Power BI Datasets (Data Models) Metadata only CI/CD

Created By: Aniket Sarkar

Creation Date: 04-JUL-2022

Contents

[Problem Description 2](#_Toc107850399)

[Tabular Object Model (TOM) 2](#_Toc107850400)

[Model.bim 3](#_Toc107850401)

[Tabular Editor 2.x 3](#_Toc107850402)

[Extract .bim file 3](#_Toc107850403)

[CI/CD using DevOps 4](#_Toc107850404)

[Tabular Database Deployment 4](#_Toc107850405)

[The Service Connection 4](#_Toc107850406)

[Configuring the task 5](#_Toc107850407)

[Tabular Editor 6](#_Toc107850408)

[Release Pipeline 8](#_Toc107850409)

# Problem Description

The usual publish / deployment option provided by Power BI Desktop deploys the entire contents of pbix file in the service. This method has several problems:

1. Data partitions of dataset (data model) in service is erased. Default partitions from local pbix development file is published. For large datasets it would be a long downtime hour to re-create partitions with data.
2. Production data in dataset (data model) in service is erased. Default data from local pbix development file is published.
3. Production roles in dataset (data model) in service is erased. Default roles from local pbix development file is published. While sometimes it is needed to publish roles from local file in service, there is no control over it as it cannot be avoided when required.

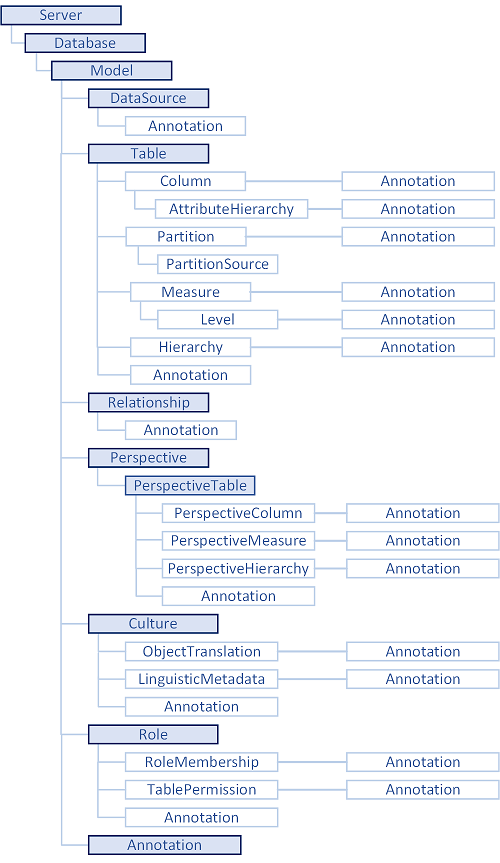
# Tabular Object Model (TOM)

The Tabular Object Model (TOM) is an extension of the Analysis Management Object (AMO) client library, created to support programming scenarios for tabular models created at compatibility level 1200 and higher. As with AMO, TOM provides a programmatic way to handle administrative functions like creating models, importing and refreshing data, and assigning roles and permissions.

TOM exposes native tabular metadata, such as model, tables, columns, and relationships objects. A high-level view of the object model tree, provided below, illustrates how the component parts are related.

TOM uses the XMLA protocol to communicate with the server and to manage objects.

Tabular metadata, which is structured as JSON documents, has a new command and object model definition syntax via the Tabular Model Scripting Language (TMSL). The scripting language uses JSON for the body of requests and responses.



# Model.bim

The Power BI file .pbix is a binary file but the metadata of the actual model in JSON format.

This file has the extension .bim

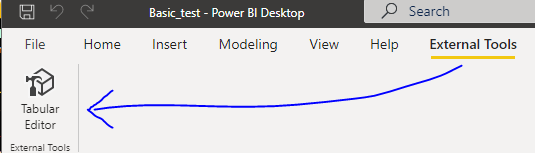
# Tabular Editor 2.x

Tabular Editor is an editor alternative to SSDT for authoring Tabular models for Analysis Services even without a workspace server.

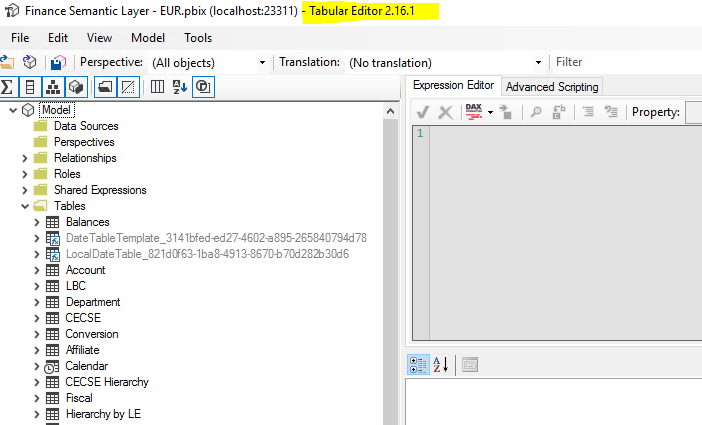
The Tabular Editor 2 is an open-source project that can edit a BIM file without accessing any data from the model. This offline capability enables quick changes to the BIM file, especially when you manipulate and manage measures, calculated columns, display folders, perspectives, and translations.

# Extract .bim file

Once the changes in the local pbix file is completed, can go to External Tools 🡪 Tabular Editor



A thin client version of Tabular Editor would open and display the Power BI metadata in object model tree format.



This tool would allow to save the entire metadata of the Power BI file in a bim file.

Go to File 🡪 Save As 🡪 Save Model.bim

# CI/CD using DevOps

Can upload the Model.bim file in DevOps repo.

From here, the deployment of the Model.bim file can be automated via Azure Release Pipeline and there are two documented Release methods.

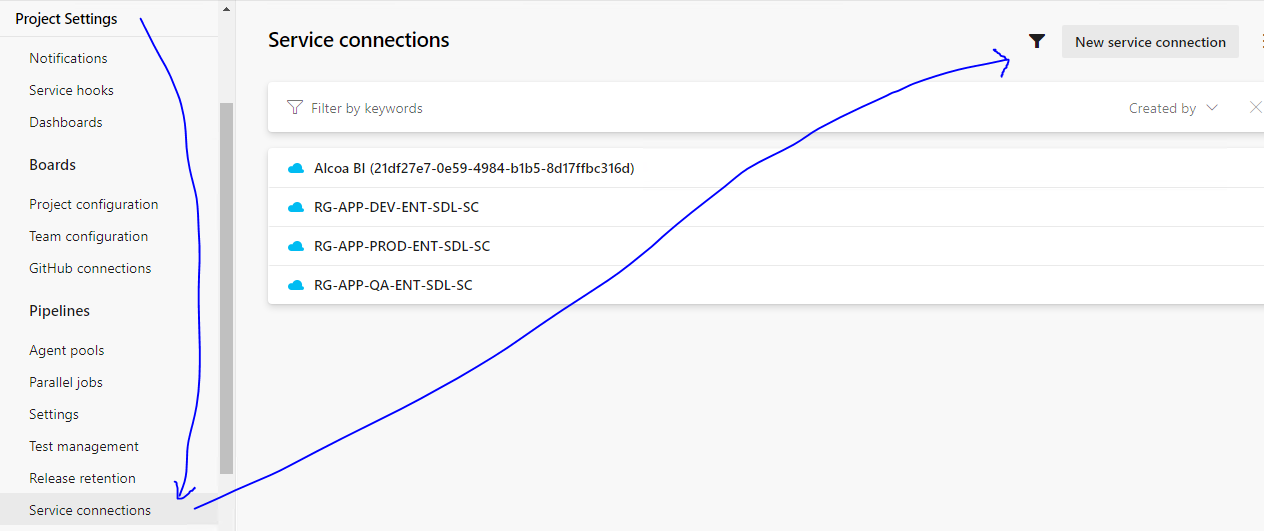
## Tabular Database Deployment

This task is available in Marketplace for free.

This task performs the deployment of metadata only from the Model.bim file from the repo to a Power BI Workspace utilizing the TOM.

### The Service Connection

A Service connection is required which can be created under DevOps Project Settings



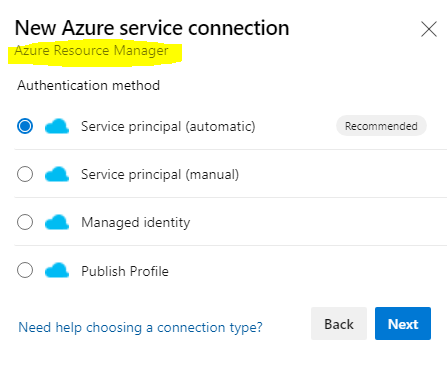
It is recommended to use the Service Principal Authentication method of Azure Resource Manager to create the Service Connection.

To perform this, an Azure App needs to be created and configured in Azure Active Directory service.

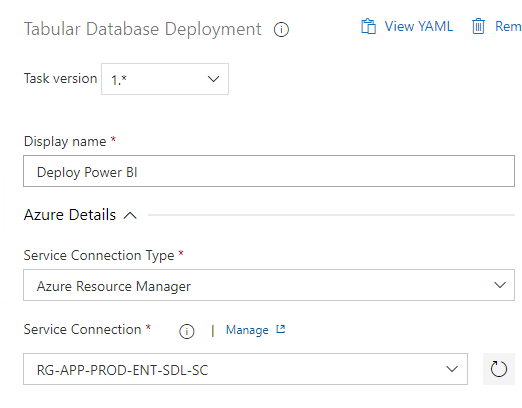
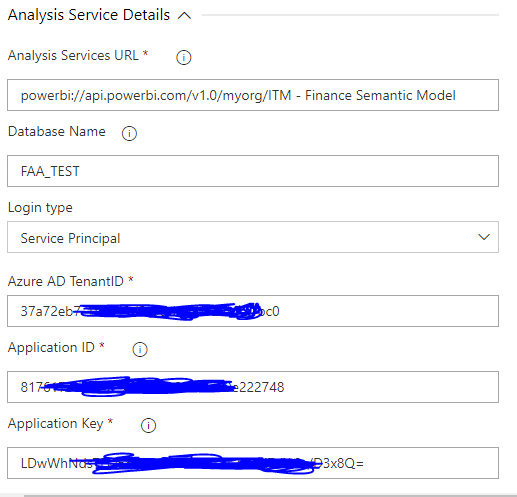
The App should be configured to have admin rights over Power BI Service.

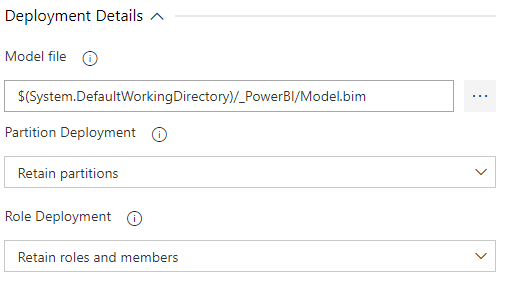
The App should have a “secret” for authentication.

The details of creation of Azure App is beyond the scope of this document.



### Configuring the task



This task provides the flexibility to choose whether user want to retain the Partitions, Roles, and Members in the service or overwrite it with what is available in the pbix file / Model.bim

Apart from that, the task takes in input the Service Connection, Power BI Service Workspace URL, Name of the Dataset by which Model.bim contents would be deployed, Azure Tenant ID, the App ID and the App Secret (Key).

## Tabular Editor

Tabular Editor has command line interfaces which can be invoked to deploy a .bim file.

Tabular Editor has a portable version which can be invoked without installation.

The command line has various switches to customize the deployment process.

The syntax is:

.\TabularEditor.exe .\Model.bim -D "Provider=MSOLAP;Data Source=<<Workspace URL>>;User ID=app:<<app id>>@<<tenant id>>;Password=<<app secret>>" <<datasetname>> -O -C -G -E -W -P -R -M

Example:

.\TabularEditor.exe .\Model.bim -D "Provider=MSOLAP;Data Source=powerbi://api.powerbi.com/v1.0/myorg/ITM - Finance Semantic Model;User ID=app:8176175d-y4876-48d9-7590-a3dj0t494748@37a72eb7-2der-9009de-9ddc-6b923b751bc0;Password=LDwWhNdsYDUE63293Kb237rhL8s/D3x8Q=" FAA\_TEST -O -C -G -E -W -P -R -M

This assumes that the Tabular Editor executable and Model.bim file is in one directory and the command is executed from the same directory.

Command Line Switches

-S / -SCRIPT Execute the specified script on the model after loading.

scriptN Full path of one or more files containing a C# script to execute or an inline

script.

-SC / -SCHEMACHECK Attempts to connect to all Provider Data Sources in order to detect table schema

changes. Outputs...

...warnings for mismatched data types and unmapped source columns

...errors for unmapped model columns.

-A / -ANALYZE Runs Best Practice Analyzer and outputs the result to the console.

rules Optional path of file or URL of additional BPA rules to be analyzed. If

specified, model is not analyzed against local user/local machine rules,

but rules defined within the model are still applied.

-AX / -ANALYZEX Same as -A / -ANALYZE but excludes rules specified in the model annotations.

-B / -BIM / -BUILD Saves the model (after optional script execution) as a Model.bim file.

output Full path of the Model.bim file to save to.

id Optional id/name to assign to the Database object when saving.

-F / -FOLDER Saves the model (after optional script execution) as a Folder structure.

output Full path of the folder to save to. Folder is created if it does not exist.

id Optional id/name to assign to the Database object when saving.

-V / -VSTS Output Visual Studio Team Services logging commands.

-G / -GITHUB Output GitHub Actions workflow commands.

-T / -TRX Produces a VSTEST (trx) file with details on the execution.

resultsfile File name of the VSTEST XML file.

-D / -DEPLOY Command-line deployment

If no additional parameters are specified, this switch will save model metadata

back to the source (file or database).

server Name of server to deploy to or connection string to Analysis Services.

database ID of the database to deploy (create/overwrite).

-L / -LOGIN Disables integrated security when connecting to the server. Specify:

user Username (must be a user with admin rights on the server)

pass Password

-O / -OVERWRITE Allow deploy (overwrite) of an existing database.

-C / -CONNECTIONS Deploy (overwrite) existing data sources in the model. After the -C switch, you

can (optionally) specify any number of placeholder-value pairs. Doing so, will

replace any occurrence of the specified placeholders (plch1, plch2, ...) in the

connection strings of every data source in the model, with the specified values

(value1, value2, ...).

-P / -PARTITIONS Deploy (overwrite) existing table partitions in the model.

-Y / -SKIPPOLICY Do not overwrite partitions that have Incremental Refresh Policies defined.

-R / -ROLES Deploy roles.

-M / -MEMBERS Deploy role members.

-X / -XMLA No deployment. Generate XMLA/TMSL script for later deployment instead.

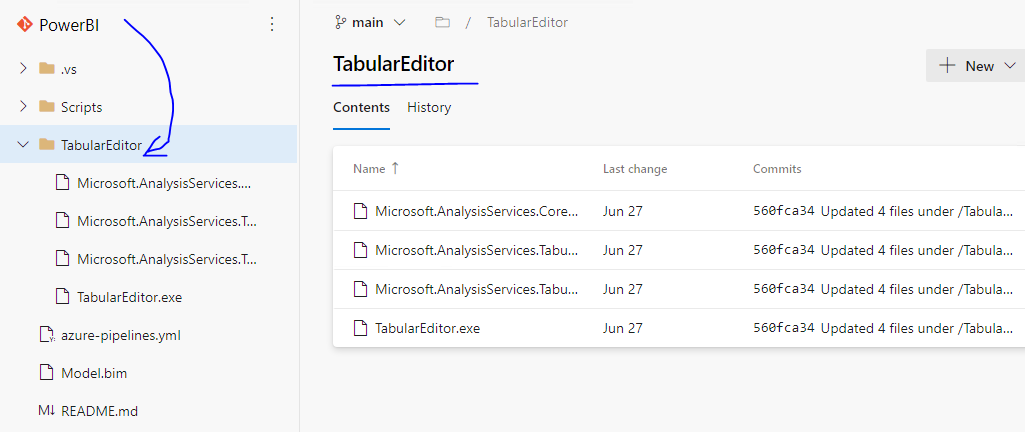
xmla\_script File name of the new XMLA/TMSL script output.

-W / -WARN Outputs information about unprocessed objects as warnings.

-E / -ERR Returns a non-zero exit code if Analysis Services returns any error messages after

the metadata was deployed / updated.

To automate it via Azure DevOps, should keep the Portable version of Tabular Editor and it’s dependent dlls in a folder in DevOps Repo.



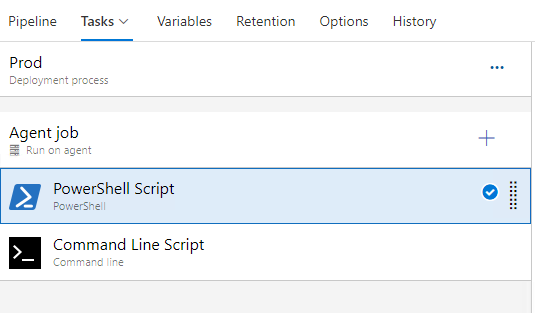
Keep a PowerShell script which would download the TabularEditor.exe in the root directory of the Repo during runtime.





### Release Pipeline

The Release Pipeline should have two tasks, viz. Executing the PowerShell script and then Executing the Command Line of Tabular Editor.exe.



The PowerShell script would only be responsible to download the TabularEditor.exe in the root directory of DevOps.

The actual deployment work would be done via the Command Line Script, invoking TabularEditor.exe

