Engine B Common Data Model (CDM) Implementation Guide

Table of Contents

[**Introduction** 2](#_Toc162461663)

[Purpose 2](#_Toc162461664)

[Scope 2](#_Toc162461665)

[Overview of Engine B CDM 2](#_Toc162461666)

[**System Requirements** 2](#_Toc162461667)

[Databricks Workspace Setup 2](#_Toc162461668)

[**Terminology** 3](#_Toc162461669)

[Common Data Folder 3](#_Toc162461670)

[Manifest File (\*.manifest.cdm.json) 3](#_Toc162461671)

[CDM Schema (model.json) 3](#_Toc162461672)

[Entities (<*entity\_name*>.cdm.json) 3](#_Toc162461673)

[Data producer 3](#_Toc162461674)

[Data consumer 3](#_Toc162461675)

[**CDM Structure and Components** 4](#_Toc162461676)

[**Data Preparation** 5](#_Toc162461677)

[Gather Entity Definitions 5](#_Toc162461678)

[Mapping the Data 8](#_Toc162461679)

[Engine B Manifest 9](#_Toc162461680)

[Explore Entities with Entity Navigator 10](#_Toc162461681)

[**Validation Procedures and Quality Checks** 10](#_Toc162461682)

[Validation Procedures 10](#_Toc162461683)

[Quality Assurance Checks 11](#_Toc162461684)

[**Maintenance and Updates** 11](#_Toc162461685)

[Regular Maintenance 11](#_Toc162461686)

[CDM Updates 11](#_Toc162461687)

[**Troubleshooting** 12](#_Toc162461688)

[Common Issues 12](#_Toc162461689)

[**Appendices** 12](#_Toc162461690)

[Appendix A: Engine-B GitHub Repo 12](#_Toc162461691)

# **Introduction**

## Purpose

This document serves as a comprehensive guide for integrating the Engine B Common Data Model (CDM) into your organization’s Databricks workspace, facilitating standardized data management and interoperability across systems.

## Scope

The document covers the entire process, from system setup and data preparation to the detailed steps involved in mapping data to the CDM and ensuring its quality and maintenance.

## Overview of Engine B CDM

The Engine B CDM provides a standardized model for organizing, storing, and managing data, ensuring consistency, accuracy, and interoperability of data across different systems and platforms.

# **System Requirements**

## Databricks Workspace Setup

The workspace should be automatically deployed along with other Azure resources.

# **Terminology**

## Common Data Folder

A folder in a data lake that conforms to specific, well-defined, and standardized metadata structures and self-describing data. These folders facilitate metadata discovery and interoperability between data producers and data consumers.

## Manifest File (\*.manifest.cdm.json)

A metadata file in a folder in a Data Lake Storage Gen2 instance that follows the Common Data Model metadata format and potentially references other sub-Manifest for nested solutions. If this file exists in such a folder, it's a Common Data Model folder. For more information, go to [Common Data Model: Introducing manifest](https://learn.microsoft.com/en-us/common-data-model/cdm-manifest).

## CDM Schema (model.json)

A metadata file in a folder in a Data Lake Storage Gen2 instance that follows the Common Data Model metadata format. If this file exists in such a folder, it's a Common Data Model folder. For more information, go to [the metadata file (model.json) for the Common Data Model](https://learn.microsoft.com/en-us/common-data-model/model-json).

## Entities (<entity\_name>.cdm.json)

A metadata file in the Common Data Model folder that contains the metadata about the specific entity, its attributes, semantic meanings of entity and attributes.

## Data producer

A service or app that creates data in Common Data Model folders in Azure Data Lake Storage Gen2.

## Data consumer

A service or app that consumes data in Common Data Model folders in Azure Data Lake Storage Gen2.

# **CDM Structure and Components**

Each Common Data Model folder contains these elements:

* The \*.manifest.cdm.json file

The \*.manifest.cdm.json file contains information about the content of Common Data Model folder, entities comprising the folder, relationships and links to underlying data files. The \*.manifest.cdm.json format allows for multiple manifests stored in the single folder providing an ability to scope data for different data consuming solutions for various personas or business perspectives. Each entity definition is in an individual file making managing, navigation and discoverability of entity metadata easier and more intuitive.

* The entity definition file

The \*.cdm.json file contains the definition for each Common Data Model entity and location of data files for each entity.

* The model.json file

The model.json metadata file contains semantic information about entity records and attributes, and links to underlying data files. The existence of this file indicates compliance with the Common Data Model metadata format; the file might include standard entities that provide more built-in, rich semantic metadata that apps can leverage.

* Data files

The data files in a Common Data Model folder have a well-defined structure and format (subfolders are optional, as this topic describes later), and are referenced in \*.manifest.cdm.json or in the model.json file. These files must be in .csv format, but we're working to support other formats.

The following diagrams show examples of a Common Data Model folder with \*.manifest.cdm.json and model.json.

Figure 1: \*.manifest.cdm.json

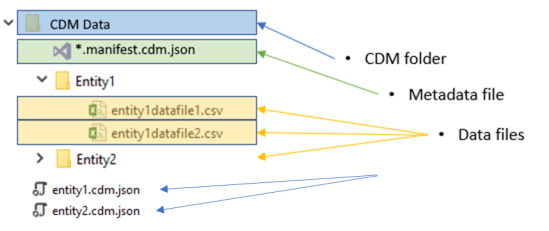
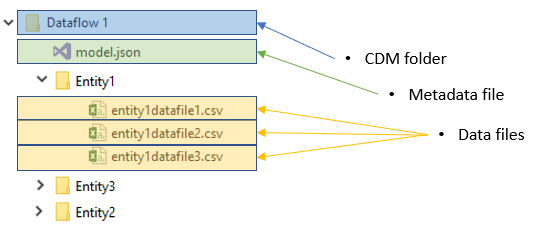


Figure 2: \*.model.json



**Data Preparation**

## Gather Entity Definitions

Consider the following data from your Trial Balance, illustrated as below. We need to find the correct data model (schema) to map out the values for your data.

Figure 3. Sample Trial Balance Data

A screenshot of a computer

Description automatically generated

As an illustrative example, we'll utilize the schema defined in **trialBalance.json**, which corresponds to the data we aim to structure. This schema is available in the GitHub repository at [EngineB-CDM/AuditCDMV3/trialBalance.json at master · Engine-B/EngineB-CDM (github.com)](https://github.com/Engine-B/EngineB-CDM/blob/master/AuditCDMV3/trialBalance.json)

Before you begin the mapping, its important to understand the Common Data Model outlined in the **trialBalance.json** file and what are the terminologies.

* **jsonSchemaSemanticVersion** identifies the version of the CDM object model.
* **imports** imports other schema documents that are needed for the current document.
* **corpusPath** is a path to a specific document that was specially configured to point to our schema document folder.
* **definitions** is where you describe the entity.
* **entityName** is the name of the current entity.
* **hasAttributes** contains the list of all the attributes thas the current entity has.

Figure 4. Sample Content of trialBalance.json schema, shortened for brevity.



# Mapping the Data

Mapping your data to this common data model involves aligning the attributes of your "**Trial Balance**" entity with the defined schema attributes in the common data model. Here's how you might go about it:

1. **amountBeginning**: This could map to the "Opening Balance" in your data. It represents the initial amount before any transactions.
2. **amountEnding**: Maps to the "Closing Balance". This is the final amount after accounting for all the transactions (debits and credits).
3. **amountCurrency**: If your data uses a single currency, you can specify it here (e.g., USD). This field won't directly come from your table but is assumed based on the entity's operational currency.
4. **balanceAsOfDate**: You might need to add this as metadata for your entity if it's not already present. It specifies the date for which the trial balance is accurate.
5. **glAccountName**: Can be mapped to the "Name" field in your data, representing the name of the account.
6. **glAccountNumber**: Maps to the "MainAccount", representing a unique identifier for the account in the general ledger.

Figure 5: Trial Balance Entity

A screenshot of a computer

Description automatically generated

To implement the mapping, you would adjust the attributes in the schema definition to align with your dataset's structure and terminology, ensuring each required field in the common data model has a corresponding source in your "Trial Balance" entity. Additionally, consider the need for transformations or additional calculations, such as converting all balances to a common currency if dealing with multicurrency transactions.

Engine B Manifest

The manifest file, a critical component, is available in the GitHub repository under the path [EngineB-CDM/Archive/AuditCDM/auditCDM.manifest.cdm.json at master · Engine-B/EngineB-CDM (github.com)](https://github.com/Engine-B/EngineB-CDM/blob/master/Archive/AuditCDM/auditCDM.manifest.cdm.json)

This file plays a key role by linking to our entity schemas, effectively acting as the access point for our entities. All manifest files are designated with a .manifest.cdm.json extension, making them identifiable.

Attention should be directed towards the **entityPath**, which signifies the direct path to an entity's definition within its schema. This path is formatted as [name of entity schema]/[name of entity], providing a clear route to the entity information. The manifest also has the capability to reference sub-manifests, expanding its range of use.

Figure 6: aduitCDM.manifest.cdm.json



Attention should be directed towards the **entityPath**, which signifies the direct path to an entity's definition within its schema. This path is formatted as [name of entity schema]/[name of entity], providing a clear route to the entity information. The manifest also has the capability to reference sub-manifests, expanding its range of use.

Top of Form

## Explore Entities with Entity Navigator

To explore and understand the schema documents within Engine B, you can utilize the [Entity navigator](https://microsoft.github.io/CDM/SchemaViz.html). Follow these steps to proceed:

1. Download the folder containing the schema documents from the GitHub repository located at [EngineB-CDM/AuditCDMV3 at master · Engine-B/EngineB-CDM (github.com)](https://github.com/Engine-B/EngineB-CDM/tree/master/AuditCDMV3)
2. Once the folder is downloaded to your local machine, open the Entity Navigator.
3. Choose the "Load from files" option.
4. Upload the previously downloaded folder.
5. Select the manifest file to initiate the exploration of entities.

# **Validation Procedures and Quality Checks**

## Validation Procedures

* **Data Integrity Checks:** Implement SQL scripts or use data validation tools to ensure data integrity. Checks should include primary and foreign key constraints to guarantee referential integrity.
* **Cross-Reference Checks:** Use cross-reference validations to ensure that data mapped from different sources aligns correctly with the corresponding entities and attributes in the CDM.
* **Logical Consistency Checks:** Validate the logical consistency of the data. This involves ensuring that the data adheres to business rules defined for your CDM entities.

## Quality Assurance Checks

* **Completeness Checks:** Develop scripts to verify that all required fields in the CDM entities are populated and that mandatory relationships between entities are established.
* **Consistency Checks:** Ensure data consistency across different entities and attributes. This can involve checking for duplicate records, validating data formats, and ensuring values are within expected ranges.
* **Accuracy Verification:** Sample data and manually verify against source systems or use automated tools that can compare CDM data against source data to ensure accuracy.

# **Maintenance and Updates**

## Regular Maintenance

* **Data Load Monitoring:** Implement monitoring tools to track the success and performance of data loads into the CDM. Identify and address load failures or performance bottlenecks promptly.
* **Performance Tuning:** Regularly review and optimize the performance of the database hosting the CDM, including indexing strategies, query optimizations, and hardware resources.

## CDM Updates

* **Version Tracking:** Utilize GitHub to track versions of the CDM schema. Monitor the Engine-B/EngineB-CDM GitHub repository for updates and changes.
* **Change Management:** Develop a process for assessing, testing, and implementing updates from the GitHub repository. This includes understanding the impact of schema changes on existing data and integration points.

# **Troubleshooting**

## Common Issues

* **Data Mapping Errors:** Issues arising from incorrect field mappings. Solution: Review mapping documents and adjust mappings as necessary.
* **Data Quality Issues:** Inconsistencies or inaccuracies in the data. Solution: Implement more stringent data validation checks and enhance data cleansing processes.
* **Performance Issues:** Slow data loads or queries. Solution: Conduct performance tuning, review indexing strategies, and consider hardware upgrades if necessary.

# **Appendices**

## Appendix A: Engine-B GitHub Repo

For the latest updates and documentation on the CDM schema, refer to the Engine-B GitHub repository at: [Engine-B/EngineB-CDM: Audit CDM](https://github.com/Engine-B/EngineB-CDM).