Semantic Model Comparison & Merging with ALM Toolkit

Summary: This whitepaper and associated samples describe how to perform model comparison and merging for Analysis Services tabular models.

Applies to: Power BI semantic models, Microsoft SQL Server Analysis Services, Microsoft Azure Analysis Services

Authored by: Analysis Services PM team

Published: August 2024

Contents

Introduction

Relational-model schema comparison and merging is a well-established market. Leading products include SSDT Schema Compare and Redgate SQL Compare, which is partially integrated into Visual Studio. These tools are used by organizations seeking to adopt a DevOps culture to automate build-and-deployment processes and increase the reliability and repeatability of mission critical systems.

Such functionality is also available for Analysis Services tabular models. This document describes how to use the ALM Toolkit open-source tool for application-lifecycle management (ALM) scenarios.

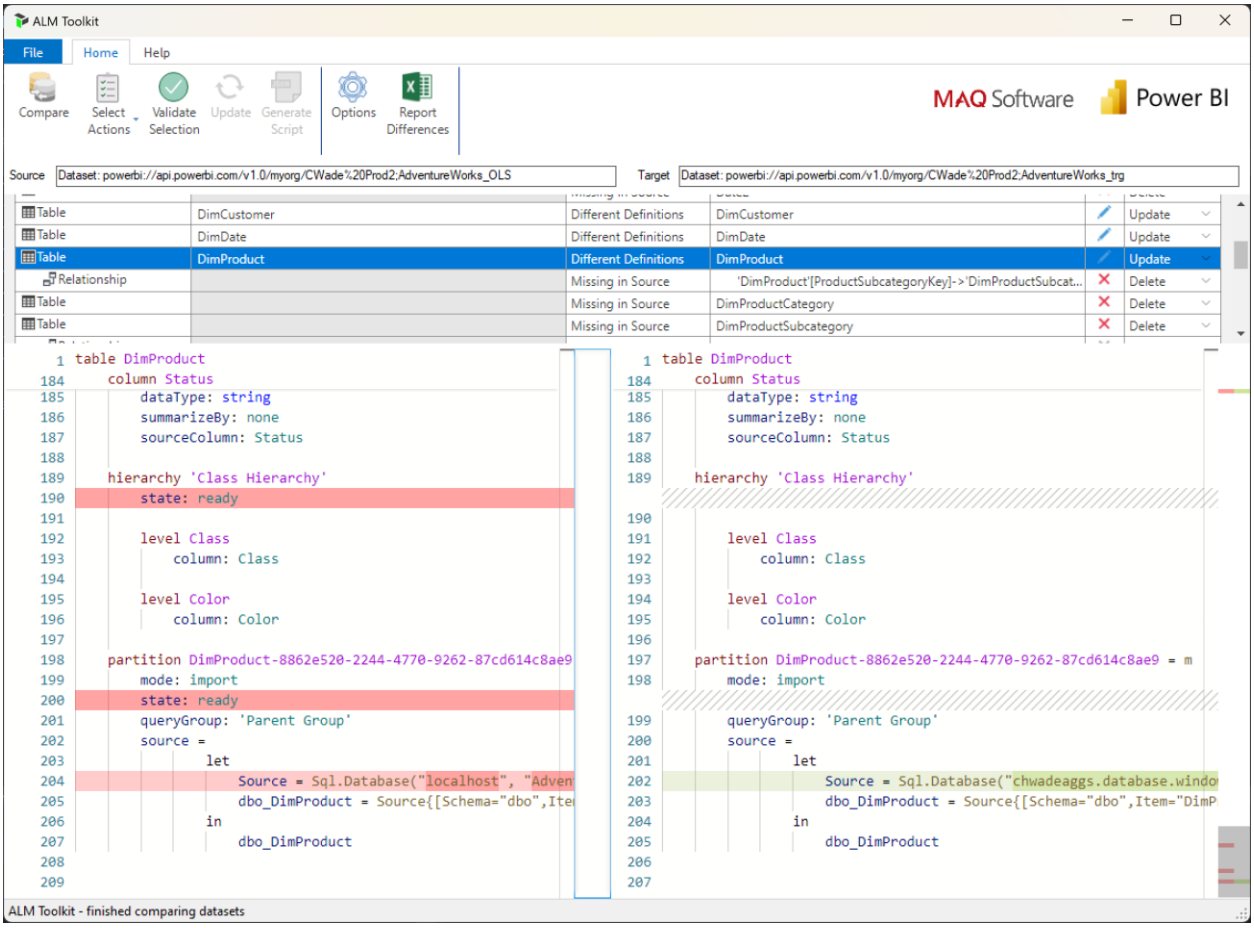
The ALM Toolkit is an open-source community tool maintained by the AS PM team. It is not a Microsoft supported product.

ALM Toolkit summary

The ALM Toolkit is a schema comparison tool you can use for ALM scenarios such as:

- Easily deploy changes only to model metadata, whilst retaining data including incremental refresh data.
- Diff & merge across repos/models with guardrails to ensure the integrity of the resulting model metadata. Pick and choose objects to reuse in other models.
- Reuse common objects across models. For example, conformed dimensions, standardized date dimensions, and common measures to be standardized across models.
- Command-line execution for automation.

Comparisons diffs are, by default, shown using the Tabular Model Definition Language (TMDL). TMDL allows you to view your tabular model metadata in a human-readable, YAML-like syntax which helps support scenarios where data professionals need to make incremental changes to models, collaborate with others, or integrate with source control. See the TMDL documentation page for more information.



Integration with Power BI Desktop

ALM Toolkit is a Power BI Desktop external tool. In addition to launching from the Windows Start menu as a standalone application, it can be launched from the External tools ribbon in Power BI Desktop.

Deployment features

For either approach, ALM Toolkit supports the following deployment features for tabular models.

- Deployment configurations are supported by saving comparison information such as source/target model, comparison options, and skipped actions to a ALMT file. It can later be used as a configuration for deployment with a model/environment combination. This is discussed below in the deployment configurations section.
- Retain partitions, role members, environment-specific data sources, etc. Partitions are normally created and managed by automated processes; they don't exist in the version of the tabular model in source control. ALM Toolkit allows partitions, and the data with them, to be retained on deployment.
- Command-line execution for integration with continuous-integration processes.
- TMSL script generation to give to a DBA for execution with elevated permissions.
- Processing (data refresh), limited to affected tables.

Prerequisites

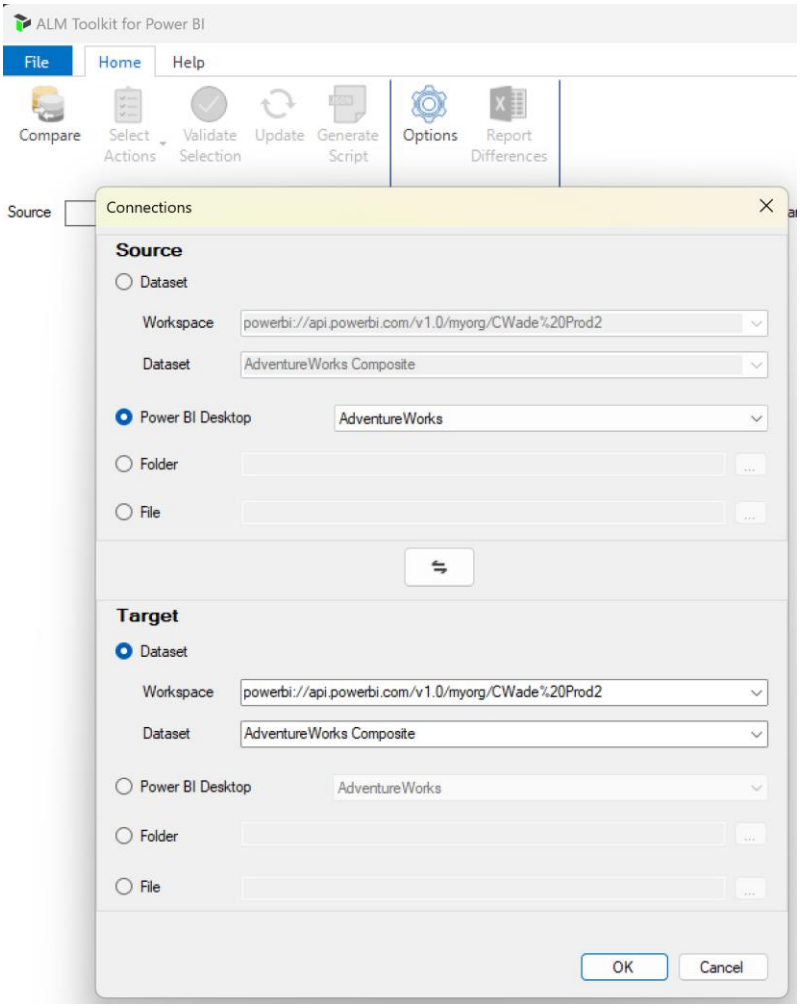
Download and install the ALM Toolkit from the Analysis Services git repo.

When working with semantic models in the Power BI service, you must ensure the XMLA endpoint is enabled for read/write operations as described here. It is also recommended that you use large model mode for faster model updates due to improved XMLA write operation performance.

Comparison workflow

New comparison

Click on the Compare button on the ribbon to launch the Connections dialog.

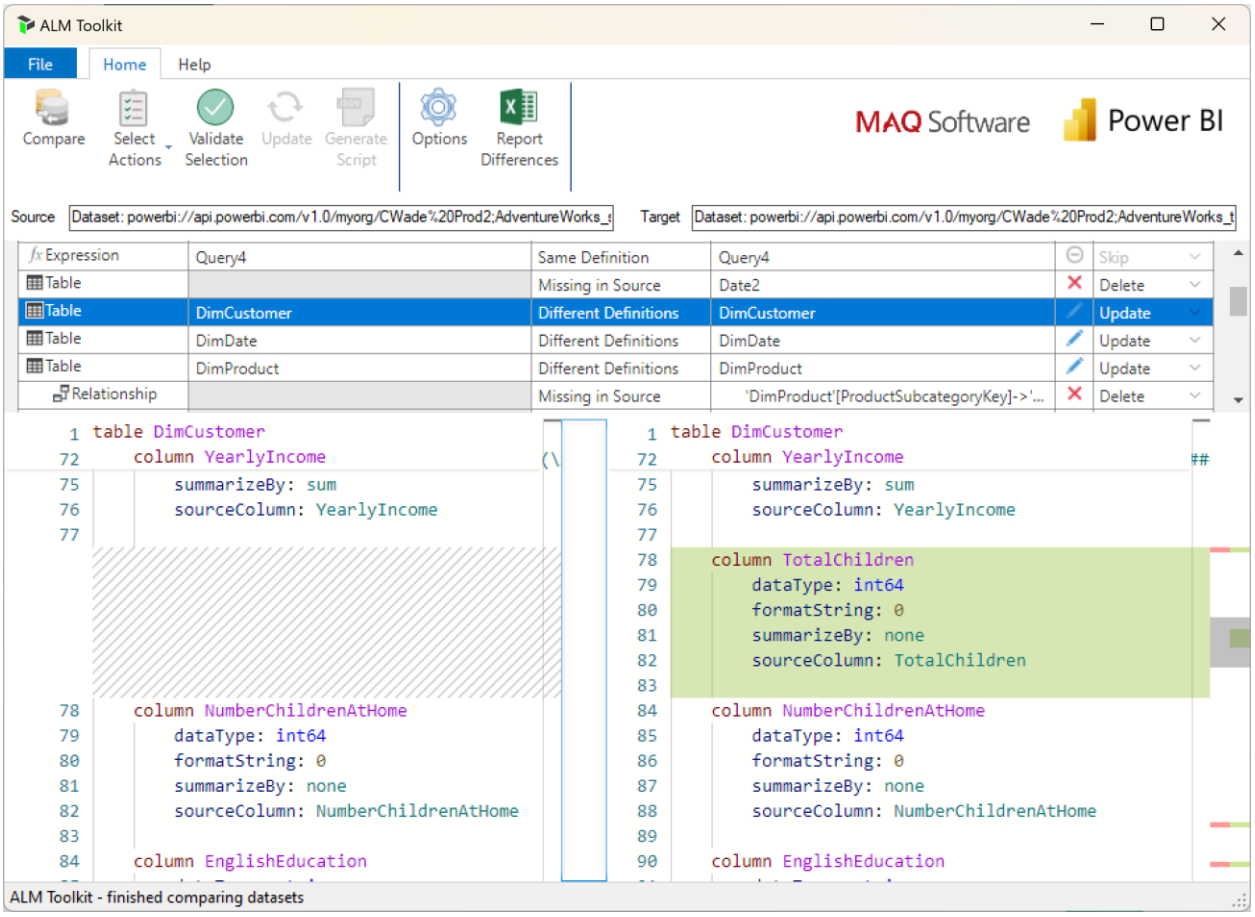


In the Connections dialog, specify the source and target, which can be any of the following:

- Semantic model in a Power BI workspace. See this document to get the workspace URL.
- Semantic model in an Azure Analysis Services or SQL Server Analysis Services server (tabular mode).
- Semantic model running locally in Power BI Desktop.
- A folder containing metadata files serialized using the TMDL format.
- Tabular metadata file such as a BIM file using the TMSL format.

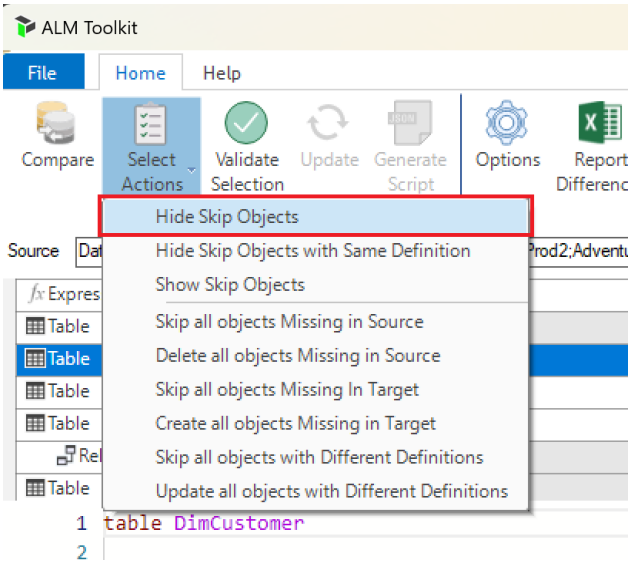
Metadata files are compared offline, without the need for an Analysis Services server instance.

With the desired source and target specified, click OK and consider the differences displayed.



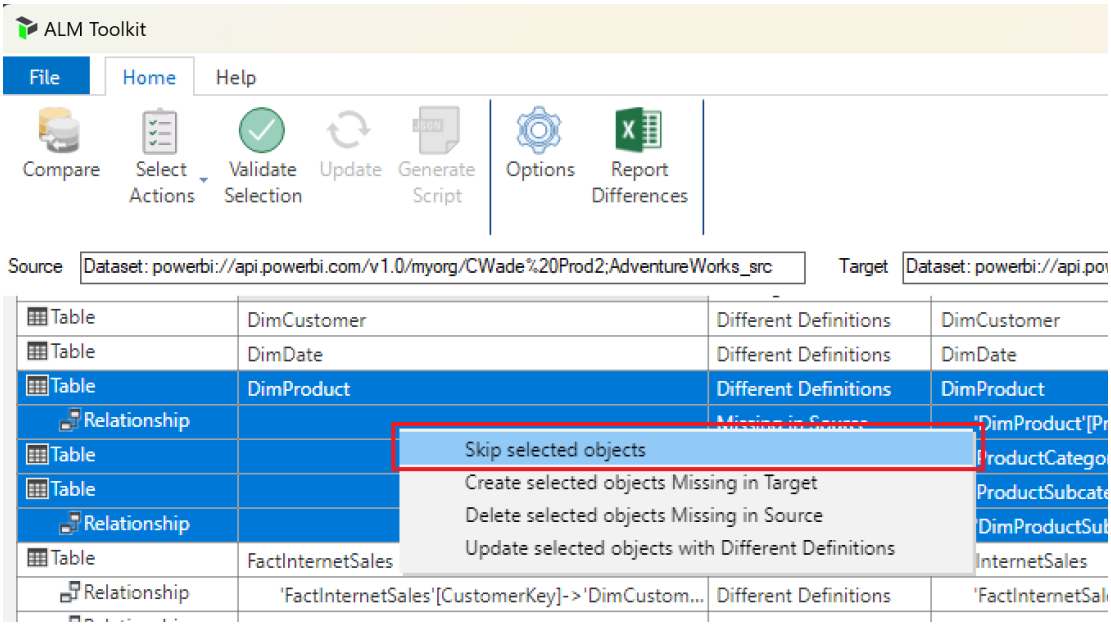
Select actions to be applied

Hide Skip Objects to focus on the differences. Objects with the same definition are automatically skipped and the action is greyed out.



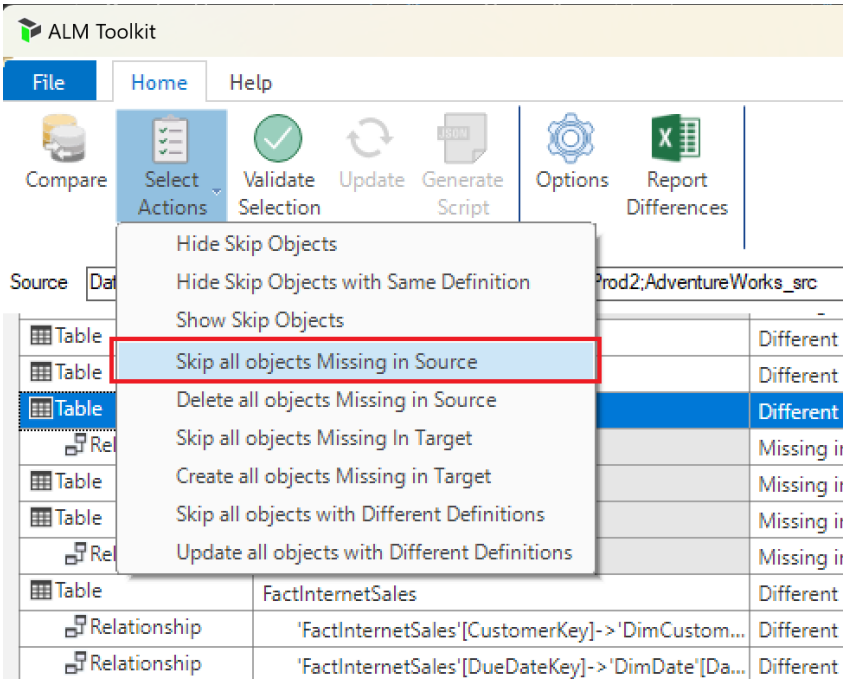
Actions available are Create, Update and Delete. Select the actions you wish to take.

- Select items individually using the dropdowns in the Action column.
- Specify multiple actions at once using the Select Actions dropdown on the ribbon (see image above).
- Specify multiple actions at once by selecting multiple items, right-click and use the context menu (see image below).



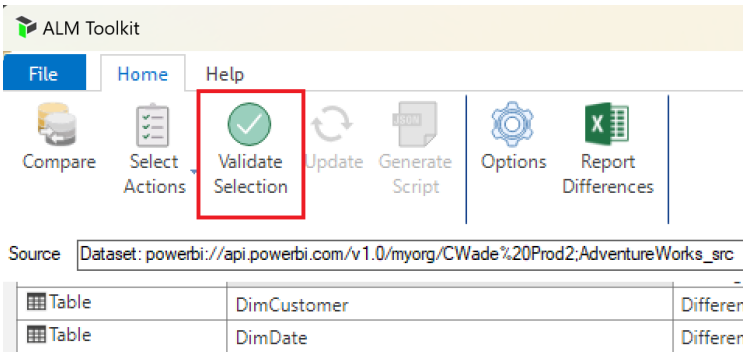
Harvesting objects for reuse

When harvesting objects – perhaps from a prototype model, we often don't want to affect any of the objects already in the target. Such objects have a Delete action by default because they exist in the target, but not the source model. Instead of skipping all the Delete actions individually, we can select Skip all objects Missing in Source to skip all deletes in one go.

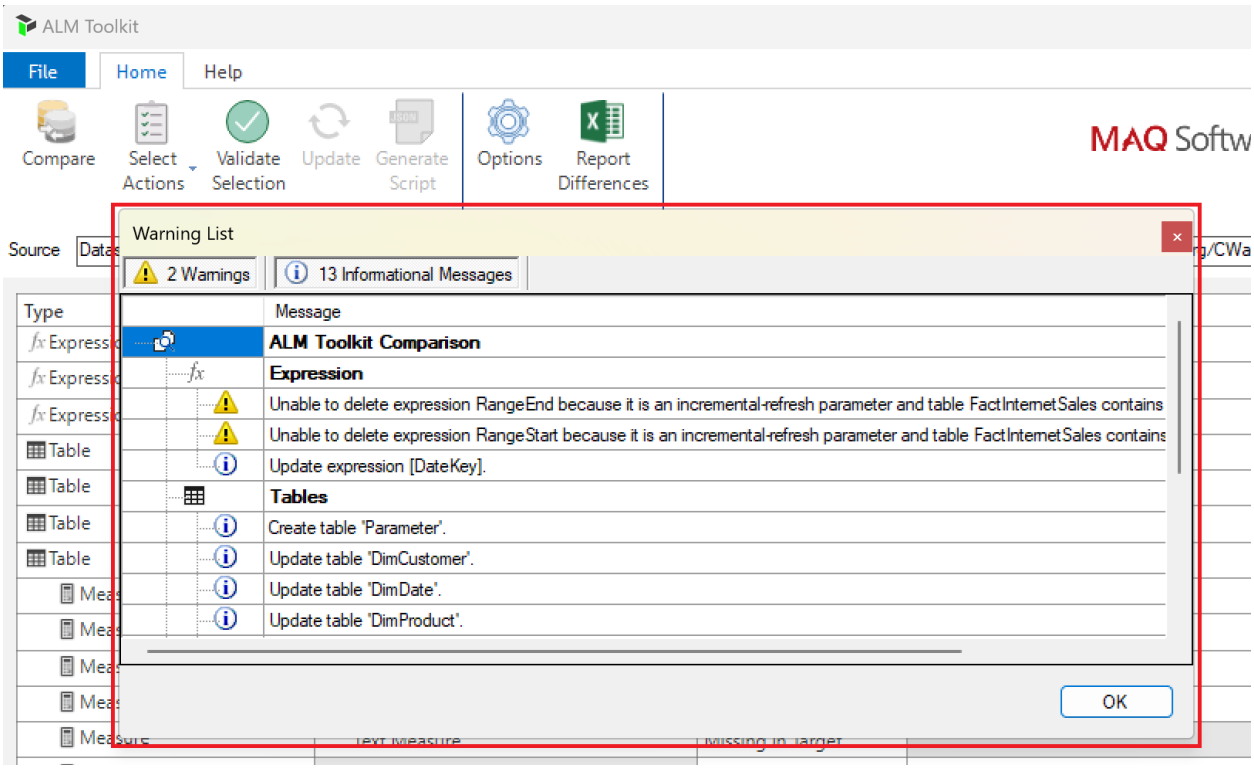


Validate selection

Click Validate Selection on the ribbon.



Check any warnings in the Warning List. This impact analysis safeguards the integrity of the target model. In the example below, the RangeStart, RangeEnd parameters can't be deleted because the FactInternetSales table (which is not being deleted) refers to them. Therefore, updating the table would introduce an invalid object reference, and ALM Toolkit disallows this.

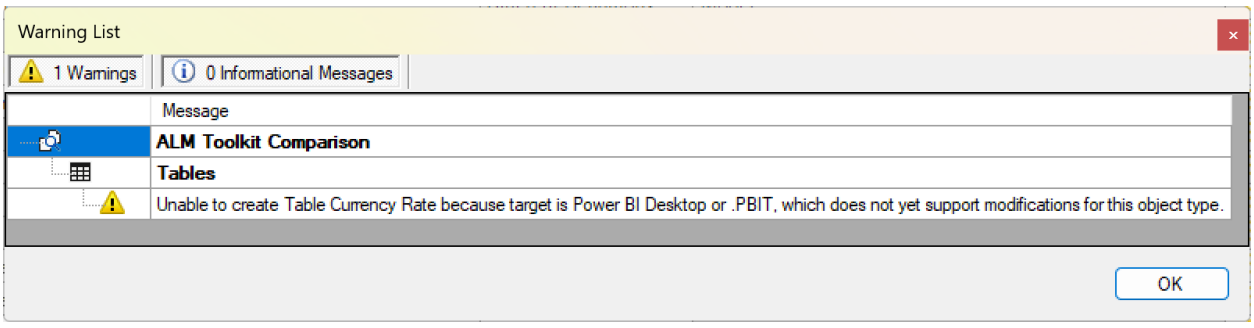


Invalid object dependencies that are handled include the following.

- Ambiguous relationship paths resulting from multiple active relationships leading to a dimension table.
- Relationships referring to nonexistent tables/columns.
- Measures/KPIs with conflicting names.
- M expression references to other M expressions and data sources that are missing.
- Table/partition dependencies on M expressions and data sources that are missing.
- Invalid DAX measure references display warnings if the option is selected (see section below), but they do not prevent objects from being created/updated/deleted.

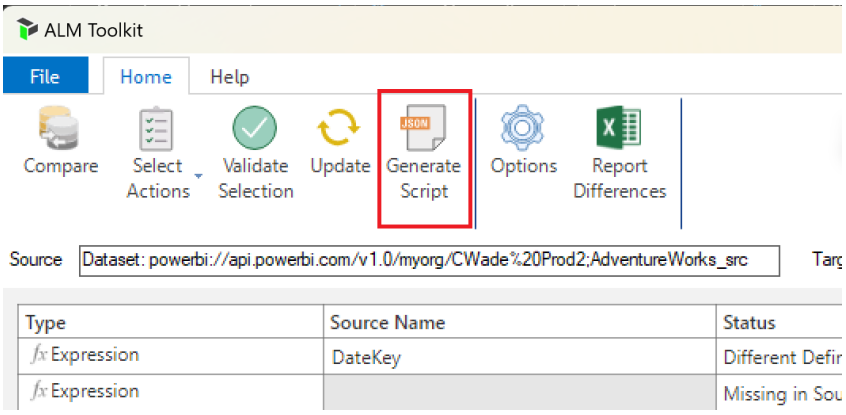
Restrictions when the target is Power BI Desktop or PBIT file

Due to the limitations imposed by Power BI Desktop on write operations, if your target is a Power BI Desktop instance or a PBIP file, validation will disallow actions on certain object types and display the following message. If/when these limitations are relaxed by Power BI Desktop "hardening", these restrictions will be removed.

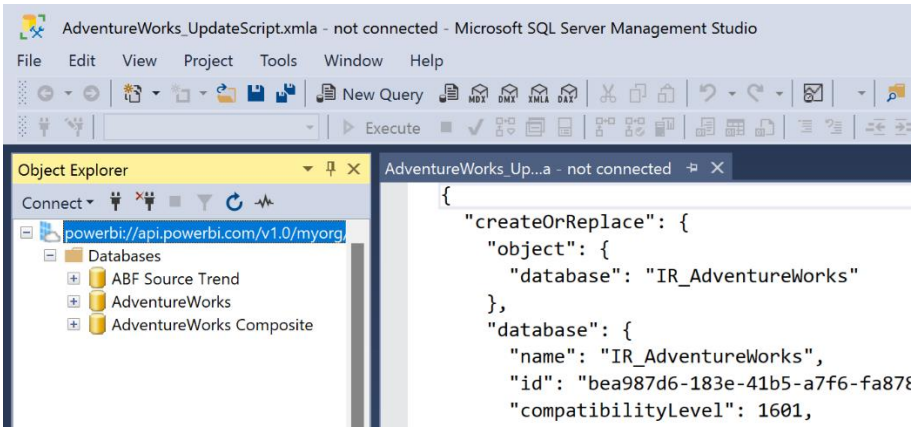


Script generation

To generate a [Tabular Model Scripting Language](#) (TMSL) script of the changes, perhaps to give to a DBA for execution with elevated permissions, click the Generate Script button.

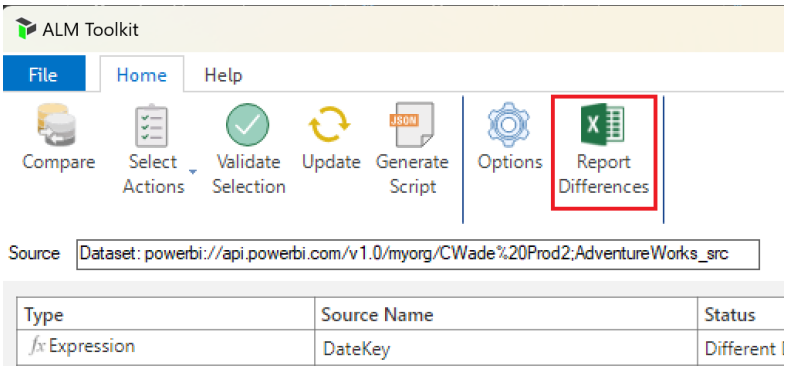


A TMSL script is generated based on the actions selected above, which can be executed from SQL Server Management Studio (SSMS).

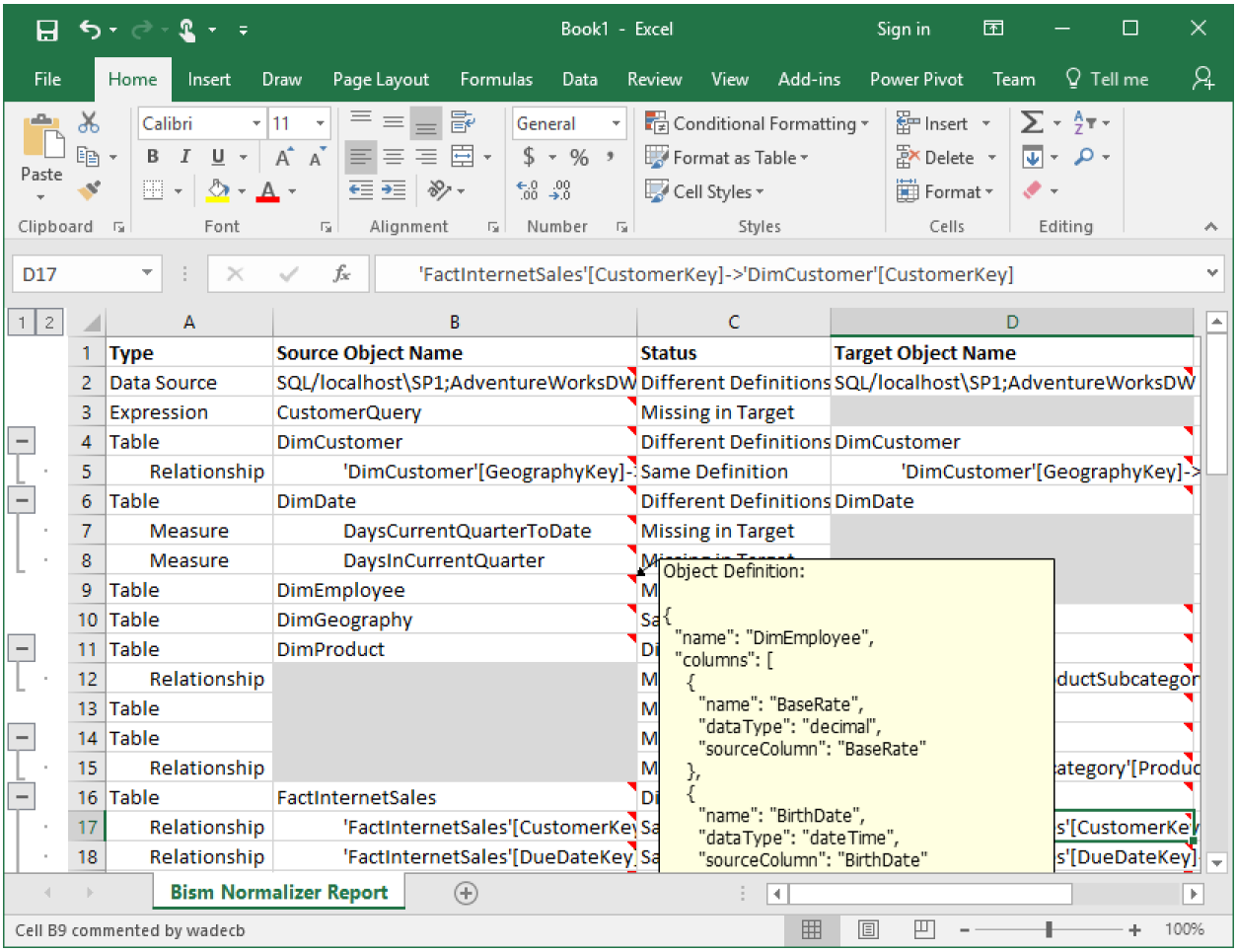


Report differences

To create a differences report, perhaps for auditing purposes, click the Report Differences button.

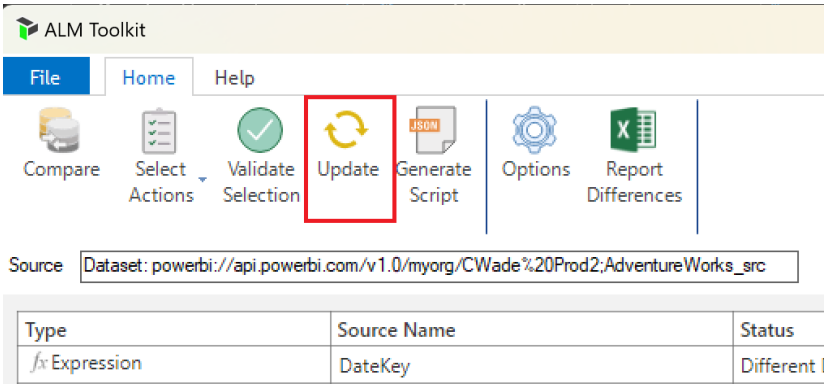


An Excel report is generated and displayed.



Update target

Click the Update ribbon button and confirm you want to update the target when prompted.

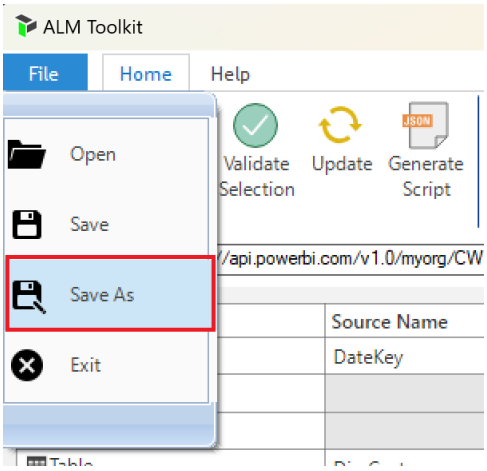


With the changes applied and the comparison refreshed, all objects now have the same definitions except those that were skipped or with changes that could not be applied due to validation constraints.

Saving comparisons

Comparisons can be saved to ALMT files to be applied/reapplied later. The following information is retained.

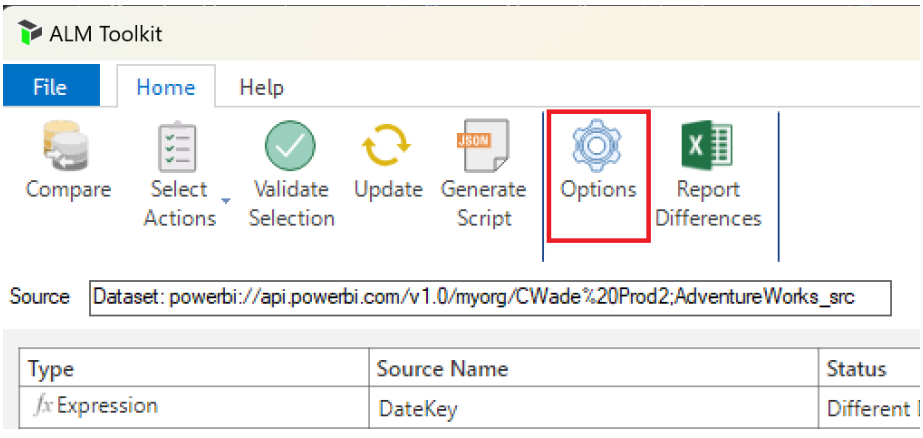
- Connection information for source and target models such as model.
- Comparison options.
- Skipped actions for model differences.



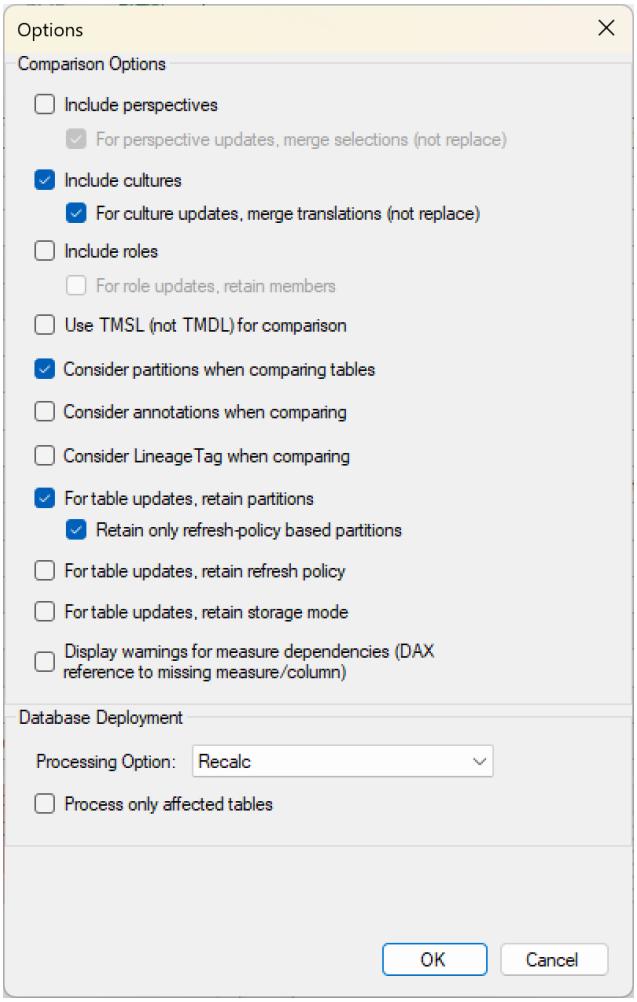
ALMT files can be used as deployment configurations. For example, you can have separate ALMT files for development, test and production environments. Saved skipped actions often include roles with different members in each environment, and data sources pointing at different instances of data sources.

Comparison options

Click Options on the ribbon.



The Options dialog is displayed.



Include perspectives

Excludes perspectives from comparison if unchecked.

For perspective updates, merge selections (not replace)

When merging models, it may be useful to create selections from a source perspective without losing existing selections that were already in the target.

Include cultures

Excludes cultures from comparison if unchecked.

For culture updates, merge translations (not replace)

The same principle is applied as when merging perspective selections. When the merge translations checkbox is checked, existing translations will not be removed. Different translations for an object property that is both in the source and target cultures will be overwritten.

Include roles

Excludes roles from comparison if unchecked.

Use TMSL (not TMDL) for comparison

Reverts to TMSL format instead of TMDL.

Consider partitions when comparing tables

When unchecked, partitions definitions are not included in the object definitions for tables. This means that tables with different partitions – but otherwise same definitions – are considered equal and automatically skipped.

With this option unchecked, partitions are not included in the JSON object definitions, and such tables are considered to have the same definition.

Consider annotations when comparing

When unchecked, annotations are not included in object definitions.

Consider LineageTag when comparing

When unchecked, LineageTags are not included in object definitions.

For table updates, retain partitions

In some cases, it may be necessary to perform an update on a table and still retain partitions in the target. For example, specifying a display folder has no structural impact on the list of columns and does not require rebuilding partitions. This option retains target partitions when checked even for table updates.

Retain only refresh-policy based partitions

Power BI models can use the incremental refresh feature for automatic partition management. This option limits partitions retained to those generated by incremental-refresh policies.

For table updates, retain refresh policy

In some cases, refresh policies may differ across environments. For example, a model instance in a Development environment may have a smaller archive range than instances on higher environments. This option retains refresh policies when checked even for table updates.

For table updates, retain storage mode

Power BI supports composite models where the storage mode is specified at the table level. This option retains the storage mode for table updates.

Display warnings for measure dependencies (DAX reference to missing measure/column)

When checked, warnings are displayed for measures and KPIs that reference nonexistent columns or measures. This option **does not affect whether actions are allowed to take place**. This is in contrast with other warnings – for example, invalid dependencies on M expressions and data sources – that do prevent actions from taking place

Database deployment

Processing option

The Processing option determines if refreshess are performed when updating the target model. Any sizeable refreshes are recommended to be run from another tool such as SSMS for additional diagnostic

capability. Process Recalc is recommended in most cases because it is lightweight and ensures the target model can be queried, without having to perform potentially time-consuming refresh operations.

The following options are provided, which align with the Analysis Services processing modes documented here.

- Recalc
- Default
- Do Not Process
- Full

Process only affected tables

When checked, only tables affected by the comparison with Create or Update actions are refreshed.

Command-line execution

Run the BismNormalizer.CommandLine.exe executable in the installation directory (typically "C:\Program Files\Power BI ALM Toolkit\Power BI ALM Toolkit") from the command line passing the ALMT file as an argument. This enables automation scenarios.

Syntax

```
BismNormalizer.CommandLine.exe AlmtFile [/Log:LogFile]
   [/Script:ScriptFile]
   [/Skip:{MissingInSource | MissingInTarget | DifferentDefinitions}]
   [/CredsProvided:{True | False}]
   [/UpgradeCompatLevel:{True | False}]
   [/SourceUsername:SourceUsername]
   [/SourcePassword:SourcePassword]
   [/TargetUsername:TargetUsername]
   [/TargetPassword:TargetPassword]
   [/WorkspaceServer:WorkspaceServer]
```

Arguments

AlmtFile

Full path to the .ALMT file.

/Log:LogFile

All messages are output to LogFile. If the log file already exists, the contents will be replaced.

/Script:ScriptFile

Does not perform actual update to target model; instead, a deployment script is generated and stored to ScriptFile.

/Skip:{MissingInSource | MissingInTarget | DifferentDefinitions}

Skip all objects that are missing in source, missing in target, or with different definitions. This is in addition to the skip actions already defined in the ALMT file.

It is possible to pass a comma-separated list of multiple skip options. For example, "/Skip:MissingInSource,DifferentDefinitions" will skip all objects that are missing in source and those with different definitions.

/CredsProvided:{True | False}

Flag to indicate that credentials are provided as command-line parameters to connect to Analysis Services.

/UpgradeCompatLevel:{True | False}

Flag to indicate whether target model compat levels should be upgraded to that of the source model if it's less than that of the source model.

/SourceUsername:SourceUsername

Username credentials to connect to the source model.

/SourcePassword:SourcePassword

Username password to connect to the source model.

/TargetUsername:TargetUsername

Username credentials to connect to the target model.

/TargetPassword:TargetPassword

Username password to connect to the target model.

/WorkspaceServer:WorkspaceServer

For SMPROJ sources/targets only, use this workspace server instead of integrated workspace.

Examples

The following example updates the target model, logging progress and error messages for later review.

```
BismNormalizer.CommandLine.exe TabularCompare1.ALMT /Log:log.txt
```

The following example does not update the target model. Instead, a script is generated and progress is logged.

```
BismNormalizer.CommandLine.exe TabularCompare1.ALMT /Log:log.txt /Script:script.xmla
```

The following example updates the target model and progress is logged. None of the objects missing in source are deleted from the target model.

```
BismNormalizer.CommandLine.exe TabularCompare1.ALMT /Log:log.txt /Skip:MissingInSource
```

Passwords and processing

ALM Toolkit in command-line mode does not set passwords or data source credentials. They need to be set separately, either manually or securely as part of a build. Automated processing (data refresh) therefore also needs to be set up separately if required.

User data

ALM Toolkit does not send or use any user data such as PII data to any server or service. User configuration data is stored locally in the installation directory for app settings and things of that nature.