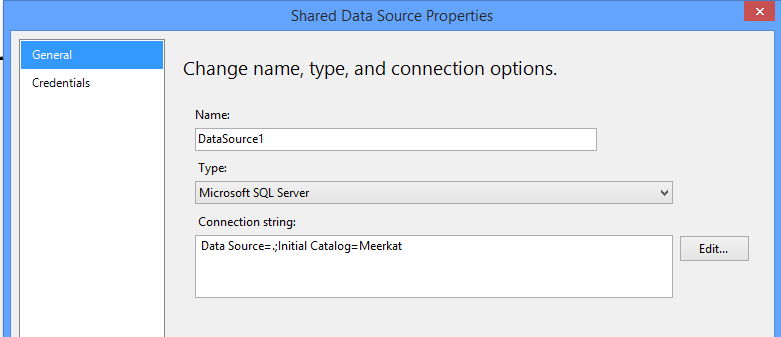


Figure 11: Example of the shared data source properties.

Once you have configured the shared data source, you need to ensure all reports are set to use the shared data source. This involves opening each report and check it’s data source properties.

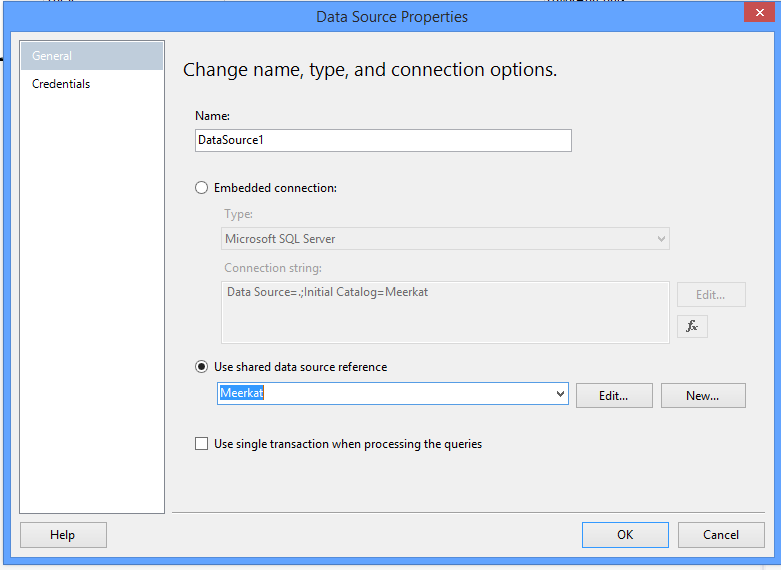


Figure 12: Confirming the shared data source is used in a report.

Finally you can right click on the project and hit deploy. You may get prompted for a login, this needs to be a valid login on the server (not a SQL login).

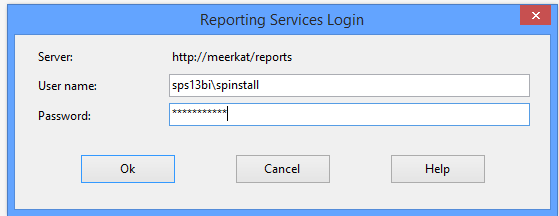


Figure 13: Example of the deploy login dialog.