

In my Power bi refresh service I have the following status: 1- Pending 2- RefreshInProgress 3- RefreshComplete when the refresh completes, I need to do some reconciliation between the data loaded to Power BI and the data source (Databricks), I need an intermediate status for the cases where numbers do not match

ChatGPT said:



You could introduce a new status like **"ReconciliationFailed"** or **"MismatchDetected"**. This status can indicate that the refresh has completed, but discrepancies were found during the reconciliation process. Here's how the statuses could look:

1. **Pending**
2. **RefreshInProgress**
3. **RefreshComplete**
4. **MismatchDetected** (new intermediate status when data between Power BI and Databricks does not match)
5. **ReconciliationComplete** (optional: if you want to indicate that reconciliation has succeeded)

With this status, you can trigger further actions like alerts or detailed logging to address the mismatch before proceeding.

You said:

I have a power bi semantic model which is getting refreshed programmatically tmsl script and powershell, in some case I have noticed that the refresh doesn't succeed but the tmsl execution doesn't return any error or information about any issue, when this happens I need to be able to detect it so that I can take an action on it. what can I do/

ChatGPT said:



To detect cases where the TMSL execution does not throw errors but the refresh isn't successful, you can implement a few strategies:

1. **Check the Refresh Status via Power BI REST API:** After executing the TMSL script, use the Power BI REST API to check the status of the refresh operation. The REST API provides a way to query the status of refresh operations, allowing you to verify if the refresh has completed successfully.
2. **Implement a Data Validation Step:** After the refresh completes, perform a data validation step to ensure that the data is updated as expected. This can include comparing row counts or checking specific measures in the dataset. If the validation fails, it can trigger a custom alert or action.
3. **Monitor Last Refresh Time:** Track the `lastRefreshTime` of the dataset before and after running the refresh. If the `lastRefreshTime` does not update after the TMSL execution, it could indicate that the refresh did not succeed even if no errors were thrown.
4. **Log Analysis:** Enhance your logging mechanism to capture more detailed information about the refresh process. While TMSL itself may not return errors, additional logging at the script or PowerShell level could help identify anomalies during the refresh execution.
5. **Retry Logic:** Implement a retry mechanism when a refresh doesn't show a clear status update. If the refresh status remains unchanged for a certain period, you can trigger a refresh retry automatically.

These methods can help ensure that you catch scenarios where the refresh appears to be successful but is not properly applied.

**You said:**

can the refresh status also be checked using c# and TOM?

**ChatGPT said:**



Yes, you can use C# and the **Tabular Object Model (TOM)** to check the refresh status of a Power BI dataset. TOM is part of the **Microsoft.AnalysisServices.Tabular** namespace, and it allows you to interact with Power BI datasets programmatically. Here's a basic example of how you can check the refresh status using C#:

1. **Add Required NuGet Packages:** You'll need to add the `Microsoft.AnalysisServices.Tabular` NuGet package to your C# project.
2. **Use the TOM Library to Connect and Monitor Refresh Status:**

```
csharp
using System;
using Microsoft.AnalysisServices.Tabular;
using Microsoft.AnalysisServices.AdomdClient;

namespace PowerBIRefreshStatus
{
    class Program
    {
        static void Main(string[] args)
        {
            string serverName = "powerbi://api.powerbi.com/v1.0/myorg/yourWorkspace";
            string databaseName = "YourDatasetName";

            // Connect to the server
            var server = new Server();
            server.Connect(serverName);

            // Get the database (dataset)
            var database = server.Databases[databaseName];

            // Check the last refresh time
            Console.WriteLine($"Last Refresh: {database.Model.LastProcessed}");

            // If you want to check the status of a refresh operation
            var refreshState = database.Model.RequestRefreshStatus;
            Console.WriteLine($"Current Refresh State: {refreshState}");

            // Logic to detect issues (for example, if the refresh state is not expected)
            if (refreshState != RefreshState.Completed)
            {
                Console.WriteLine("Warning: The refresh may not have completed successfully.");
                // Take appropriate actions like logging or triggering a retry
            }

            // Close the connection
            server.Disconnect();
        }
    }
}
```

## Key Parts of the Code:

- **Connect to the Power BI Service:** You connect to the Power BI service using the `Server` object with the appropriate server name (your workspace URL).

- **Retrieve the Database (Dataset):** Use the `server.Databases` collection to access the specific dataset you want to monitor.
- **Check `RequestRefreshStatus`:** This property tells you the current state of the refresh operation.
- **LastProcessed Property:** This gives the last time the model was processed, which you can compare to the expected refresh time.

## Things to Consider:

- **Authentication:** Ensure that your application has the necessary authentication to connect to the Power BI service. You may need to use service principal authentication or integrate Azure Active Directory (AAD) OAuth flows.
- **Error Handling:** Add appropriate error handling to catch connection issues or exceptions during refresh checks.

This approach allows you to integrate status checks directly into your C# application, offering more control over how you monitor refresh processes.

You said:

what is good name for a column in table used to log errors?

ChatGPT said:



A good name for a column in a table used to log errors should be descriptive and convey the purpose clearly. Here are a few suggestions:

1. **ErrorMessage** - This is straightforward and indicates that the column holds error messages.
2. **ErrorDetails** - This can be used if the column includes detailed information about the error.
3. **ErrorDescription** - A slightly more descriptive alternative to "ErrorMessage."
4. **ErrorLog** - A general term that suggests the column contains logs of errors.
5. **ErrorInfo** - A concise name that indicates the column holds information related to an error.
6. **ErrorSummary** - Useful if you intend to store a brief description or summary of the error.

The choice depends on the level of detail you plan to store and your naming conventions. If it's just a short message or code, **ErrorMessage** or **ErrorDescription** is suitable. If it includes more detailed information, like stack traces or extended descriptions, **ErrorDetails** might be more appropriate.

You said:

in which version of `Microsoft.AnalysisServices.Tabular` is "`RequestRefreshStatus`" available?

ChatGPT said:



The `RequestRefreshStatus` property is not a direct property available in **Microsoft.AnalysisServices.Tabular**. Instead, tracking the refresh status using TOM (Tabular Object Model) typically involves monitoring properties like:

- `Model.LastProcessed`: This indicates the last time the model (dataset) was processed.
- `Model.LastUpdate`: Similar to `LastProcessed`, it gives details about the last update.

For monitoring refresh operations in Power BI specifically, you would use the Power BI REST API for datasets, as TOM does not directly expose a `RequestRefreshStatus` property for ongoing refresh monitoring.

To handle refresh statuses programmatically, here's what you can do:

1. **Power BI REST API:** If you want to monitor a refresh status in a more detailed manner, you should use the Power BI REST API in combination with C#. You can make HTTP requests to endpoints like:

- GET  
`https://api.powerbi.com/v1.0/myorg/groups/{groupId}/datasets/{datasetId}/refreshes`

This endpoint provides details about refresh history, including status (Completed, Failed, etc.).

2. **Monitor Model State:** Using TOM, you can only observe the model's state and the last processed time, but it won't give real-time refresh

**You said:**

can I use xmla to get the refresh status?

**ChatGPT said:**



Yes, you can use **XMLA (XML for Analysis)** to get the refresh status of a Power BI dataset. XMLA endpoints are powerful and provide detailed control and information about Power BI datasets and models, similar to how you interact with SQL Server Analysis Services (SSAS).

To retrieve the refresh status using XMLA, you can execute an XMLA query against the dataset to check the state or status of partitions or processing events. Here's how you can approach this:

## 1. Connecting to the XMLA Endpoint:

You need to connect to your Power BI workspace using the XMLA endpoint. The endpoint typically looks like this:

```
bash
powerbi://api.powerbi.com/v1.0/myorg/YourWorkspaceName
```

## 2. Using Tools like SSMS (SQL Server Management Studio):

If you're testing or working interactively, you can connect to the XMLA endpoint of your Power BI workspace using SSMS and run queries directly to inspect the model, tables, and partitions.

## 3. Example XMLA Query for Status:

To check the refresh status of a specific partition or table, you can use the following XMLA query structure:

```
xml
<Discover xmlns="urn:schemas-microsoft-com:xml-analysis">
  <RequestType>DISCOVER_XML_METADATA</RequestType>
  <Restrictions>
    <RestrictionList>
      <DATABASE_NAME>YourDatasetName</DATABASE_NAME>
    </RestrictionList>
  </Restrictions>
  <Properties>
    <PropertyList>
      <Catalog>YourDatasetName</Catalog>
    </PropertyList>
  </Properties>
</Discover>
```

This query will provide metadata about the dataset, including information on tables and partitions. You can inspect this metadata for refresh-related details.

## 4. Programmatically Using C# with XMLA:

If you want to execute XMLA queries programmatically, you can use the AdomdClient library in C# to connect and run the XMLA command:

```
csharp
using Microsoft.AnalysisServices.AdomdClient;

string connectionString = "Data Source=powerbi://api.powerbi.com/v1.0/myorg/YourWorkspaceName;";
using (AdomdConnection connection = new AdomdConnection(connectionString))
{
    connection.Open();

    string xmlaQuery = @"
        <Discover xmlns='urn:schemas-microsoft-com:xml-analysis'>
            <RequestType>DISCOVER_XML_METADATA</RequestType>
            <Restrictions>
                <RestrictionList>
                    <DATABASE_NAME>YourDatasetName</DATABASE_NAME>
                </RestrictionList>
            </Restrictions>
            <Properties>
                <PropertyList>
                    <Catalog>YourDatasetName</Catalog>
                </PropertyList>
            </Properties>
        </Discover>";

    using (AdomdCommand command = new AdomdCommand(xmlaQuery, connection))
    {
        using (XmlReader reader = command.ExecuteXmlReader())
        {
            // Process the XML response to extract the refresh status or metadata details.
        }
    }
}
```

Replace "YourWorkspaceName" and "YourDatasetName" with your actual workspace and dataset names.

## 5. Inspecting DISCOVER\_TRACE for Detailed Information:

For even more detailed information about refresh operations, you can use the DISCOVER\_TRACE query type to inspect processing events, which might show more details about the refresh process.

### Summary:

- XMLA provides access to detailed metadata and information about Power BI datasets and models.
- For tracking refresh statuses specifically, you can query partitions or use DISCOVER\_TRACE to get processing details.
- This can be done interactively through tools like SSMS or programmatically using libraries like AdomdClient in C#.

Using XMLA gives you a deeper level of access compared to the REST API, allowing you to inspect model metadata directly, but it may require more detailed query handling.

**You said:**

"For tracking refresh statuses specifically, you can query partitions or use DISCOVER\_TRACE to get processing details." How can I track partition refresh specifically?

2/2

**ChatGPT said:**



To track the refresh status of specific partitions using XMLA, you can query the status of partitions directly, which helps you understand whether each partition has been processed as expected. Here's a more detailed guide on how to track the refresh status of partitions using XMLA:

## 1. Understanding the XMLA Structure for Partition Queries

To check the status of partitions in a Power BI dataset, you typically use an XMLA DISCOVER request or a SELECT statement that targets the processing history or metadata of partitions.

A common XMLA query to retrieve partition processing information would look like this:

```
xml
<Discover xmlns="urn:schemas-microsoft-com:xml-analysis">
  <RequestType>DISCOVER_PARTITIONS</RequestType>
  <Restrictions>
    <RestrictionList>
      <DATABASE_NAME>YourDatasetName</DATABASE_NAME>
    </RestrictionList>
  </Restrictions>
  <Properties>
    <PropertyList>
      <Catalog>YourDatasetName</Catalog>
    </PropertyList>
  </Properties>
</Discover>
```

This query retrieves metadata related to partitions within a dataset, including their names, statuses, and other properties.

## 2. Example XMLA for Partition Processing Status

To focus on tracking the last refresh time and state of each partition, you can inspect the properties of each partition using XMLA. This would provide details like `LastProcessed` (indicating when the partition was last processed) or `State`.

Here's an example of querying metadata to get partition processing details:

```
xml
<Discover xmlns="urn:schemas-microsoft-com:xml-analysis">
  <RequestType>DISCOVER_XML_METADATA</RequestType>
  <Restrictions>
    <RestrictionList>
      <DATABASE_NAME>YourDatasetName</DATABASE_NAME>
      <CUBE_NAME>YourModelName</CUBE_NAME>
    </RestrictionList>
  </Restrictions>
  <Properties>
    <PropertyList>
      <Catalog>YourDatasetName</Catalog>
      <Format>Tabular</Format>
    </PropertyList>
  </Properties>
</Discover>
```

This retrieves metadata for all partitions under a specified model (`CUBE_NAME`). The response will contain XML data detailing each partition's properties, including `LastProcessed`.

## 3. Interpreting the XML Response

The response from the XMLA query will be in XML format. It will include details about each partition such as:

- PartitionID or Name: The name of the partition.
- LastProcessed: The date and time when the partition was last processed.
- State: The current state of the partition, such as Processed or Unprocessed.

By parsing this response, you can determine if a specific partition was successfully processed during the last refresh.

## 4. Using SSMS for XMLA Queries

If you want to interactively run XMLA commands, you can use **SQL Server Management Studio (SSMS)**:

- Connect to your Power BI workspace via its XMLA endpoint.
- Open a new XMLA query window.
- Paste the DISCOVER XMLA query and execute it.
- The result will provide information about the partitions of your dataset.

## 5. Programmatic Example Using C#

To perform this check programmatically using C#, you can use `AdomdClient` as shown in this simplified example:

```
csharp
using Microsoft.AnalysisServices.AdomdClient;
using System;
using System.Xml;

class Program
{
    static void Main(string[] args)
    {
        string serverName = "powerbi://api.powerbi.com/v1.0/myorg/YourWorkspaceName";
        string databaseName = "YourDatasetName";

        string xmlaQuery = @"
        <Discover xmlns='urn:schemas-microsoft-com:xml-analysis'>
            <RequestType>DISCOVER_XML_METADATA</RequestType>
            <Restrictions>
                <RestrictionList>
                    <DATABASE_NAME>YourDatasetName</DATABASE_NAME>
                    <CUBE_NAME>YourModelName</CUBE_NAME>
                </RestrictionList>
            </Restrictions>
            <Properties>
                <PropertyList>
                    <Catalog>YourDatasetName</Catalog>
                    <Format>Tabular</Format>
                </PropertyList>
            </Properties>
        </Discover>";

        using (AdomdConnection connection = new AdomdConnection($"Data Source={serverName};"))
        {
            connection.Open();
            using (AdomdCommand command = new AdomdCommand(xmlaQuery, connection))
            {
                using (XmlReader reader = command.ExecuteXmlReader())
                {
                    {
                        while (reader.Read())
                        {
                            // Parse the XML response to get partition details such as "LastProcessed" or "State".
                            Console.WriteLine(reader.ReadOuterXml());
                        }
                    }
                }
            }
        }
    }
}
```

This code snippet connects to the Power BI XMLA endpoint, runs an XMLA query to retrieve metadata about partitions, and outputs the details. You can parse the XML response to check for partition statuses.

## **Summary**

By using XMLA queries targeting partition metadata, you can track whether each partition of a Power BI model was processed correctly. This approach allows you to detect situations where specific partitions were not processed as expected, even if the overall refresh appears successful. For real-time monitoring or deeper insights, combining XMLA with REST API checks can provide a more comprehensive view of the refresh status.