Power BI Solution Template Deployment Guide - Sales Management

# Overview

The Power BI Solution Template brings together opportunity, target, and actual sales data to provide a complete sales management reporting solution.

Two deployment options are available depending on the customer requirements:

* **Option 1 – Power BI Model:**
* Incremental loads – fast ETL performance.
* Easily extensible with custom CRM data or data sourced from other applications.
* **Option 2 – SSAS Model:**
* Secure – rules driven by the data.
* For large data volumes exceeding the Power BI limit and/or very demanding workloads where a dedicated instance of SSAS is required.

The following instructions will help you configure and deploy the Solution Template.

# Prerequisites

* An Azure Virtual Machine with a minimum recommended size of A3 or an on premise server with equivalent or greater technical specifications.
* PowerShell version 3 or greater must be available on the server.
* SQL Server Enterprise Edition with a SQL Server Database (2012 service pack 3 or later).
* A Power BI Pro subscription.
* If using Salesforce as a data source:
* An [Informatica account](https://www.informatica.com/).
* An on premise [Informatica Agent](https://network.informatica.com/docs/DOC-14954).
* If using Dynamics CRM as a Data Source:
* A [Scribe Online account](http://www.scribesoft.com/products/scribe-online/).
* An on premise [Scribe Agent](http://help.scribesoft.com/scribeonline/en/sol/agent/agentinstall.htm).

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), and scripts used by setup.ps1 (Scripts).
* 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.

# Environment Setup

* If using the **Option 2** approach with SQL Server Analysis Services:
* SQL Server Analysis Services must be available and configured in tabular mode.
* SQL Server Agent must be enabled.
* The Active-Directory PowerShell module must be available (This is installed alongside [Remote Server Administration Tools](https://www.microsoft.com/en-us/download/details.aspx?id=45520)).
* If using a Scribe Online Account:
* API access range must include the server’s IP Address (Organization tab -> Security).
* Replication Services must be enabled.
* Dynamics CRM and SQL Server Database connectors must be enabled.

# Step 1: Configure the INI file

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

## INI Root

|  |  |
| --- | --- |
| **Key/Value** | **Meaning** |
| sql\_server=servername | SQL Server host and instance name to use (in the form of <host>\<instance>, or <instance> for on premises connections)) |
| sql\_database=CRM | SQL Server database name to create |
| sql\_user\_id=sa  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 | **Option 2 Deployment:** 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 | **Option 2 Deployment:** SQL Server Analysis Services database name to create |
| type\_etl= | ETL tool to use (informatica | scribe) |
| type\_source= | Data source to replicate (dynamics | salesforce) |

## Informatica and Informatica Connections

|  |  |
| --- | --- |
| **Key/Value** | **Meaning** |
| user=user@company.com  password=password | Informatica username and password |
| url=https://app.informaticaondemand.com | Informatica API URL |
| organization\_id=11111 | Informatica organization ID |
| task\_name=SampleTask | Informatica task name to use |
| source.agent\_name=SampleAgent | Informatica agent name to use |
| source.name=SampleSource | Informatica source connection name to create |
| source.user=user@company.com  source.password | Salesforce username and password |
| source.token=0000000000000000000000000 | Salesforce API token to use |
| target.agent\_name=SampleAgent | Informatica agent name to use |
| 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=sa  target.password=password | SQL Server username and password |

## Scribe and Scribe Connections

|  |  |
| --- | --- |
| **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 |
| source.name=SampleSource | Scribe source connection name to create |
| source.type=CRM | Scribe source type (CRM) |
| source.user=user@company.com  source.password=password | Dynamics CRM username and password |
| source.organization=Microsoft | Dynamics CRM organization name |
| source.deploy=Online | 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=CRM | The same SQL Server database name as used by the top level entry (see sql\_database=) |
| target.authentication=SQL Server | SQL Server authentication type |
| target.user=sa  target.password=password | SQL Server username and password |

# Step 2: 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 3: 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. For **Option 2** deployment, 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 4: Configure the Enterprise Gateway

If you want to publish to PowerBI.com, download and set up the [Power BI Enterprise Gateway](https://powerbi.microsoft.com/en-us/documentation/powerbi-gateway-enterprise/).

# Step 5: Setting up the Dashboards using Power BI

1. Open PowerBIReport\SalesManagementReport.pbix and update the connection to SQL Server Database or SQL Server Analysis Services based on the deployed environment (this will repopulate the graphs inside Power BI).
2. Optionally publish the desktop file to <https://www.powerbi.com> and configure the gateway connection (refer to the link in step 4).

Congratulations - you have successfully deployed the solution template!

# Populating Actual, Quota and Target Values

The template contains the following tables to be separately populated with targets, actuals and quotas. 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) and keep them up to date.

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

ActualSales

|  |  |  |
| --- | --- | --- |
| **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 |  |

Targets

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