Power BI Solution Template Deployment Guide - Sales Management v1.1

Table of Contents

[Overview 2](#_Toc446313580)

[**Option 1 – Power BI Model** 2](#_Toc446313581)

[Option 1 Prerequisites 2](#_Toc446313582)

[Step 1: Configure the INI file 3](#_Toc446313583)

[Basic INI Contents 3](#_Toc446313584)

[Informatica INI Contents 3](#_Toc446313585)

[Scribe INI Contents 4](#_Toc446313586)

[Step 2: Run setup.ps1 4](#_Toc446313587)

[Step 3: Post Deployment 5](#_Toc446313588)

[Step 4: Configure the Enterprise Gateway 5](#_Toc446313589)

[Step 5: Setting up the Dashboards using Power BI 5](#_Toc446313590)

[Step 6: Publish the report to PowerBI.com 7](#_Toc446313591)

[**Option 2 – SQL Server Analysis Services Model** 7](#_Toc446313592)

[Prerequisites 7](#_Toc446313593)

[Step 1: Configure the INI file 8](#_Toc446313594)

[Base INI Contents 8](#_Toc446313595)

[Informatica INI Contents 8](#_Toc446313596)

[Scribe INI Contents 9](#_Toc446313597)

[Step 2: Run setup.ps1 9](#_Toc446313598)

[Step 3: Post Deployment 10](#_Toc446313599)

[Step 4: Configure the Enterprise Gateway 10](#_Toc446313600)

[Step 5: Setting up the Dashboards using Power BI 10](#_Toc446313601)

[Addendum A: Populating Actual and Quota Values 11](#_Toc446313602)

# Overview

The Power BI Sales Management Solution Template reduces the time to implement a Power BI solution on Dynamics CRM or Salesforce. This document provides instructions for deploying the solution template.

Two deployment options are available depending on customer requirements. Both approaches benefit from fast ETL loads with incremental updates. Where they differ is where the model and data reside.

**Option 1 – Power BI Model**

In the Power BI Model, the model and data reside within the .pbix file. When published to PowerBI.com, the data can be refreshed from the source. Advantages of the Power BI Model approach include Cortana integration and natural language querying. The model data volume is limited to 250 MB (although this is the compressed figure – this restriction only becomes a factor for large CRM implementations).

**Option 2 – SSAS Model**

With this approach, the data and model reside in an instance of SQL Server Analysis Services separate from Power BI. This is recommended when data volumes exceed the 250MB limit or when performance requirements exceed what Power BI can provide.

The SSAS Model also supports row level security with rules derived from the source application (this feature will be supported in the Power BI Model in the summer of 2016).

# **Option 1 – Power BI Model**

## Prerequisites

The following software must be available:

* A Power BI Pro subscription.
* Salesforce data source requirements:
  + An [Informatica account](https://www.informatica.com/).
  + An on premise [Informatica Agent](https://network.informatica.com/docs/DOC-14954).
* Dynamics CRM data source requirements:
  + [Scribe Online account](http://www.scribesoft.com/products/scribe-online/) is required.
  + On premise [Scribe Agent](http://help.scribesoft.com/scribeonline/en/sol/agent/agentinstall.htm) must be installed.
  + Scribe API access range must include the server’s IP Address (Organization tab -> Security).
  + Scribe Replication Services must be enabled.
  + Scribe Dynamics CRM and SQL Server Database connectors must be enabled.
* A destination database - either:
  + Azure SQL DB, or
  + SQL Server 2012 service pack 3 or later
* PowerShell version 3 or later must be available on the machine where the installation script is run.

The following are provided alongside this guide:

* Automated Deployment Script (setup.ps1).
* Configurable INI file for setup (Scripts\sample.ini).
* Database Deployment Scripts (DBScripts).
* Documentation (Documents).
* Power BI .pbix file (PowerBIReport\SalesManagementReportwithData.pbix).

Details of how to deploy the Power BI Solution Template with the prerequisites and the material provided separately follows.

## Step 1: Configure the base INI file settings

A sample INI file has been provided (Scripts\sample.ini) alongside the setup.ps1 script to configure the Solution.

### Basic INI Contents

|  |  |
| --- | --- |
| **Key/Value** | **Meaning** |
| sql\_server=[server name] | SQL Server host and instance name to use (in the form of <host>\<instance>, or <instance> for on premises connections) |
| sql\_database=[database name] | SQL Server database name to create |
| sql\_user\_id=[username]  sql\_password=[password] | SQL Server username and password for SQL Server Authentication (comment or remove these two lines to use integrated authentication) |
| use\_ssas= [ true | false ] | **Special Note:** Set this to false for Power BI deployments |
| ssas\_server=[server name] | Not applicable for this option |
| ssas\_database=[AS database name] | Not applicable for this option |
| type\_etl=[informatica | scribe] | ETL tool to use |
| type\_source=[dynamics | salesforce] | Data source to replicate |

## Step 2: Configure the ETL INI file settings

Refer to the addendums depending on the ETL tool selected – Informatica or Scribe.

## Step 3: Run setup.ps1

1. Open a new PowerShell console running as administrator.
2. Run the following PowerShell command to enable scripting for this session:

Set-ExecutionPolicy Bypass

1. Navigate to the directory containing setup.ps1
2. Execute setup.ps1 with the following command

.\setup.ps1

## Step 4: Post Deployment

1. **If using Informatica, create a new Task using the newly created connections and select only the following entities to pull:**
   1. account
   2. lead
   3. opportunity
   4. OpportunityLineItem
   5. OpportunityStage
   6. product2
   7. user
   8. UserRole
2. **If using Scribe, edit the newly created Scribe Solution to pull only the following entities:**
   1. account
   2. businessunit
   3. lead
   4. opportunity
   5. opportunityproduct
   6. product
   7. systemuser
   8. systemusermanagermap
   9. territory

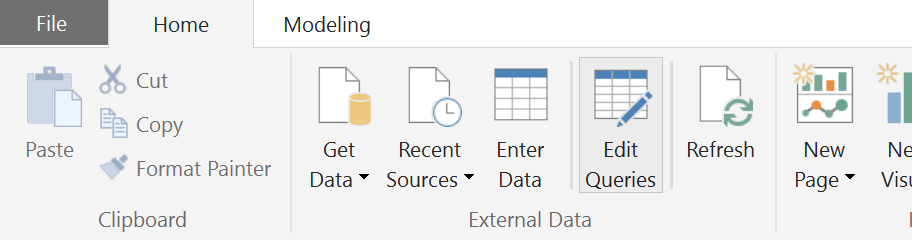
## Step 5: Configure the Enterprise Gateway

[Power BI Enterprise Gateway](https://powerbi.microsoft.com/en-us/documentation/powerbi-gateway-enterprise/) must be installed and configured if the database resides on an on-premise server or in a VM (either on-premise or Azure).

## Step 6: Setting up the Dashboards using Power BI

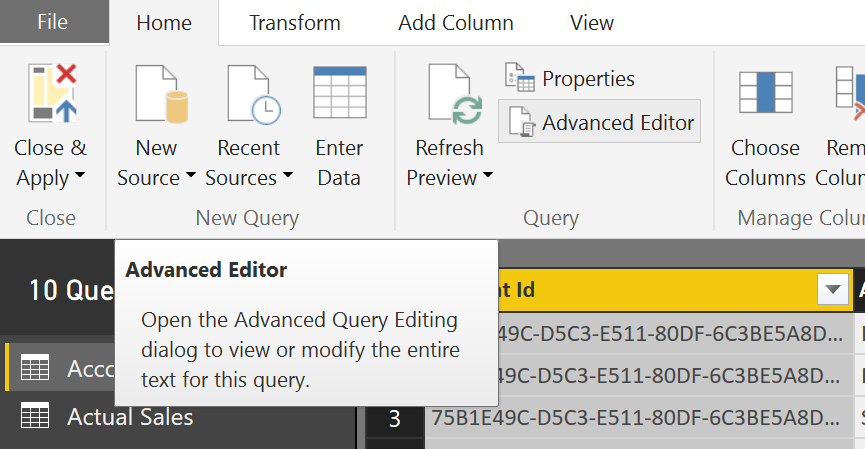
Open PowerBIReport\SalesManagementReportwithData.pbix.

To refresh the report with data from the source CRM application, the data sources in the Power BI file must first be updated. From the task bar, select “Edit Queries”:



For each and every table in the model, repeat the following steps:

Select “Advanced Editor”:



In the Advanced Editor, update the M script to refer to the data source instance and database name by replacing the highlighted areas:

let

Source = Sql.Databases("[server name]"),

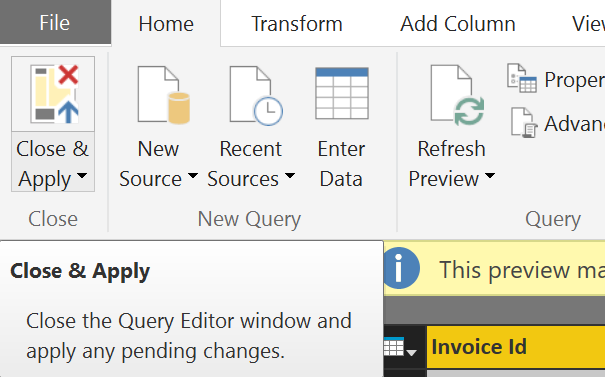
CRM = Source{[Name="[database name]"]}[Data],

Smgt\_AccountView = CRM{[Schema="Smgt",Item="AccountView"]}[Data]

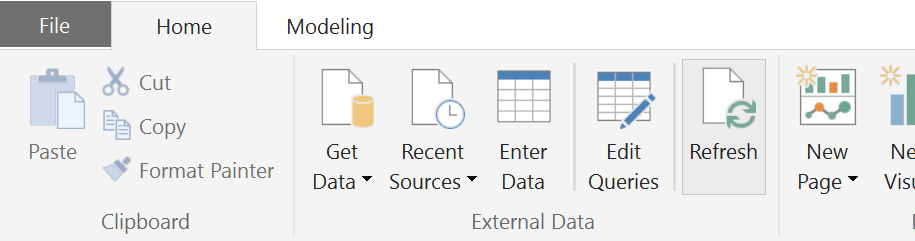
in

Smgt\_AccountView

After all tables’ datasources have been updated, select “Close and Apply”:



From the Report Editor, select “Refresh”:



The data retrieved from the source system will be shown.

## Step 7: Publish the report to PowerBI.com

To publish the report, select the “Publish” icon in the task bar. For on-premise data sources or data sources in Azure virtual machines, the Enterprise Gateway must be installed and configured.

**Congratulations** - you have successfully deployed the solution template!

# **Option 2 – SQL Server Analysis Services Model**

## Prerequisites

The following software must be available:

* A Power BI Pro subscription
* Salesforce data source requirements:
* An [Informatica account](https://www.informatica.com/).
* An on premise [Informatica Agent](https://network.informatica.com/docs/DOC-14954).
* Dynamics CRM data source requirements:
* [Scribe Online account](http://www.scribesoft.com/products/scribe-online/) is required
* On premise [Scribe Agent](http://help.scribesoft.com/scribeonline/en/sol/agent/agentinstall.htm) must be installed.
* Scribe API access range must include the server’s IP Address (Organization tab -> Security).
* Scribe Replication Services must be enabled.
* Scribe Dynamics CRM and SQL Server Database connectors must be enabled.
* An Azure Virtual Machine with a minimum recommended size of A3 or an on premise server with equivalent or greater technical specifications.
* SQL Server Enterprise Edition with a SQL Server Database (2012 service pack 3 or later).
* SQL Server Analysis Services must be available and configured in tabular mode.
* SQL Server Agent must be enabled.
* PowerShell version 3 or greater must be available on the server.
* The Active-Directory PowerShell module must be available (installed with [Remote Server Administration Tools](https://www.microsoft.com/en-us/download/details.aspx?id=45520)).

The following are provided alongside this guide:

* Automated Deployment Script (setup.ps1).
* Configurable INI file for setup (Scripts\sample.ini).
* Database Deployment Scripts (DBScripts), Analysis Services Configuration (Model), Documentation (Documents)
* Power BI .pbix file (PowerBIReport\SalesManagementReport.pbix).

Details of how to deploy the Power BI Solution Template with the prerequisites and the material provided separately follows.

## Step 1: Configure the Base INI file settings

A sample INI file has been provided (Scripts\sample.ini) alongside the setup.ps1 script to configure the Solution.

### Base INI Contents

|  |  |
| --- | --- |
| **Key/Value** | **Meaning** |
| sql\_server=[server name] | SQL Server host and instance name to use (in the form of <host>\<instance>, or <instance> for on premises connections) |
| sql\_database=[database name] | SQL Server database name to create, eg CRM |
| sql\_user\_id=[user name]  sql\_password=[password] | SQL Server username and password for SQL Server Authentication (comment or remove these two lines to use integrated authentication) |
| use\_ssas=true | Optionally toggle using SQL Server Analysis Services |
| ssas\_server=servername | SQL Server Analysis Services host and instance name to use (in the form of <host>\<instance>, or <instance> for on premises connections) |
| ssas\_database=SalesManagementTabularModel | SQL Server Analysis Services database name to create |
| type\_etl= | ETL tool to use (informatica | scribe) |
| type\_source= | Data source to replicate (dynamics | salesforce) |

## Step 2: Configure the ETL INI file settings

Refer to the addendums depending on the ETL tool selected – Informatica or Scribe.

## Step 3: Run setup.ps1

1. Open a new PowerShell console running as administrator.
2. Run the following PowerShell command to enable scripting for this session:

Set-ExecutionPolicy Bypass

1. Navigate to the directory containing setup.ps1
2. Execute setup.ps1 with the following command

.\setup.ps1

## Step 4: Post Deployment

1. **If using Informatica, create a new Task using the newly created connections and select only the following entities to pull:**
   1. account
   2. lead
   3. opportunity
   4. OpportunityLineItem
   5. OpportunityStage
   6. product2
   7. user
   8. UserRole
2. **If using Scribe, edit the newly created Scribe Solution to pull only the following entities:**
   1. account
   2. businessunit
   3. lead
   4. opportunity
   5. opportunityproduct
   6. product
   7. systemuser
   8. systemusermanagermap
   9. territory
3. Configure the SQL Server Agent Jobs:
   1. Open SQL Server Configuration Manager and configure the SQL Server Agent to run as a domain account.
   2. In SQL Server Management Studio:
      1. Open the Properties of the newly created “Save credential” SQL Server Agent job and edit the job at the “Encrypt” step.
      2. Replace the following line with your ETL account password (Informatica or Scribe):

“put password here, run, then set this back to empty string”

* + 1. Run the “Save credential” job, then remove your password by editing the job again.
    2. Run the “Data load and processing” job to pull data from your source or create a schedule for when it should run.

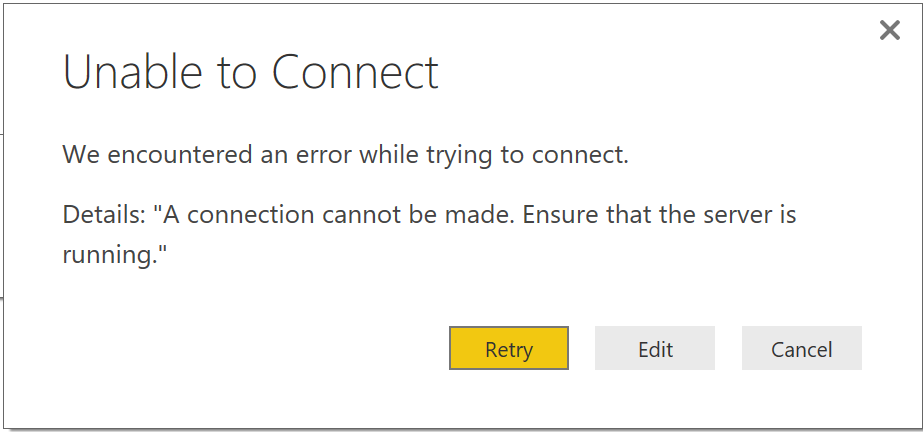
## Step 5: Configure the Enterprise Gateway

The [Power BI Enterprise Gateway](https://powerbi.microsoft.com/en-us/documentation/powerbi-gateway-enterprise/) must be installed on the same physical or virtual machine as SQL Server Analysis Services.

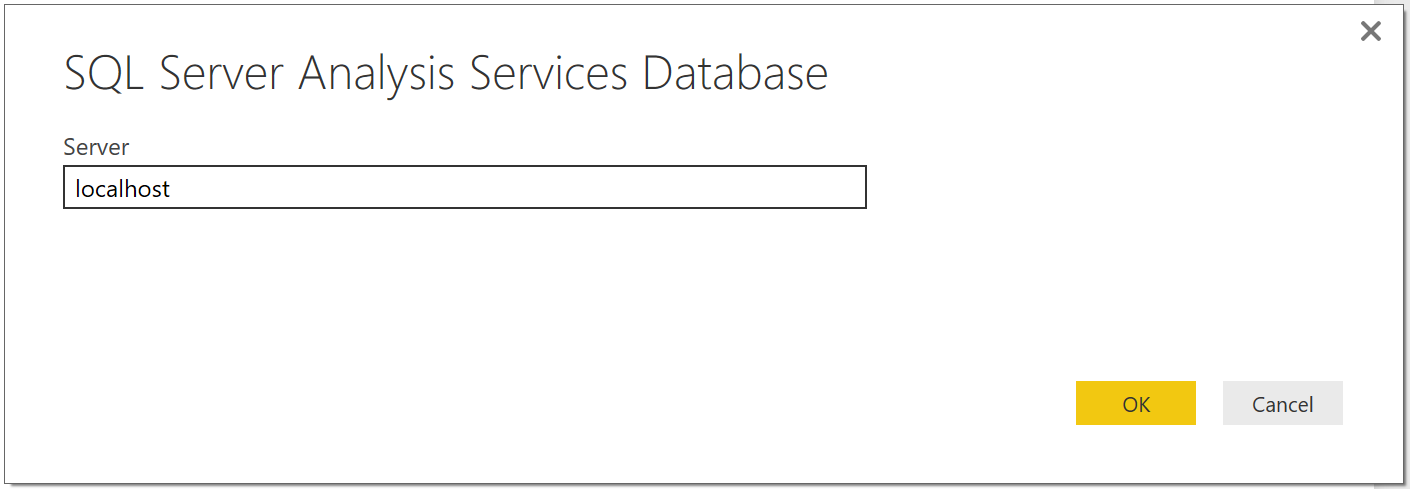
## Step 6: Setting up the Dashboards using Power BI

1. Open PowerBIReport\SalesManagementReport.pbix.

The connection to SQL Server Analysis Services generally needs to be updated:



Click “Edit” and enter the server name where SQL Server Analysis Services was installed:



and update the connection to SQL Server Database or SQL Server Analysis Services based on the deployed environment.

**Congratulations** - you have successfully deployed the solution template!

# Addendum A: Populating Actual and Quota Values

The template contains the following tables to be separately populated with actuals and quota values. It is the responsibility of the end customer to provide the mapping of source identifiers to the CRM identifiers (e.g., the account IDs may not match between an ERP and CRM system), configure the settings to pull actual values from the actuals table instead of closed opportunities, and keep values in both tables up to date.

When the empty tables are populated, these values appear in the template Power BI dashboard.

## ActualSales

The default configuration is to pull actual sales from closed opportunities. To change this behavior, first populate the smgt.ActualSales table (defined below) and update the configuration table as follows:

update Smgt.configuration set [value]=1 where configuration\_subgroup='actual\_sales' and [name]='enabled'

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype** | **Description** |
| InvoiceID | varchar(50) | Invoice identifier from source system |
| Revenue | Decimal | Actual revenue amount. |
| InvoiceDate | Date | Date when revenue realized. |
| AccountID | Uniqueidentifier | The CRM account identifier. Where original source system accountIDs do not match those in CRM, the value must be provided. |
| ProductID | Uniqueidentifier | The CRM product identifier. Where original source system product identifiers do not match those in CRM, the value must be provided. |

## Quota

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype** | **Description** |
| Date | Date |  |
| OwnerID | Uniqueidentifier | The CRM system user identifier. Where original source system product identifiers do not match those in CRM, the value must be provided. |
| ProductID | Uniqueidentifier | The CRM product identifier. |
| Quota | decimal |  |

# Addendum B: Informatica INI Contents

|  |  |
| --- | --- |
| **Key/Value** | **Meaning** |
| user=[Informatica account]  password=[password] | Informatica username and password |
| url=https://app.informaticaondemand.com | Informatica API URL |
| organization\_id=11111 | Informatica organization ID |
| task\_name=[Informatica task name] | Informatica task name to use |
| source.agent\_name=[ Informatica agent] | Informatica agent name to use |
| source.name=[SampleSource] | Informatica source connection name to create |
| source.user=[user@company.com]  source.password=[password] | The Salesforce username and password used by Informatica. |
| source.token=0000000000000000000000000 | The Salesforce Security Token – a 24 character string available in salesforce |
| target.agent\_name=SampleAgent | The name of the Informatica agent used to replicate the data. |
| target.name=SampleTarget | Informatica target connection name to create |
| target.hostname=servername | The same SQL Server name as used by the top level entry (see sql\_server=) |
| target.database=CRM | The same SQL Server database name as used by the top level entry (see sql\_database=) |
| target.user=[username]  target.password=[password] | SQL Server username and password |

# Addendum C: Scribe INI Contents

|  |  |
| --- | --- |
| **Key/Value** | **Meaning** |
| user=user@company.com  password=password | Scribe username and password |
| salt=ac103458-fcb6-41d3-94r0-43d25b4f4ff4 | Scribe salt to use (do not modify unless requested by Scribe) |
| key=00000000-0000-0000-0000-000000000000 | Scribe encoding key to use (accessible from the Organization tab -> Security -> API Cryptographic Token) |
| organization\_id=11111 | Scribe organization ID (accessible from the Organization tab) |
| agent\_name=SampleAgent | Scribe agent name to use |
| solution\_name=SampleSolution | Scribe solution name to create. Maximum number of characters: 25. |
| source.name=SampleSource | Scribe source connection name to create. Maximum number of characters: 25. |
| source.type=CRM | Scribe source type (CRM) |
| source.user=[username]  source.password=[password] | Dynamics CRM username and password |
| source.organization=Microsoft | Dynamics CRM organization name |
| source.deploy=[deployment type] | Dynamics CRM deployment type (Online | On-Premise | Partner-Hosted (IFD)) |
| source.url=https://disco.crm.dynamics.com | Dynamics CRM url |
| target.name=SampleTarget | Scribe target connection name to create |
| target.type=MSSQL | Scribe target type |
| target.server= [servername] | The same SQL Server name as used by the top level entry (see sql\_server=) |
| target.database=[database name] | The same SQL Server database name as used by the top level entry (see sql\_database=) |
| target.authentication=SQL Server | SQL Server authentication type. Only SQL authentication is currently supported. |
| target.user= [user name]  target.password=[password] | SQL Server username and password. |

# Addendum D: Changing the Fiscal Calendar

In order to change the fiscal start month, you will need to go to the Smgt.Configuration table and update the record FiscalMonthStart. By default, it is set to 1. To update it you would need to run the following script.

UPDATE [Smgt].[configuration] SET [value] = 1 WHERE [name] = 'FiscalMonthStart'

To support other types of fiscal calendars you would need to reinsert the data into the date table and ensure the fields beginning with Fiscal are populated including the calculated columns and measures inside the date table within the model.