Metadata Management Tutorial

Fundamentals Using Meta Integration® Metadata Management (MIMM)

Version 9.1.0 (8/25/2017)

TABLE OF CONTENTS

TABLE OF FIGURES

1 Introduction

The need for more sophisticated and precise metadata management is a growing concern for most large organizations. Nearly all components that comprise modern information technology, from CASE tools, ETL engines, Warehouses, BI, EAI environments, as well as metadata repositories, contain, and often derive their processing from, metadata. The metadata for these environments is distributed and duplicated, often times active, and generally represented in a variety of methodologies, depending upon the underlying technology they represent.

Meta Integration® Metadata Management (MIMM), provide strikingly expansive set of capabilities in many facets of metadata management, including:

- Business Glossary
- Data Governance
- Metadata comparison, integration, and mapping
- Data life cycle related metadata management
- Lineage and impact analysis
- Enterprise architecture development, management and deployment.
- Data documentation
- Data Mapping Specifications, design and forward-engineering

This document is the culmination of more than fifteen years of experience in supporting the enterprise metadata management and integration requirements of numerous clients. It presents in detail and with supporting tutorials metadata management process and methods, best practices, as well as, many strategic scenarios that leverage the Meta Integration® Metadata Management (MIMM) suite. These examples are comprehensive and directed at real-world examples tied to business-oriented goals and return on investment. In all, anyone who completes the exercises in this document should find it straight-forward to implement and deploy an effective and comprehensive metadata management environment.

Disclaimer

Some of the features detailed in this document may not apply and/or be available for the particular Meta Integration® Metadata Management (MIMM) edition you may have.

1.1 How to use this document

It is certainly possible to skip through the tutorials, and thus simply glean an "management-level understanding" of Meta Integration® Metadata Management (MIMM) and its use within a metadata management environment. However, it is not recommended that one try to skip parts of the tutorials and then try to go through later parts. When following through the tutorial sections, it is very important to respect the order of the steps (and the order sections/labs within each section). The results of preceding tutorials are re-used and built upon in each successive lesson.

In addition, it is important to ensure complete understanding of the conceptual background provided in the sections leading up to and supporting the tutorial material. Thus, one should not simply jump into the tutorials with carefully reviewing the concepts presented in that section.

As this document include hand-on tutorials, a great deal of specificity is required. This detail includes specifying particular CASE, ETL, BI, etc., vendor's tools. While Meta Integration® Metadata Management (MIMM) environment itself is capable of working with over 100 different versions of third-party tools (see http://www.metaintegration.net/Products/MIMB/SupportedTools.html), it is necessary for the clarity conciseness of the tutorials to limit the cadre of tools that will be referred to. Please note that it is not necessary to have these tools on-hand to get the full benefit of the tutorials. Remember also, though you may intend to use Meta Integration® Metadata Management (MIMM) suite of tools with many of the supported third-party tools not specified in the tutorial, it is still quite valuable to learn the processes, methods and best practices presented here. Then one may reuse what one has learned and apply that knowledge and skill to the particular set of tools that are critical to one's own enterprise.

1.2 Conventions used in the tutorial

The following font conventions will be used throughout the tutorial.

- User Interface item – New
- Submenu item – New › Folder
- Terminology item – *model content* item
- Name or label reference – Accounts Payable

2 About Metadata Management

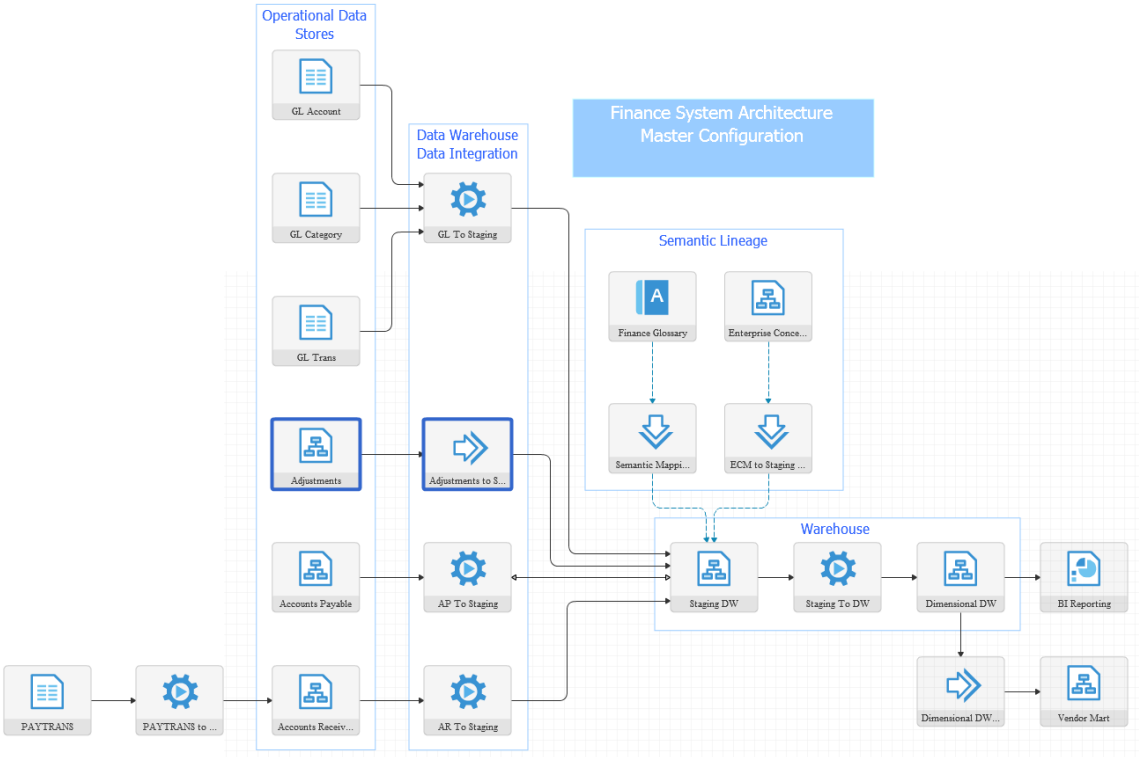


Figure 1 - Model Connection Diagram for System Metadata Management Exercises

The above figure represents an overview of the tutorial that this document will lead one through. While it appears here to be a fairly complicated diagram, this document will decompose and will walk one through the data and metadata management concepts encompassed.

The sample metadata represents what would be captured from operational finance based systems, including the operational data stores (ODS), data movement (ETL), warehouse (DW and DWS), and business intelligence (BI) reporting. In addition, a business glossary and enterprise conceptual data model (CDM) are also included and mapped to the operational system metadata.

2.1 General data and metadata management discussion

Metadata management is not an end in and of itself. There has to be real return on investment in what can be a very expensive and long term undertaking. The promise in terms of having a well architected and interoperable information systems environment is well understood by most people in the information management community. However, as with any foundation, it must be solid, well thought out, and appropriate to the need. In order to have an outrageously successful effort, then, the specific objectives must be very clearly defined, the organization as a whole must be able to leverage the results of the metadata efforts, and the infrastructure must be able to support all of the many methods, design tools, etc., that an organization is using and will adopt into the future.

It has been said many times that "Metadata and Architecture assets will outlive the tools, methodologies, and individuals used to produce them." Thus, it is *imperative* that your metadata be as independent as possible from a particular tool format, methodology or process. Having incredibly valuable and expensive work "locked" into a particular tool and methodology or process can spell doom for any metadata and data management effort. Metadata *must* also be accessible to the widest possible audience within an organization, in order to conclusively demonstrate to the technical and business communities that metadata is a valuable (and accessible) corporate asset.

2.1.1 Data and metadata are everywhere!

One of the key reasons seen for integrating and managing metadata across an enterprise is that, as we know, *data is everywhere*. The proliferation of different software applications that can accomplish more and more specific tasks, and the incredible cost to develop new ones, means that inter-operation, or enterprise application integration, with many disparate data sources is the only way to provide an integrated enterprise. After decades of trying, we have given up on the concept that there will be just one data source, virtual or otherwise, and interoperability is now known to be the key to integration. If data must exist in physically separate data stores, on a diverse array of platforms, then integrated enterprise data architecture will be required to "link" the disparate and possibly redundant data stores together via well-defined processes. Losing track of these "links" costs organizations millions of dollars annually, by increasing the risk of system errors, extending application development project timelines, increasing the cost/complexity of building decision support systems, and countless other ways. The inability to trace information across disparate processes and technologies has lead to a lack of trust by business management about the reliability of the content of corporate information. Having to then assure the quality of the information used to make the decision, inevitably causes delays in key business decision making. Thus is born the need for metadata repositories, metadata integration and management.

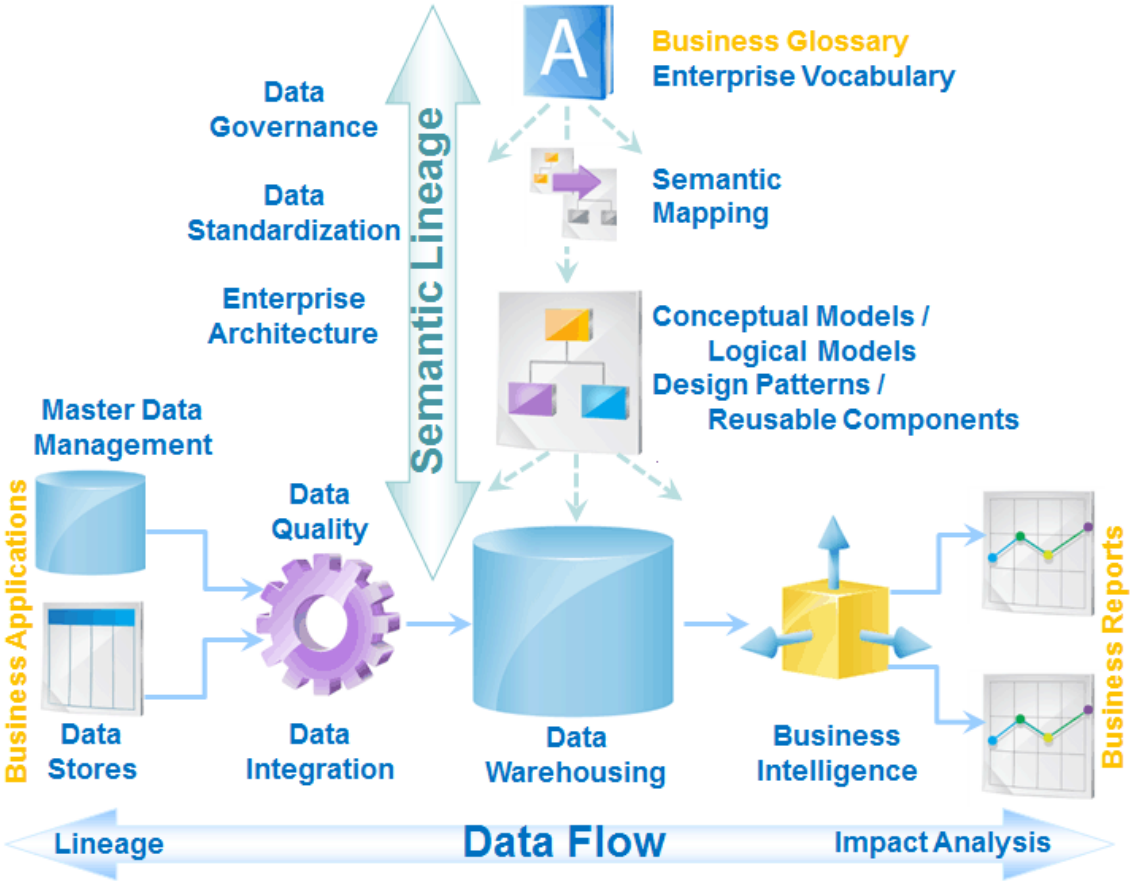


Figure 2 - Metadata is everywhere

What needs to be recognized, though, is that *metadata is also everywhere*. Data warehousing, business intelligence, CASE and ETL tools all have their own repositories, just about every application has its own data dictionary, XML carries the metadata with it in the message or document, and enterprise application integration environments have their own repositories and metadata mapping and integration facilities. Thus, the same difficult lesson applies: the truly effective and future proof way to integrate and manage this metadata is interoperability and enterprise repository integration. There is never going a single repository (metadata source), virtual or otherwise, and thus, metadata exchange and repository interoperability are key to metadata integration.

2.1.2 Strategy to support data and metadata management

In order to succeed, then, one must have a good enterprise repository integration environment. This philosophy is what the Meta Integration® Metadata Management (MIMM) repository solution and products are based upon. It is an integrated whole that can bridge the technical and non-technical aspects of metadata, while simultaneously bridging the chasm between the different metadata source and target systems that constitute any modern information management environment.

Metadata is everywhere. In all development scenarios, using CASE, ETL, BI, DW, ERP, EAI, etc., tools, MITI provides the glue that allows for integration and exchange of this information in as complete and robust manner possible.

As the metadata and data management market has evolved from CASE tools and repositories, to ETL tools with repositories, to BI tools with metadata management, Meta Integration® Metadata Management (MIMM) incorporated these metadata into an integrated meta-model supporting standards such as CDIF, IDEF1X, UML, and other more proprietary methodologies and related metadata in order to provide and integrated approach to the metadata integration need.

Along with the usefulness of metadata movement and integration, came the need for metadata management, change management, mappings, transformations and stitching.

In this way, Meta Integration® Metadata Management (MIMM) provides the only metadata repository on the market that can import and export the most complete metadata via so many bridges, and fully manage the current configuration, and mappings of the managed metadata. Meta Integration® Metadata Management (MIMM) allows one to manage and analyze the complete enterprise metadata life cycle across heterogeneous vendors. Solutions range from reusable metadata movement components to repository. Over 50 Model Bridges integrate standards like CWM, and the most popular modeling, ETL, and OLAP/BI tools. The open repository implements powerful Metadata, Browsing, Comparison, Integration, Mapping, and Reporting.

2.2 Metadata Repositories are Everywhere

Just as the idea of a single database failed for obvious reasons, the concept of a sinlge metadata repository has also proven unrealistic. As the metadata is everywhere, so are the repositories.

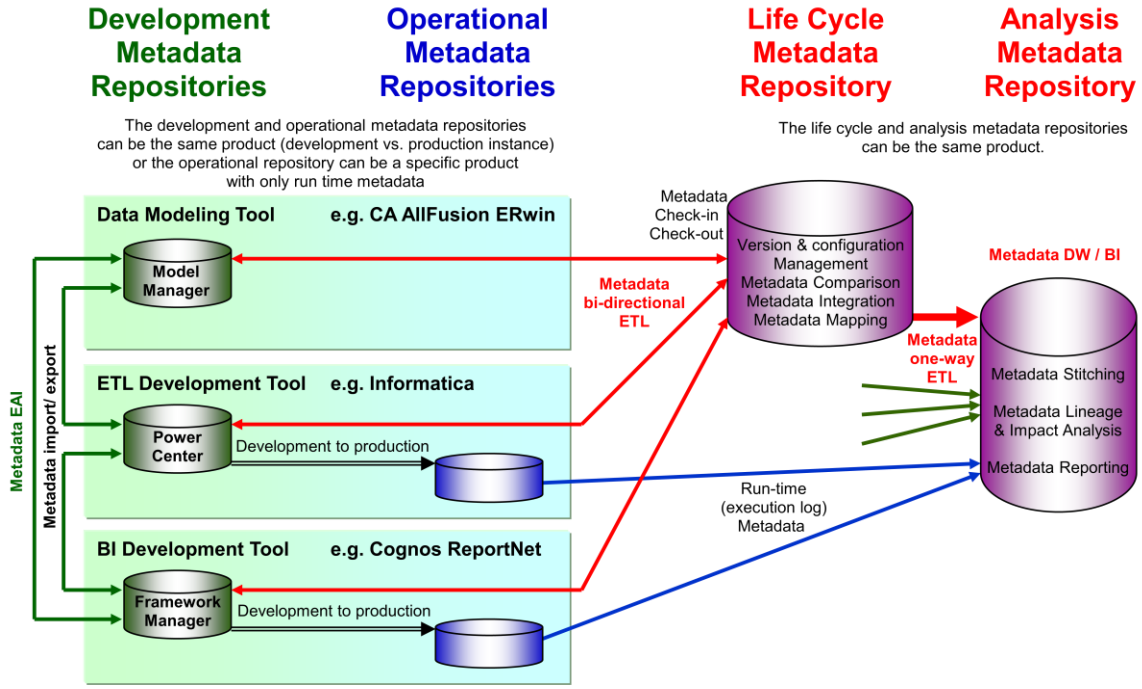


Figure 3 - Metadata Repositories are everywhere!

What becomes critical then, is not the replacement or merging/integration of these very necessary repositories into one large "meta-repository". Instead, the important keywords are *integration*, *support for change*, *and change management*, and *metadata lifecycle management*.

In the next section, this document will outline what the Meta Integration® Metadata Management (MIMM) comprises. The following section of this part will then continue to detail the steps required to develop an effective enterprise architecture based metadata management environment, including the following steps:

- Identify data stores & associated design tools
- Identify data processes within ETL, BI and their associated tools
- Build the enterprise architecture and data flow

2.3 Meta Integration® Metadata Management (MIMM) Toolset

2.3.1 Metadata Movement Lifecycle

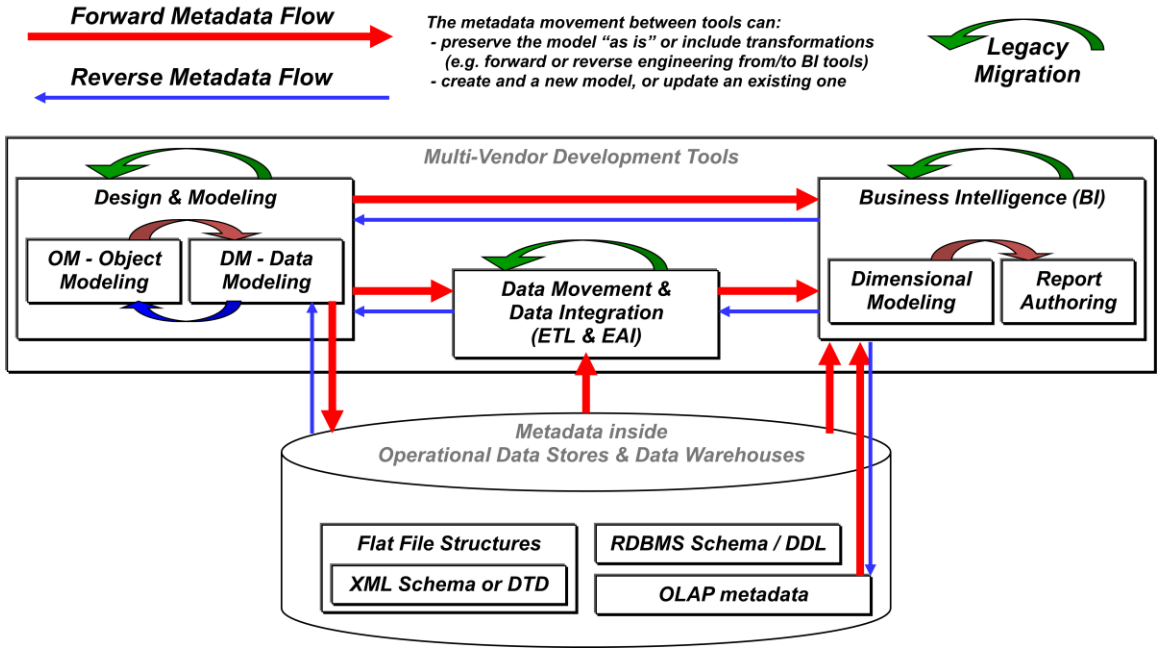


Figure 4 - Meta Integration® Metadata Management (MIMM) data and metadata movement lifecycle

Another perspective on the flow of metadata is to follow the flow of data. This figure depicts common examples where design and modeling tools are used to develop systems, databases and interfaces/messaging using data and object modeling methodologies. These models or metadata may then be moved or migrated (translated) forward in the data flow, as in this example where a CASE design model is reused by the ETL repository or the BI designer tool. Additionally, metadata may be migrated backward (in reference to data flow), again in this example due to a change in the reporting metadata that requires similar changes to the ETL and source system metadata.

In keeping with the flow, or "heart-beat", of the metadata lifecycle, these models are continually imported and exported into and out of the repository, mappings among them created, translated to other tools in the organization, mapped to standards, and given other business metadata, such as stewardship assignments.

It is critical that anyone managing change in the repository understand this metadata lifecycle. It is also essential that good metadata management strategies and practices are codified and followed within the organization. The tutorials will provide examples of these. It is up to the reader to define the specifics that best fit the organization.

2.3.2 Metadata Integration Capabilities

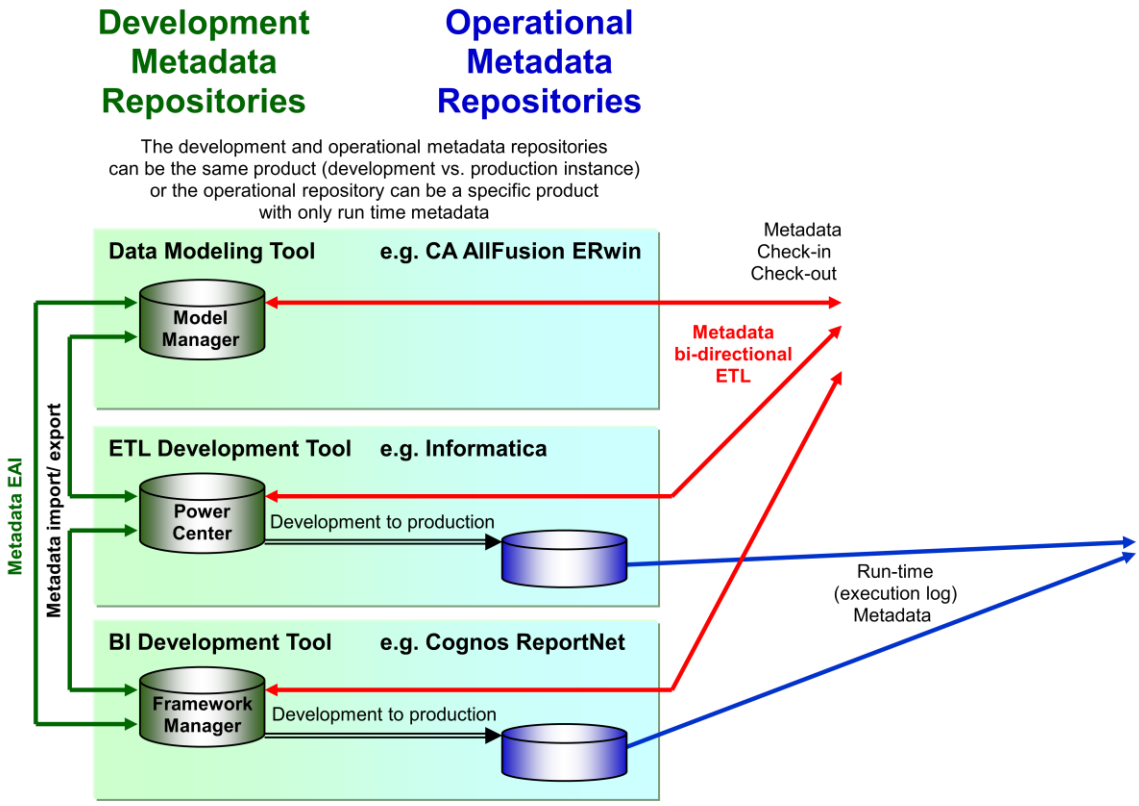


Figure 5 - Metadata management from a repository perspective

Metadata management means tool integration. Metadata is indeed everywhere must be extracted from repositories, design tools, file formats, database specifications, and the list goes on.

2.3.3 Metadata Management Capabilities

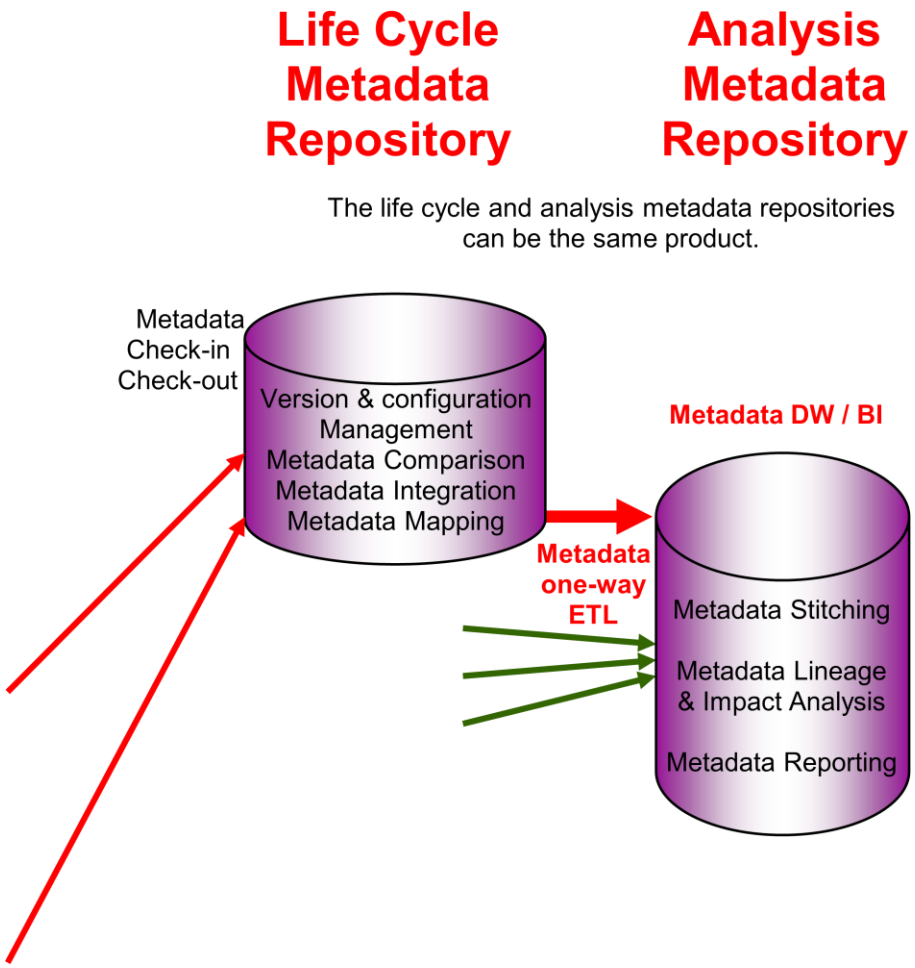


Figure 6 - Metadata Management Capabilities

2.3.3.1 Metadata Comparison

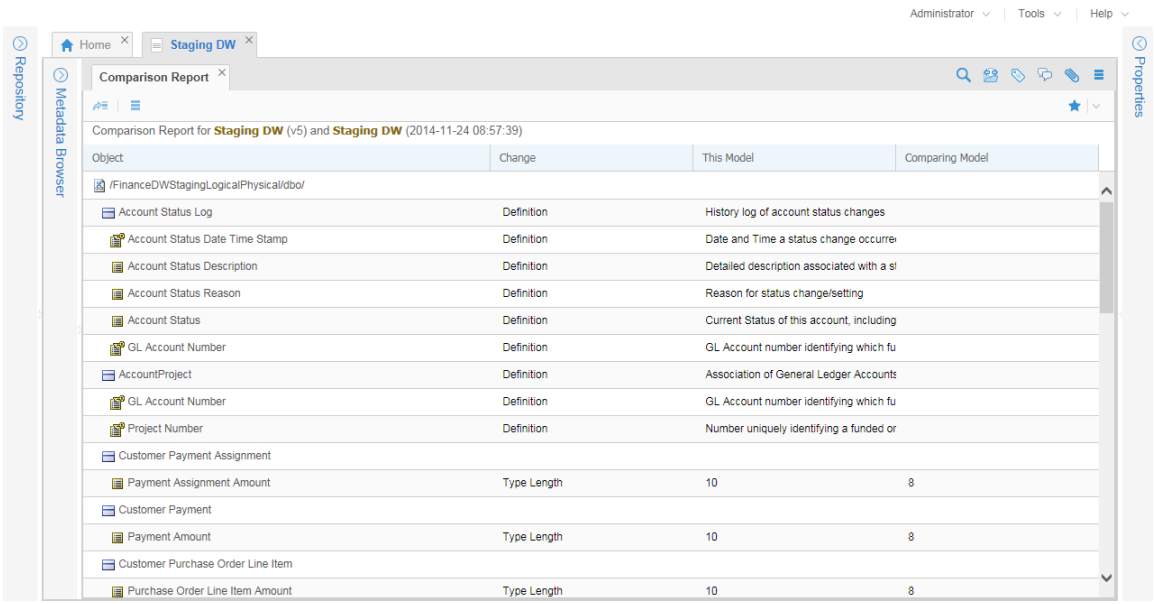


Figure 7 - Model comparison report

Complete metadata comparison capabilities are provided with Meta Integration® Metadata Management (MIMM). As all metadata is represented by an integrated metamodel at the core of Meta Integration® Metadata Management (MIMM), this comparison facility will provide comparisons across metadata from ALL of the sources supported, including design tools, ETL, BI, database, file formats, etc.

2.3.3.2 Metadata Mapping

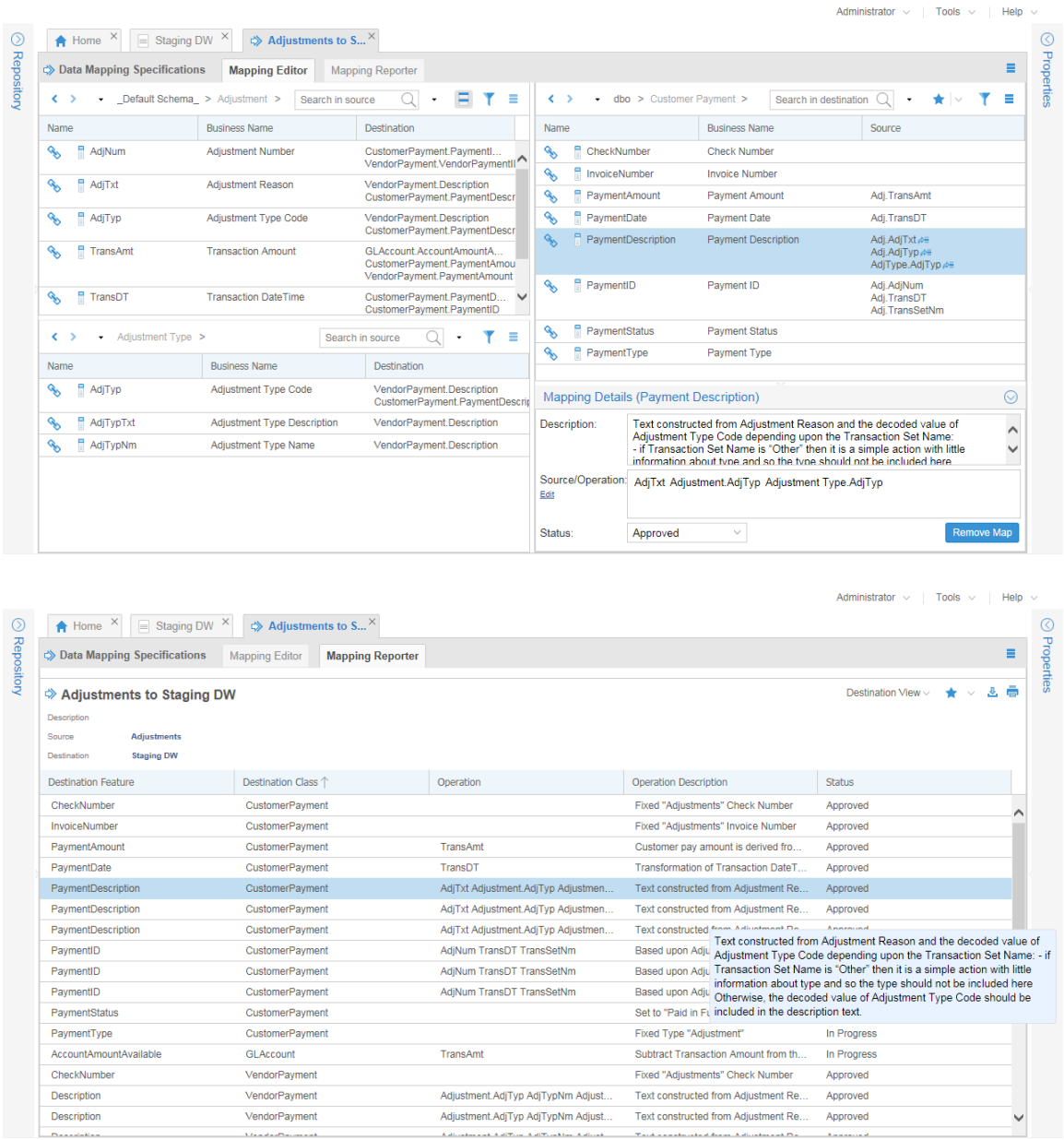


Figure 8 - Metadata Mapping

Once harvested, metadata captured within Meta Integration® Metadata Management (MIMM) may be mapped in a myriad of ways to any other metadata within Meta Integration® Metadata Management (MIMM). This ability is critical to the success of any metadata management solution. In particular one may define *data flow mappings* describing data movement type relationships and when a database is read and the results written to another database, as well as *semantic mappings* which identify semantic relationships between elements, oftentimes *conceptual* or *logical* in nature, such as for a data dictionary or conceptual model such as a UML model.

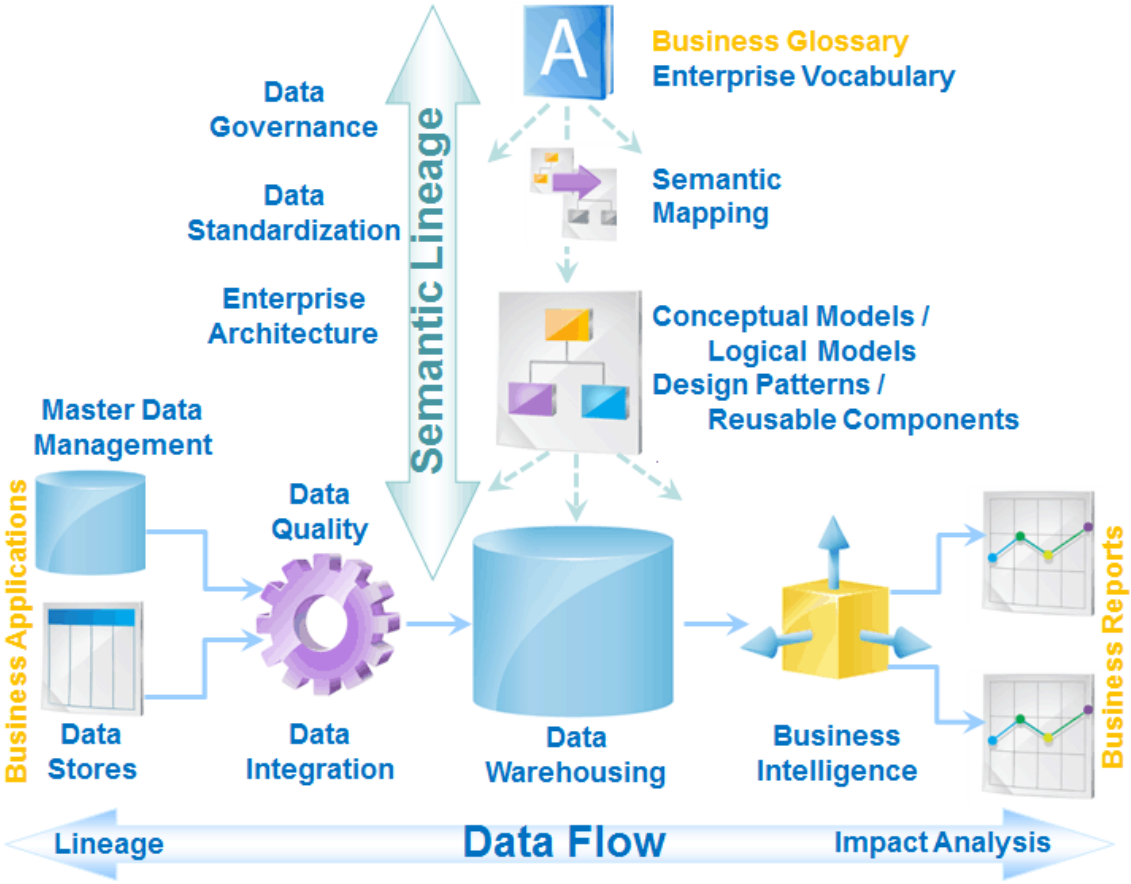


Figure 9 - Data flow vs. semantic mapping

2.3.3.3 Metadata Stitching

While a fairly technical subject, that is covered extensively in the tutorial, metadata stitching is fundamental to the correct and completely automated determination and analysis of the data flow and semantic lineage of metadata in the repository. It is also a critical concept supporting change management and the constant rate of updates and change which a repository must survive and even thrive in.

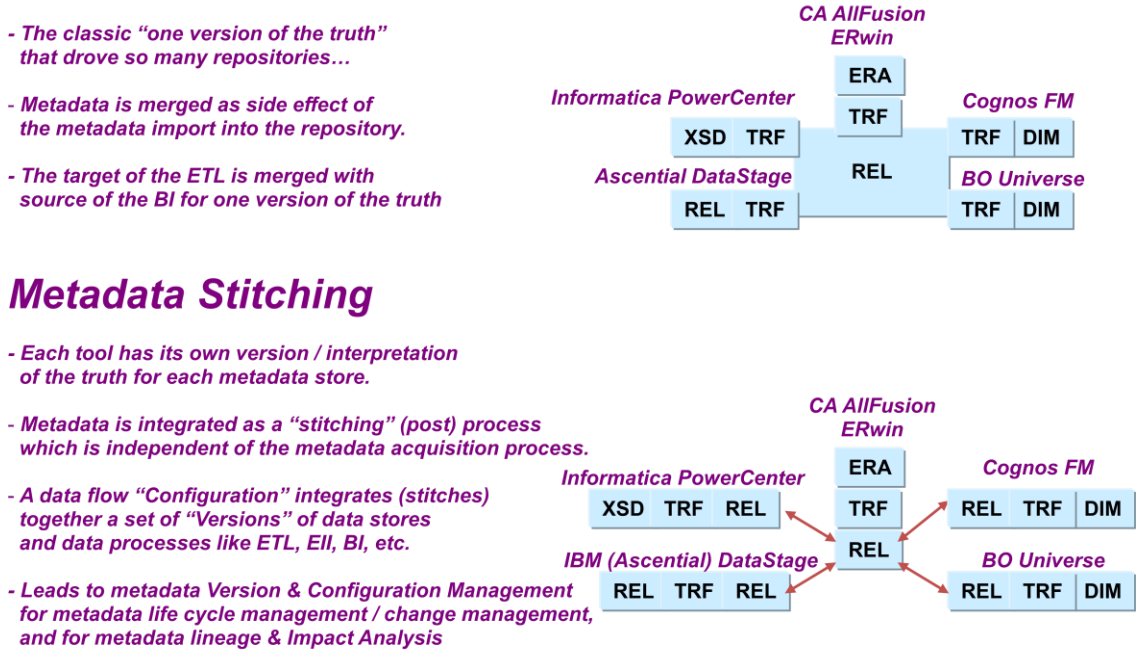
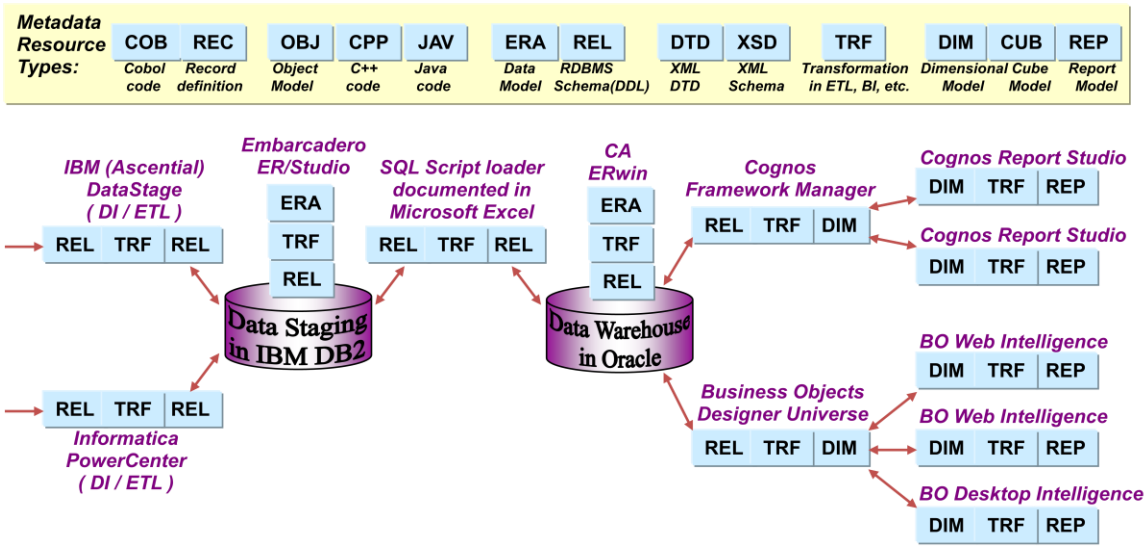


Figure 10 - Metadata Stitching versus Metadata Merging to represent and trace lineage

In addition, the metadata harvested by Meta Integration® Metadata Management (MIMM) is a tightly matched representation of the actual physical architecture (databases, ETL/BI/EAI/DII transformation and connection definitions, message schemas, etc.). Thus, the Configuration Editor, exploits this fidelity by allow the use to simply "define the connection rules" from the different data flow mappings defined by ETL/BI/etc. and their connection definitions to the different data stores (databases, design models, etc.).



In this way, the enterprise architecture is faithfully modeled, and data flow lineage is completely and accurately derivable.

2.3.3.4 Lineage and Impact Analysis

Once well managed, metadata is then open for detailed analysis, and true business level use cases may be solved. Meta Integration® Metadata Management (MIMM) support full technical and business level lineage and impact analysis to a degree only possible from an environment based upon the hundreds of bridges which scour the metadata from so many and varied a sources.

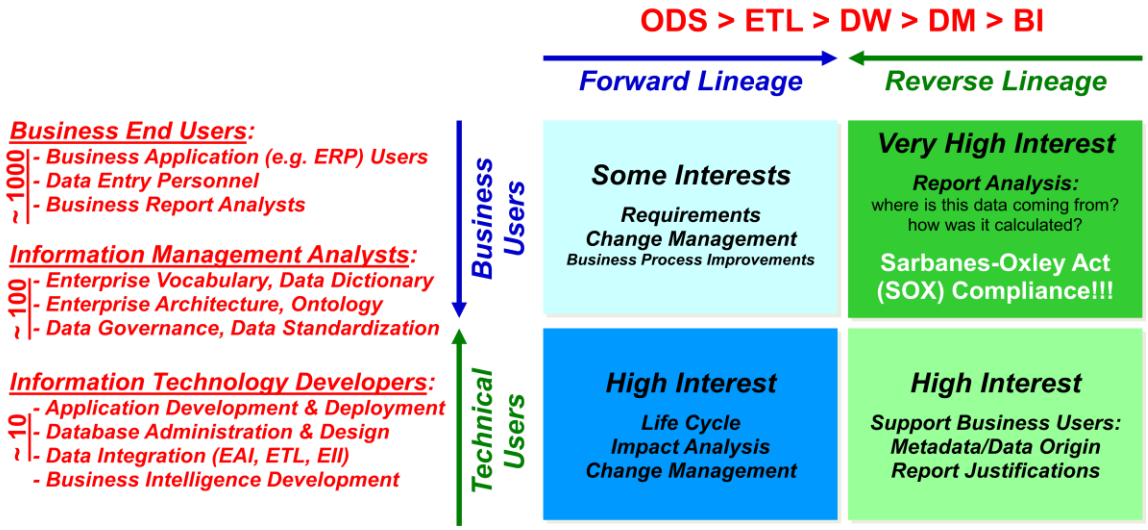


Figure 11 - Metadata Lineage and Impact Analysis Use Case Quadrant

The above quad chart shows four major use case categories in relation to the dimensions of *lineage analysis* and *user type*.

2.3.3.4.1 Business Users – Reverse Lineage

Of very high interest are the *reporting analysis* type use cases, generally posed as questions such as:

- Given an item on a report, what data entry system fields impact these results?
- Why are the numbers on this report the way that they are?
- How do change the system data to get the correct the results of this report?

This type of analysis, i.e., asking where the information comes from, is a question posed "upstream" in the dataflow. We refer to it as a *reverse lineage* question. When consumers of these reports ask these questions, a correct and responsive answer may be the most valuable information provided by a metadata management solution.

2.3.3.4.2 Technical User – Forward Lineage

Of especially high interest to the technical user are questions like:

- If I must change these elements (data type, code sets, etc.) in my operational data store, what are impacts downstream?

- This new ETL process is populating my staging warehouse in new ways, how does this change impact the OLAP model in my reporting services?

This type of analysis, i.e., asking what this information impacts, is a question posed "downstream" in the dataflow. We refer to it as a *forward lineage* (or *impact analysis*) question. It is a critical part of change management in any IT organization.

2.3.3.4.3 Technical Users – Reverse Lineage

Reverse lineage type questions may also be asked by more technical user, such as:

- How many systems are required to determine the dimensions for this portion of the OLAP model?
- A business report use is asking about the reasons for particular values on a report, so where does the data come from and how is it manipulated?

These questions are important for the IT organization as well, though often not seen to be as important as the preceding use case.

2.3.3.4.4 Business Users – Forward Lineage

Finally, business users may ask the forward lineage or impact analysis type of questions:

- If I make a change to this field, what reports will be impacted?
- How is this identity information merged with the personnel system information on these other reports?

2.3.3.5 Business Glossary

2.3.3.6 Model Documenter

2.3.4 Architecture

MIR is a three tier application

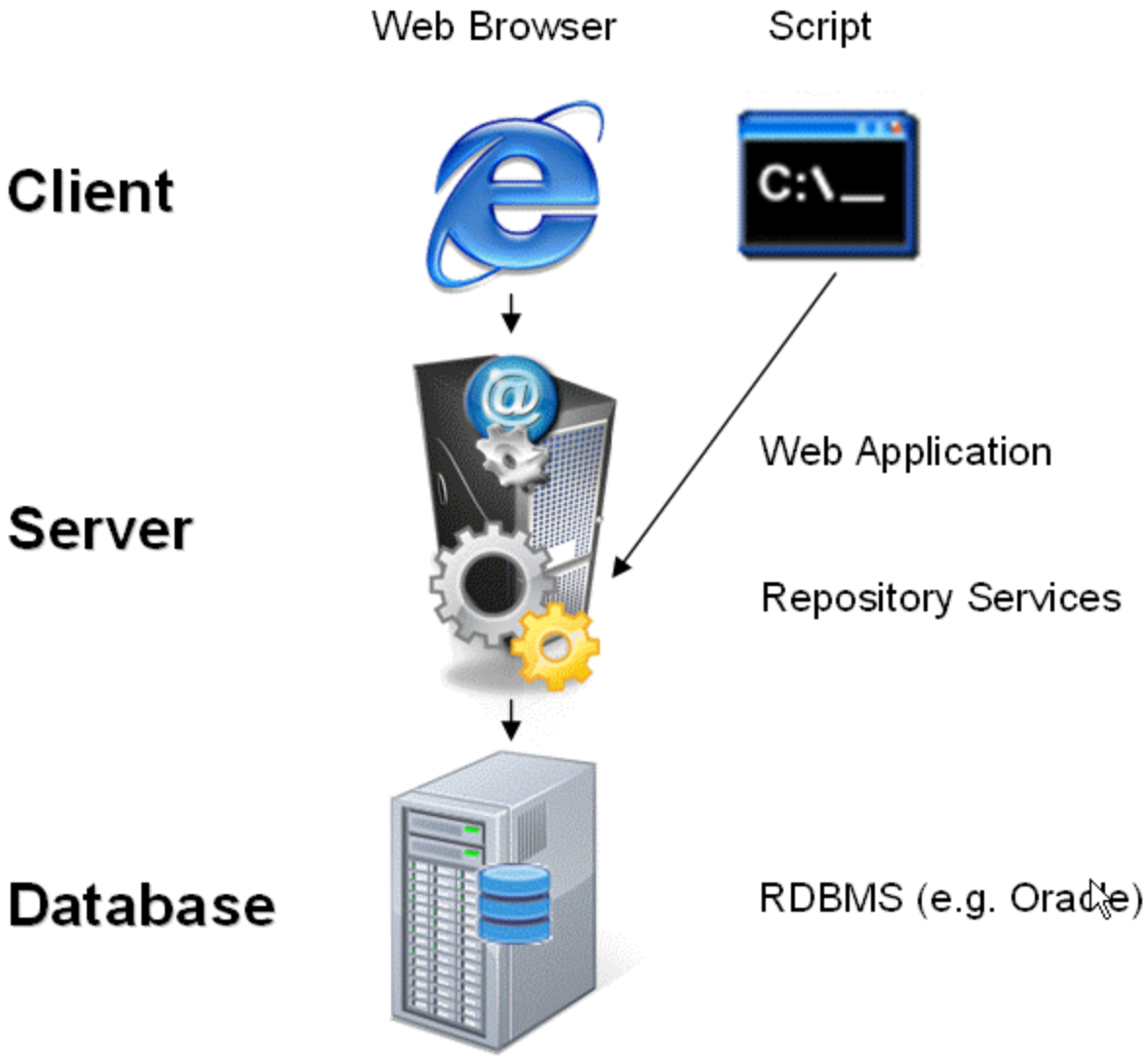


Figure 12 - Meta Integration® Metadata Management (MIMM) Three Tier Architecture

2.3.4.1 Client Tier

MIR provides Web browser and Scripting clients.

2.3.4.1.1 Web browser client

The Web browser client allows one to administer and utilize everything that Meta Integration® Metadata Management (MIMM) offers. The user interface is a rich web application that works across all major web browsers, built using a client-side web application JavaScript framework.

2.3.4.1.2 Scripting client

The Scripting client allows customers to automate administration task (e.g. create a [folder] and harvest a model upon change) and execute model management operations

(e.g. compare, migrate or stitch models). All basic capabilities available to the user interface are scriptable.

2.3.4.2 Server Tier

MIR Server is a web J2EE application (WAR) that can run in a web container/application server, like Tomcat or WebSphere. The server is logically divided into a Web application and Repository Services components.

2.3.4.3 Database Tier

Meta Integration® Metadata Management (MIMM) uses a database not only to store data but to run computations on. As a rule, Meta Integration® Metadata Management (MIMM) uses the database to perform computations on data that can potentially exceed the amount of available server memory. For example, a configuration may contain metadata totaling 10Gb or more, which cannot fit into server memory. Therefore, Meta Integration® Metadata Management (MIMM) traces lineage through a configuration using a database stored procedure (function).

When multiple users access Meta Integration® Metadata Management (MIMM) simultaneously the database can become the key feature for client performance. In this case, one should consider scaling up the database performance by adding more computational resources (e.g. CPUs) or speeding up the database storage (e.g. RAID, faster disks, etc.).

2.3.5 Deployment Architecture

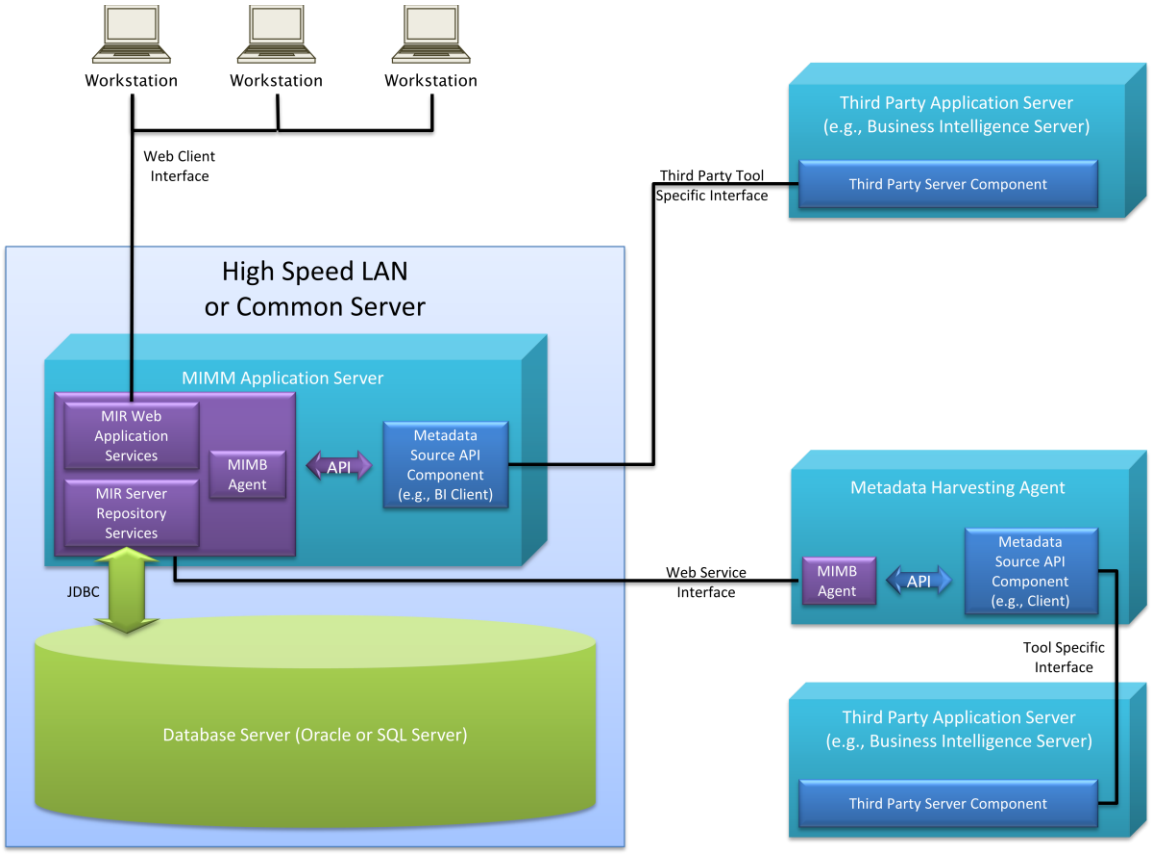


Figure 13 - Example Meta Integration® Metadata Management (MIMM) Deployment Architecture

Common deployment architecture for Meta Integration® Metadata Management (MIMM) includes:

- Meta Integration® Metadata Management (MIMM) Application Server
- Database Server
- One or more Metadata Harvesting Agents
- Workstations

2.3.5.1 Meta Integration® Metadata Management (MIMM) Application Server

The application server comprises the MIRServer providing most services used, a web server based MIRWeb server (by default, this is a Apache Tomcat installation), all of the MIMB bridges for metadata import and export to and from the repository, and generally an MIMBServer providing direct model to model metadata conversion capabilities.

2.3.5.2 Database Server

An Oracle or Microsoft SQL Server based database server where all repository information (metadata and otherwise) will be stored. It should be in "close proximity" to the application server in terms of network connectivity, as there is a high degree of traffic between these two.

2.3.5.3 Meta Harvesting Agent

A metadata harvesting agent is an MIMB server, installed on a machine remote to the Meta Integration® Metadata Management (MIMM) Application Server. These are often necessary when it is not possible to install APIs, which are required for harvesting bridge function, on the Meta Integration® Metadata Management (MIMM) Application Server itself. In this case an MIMB Agent is installed on this machine as well as the API for the metadata source technology (generally a client installation). Meta Integration® Metadata Management (MIMM) then utilizes this machine as an agent to harvest from or export to that source technology.

2.3.5.4 Workstations

All administrative and analysis activities are generally conducted from personal workstations with a supported web browser.

3 Metadata Harvesting

In order to design, build and maintain a metadata management environment, one must identify and *catalog* the assets in the current environment. In general, this process is broken into the cataloging of *data stores*, *data processes* and *conceptual models*.

3.1 From Data Stores and Data Processes To Enterprise Architecture

3.1.1 Identify data stores & associated design tools

With that overview of the Meta Integration® Metadata Management (MIMM), we may now return to the general metadata management discussion. One of the first steps in populating the repository and thus building up one's metadata management environment is an inventory of the metadata in your organization.

For the purposes of this document, the specific methodology used is immaterial. Instead, from the MIR perspective it is most important to analyze

- where the metadata resides
- what technology is required to extract it (and ultimately forward-engineered)
- what process or set of steps must be followed in order to ensure proper extraction

As was stated in previous sections, metadata is everywhere within any modern information management organization. Also, there are many repositories that are required to exist for each technology environment in order for it to perform properly. In order for these sources and targets to provide a coherent and coordinated environment, this metadata must be exchanged.

3.1.1.1 Interface technology issues

Fundamental to the repository is the ability to exchange metadata among the disparate sources and targets that exist in the enterprise and those that will be adopted into the future. The following table addresses how metadata is extracted from and returned to the different sources.

Metadata Source / Target	**POC**	**Reverse Engineering Method**	**Reverse Engineering Contents**	**Forward Engineering Method**	**Forward Engineering Contents**
Tool Name and Description	Point of Contact / Steward for Tool	(Tool-specific metadata bridge)	(Tool-specific contents, see http://www.metaintegration.net/Products/MIMB/Documentation/SupportedTools.html for specific details by source)	(Tool-specific metadata bridge)	(Tool-specific contents, see http://www.metaintegration.net/Products/MIMB/Documentation/SupportedTools.html for specific details by target)

Figure 14 - Classifying metadata sources and targets

Depending upon the particular source, there will most likely be different technology related issues to consider when extracting metadata. Generally these issues revolve around the type of interface required by the particular tool vendor. These interfaces may be divided into three categories:

Issues related to these different technology approaches are summarized in the following table and are also documented on the website:

Metadata Interface Type	Examples	Issues
Direct application programming interface (API)	• Repository bridges (e.g., BI tools, ETL tools)	• Generally must have client for repository with API installed on repository server • Sometimes will not cross OS-specific lines, e.g., COM/DCOM based API requires a Windows-based repository server
Import/Export file format	• Specific text or binary formats • CSV text files dumping tables from repositories • XML files (standards like OMG UML or CWM XMI or proprietary	• Sometimes incomplete or unreliable (try the export/re-import roundtrip test) • Sometimes multiple interpretations of the standards like CDIF and XMI
Native File Format	• Binary files of design tools or repositories • Text files, possibly structured in a language like the Rational Rose MDL • Direct database access (underlying repository)	• Reliable metadata when exporting. • Possibility of corrupting metadata when importing. • Proprietary solution specific to the tool vendor • May again require vendor API

Figure 15 - Types of metadata interfaces

Again, one should refer to the web site for specific details by source. It is highly encouraged that any organization building a metadata management environment develops a detailed set of tables like the ones above.

3.1.1.2 Metadata package structure

In order to capture that inventory as metadata within the Meta Integration® Metadata Management (MIMM), it is essential to develop a good product structure in the repository. The repository is capable of storing information about metadata within the repository using any [folder] structure that one defines. Folders may contain [folder]s,

and so on, providing one with a directory structure within the repository. The top directory is presented as a node with the name of the MIR server.

It is important that these structures reflect the enterprise, its processes and business rules. Different organizational structures may be necessary for various types of users of the repository. Meta Integration® Metadata Management (MIMM) provides an organizational structure based upon a generic large organization, as a repository load script for automatically generating the appropriate product structure. In this way, one can tailor the script to more closely reflect one's organization and generate the appropriate structure automatically.

A recommended product structure is given in the following table:

Product Organization Category	**Description**
Information Management Architecture	Organization of the metadata assets by data stores, data processes, and data flows within the IM architecture
Business Functional Architecture	Organization of the metadata assets by specific business function, e.g., by specific initiatives, product lines, etc.
Business Organization	Organization of the metadata assets by departments, divisions, etc.
Development Tool or Metadata Source/Target	Organization of the metadata assets by specific tools used within the organization that contain metadata

Figure 16 - Classifying metadata sources and targets

As part of the metadata harvesting described in Part II of this document, metadata is captured from the identified metadata sources and is imported into the repository as child products (contents) within this structure. This structure then serves as the master roadmap, or metadata architecture for the organization. Generally, at this stage, the Sources and Targets product structure will be the only portion that is well-defined.

3.1.2 Identify data processes within ETL, BI and their associated tools

Once one has identified and categorized metadata sources and technologies, it is then possible to identify the *data processes* within the enterprise. There are several types of data processes in a typical organization:

Data Process Categories	**Description**
Extract, transform and load (ETL)	Extract, transform and load (ETL) type processes where data is extracted from one or more sources, potentially transformed, and then delivered to another persistent data location (e.g., a database)
Reporting, analysis, business intelligence (BI), and OLAP	Reporting, analysis, business intelligence (BI), and OLAP type processes where data is presented in a number of ways, some more persistent than others

Data exchange or integration (EAI or EII)	Data exchange or integration where data is continually exchanged, integrated, or distributed, such as in EAI and EII.

Figure 17 - Data process types by category

Each of those types of data processes can then be classified according to frequency or duration. These include:

Data Process Frequency	Description
Continuous processes	Continuous processes such as operational data store interfaces, EAI operations, or messages
Scheduled data processes	Scheduled data processes, such as data warehouse ETL processes, database duplication, external data feeds, or data mart loads
Special cases	Special cases, such legacy data migration (LDM), version data migration associated with the release of a new version of an application which requires a database update, or other unique circumstances that require a movement and/or transformation of data.

Figure 18 - Data process types by frequency

3.1.2.1 Data Processes in the Catalog

Cataloging the data process that make up and organization is a good exercise that will lead to the development of an accurate repository product structure, help gauge the level of that will be required to populate and maintain the repository, highlight critical data process flows, and determine priorities for the metadata management program. A sample table is given below:

Data Process Name	Description	Process Type	POC	Metadata Source(s)	Data Source(s)	Mappings
Name of data process	Descriptive information	Category and Frequency from tables above	Point of Contact or Steward	Source of mapping metadata, especially if it is to be imported directly from a tool, such as the ETL or BI tool	Metadata mapping type to be used within the repository for representing the data process from the table above	Mappings that must be generated in the repository in order to capture the data process and stitch products together

Any metadata management and/or enterprise architecture initiative must pay scrupulous attention to identification and classification of the data processes within the organization. Having completed the above data process categorization table, one will then be able to define the Data Processes product structure in the MIR. These data processes will then become metadata mappings among the metadata schema that are harvested in Part II of this document.

3.1.3 Build the enterprise architecture and data flow

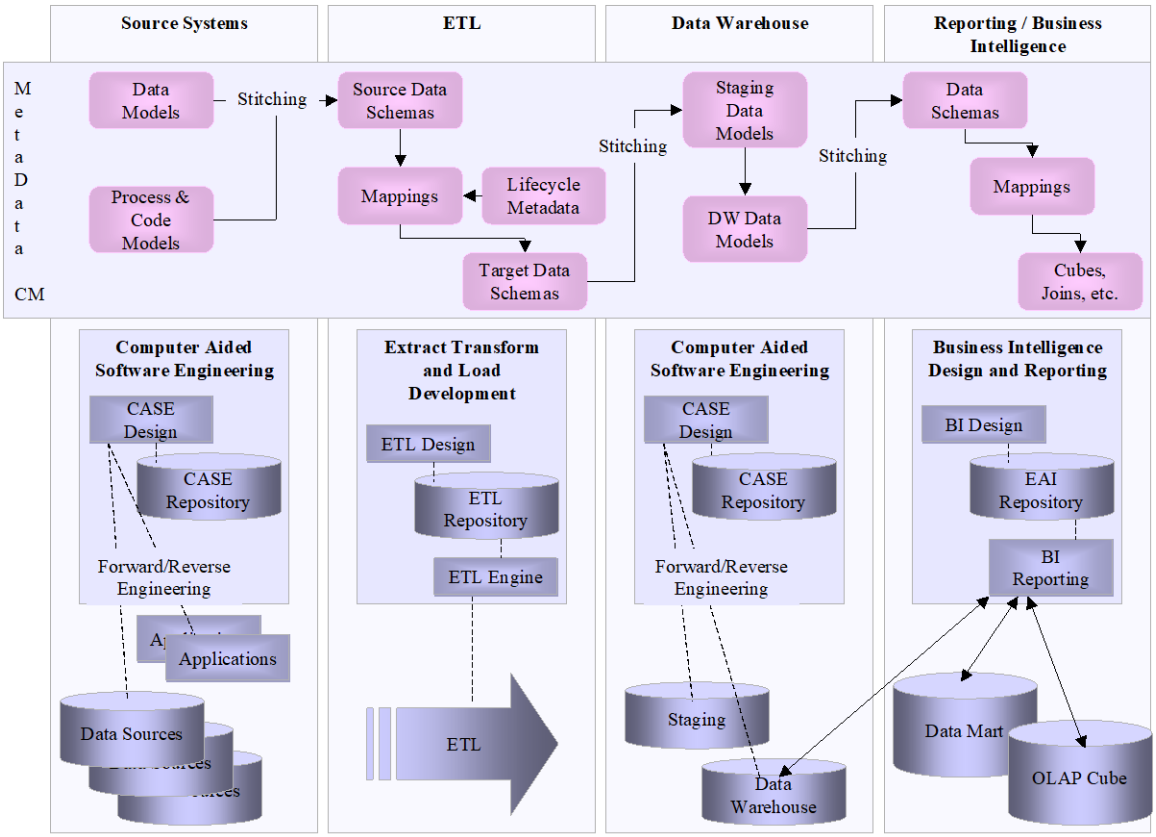


Figure 19 - Sample data-flow based enterprise architecture

Having cataloged the metadata sources and data processes, one may now build the enterprise architecture at a conceptual level. At this level, the enterprise architecture describes the large-scale metadata sources by type and relates them to each other by their roles in the data flow processes. A common example is that of data flow from several ODSs via an ETL engine into a data warehouse (consisting of staging and operational databases), along with data extraction into a BI reporting tool and the ultimate reports, as in the above diagram.

The level of detail that one wishes to develop this to relates directly to the level of complexity and detail of organizational and programmatic structure within an enterprise. Again, one may use the sample product structure in a previous section as a guide.

3.2 Understanding This Tutorial's Enterprise Architecture

As a business case, this document refers to the data management and metadata management lifecycle associated with a financial system for a large organization, simply referred to as "the company." It is comprised of several components and activities, including:

- Finance System:
  - PAYTRANS finance payment transaction format
  - Physical legacy GL flat file-based system and related Copybook schemas
  - Physical legacy Accounts Payable (AP) system and reverse-engineered data model
  - Logical design and physical data model for Accounts Receivable (AR) system
  - Logical-Physical DW Staging and Physical dimensional Star-Schema DB Model
  - Talend Data Integration and Informatica PowerCenter ETL
  - Microsoft Excel documented ETL for some data movement processes not directly imported from a tool
  - SAP Business Objects and Tableau business intelligence and reporting
- A conceptual layer
  - Business Glossary maintained in Meta Integration® Metadata Management (MIMM)
  - And Enterprise Conceptual Data Model (CDM) created using a logical design tool (ERwin Data Modeler).

Metadata harvesting is the process of populating the repository with metadata from all of the sources identified in the last section after cataloging the metadata sources and data processes and associated metadata mappings. In order to do so in a well managed process, one must first develop a catalog of the metadata in your organization.

3.2.1 Catalog the Operational Data Stores

Below is the completed Data Store Catalog for the tutorial scenarios. Normally, such a table would take several iterations to complete in a large organization.

Metadata Source / Target	POC	Reverse Engineering Method	Harvesting Method	Forward Engineering Method	Forward Engineering Contents
PAYTRANS Flat File	PAYTRANS POC	Excel Bridge	Flat file structure of PAYTRANS transactions as a table with columns and datatypes	N/A	N/A
Finance General Ledger (GL) Flat File	GL POC	Excel Bridge	flat file structure of GL transactions including data-types	N/A	N/A
Finance Accounts Payable (AP) ERwin Reverse Engineering	Finance AP-AR POC	ERwin v 9.x ER1 bridge	Physical RDBMS design reverse engineered from the SQLServer DB and commented in ERwin.	Extract from repository using ERwin v 9.x ER1 bridge and Generate DDL using ERwin for new database.	Physical RDBMS design as DDL as generated from ERwin physical RDBMS design
Finance Accounts Receivable (AR) ERwin Logical Physical Design	Finance AP-AR POC	ERwin v 9.x ER1 bridge	Logical E-R and derived Physical RDBMS design for the SQL Server DB in ERwin	Extract from repository using ERwin v 9.x ER1 bridge and Generate DDL using ERwin for new database.	Physical RDBMS design as DDL as generated from ERwin physical RDBMS design
Finance Data Warehouse Staging DB Logical Physical Design	Finance DW POC	ERwin v 9.x ER1 bridge	Logical E-R and derived Physical RDBMS design for the SQL Server DB in ERwin	Extract from repository using ERwin v 9.x ER1 bridge and Generate DDL using ERwin for new database.	Physical RDBMS design as DDL as generated from ERwin physical RDBMS design
Finance Data Warehouse DB Logical Physical Design	Finance DW POC	ERwin v 9.x ER1 bridge	Physical Dimensional E-R and derived Physical RDBMS design for the SQL Server DB in ERwin	Extract from repository using ERwin v 9.x ER1 bridge and Generate DDL using ERwin for	Physical RDBMS design as DDL as generated from ERwin physical RDBMS design

				new database.	
Finance Data Warehouse BI Design	Finance BI POC	SAP Business Objects Repository XI bridge	RDBMS connectivity, OLAP and reporting design metadata	N/A	N/A

3.2.2 Catalog the Data Processes.

Below is the completed Data Process Catalog for the tutorial scenarios. Normally, such a table would take several iterations to complete in a large organization.

Data Process Name	Description	Process Type	POC	Harvesting Method	Data Stores(s)/Stitching
PAYTRANS EAI	Data feeds from external sources of payment using the PAYTRANS format	Batch process that runs nightly	PAYTRANS PoC	Microsoft Excel Bridge	**Source:** PAYTRANS **Target:** Accounts Receivable
Finance General Ledger (GL) to Finance DW Staging ETL	flat file transactions from GL system loaded into DW Staging DB	Mainframe triggered ETL	GL POC	Microsoft Excel Bridge	**Source:** GL Account and GL Trans **Target:** Staging DW
Finance Accounts Payable (AP) to Finance DW Staging ETL	Extracts from the AP RDBMS are loaded into DW Staging DB according to what is already populated in the Staging DB	Scheduled ETL via Informatica	DW POC	Informatica PowerCenter Bridge	**Source:** Accounts Payable and Staging DW **Target:** Staging DW
Finance Accounts Receivable (AR) to Finance DW Staging ETL	Extracts from the AR RDBMS are loaded into DW Staging DB	Scheduled ETL via Informatica	DW POC	Talend Data Integrator Bridge	**Source:** Accounts Receivable **Target:** Staging DW
Finance DW Staging to DW ETL	Extracts from the DW Staging RDBMS are loaded into DW	Scheduled ETL via PL SQL	DW POC	Talend Data Integrator Bridge	**Source:** Staging DW **Target:** Dimensional DW

	RDBMS				
Finance DW to BI	Extracts from the DW RDBMS are Used by Cognos for BI	Real-time RDBMS access	BI POC	N/A (stitching)	**Source:** Dimensional DW **Target:** BI Reporting

3.3 Harvesting the Finance System Metadata

3.4 Setting Up the Tutorial

In order to create this structure in the repository for the tutorial, a Javascript based backup of the Metadata is provided to you. To execute this script, sign in as Administrator (generally with password "Administrator" in a new installation), open Meta Integration® Metadata Management (MIMM) and right-click on the top of the *Repository Tree* (far left panel). Select Scripts › Metadata Management Tutorial – From the Start (Create the Metadata Management Tutorial initial structure in order to begin at the beginning of the Metadata Management Tutorial) from the list of scripts.

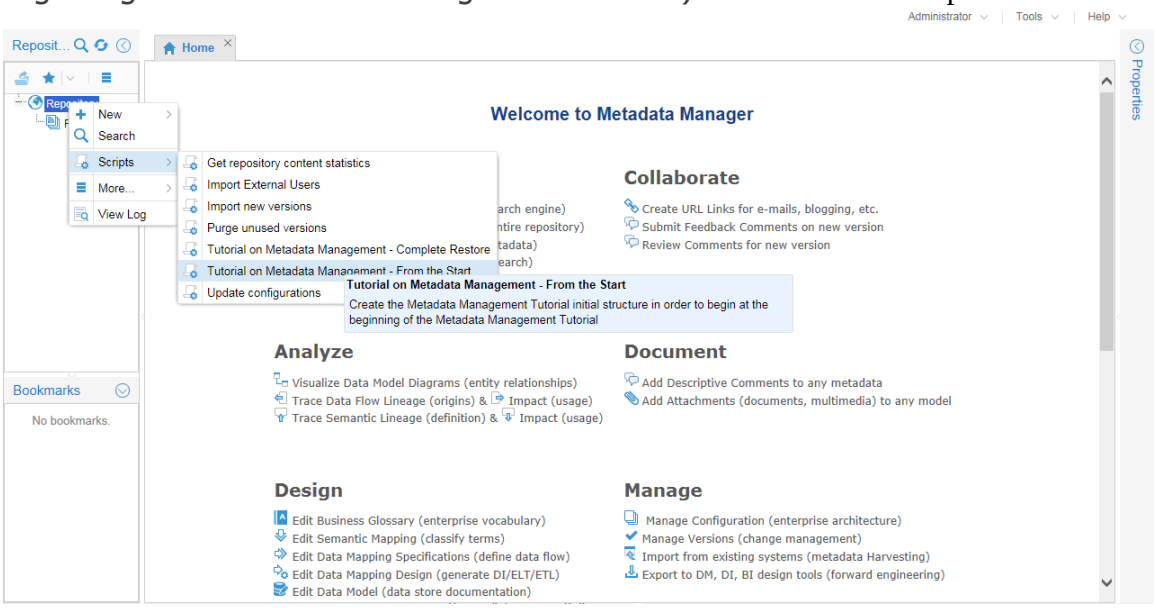


Figure 20 - Selecting the tutorial restore script

Click on the Run Script button in the Run Custom Script dialog that is presented.

Once the script has completed close the dialog. You may need to click on the Refresh control ( ) at the top of the *Repository Tree* panel. Note how the metadata cataloged in the previous section are reflected in the tutorial repository structure under the [folder]s Tutorial/Metadata Management.

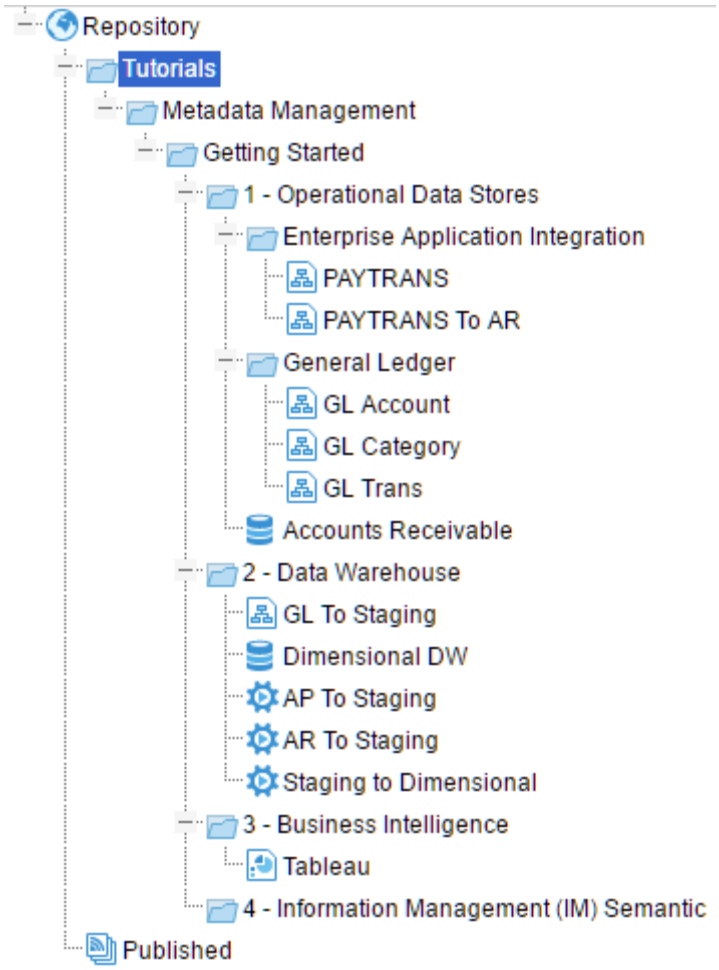


Figure 21 - Tutorial repository structure after executing the restore script

After executing the tutorial restore structure script, the basic [folder] structure of the repository is created. Now it is time to define models to represent the data stores and data processes that were cataloged in the previous sections. This section of the tutorial will deal with the mechanics of harvesting the metadata embedded in data stores and data process tools.

Harvesting may be as automatic a process as one wishes to specify it. Most of the effort involved is in the definition of connection information or *bridge import parameters* to define how the particular bridge will import, or harvest, the metadata from that source. Beyond that, harvesting is a scheduled or manual process, depending upon your work-flow. The next sections walk one through the process of creating and harvesting the appropriate *models* for The Company.

In the above diagram, one can see a lineage diagram at the model level of detail of the complete Finance System for The Company. As described earlier, it is a fairly typical architecture consisting of

- *Operational data stores* (ODS) consisting of data entry and data management systems
- *Data integration* or *extract, transform and load* (ETL) feeds to
- A *data warehouse* consisting of
  - A *data warehouse staging* (normalized) database
  - A *snowflake or star-schema dimensional* database
- With *business intelligence OLAP* schema and reporting.

We will now refer back to the metadata catalog that we developed earlier in order to populate the repository with metadata.

As one may note from the metadata catalog, harvesting from the sources of metadata in some cases require connectivity directly to a source tool repository and/or API. However, for the purposes of this demonstration, all the metadata required for harvesting will be either available directly in the tool format that a bridge can read (e.g., ERwin Data Modeler in XML format or Embarcadero ER/Studio Data Architect DM1 format), or will be provided as MIR XML Metadata Interchange (MIR XMI), a proprietary format of Meta Integration® Metadata Management (MIMM) that faithfully represents the metadata captured in the repository and via any of the bridges.

3.4.1 Harvest the Finance System ODS Metadata

In Meta Integration® Metadata Management (MIMM), expand the Repository Tree in the left hand pane to the [folder] at the path:

1 - Operational Data Stores

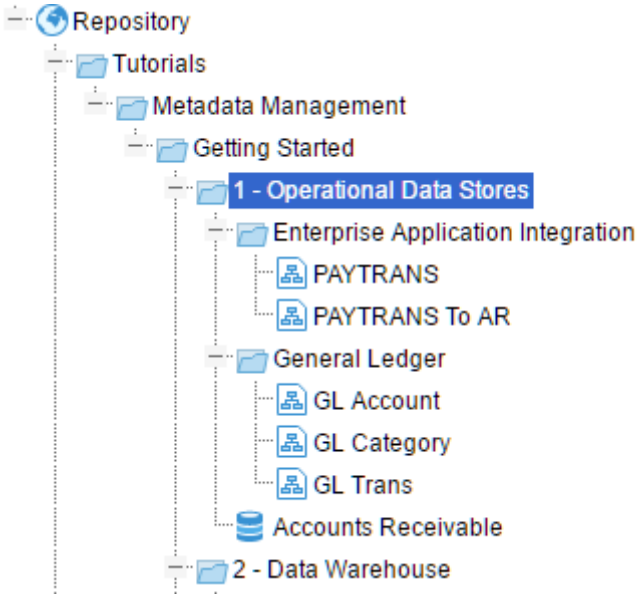


Figure 22 - Repository tree expanded to show the Finance System ODS [folder]

Note also, many model contents are already present in the 1 – Operational Data Stores [folder]. Comparing with the Finance System metadata catalog, only the Accounts Payable model content is not already defined there. Thus, we must first create it.

3.4.1.1 Create Accounts Payable Model Content

Right-click on the "1 – Operational Data Stores" [folder] and select New › Model:

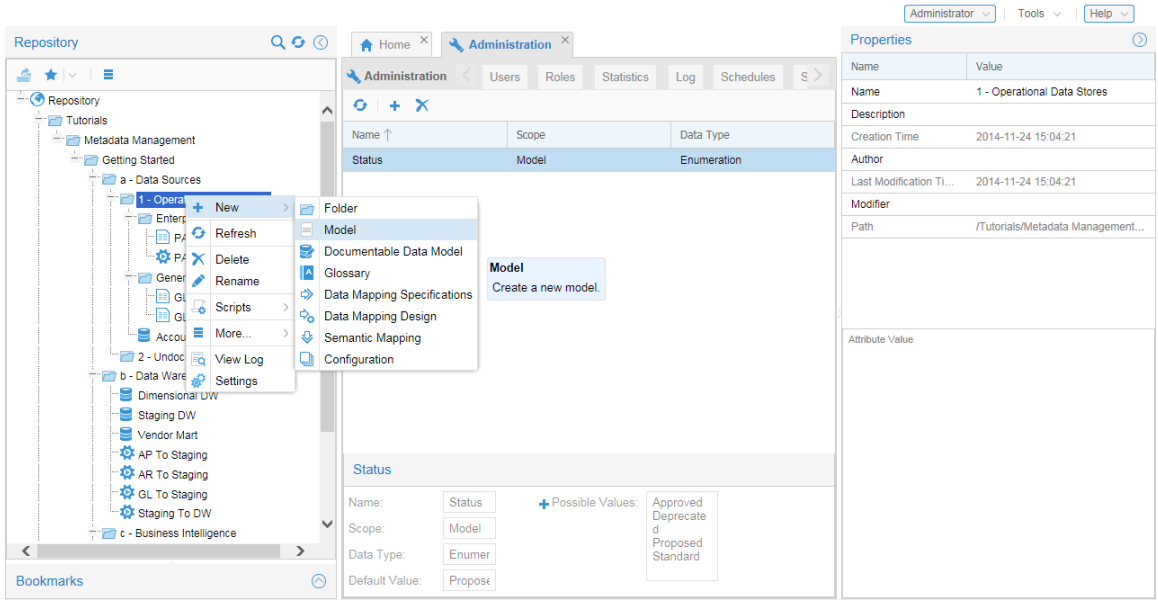


Figure 23 - Create new model content in the Finance ODS [folder]

In the Create Model dialog enter the name of the model content to be added: Accounts Payable. Also, enter a description for the model content:

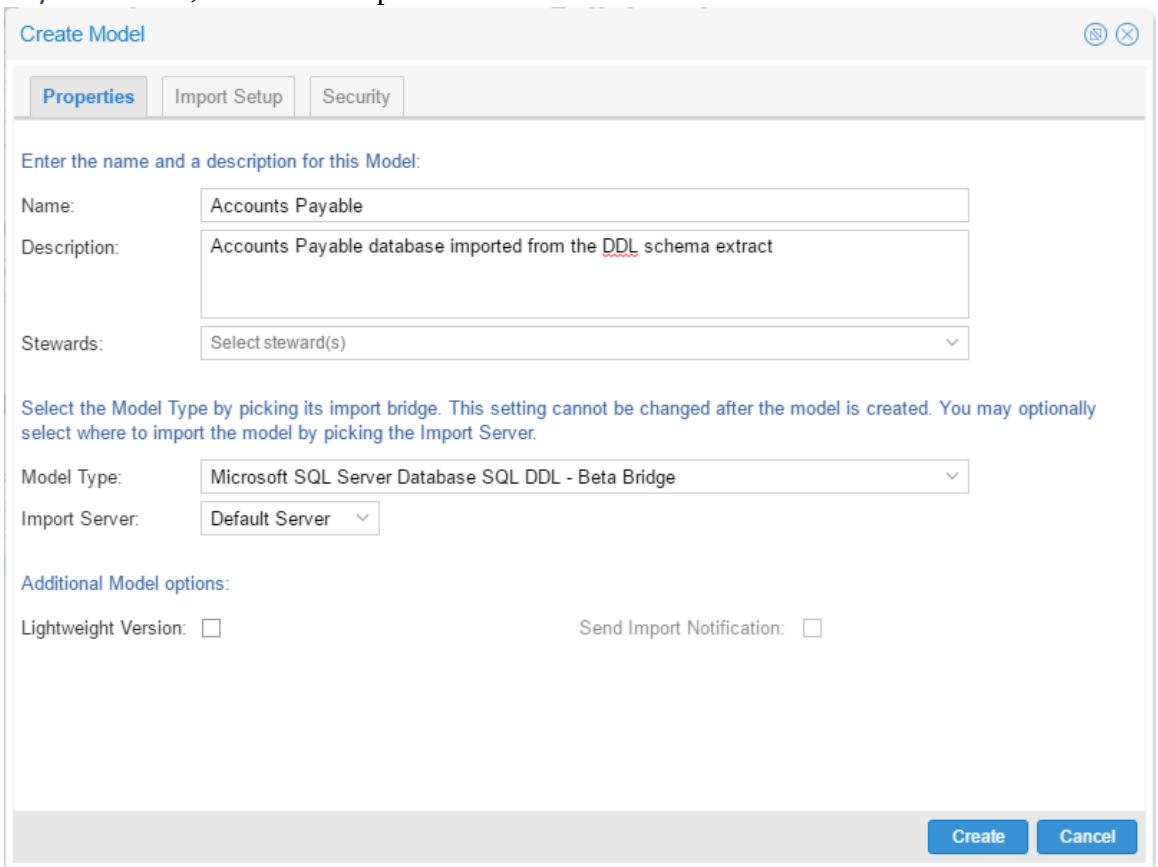


Figure 24 - Create model content dialog for Accounts Payable

In the dialog, the Import from data entry field allows one to specify the *metadata bridge* to be used to import this model.

Finally, in the lower section of the dialog are two check boxes:

- Lightweight – when set, indicates that the result of the import will only be usable as a basis for other objects. We will use this option later and should NOT be checked at this time.
- Send Import Notification – when set, indicates that each time the model content is successfully harvested, notification will be sent according to the rules defined. We will use this option later and should NOT be checked at this time.

Click on the pull-down for the Import from combo box and you will see the complete list of bridges available for a model content to use to harvest metadata.

This list is continuously being updated and expanded as support is developed for more bridges to more formats and tools. For an official list of currently supported bridges and detailed documentation, mapping specifications and limitations, please refer to the website at http://www.metaintegration.net/Products/MIMB/SupportedTools.html.

Select the import bridge as shown below:

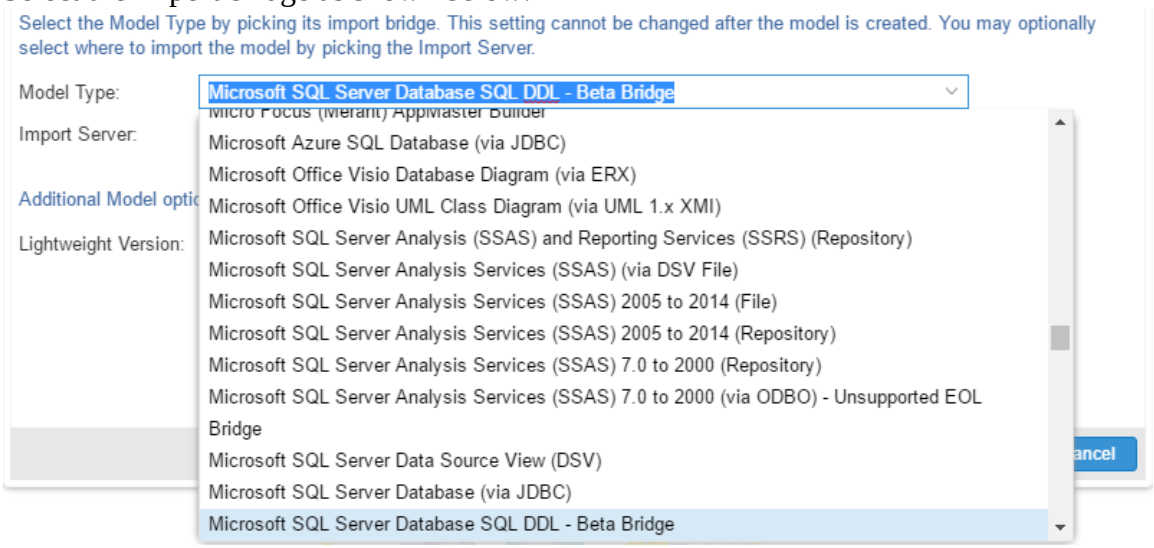


Figure 25 - Select the import bridge

We select this bridge because we are importing metadata from a database, but we do not have the database available for JDBC connectivity. In this case we use a file based bridge that does not require live connectivity to a database.

Note also, the Server combo box refers to the Default Server. This is the Meta Integration® Metadata Management (MIMM) Application Server. If you wish to execute

a bridge on another server, refer to the MIMBReadme.html documentation for instructions on how to set up remote bridge servers.

Now click on the Import Setup tab. In this tab, one specifies all the criteria, or *bridge parameters*, required to harvest metadata for this model content. Most of the bridge parameters are determined by the choice of bridge, in the Import from combo box.

Now, click on the show/hide control ( ) for the Help panel on the right side of the dialog. The tool tip, or help text, for these selected bridge is then presented.

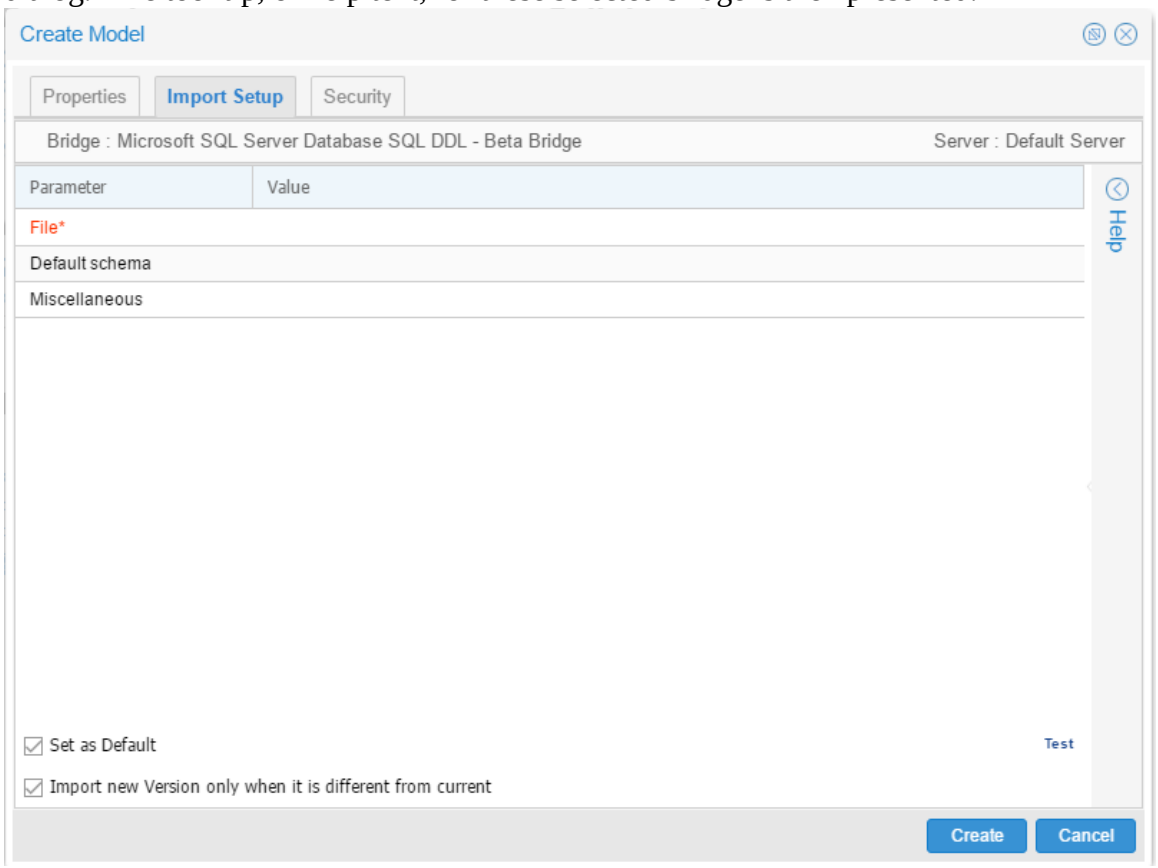


Figure 26 - Tool tip, or help text, for the selected import bridge

Note, there is a great deal of detailed information about the use of this bridge, including requirements, special considerations, frequently asked questions, etc.

Now, click on the File* parameter under the Value heading. Note, the tool tip changes to this new scope.

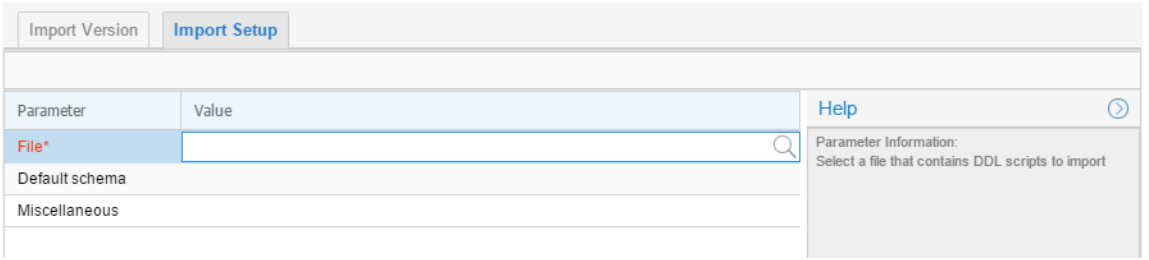


Figure 27 - Tool tip for File parameter of the import bridge*

As the tool tip states, this is where one specifies the directory path for the bridge to locate the file to be parsed for metadata. The asterisk (*) indicates that this parameter is mandatory.

Also, note the browse control ( ) on the right hand side of the File* data entry box. Click on this control and the Select a File dialog appears.

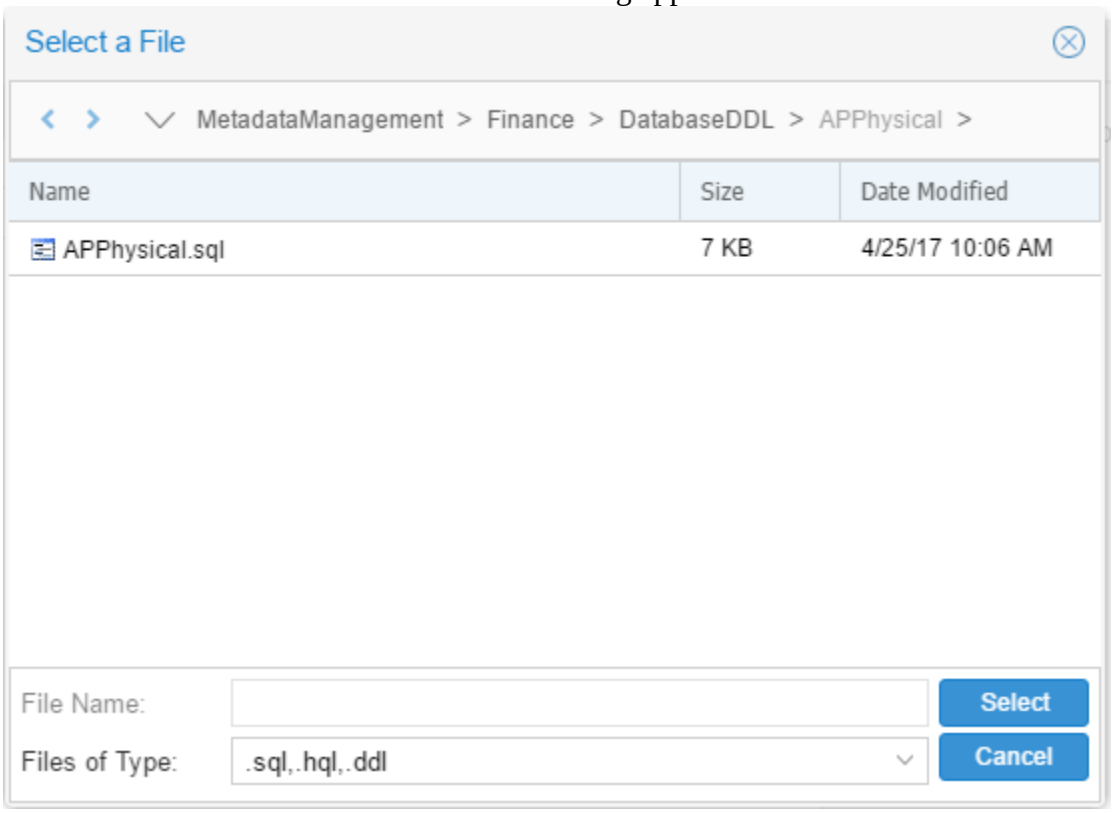


Figure 28 - Select a File dialog

This dialog works quite like any browse dialog for your operating system. Note, the location being browsed is on the server where the bridge will run, in this case Meta Integration® Metadata Management (MIMM) Application Server, not your local drive. All file and path parameters used for harvesting into Meta Integration® Metadata Management (MIMM) are relative to this server. Thus, these files must be visible to the Application Server, and the process it is running under (for more sys admin type information about paths and processes on the Application Server, please refer to the Meta Integration® Metadata Management (MIMM)Readme.html documentation).

In this case, the file is at the following location within the temp directory the C: Drive on the application server (you may need to contact your sys admin staff to discover where it was installed in your environment):

C:\Temp\Models\MetadataManagement\Finance\DatabaseDDL\APPhysical.sql

Note also, to find this file in the Select a File dialog, you will need to specify All Files in the Files of Type: pulldown, as the file you are looking for has an extension.

Additional parameters may be specified as needed in order to harvest the metadata in the manner required for a particular application. In this case, the only change you need is to enter a Default schema, as the DDL in the above SQL file does not contain a schema name and the simple default for Microsoft SQL Server databases is "dbo". Otherwise, the default settings are what we want here.

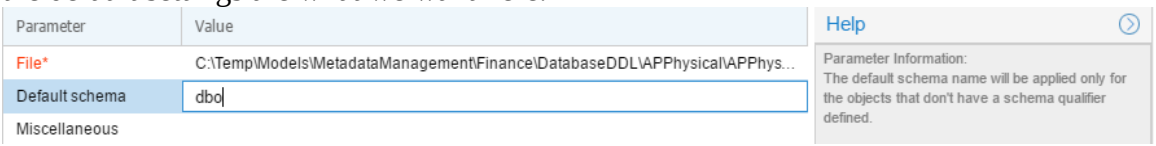
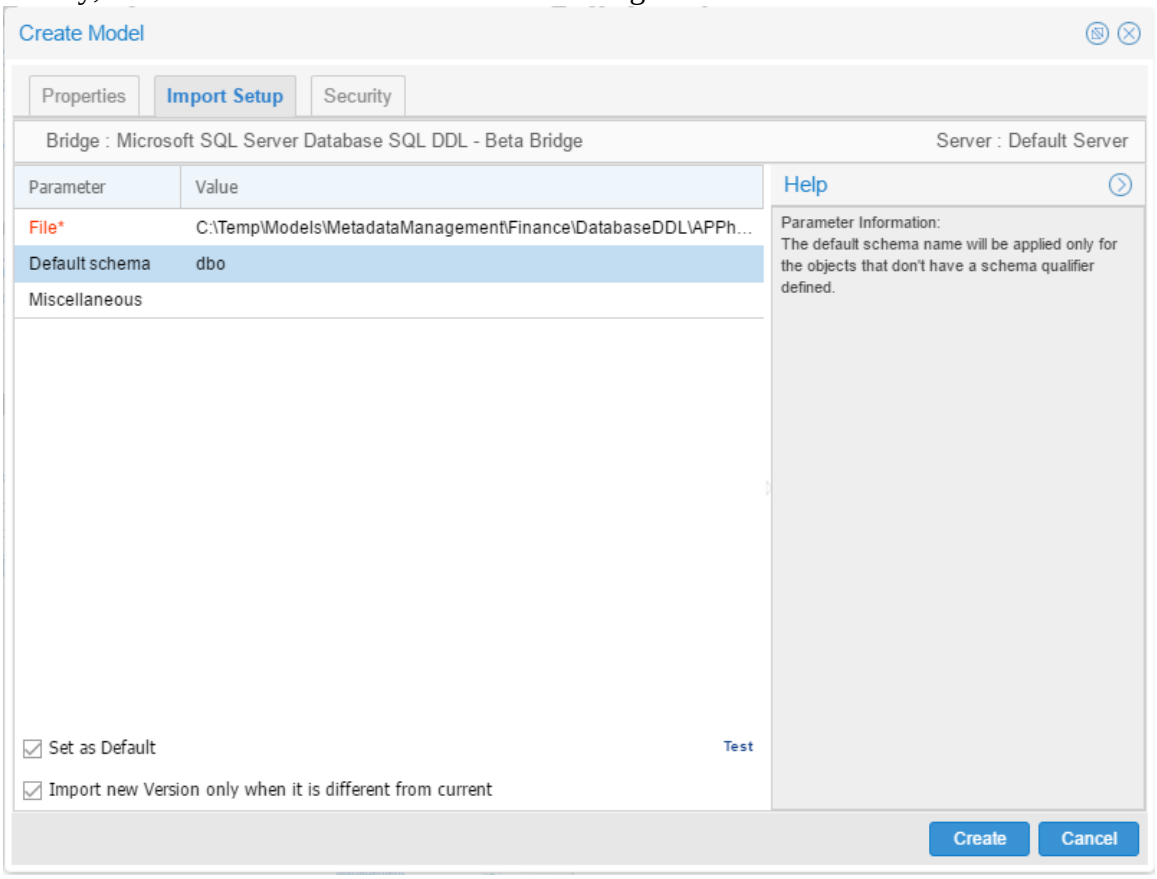


Figure 29 - Import Setup

Finally, in the lower left-hand corner of the dialog is a check box:



- Re-import only when changes are detected – when set, indicates each time the model content is successfully harvested, a comparison is made and it only updates the metadata in the repository if there are changes.

In this case, this option should be checked.

Then, click the Create button and the model content will be created. You will be offered the chance to import the model immediately.

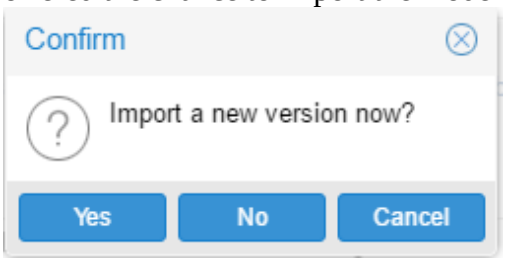


Figure 30 - "Import now" dialog

Click the No button.

3.4.1.2 Import Accounts Payable Model Content

As we did not specify any scheduling or triggering for this model content, we will manually request that the bridge import the model content. As we just created the model, you may simply click the Yes button. Otherwise, if you have a model and wish to import it later, right-click on the Accounts Payable model content and select Import.

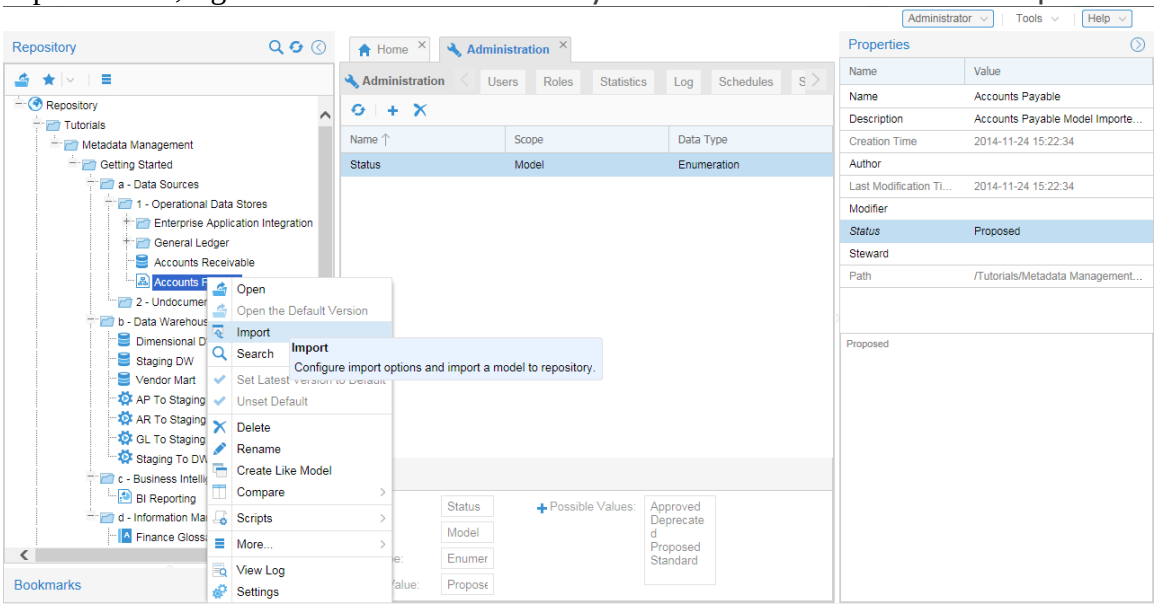


Figure 31 - Importing the Accounts Payable model content

The Import: Accounts Payable dialog is presented.

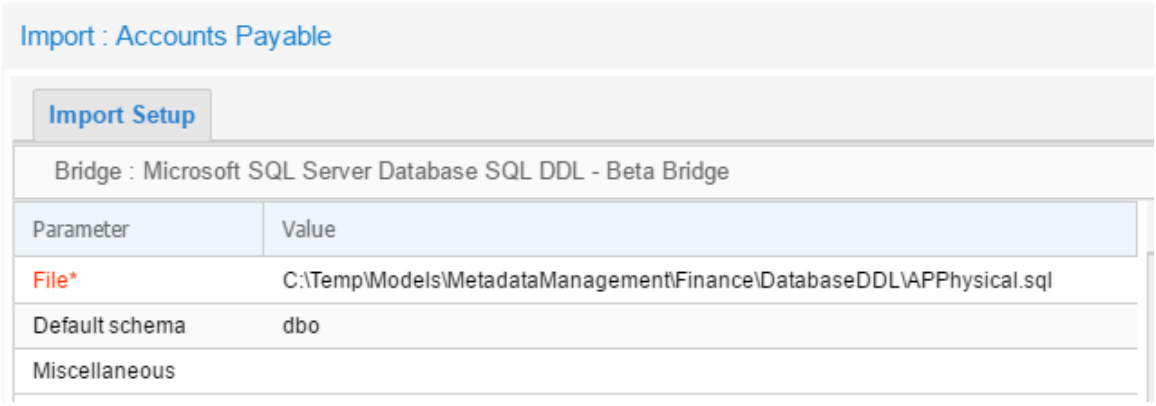


Figure 32 - Import: Accounts Payable dialog

Click on the Import button.

If you are using a single version / single configuration license, you will be presented with the following dialog:

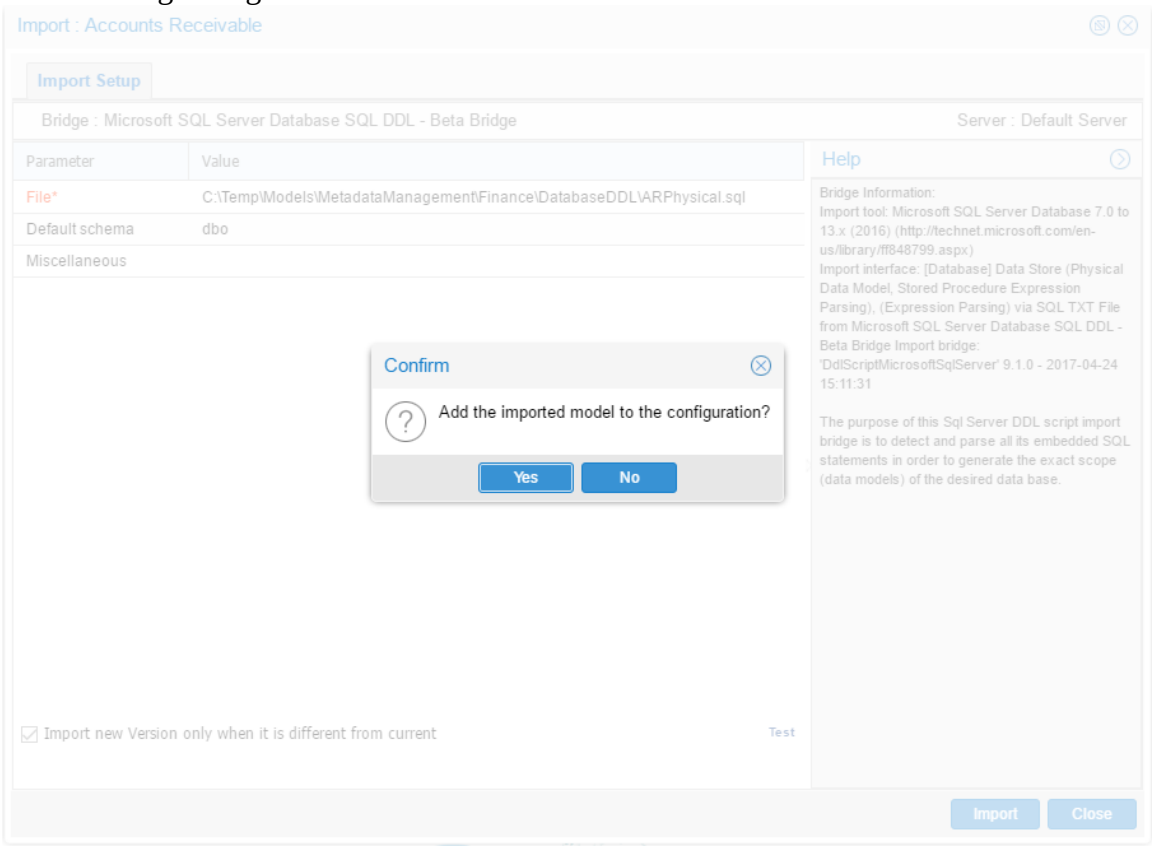


Figure 33 - Add to configuration dialog

Click on the No button.

Then a confirmation dialog is presented when the import has completed.

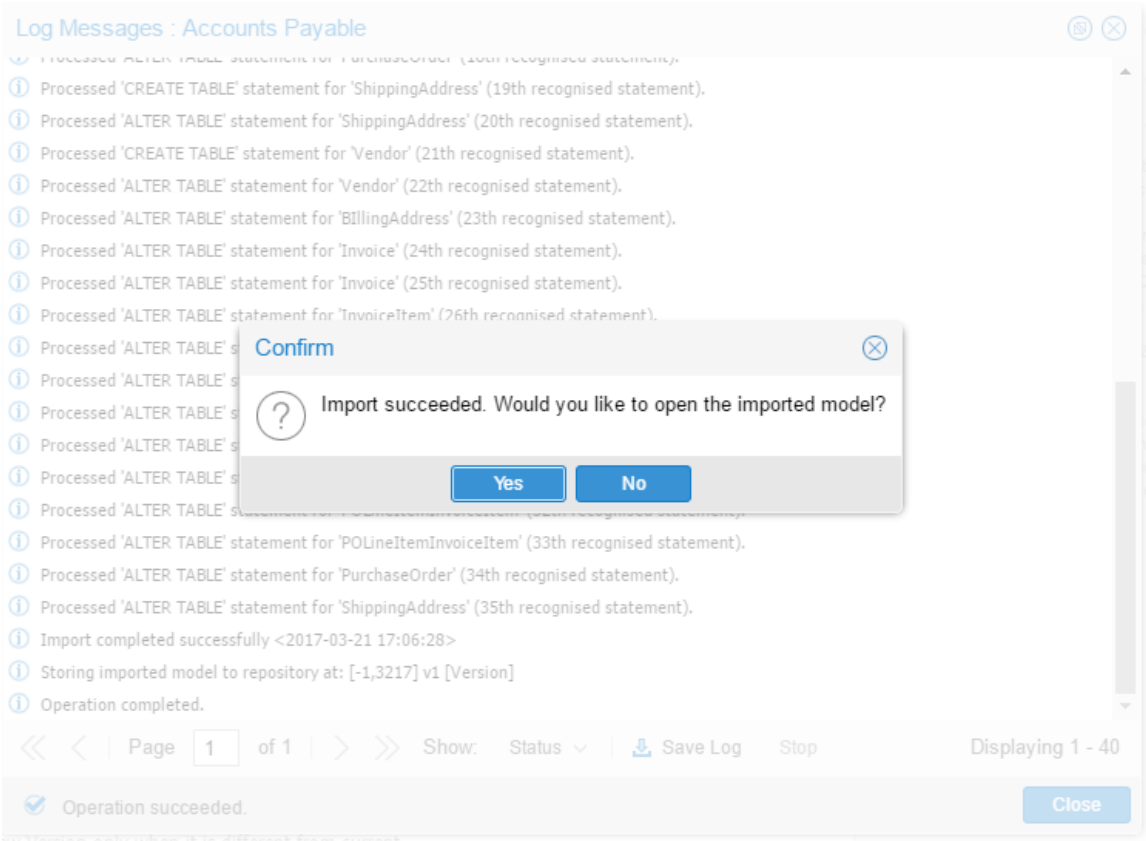


Figure 34 - Log Messages dialog for harvesting of the Accounts Receivable model content

Click on the No button and the log will be presented.

3.4.1.3 Review Accounts Payable Import Log

It is generally a good idea to review the log of the harvest operation (import and writing to the repository database).

In this case, we see the log in front of us. However, if we closed that window, to view the log after the import has completed, simply right-click on the [repositoryobject] you used to initiate the action (in this case the Accounts Payable model content) and select View Log.

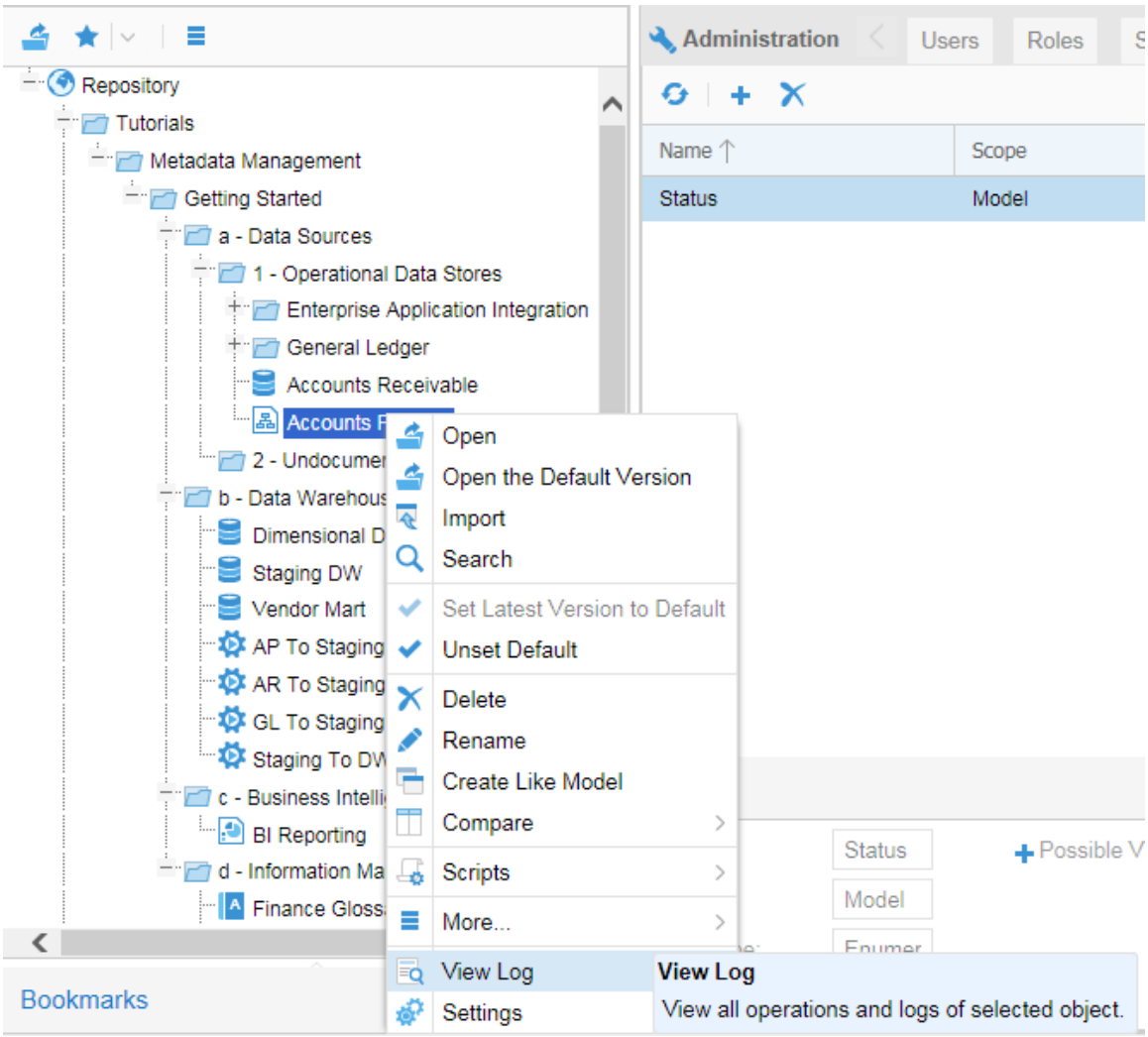


Figure 35 - Recalling log messages for harvesting of the Accounts Receivable model content

Now, the dialog is again presented with a list of all logs available.

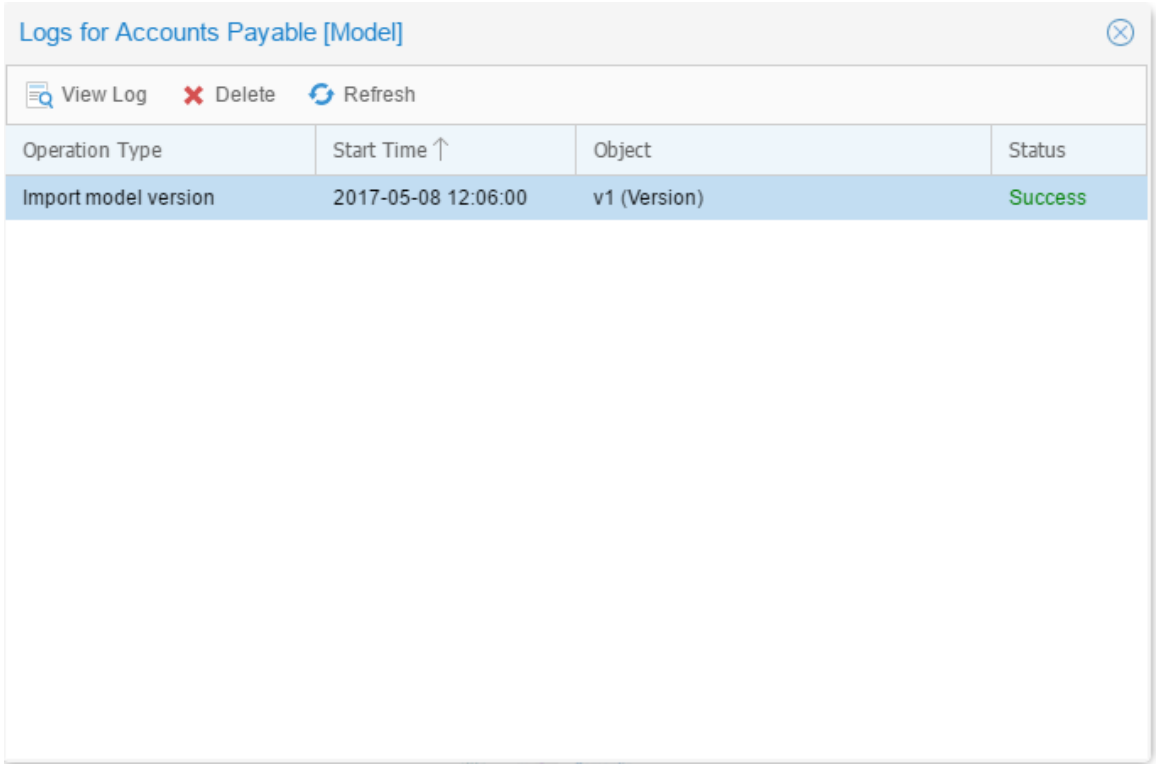


Figure 36 - List of available logs the Accounts Receivable model content

Click on View Log for the only one in the list. The Log Messages dialog with the last harvest log is presented.

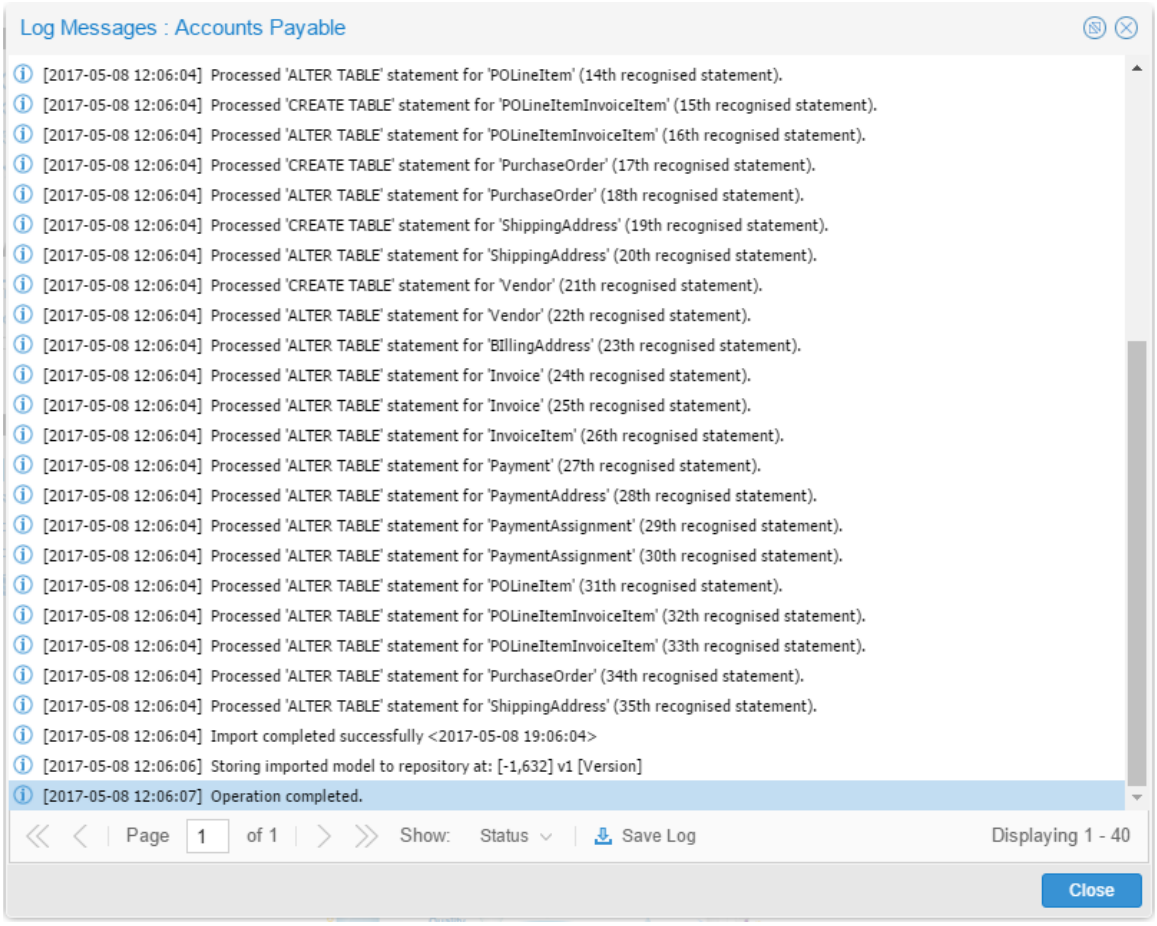


Figure 37 - Status level log messages for the harvesting of the Accounts Receivable model content

The log messages dialog is may be used to navigate through the log messages by using the control on the bottom row of the dialog. The Show: selection control allows one to filter the log messages to show only those of the specific level (*error*, *warning*, *status*, or *all*) that one wishes to see.

Select All v.

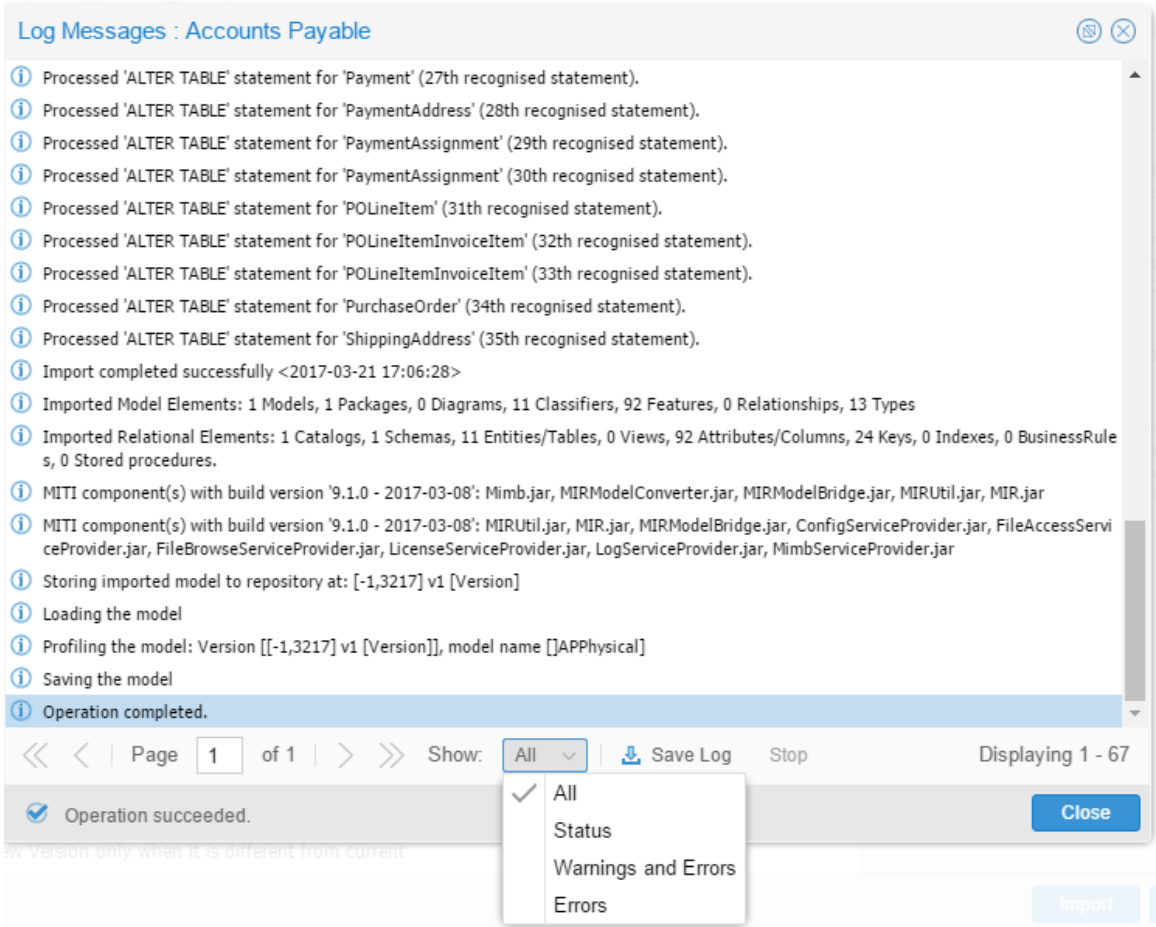


Figure 38 - Status level log messages for the harvesting of the Accounts Receivable model content

In addition, one may page through the log messages (if there is more than one page), refresh the dialog (to see more messages, thought the dialog refreshes regularly on its own), and save the log messages to a file.

3.4.1.4 Review Accounts Payable Harvested Metadata in the Repository

3.4.1.4.1 Metadata Browser

After reviewing the logs and closing those dialogs, one may see the model in the Metadata Browser. This time double click on the Accounts Payable model in the Repository Panel.

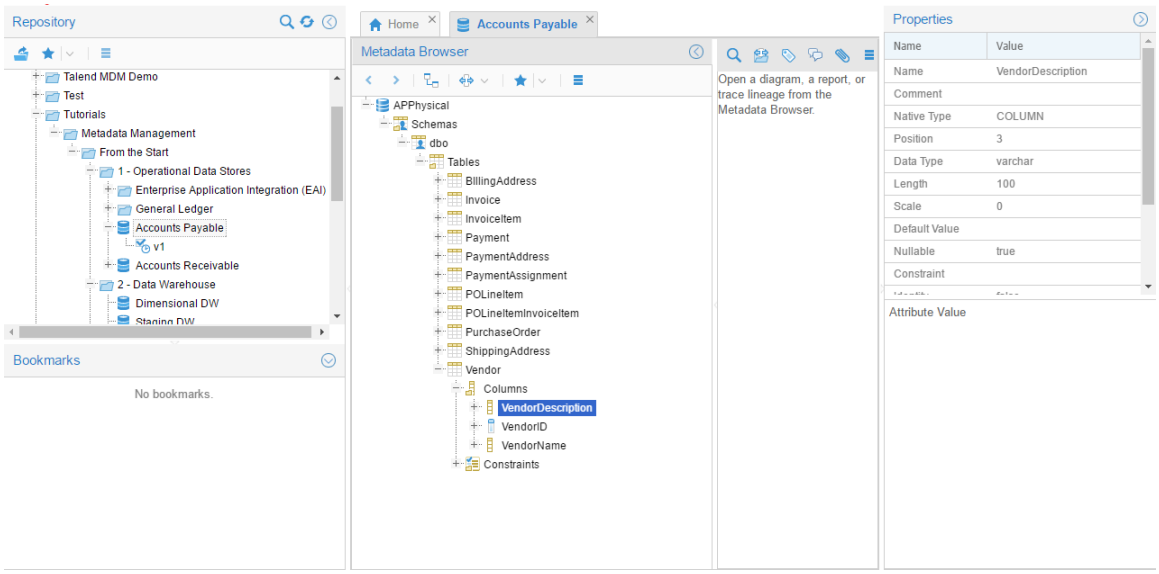


Figure 39 - Accounts Payable model in the Model Browser tab with Repository Tree hidden

The *Model Browser* present the metadata harvested in a tree-structured and expand/contract navigation format, controlled by a *profile* (in this case the relational database profile). The icons, naming conventions, properties of the metadata objects, and organization of the model is all presented in a manner that would be familiar to the actual tool user.

Like the Repository panel, this is a hyperlinked tree, representing what is actually more than just a hierarchy of objects in a tree format, with links to various associated objects.

Now, hide the Repository panel:

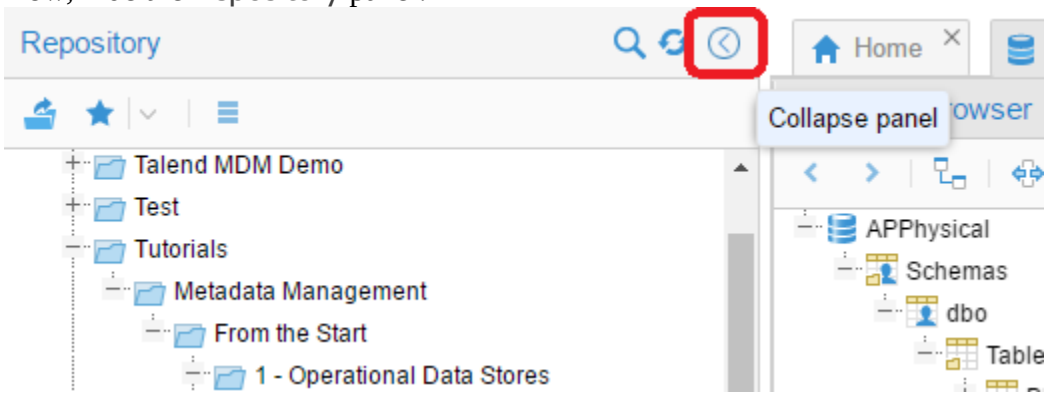


Figure 40 - Hide the Repository panel

And we have more room to view the model:

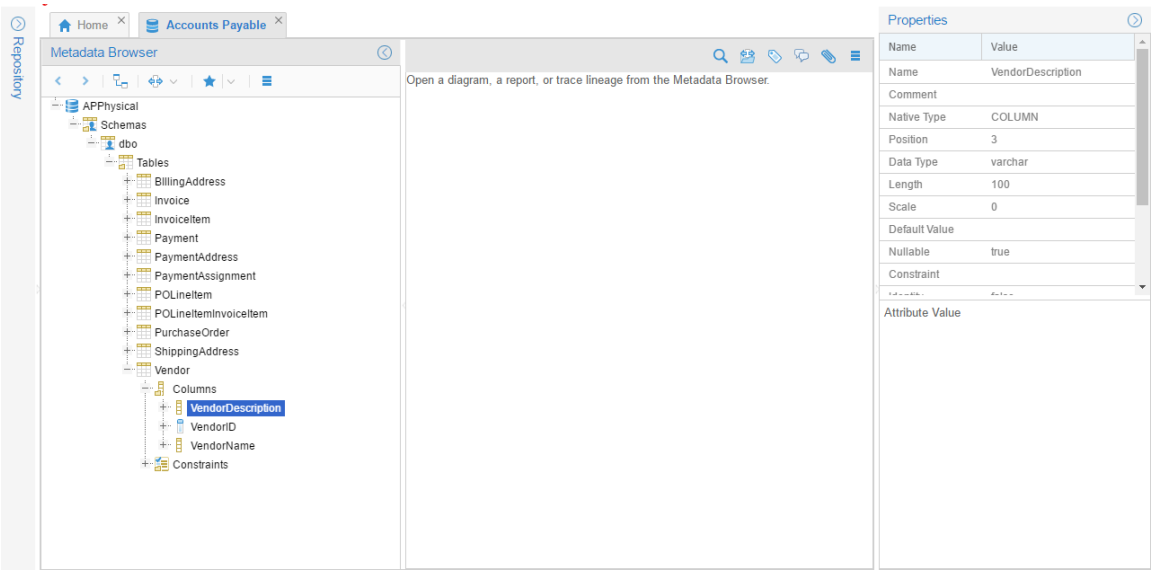


Figure 41 - Browsing to a column in the Accounts Payable model

Note that the Properties panel on the right may be expanded and contracted (show and hide) as needed to show metadata properties of the object selected in the model tree. In the above screen capture, note the VendorDescription column is selected and the properties like Data Type and Length are presented in the Properties panel.

The Actions control provides access to the same actions as would be available if you right-clicked on the object in the Repository panel.

3.4.1.5 Harvest Other Finance Operational Data Stores

Several more ODS models must be harvested in the Finance System. For the purposes of this tutorial, the model contents for each of these operation data stores have already been created. Thus, simply import them as you did the Accounts Payable model content manually but right-clicking on each and selecting Import.

Please import the Accounts Receivable model now.

There are also two different folders inside of the ODS folder:

- Enterprise Application Integration
- General Ledger

The Enterprise Application Integration folder contains two models, the PAYTRANS flat file message and the ETL process to load PAYTRANS into the Accounts Receivable model. The General Ledger folder contains two flat file based models which contain general ledger account and transaction information.

Instead of right-clicking on each model content and harvesting manually, we will use a *scripted* harvesting method. Every action and every function that Meta Integration® Metadata Management (MIMM) can perform is accessible via *JavaScript*, and are

accomplished via calls to the *MIR SDK* which are made available to JavaScript. In addition, a script, once written, may be uploaded to Meta Integration® Metadata Management (MIMM) and associated with a particular metadata object type (e.g., a model content or a [folder]). Once associated, the Meta Integration® Metadata Management (MIMM) user interface provides access to these scripts via the right-click options as shown below.

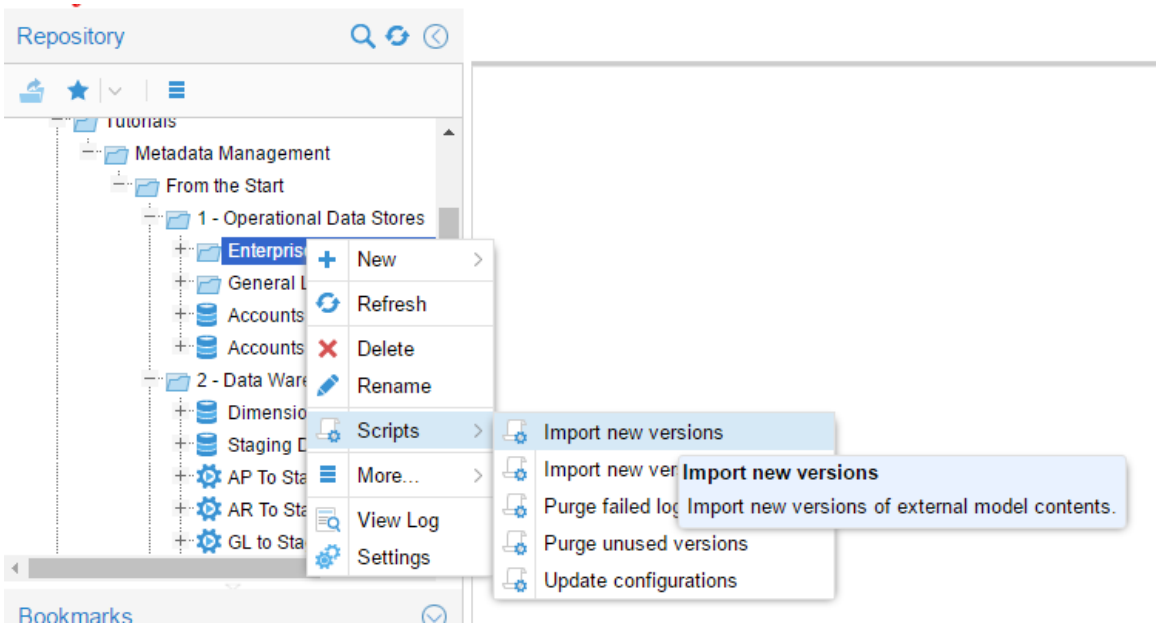


Figure 42 - Selecting the Import model(s) script for the Finance System Data Warehouse [folder]

A number of scripts are provided "out of the box" with Meta Integration® Metadata Management (MIMM), for convenience. We have already seen one of these scripts – the Create Metadata Management Tutorial script associated with the repository as a whole, which we used to populate the repository structure. Now we will use another to harvest the rest of our metadata.

Right-click on the b-Data Warehouse [folder] and select Scripts › Import model(s). A Run Customer Script dialog is presented.

This dialog presents the script code and the option to run the script. Click on Run Script to harvest the [folder]. Again, a Log Messages dialog is presented and the harvesting process proceeds until completed.

Please do the same now with the General Ledger folder.

3.4.2 Harvest Finance Data Warehouse Metadata

The next [folder] in the Finance System is the 2 - Data Warehouse [folder]. In this [folder] there are several model contents to be harvested. Some, like the Dimensional DW, are data stores. Some, like the Staging To Dimensional, are data processes.

Use the Import model(s) script to harvest all these models:

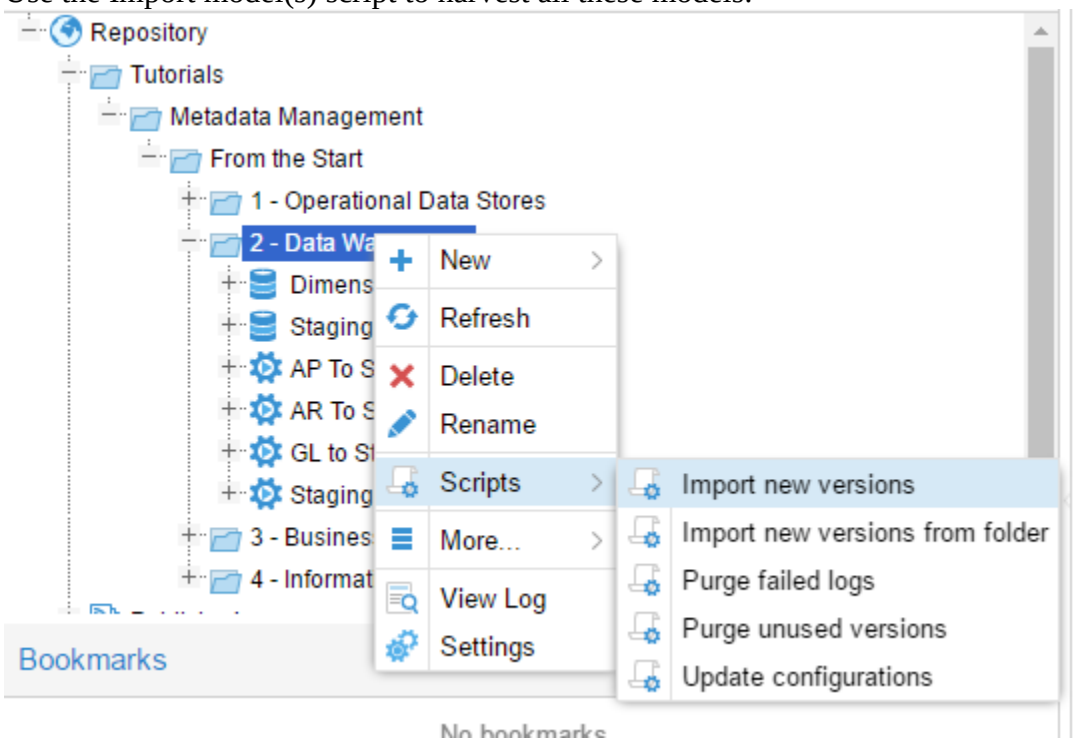


Figure 43 - Import models(s) for 2 – Data Warehouse

3.4.2.1 The Staging [PhyscalDataModel]

So far, all the models we have seen are *harvested* models. This means that the actual metadata for these models is extracted from external sources. In general, this means that the system of record for the metadata is actually the source tool. However, in the Data Warhouse folder, we also have a physical data model, which is a special type of model that allows one to document (or author) the schemas, tables, columns and additional attribution, diagrams, etc., in that model.

In addition, a physical data model may be based upon a harvested data store. In this case, the physical data model starts out with the schemas, tables and columns as defined in that datastore and then may be extended to include relationships, diagrams, subject areas, logical attribution (business name and description, e.g.) of a complete data model. This physical data model may then have a life of its own, or it may be *synchronized* with changes (due to harvesting the models) to the underlying physical metadata.

Thus, one may point to a data store (like a database, big data cluster via Hive, etc.) and then produce a full data model based upon that underlying data structure. One may then set that model to be synchronized with any changes to that underlying data structure as they are harvested.

Let's create a physical data model here for the Staging Data Warehouse.

Right-click on the 2 – Data Warehouse folder, select New ➔ Physical Data Model, and confirm the following details:

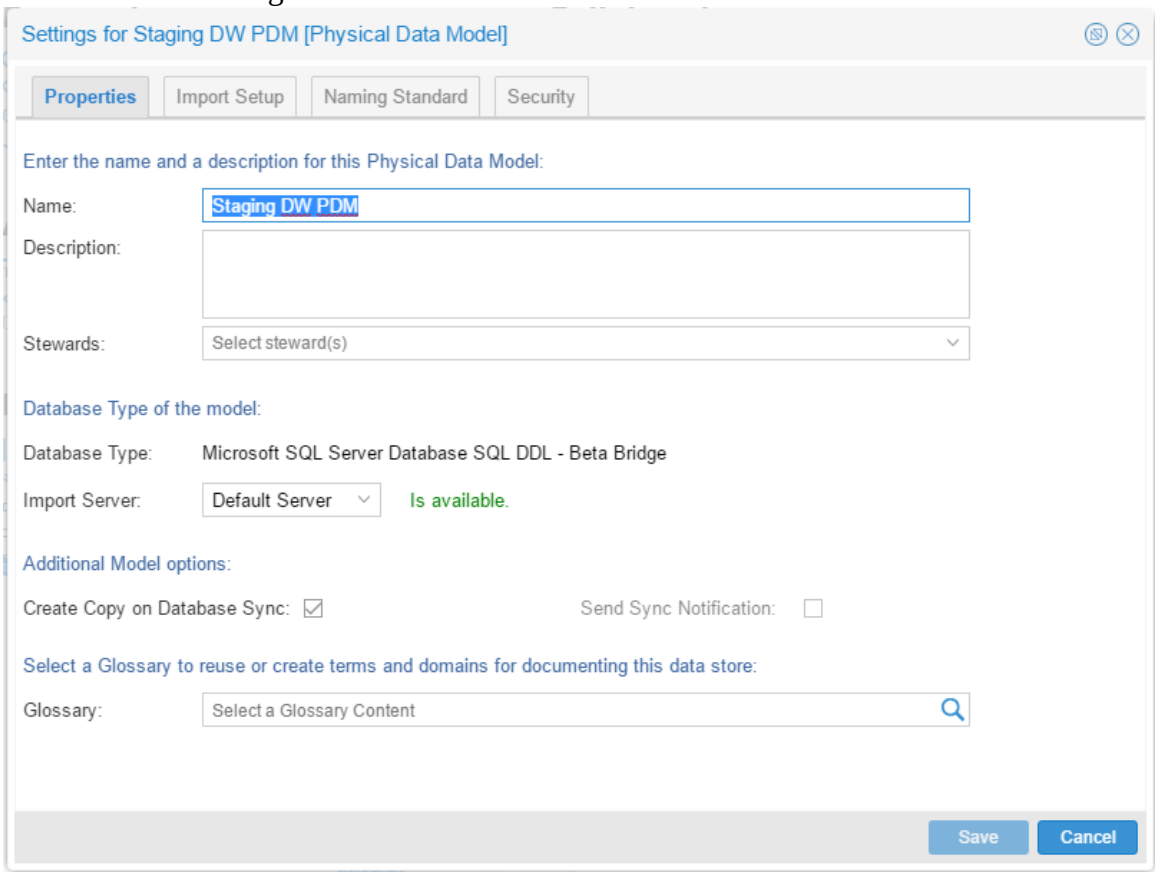


Figure 44 - Settings for the Staging DW model

Note, the Create Copy on Database Sync: checkbox is checked. In this case, when the model is harvested, any changes will by synchronized with the physical data.

Now, go to the Import Setup tab.

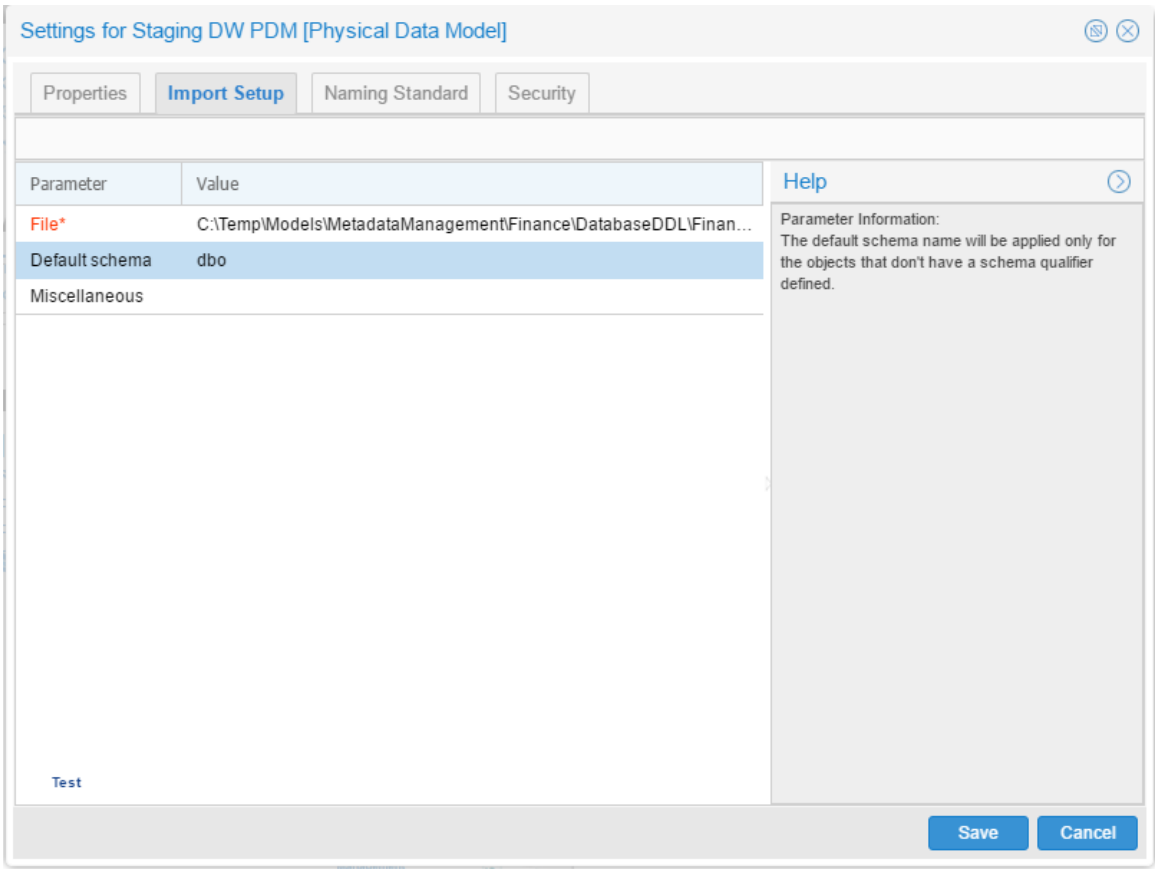


Figure 45 - Import setup

These are the properties to be used when importing this base data store for the physical data model.

Then, click the Create button and the model content will be created. You will be offered the chance to import the model immediately.

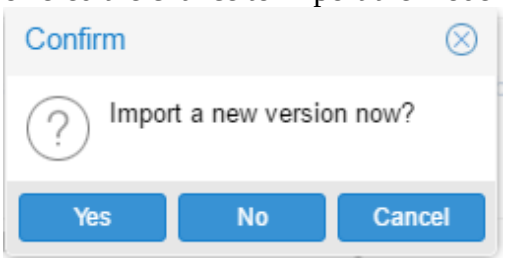


Figure 46 - "Import now" dialog

Click the No button.

Now, the last step is to harvest the PDM with the database content. To do so, simply specify Sync with Database:

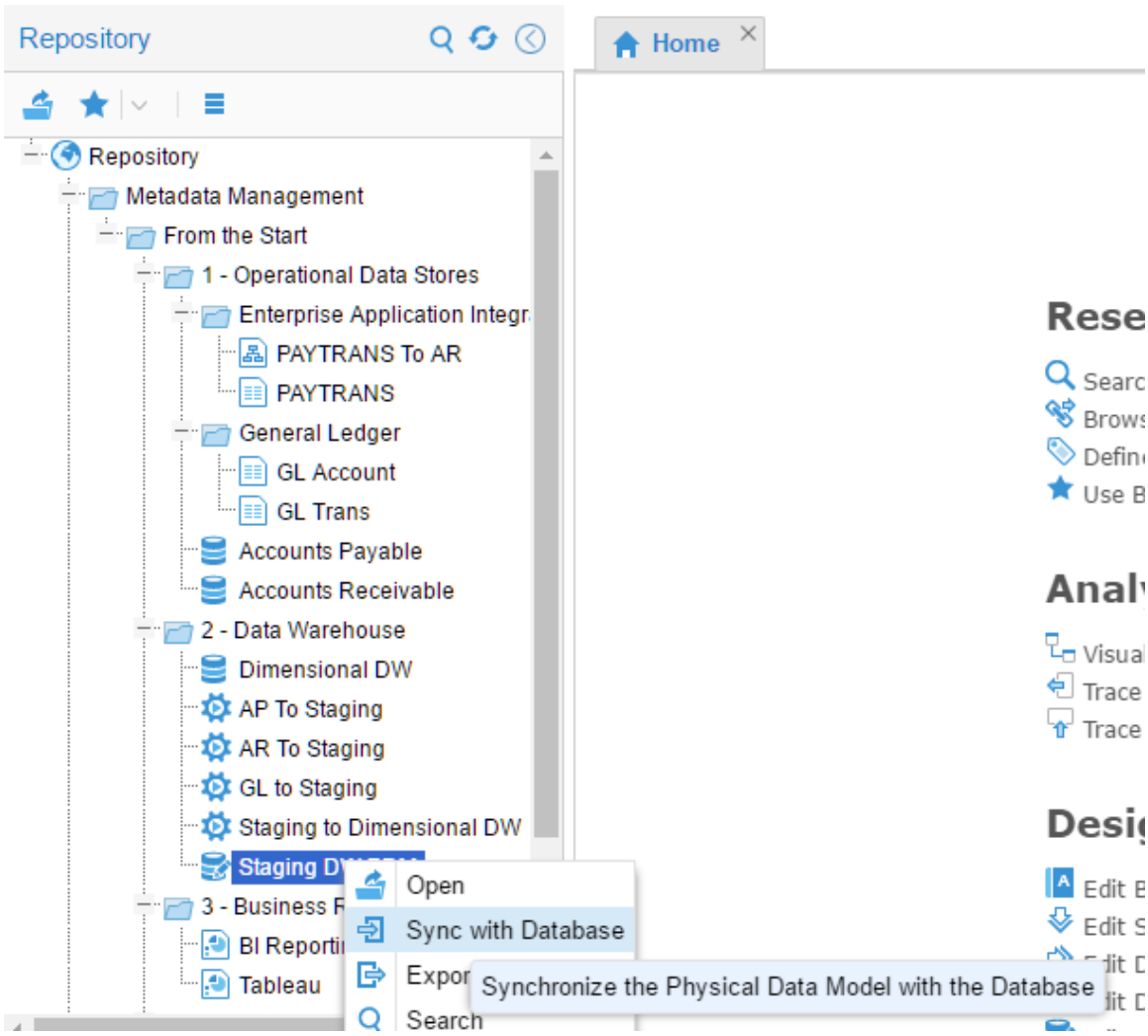


Figure 47 - Sync with Database

3.4.2.2 Harvest the rest of the warehouse models

Now, harvest the other models in the 2 – Data Warehouse folder, using the methods above.

3.4.3 Harvest Finance Business Intelligence Metadata

Two models reside in this folder:

- BI Reporting which is a model originally harvested by connecting to an SAP Business Objects business intelligence repository and retrieving an OLAP model derived from the dimensional warehouse we just harvested, along with several reports on that OLAP model.
- Tableau which is a model originally harvested by connecting to a Tableau based web server and retrieving connection definitions to the dimensional warehouse we just harvested, along with several reports.

As these bridges requires a live connection to SAP BusinessObjects and Tableau, we will simply skip this step. When we restore again via scripts, this metadata will be populated for us. However, for reference, here are the bridge parameters used:

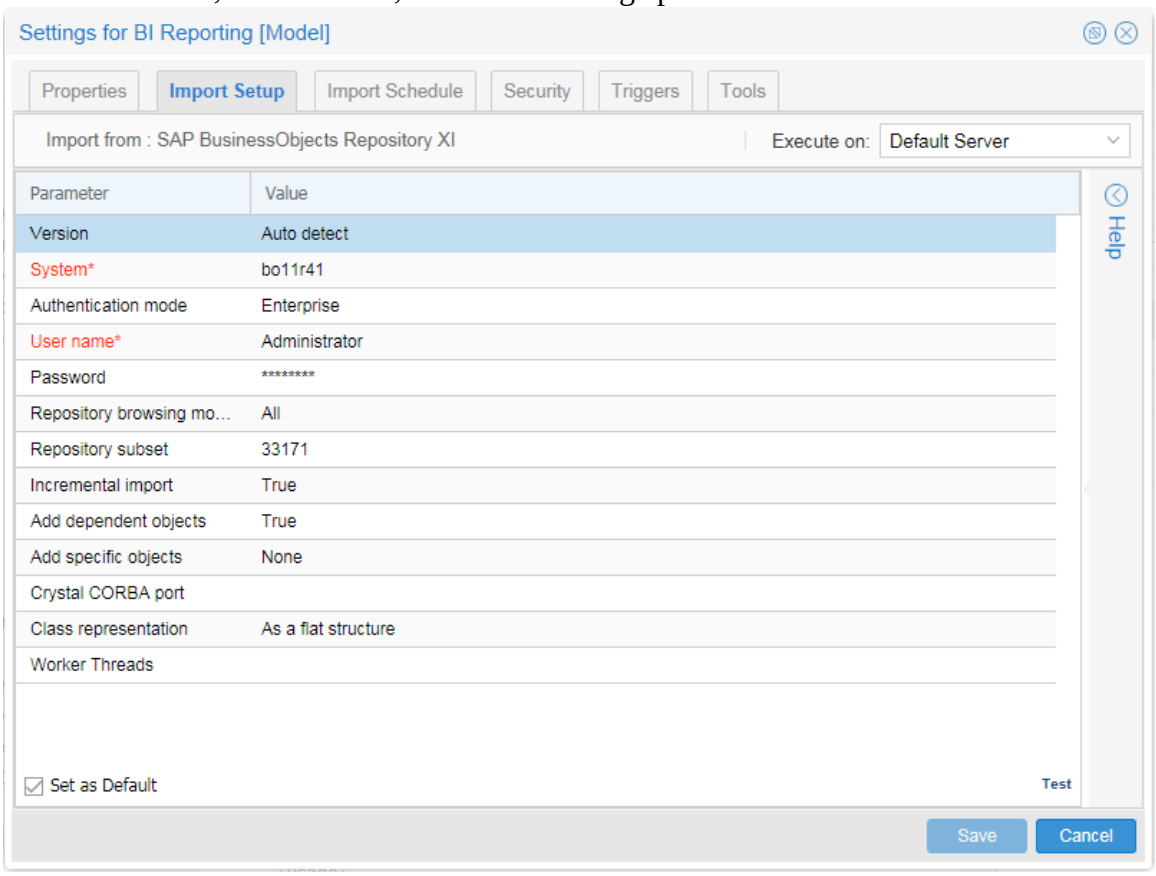


Figure 48 - Harvesting the SAP Business Objects model

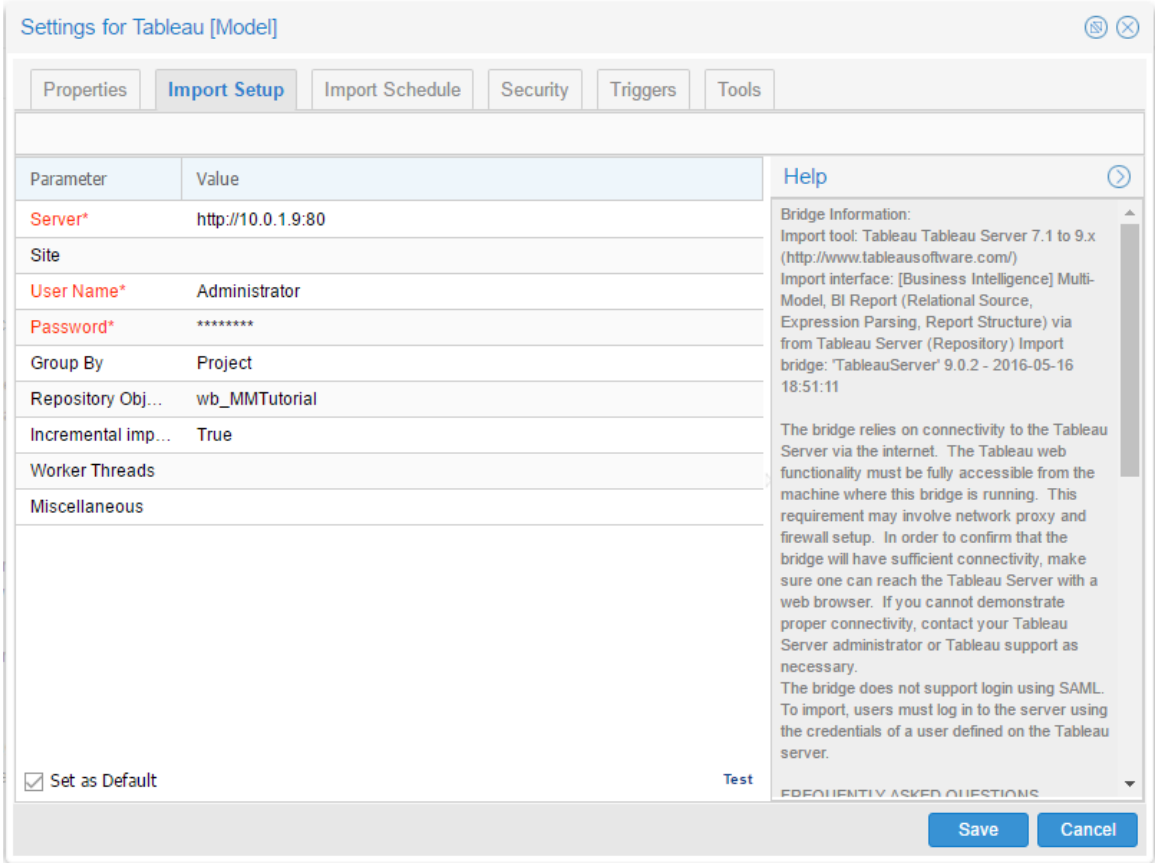


Figure 49 - Harvesting the Tableau model

There is no need to restore this model as it was already restored in the script restore at the beginning of the Tutorial.

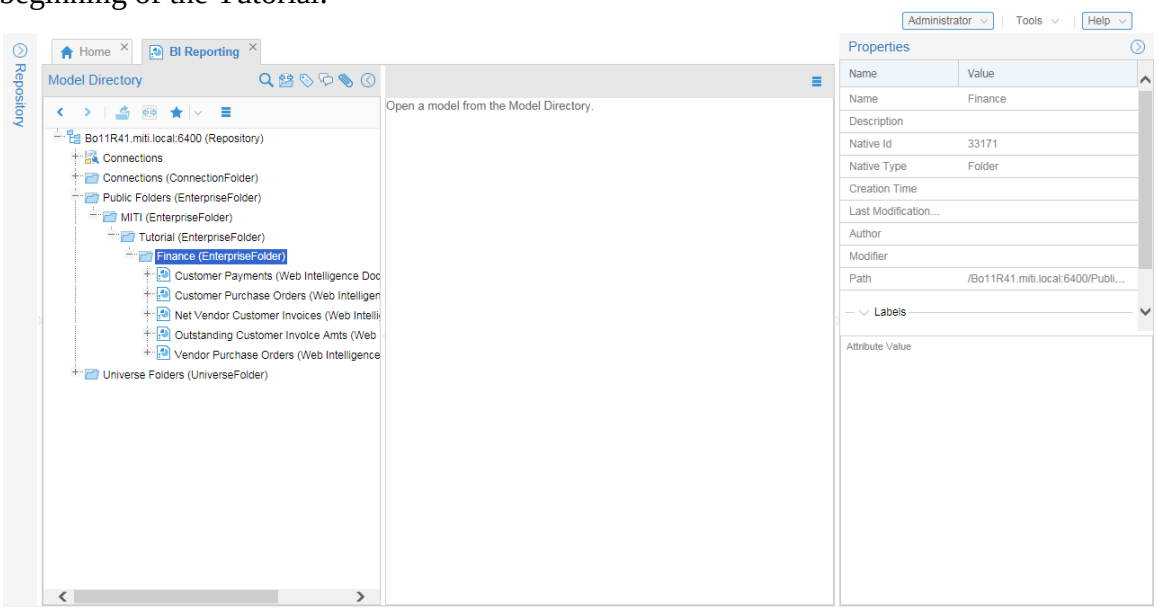


Figure 50 - Finance Business Intelligence multi-model in the Model Browser tab

Once harvested, you may see above that the model actually contains several models. Again, internally this is referred to as a *multi-model*. Model_1 is a *universe* file, the OLAP model in SAP BusinessObjects terminology, and the five reports

- Customer Payments
- Customer Purchase Orders
- Net Vendor Customer Invoices
- Outstanding Customer Invoice Amts
- Under-Invoiced Customer Purchase Orders
- Vendor Purchase Orders

Opening the report model in another tab shows the following:

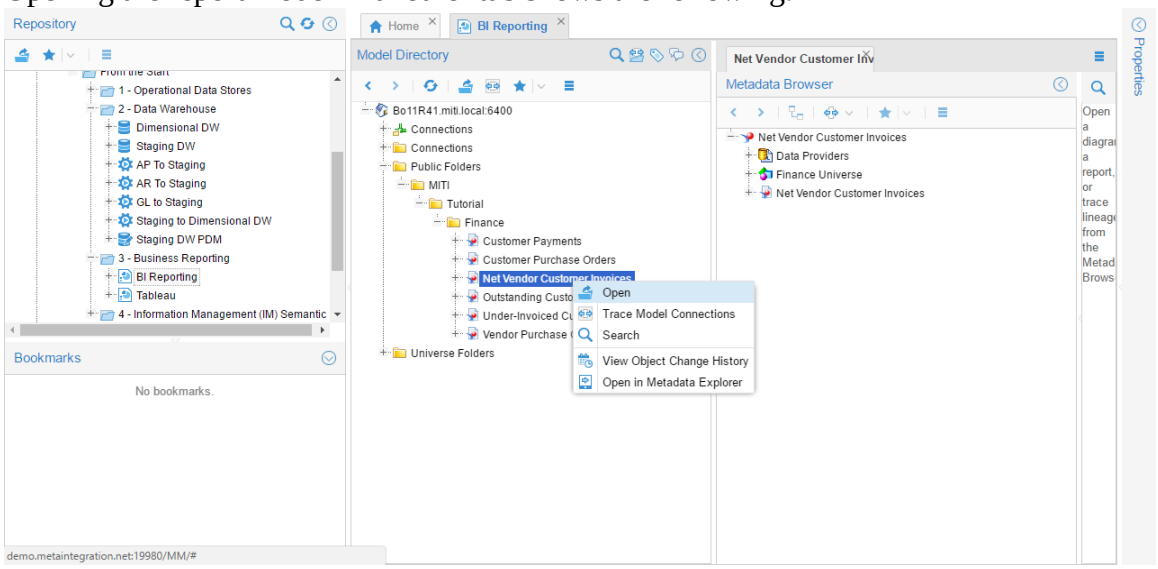


Figure 51 - Finance Business Intelligence report in the Model Browser tab

Are also models contained within the multi-model. Clicking on the Model Connection Overview tab will also show that these models are already represented in terms of their lineage. I.e., they are already stitched (see the section on configuration management for a more detailed definition of stitching and lineage). As the lineage information was already available in the source environment, it is maintained in Meta Integration® Metadata Management (MIMM) as well.

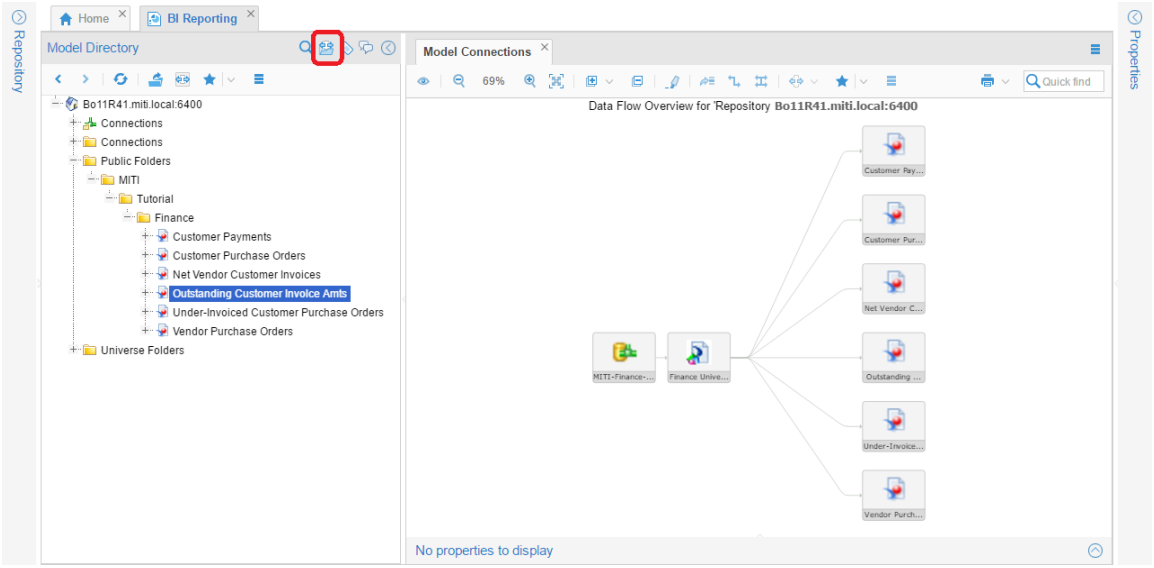


Figure 52 - Finance Business Intelligence multi-model in the Architecture Diagram tab

4 Configuration Management

The data processes and data stores that were cataloged in a previous section are now represented as models within the repository and they have been populated with metadata (harvested). This section of the tutorial will deal with the mechanics of creating *configurations* and then *stitching* the metadata extracted from the data stores to the metadata extracted describing the data processes.

4.1 Configuration Management

The configuration is an extremely important concept in Meta Integration® Metadata Management (MIMM). It is the scope for many operations, including lineage analysis, search, version management, etc. In this way, what would otherwise be an overload of information (everything in the repository), is instead well managed according to the configuration of metadata one is interested in analyzing or manipulating.

A valid *configuration* consists of relationships between the configuration and a collection of model contents and *stitchings.* Just as we have seen in the previous sections, the model contents relate to data stores and data processes that have been harvested into the repository. The stitchings are repository generated mappings between the model contents (data stores and data processes) as in the diagram below.

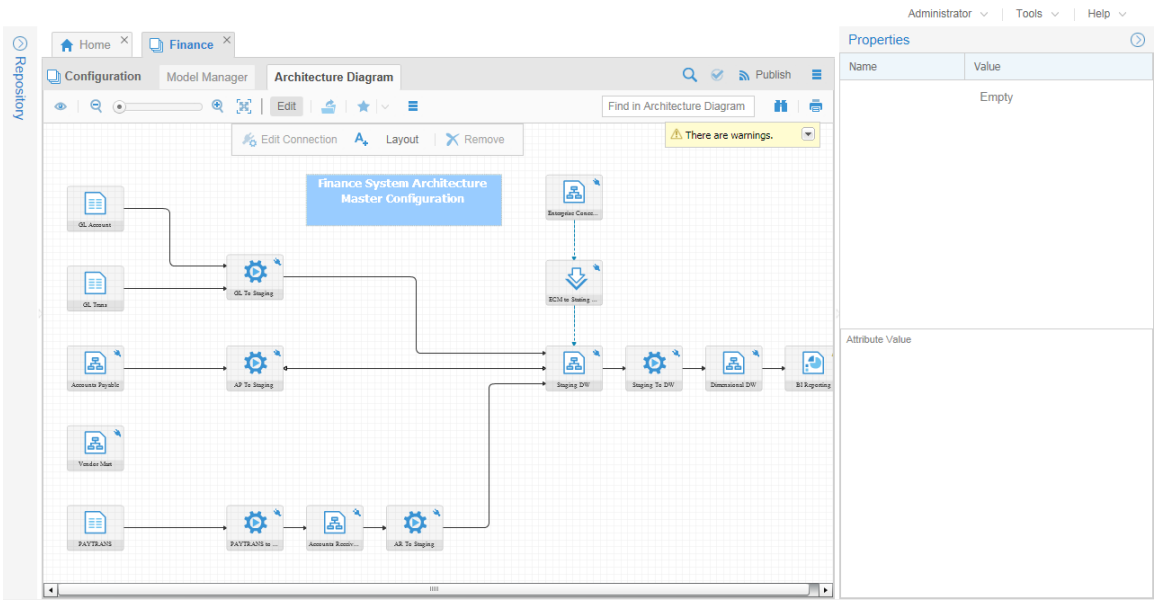


Figure 53 - Configuration lineage

The Meta Integration® Metadata Management (MIMM) accomplishes this management by the definition and use of repository configurations. Imbedded within the UI is the *Configuration Manager*. This tool provides a drag and drop based visual interface for constructing configurations of models and mappings within the repository. A configuration is represented by a *configuration content* within the repository.

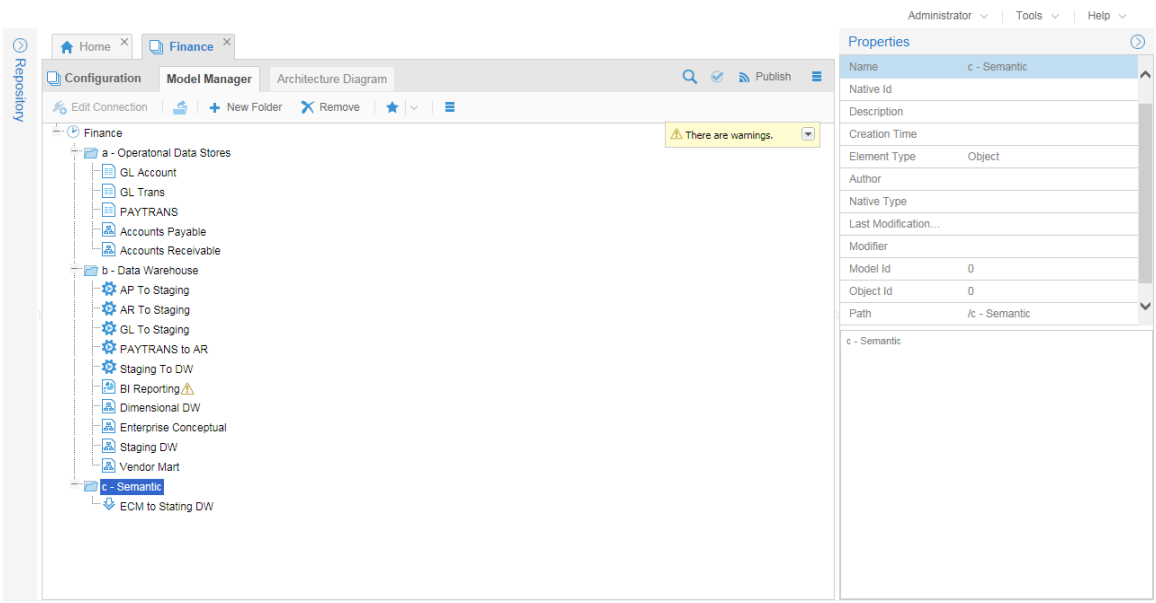


Figure 54 - Configuration Content

This section describes the process of assembling those content objects into a configuration and their use.

4.1.1 User Interface Considerations

Appropriately, the configuration management user interface provided by Meta Integration® Metadata Management (MIMM) is very graphical. Thus, as with any metadata design tool (modeling, ETL mapping design, BI OLAP model design), screen real estate is at a premium. To support this requirement, it is recommended that one:

- Change the screen orientation (landscape) and resolution to the maximum horizontal resolution that is comfortable. Configuration and lineage diagrams are generally linear, from left to right and horizontal space is important.
- Use the show/hide feature of the side panels in the user interface, as in the last screen capture.
- Most commonly, one may use the *overlay mode* for these panels by simply clicking on the Repository bar or Properties bar, which will expand the side panel as a overlay, without resizing the center pane. This is the recommended approach when one merely wishes to view properties quickly for an element or perform one action in the Repository Tree.
- Activate the Meta Integration® Metadata Management (MIMM) user interface *full screen mode* by simply right-clicking on the header portion of the UI and selecting or Full Screen.
- Activate the web browser's *full screen* mode, so that much of the header information and controls are hidden when not needed.

4.2 Building the Published Configuration

4.2.1 The Configuration Model Manager

We will work with the already defined Published configuration. To work with it, either double-click on this configuration or right-click on the configuration and select Open.

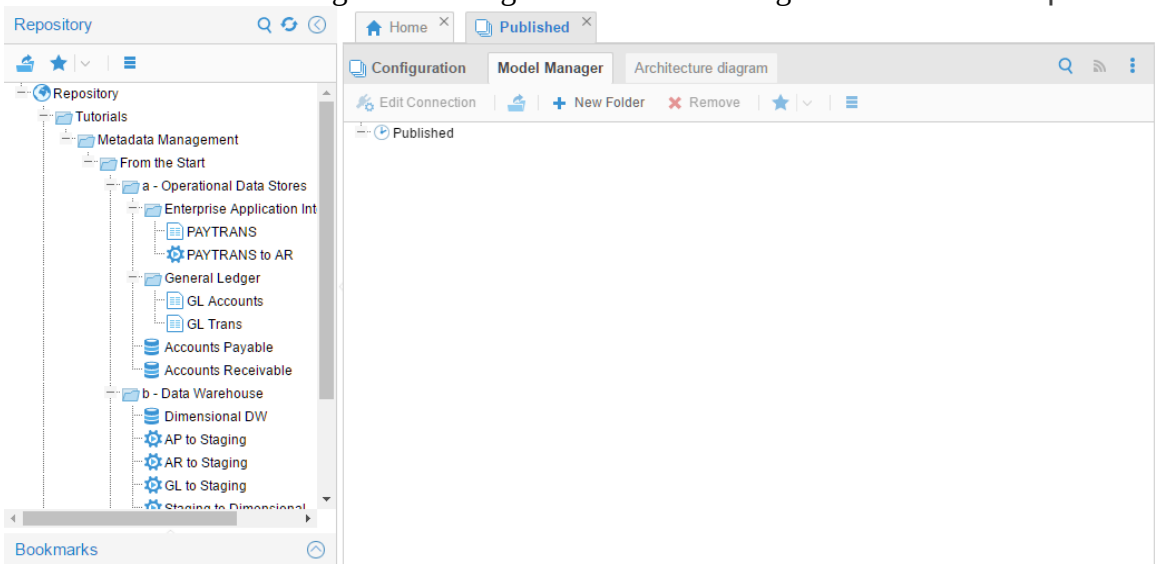


Figure 55 - Editing the Published configuration content

Note, in the image above, the properties panel is already hidden, to make more room for the graphical nature of the configuration editing process, which consists of adding model and mapping contents by drag and drop, and then stitching these models (and multi-models) together in data flow order.

4.2.1.1 Bringing Model Formats into a Configuration

Expand the 3 – Business Reporting [folder] under the Metadata Management [folder] and then drag the Tableau model into the configuration Model Manager tab.

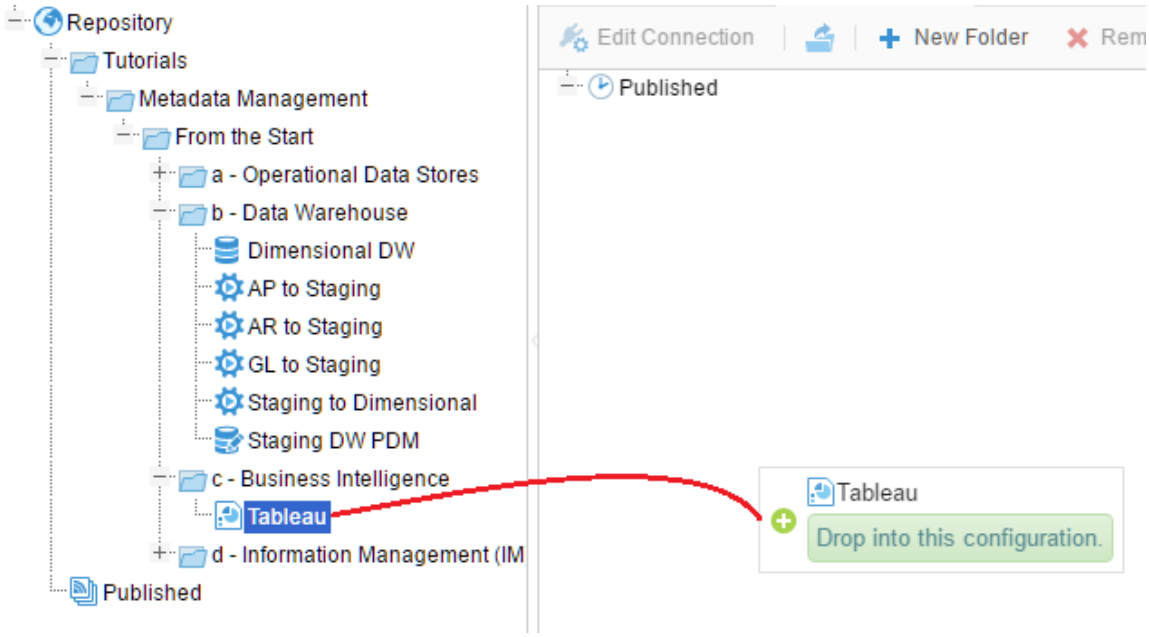


Figure 56 - Drag and drop the Tableau model content into a configuration

Now, expand the 2 – Data Warhouse [folder] under the Metadata Management [folder] and drag the Dimensional DW model into the configuration as well.

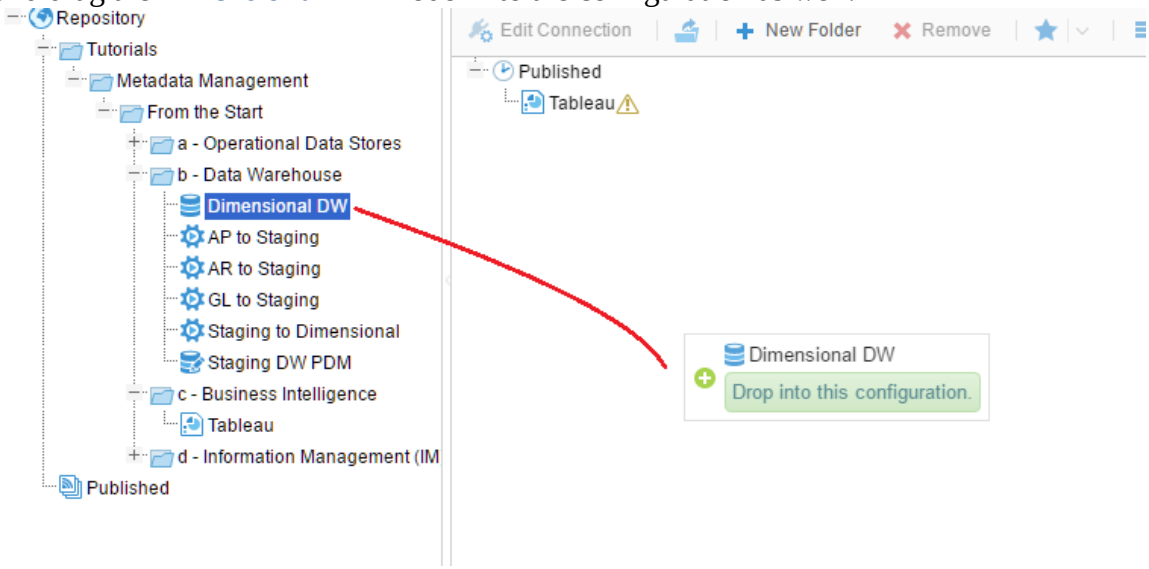


Figure 57 - Add the Dimensional DW model

- All the models are represented as items in a list with the model content name and all additional information is presented in the Properties pane.
- Note, the model with the icon (⚠) next to it, indicating that there are connection definitions to be resolved. This is a business intelligence model, as we discussed in the last chapter, and it represents a reporting environment that read from one or more databases (a *connection*) and then use that information for the reporting. In this case, it reads from the Dimensional DW database, and thus we added this

model also to the configuration. We now need to *stitch* the connection to the database model.

Right-click on the Tableau model and select Edit Connection.

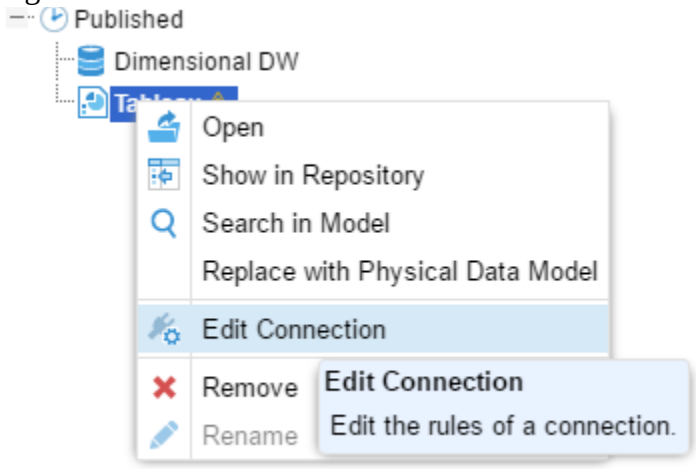


Figure 58 - Edit Connection

You are presented with the connection editor showing any connections which need to be resolved.

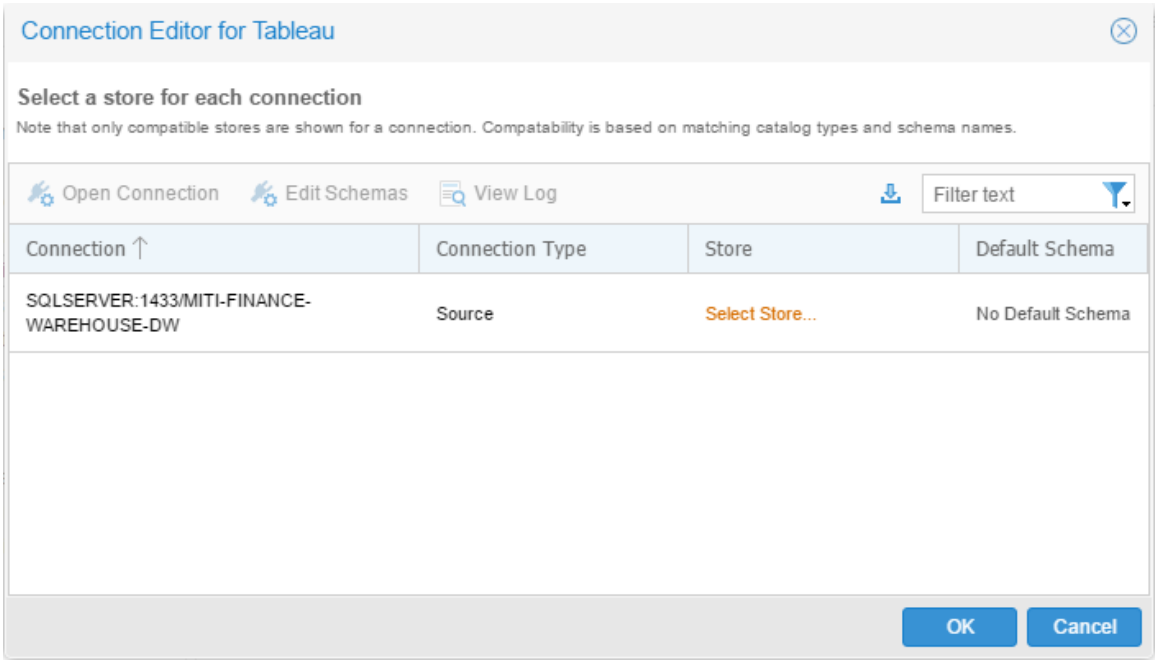


Figure 59 - Connection editor for BI Reporting

Then pick the correct data store (only one to choose from as that is all we added to the configuration). Note, for Default Schema, the UI reports that there is No Default Schema. This means that the BI specified the fully the schema name that it was reading from. In other case, an BI process may not specify a schema name. In that case, the default was assumed and it may be necessary for you to specify which schema should be taken as the default.

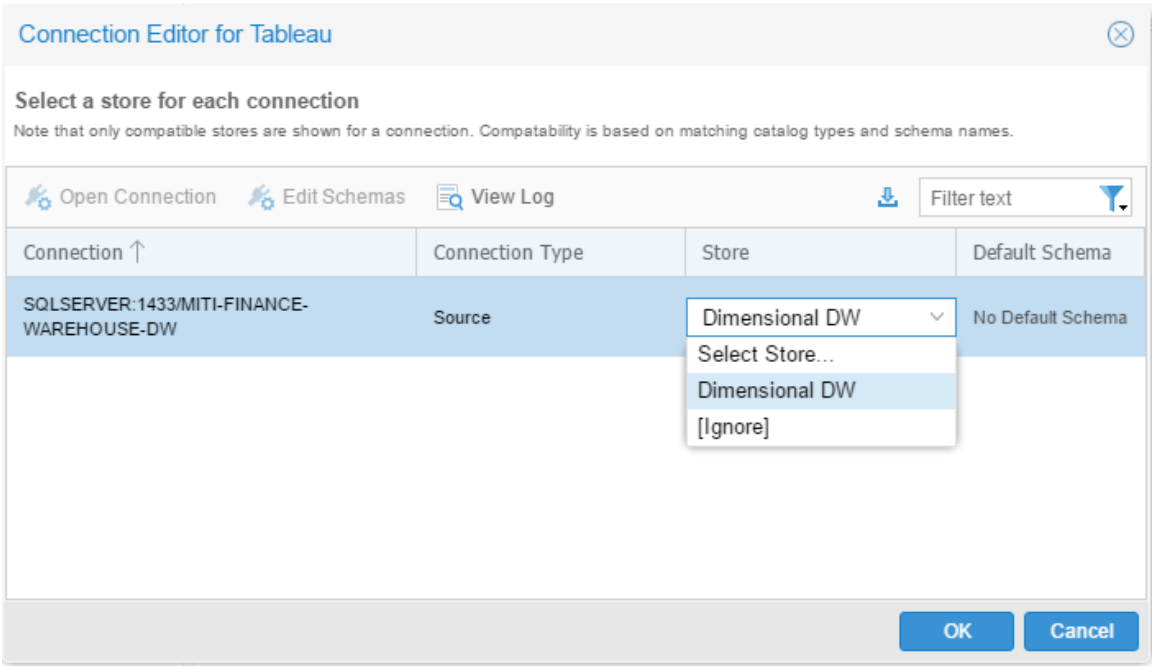


Figure 60 - Define the connection store

Then click on the OK button.

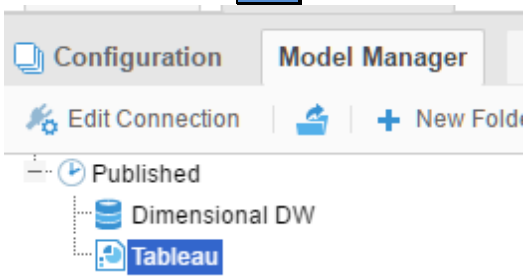


Figure 61 - Configuration

Note the warning icon (⚠) is no longer next to the Tableau model, indicating that all the connection definitions are resolved.

We now have all the business intelligence model connection definitions properly associated with the Dimensional DW data store.

Now, click on the Architecture Diagram tab.

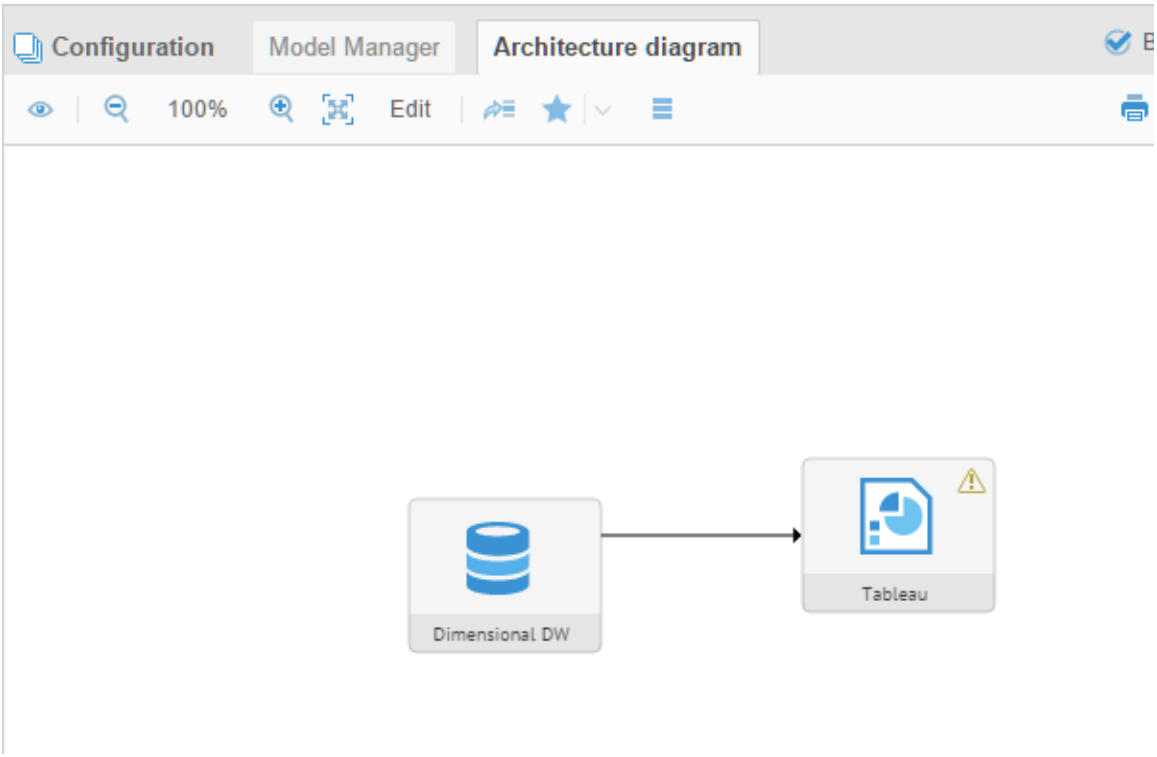


Figure 62 - Architecture diagram for BI

Note, the arrows connect the Dimensional DW model to the two BI models, pictorially depicting the stitching and pointing in the direction of data flow (left to right, or data store to BI).

In addition, we have the warning icon (⚠) back next to the two BI models (with connection definitions). Here, this icon indicates that those stitchings have not yet been *built*.

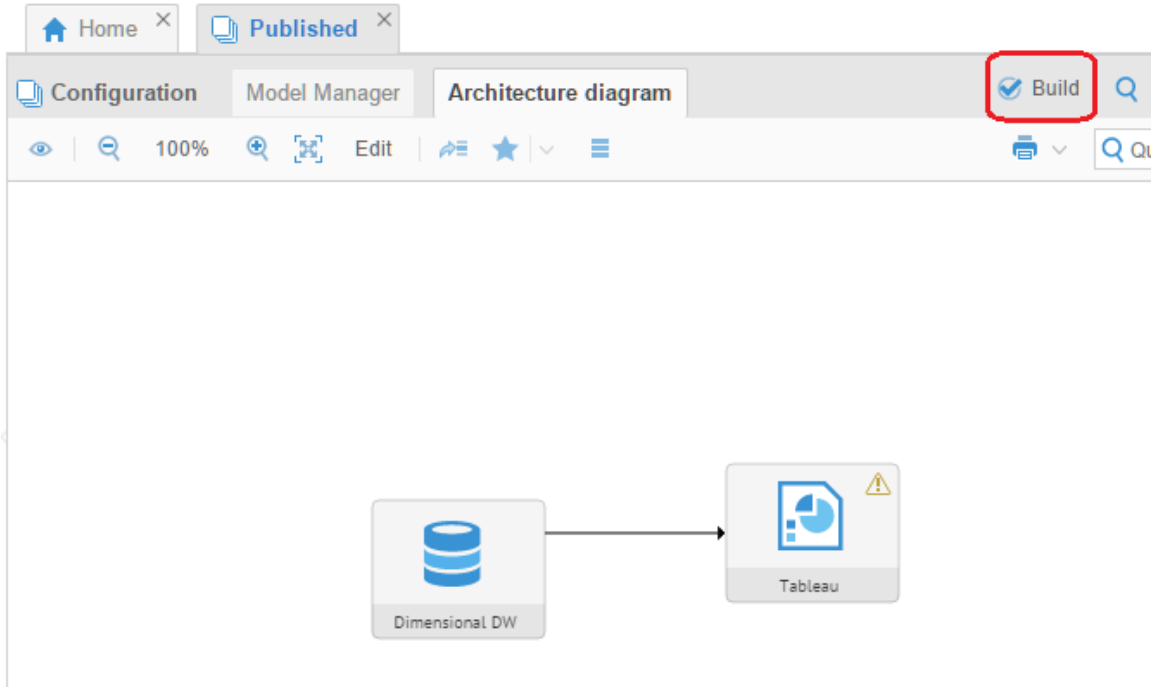


Figure 63 - Build icon

Click on the Build icon.

During this process Meta Integration® Metadata Management (MIMM) is validating and then pre-calculating and caching the lineage in the repository database. Meta Integration® Metadata Management (MIMM) caches this information, so that it does not need to be recomputed on demand, providing very quick presentations of lineage when viewed later.

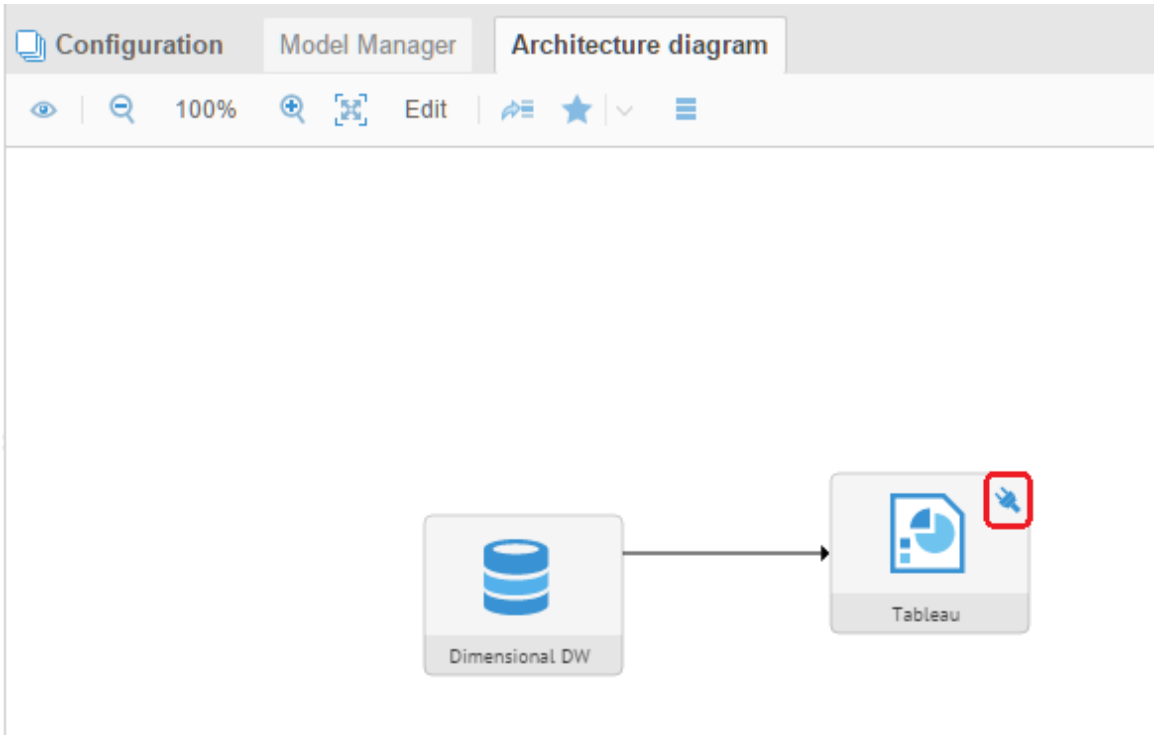


Figure 64 - Warning in configuration

If there were issues in the stitching, you will notice that there is a warning where the Build icon was and you would see warnings on the model itself. Instead, here we see the connection icon that they are defined correctly.

Now, let's move back from right to left ultimately back to the source systems. The next data store backwards through the data flow process is the Staging data warehouse. Recall, this is a special model though, as it is a Physical Data Model. Thus, we include the Staging DW PDM.

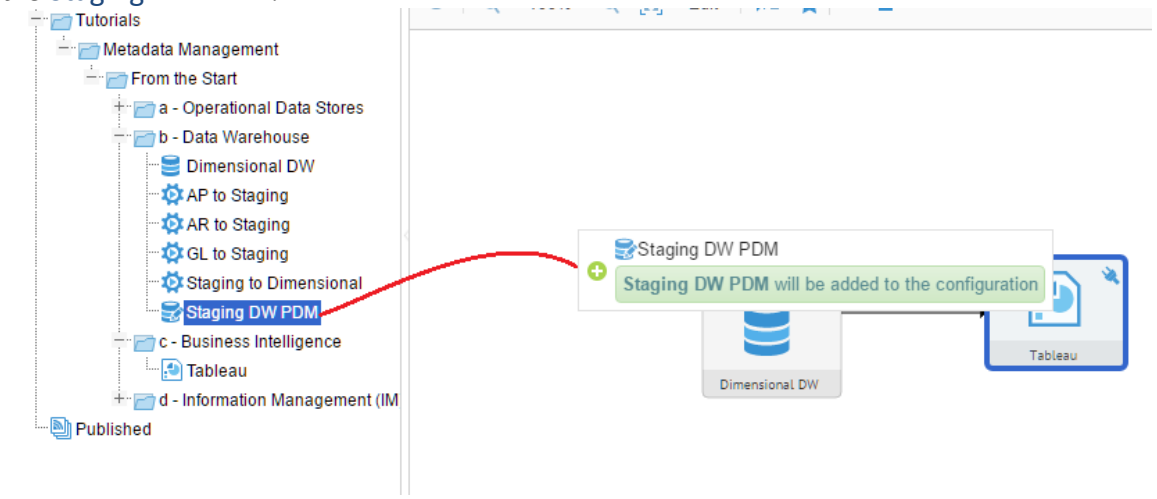


Figure 65 - Adding Staging DW PDM

Now, there is an data integration process that must existing to move the data from the Staging DW database to Dimensional DW database. This is call Staging to Dimensional DW and it must be included in the configuration in order to stitch it.

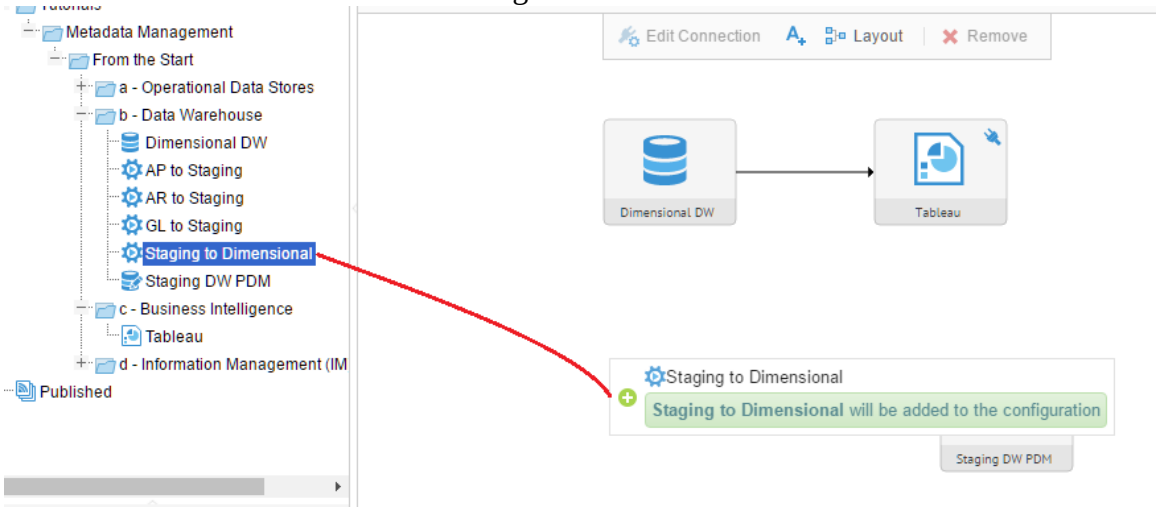


Figure 66 - Add the Staging to Dimensional DW data integration process

Once that is done, you see the warning about connection definitions on only one of the two new models (Staging to Dimensional). This model is an ETL model, and thus had connection definitions to at least one source, one destination and transformations in between.

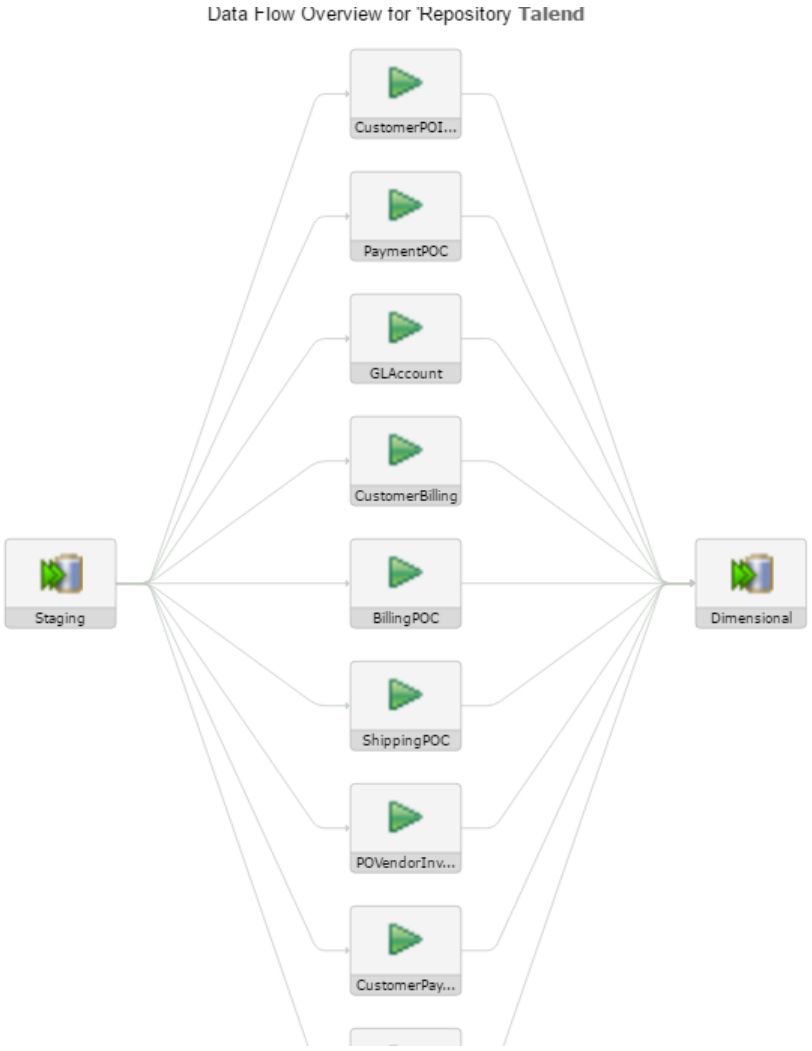


Figure 67 - ETL with source and target connections

To stitch these connections, all we need to do is right-click on the Staging to Dimensional DW model and select Edit Connection

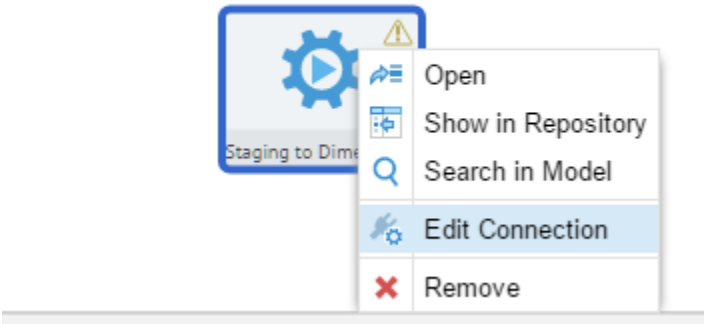


Figure 68 - Edit connection

and you are presented with the connection editor showing any connections which need to be resolved.

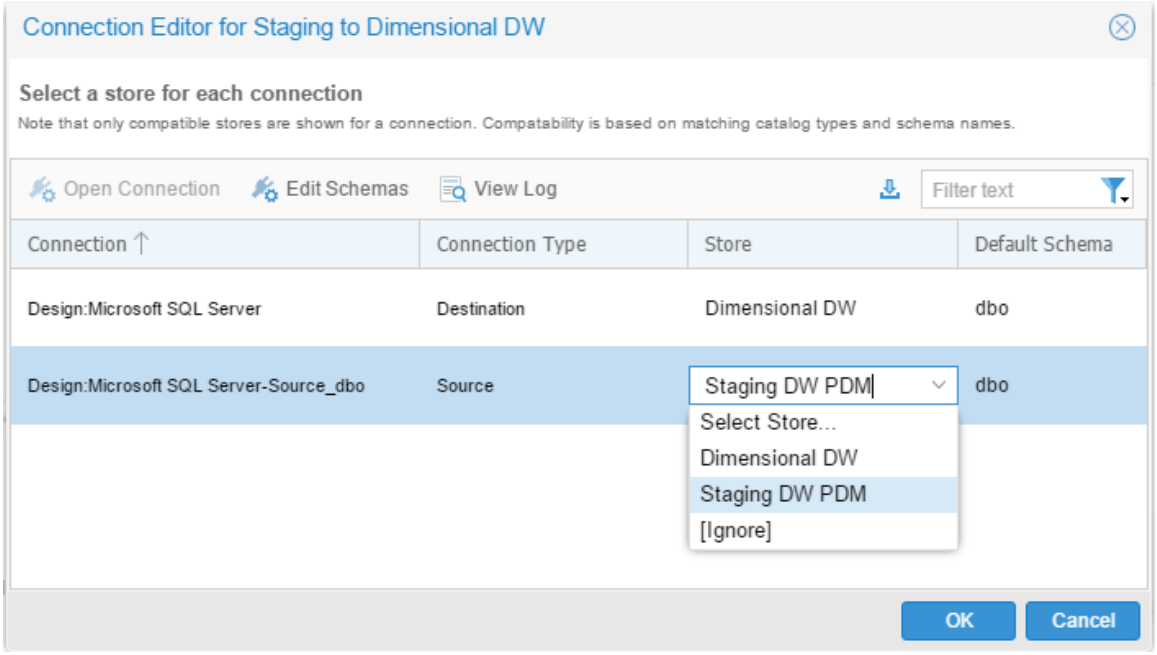


Figure 69 - Edit connection for Staging to Dimensional DW

Note, the names of the connections are not that clear. This is very common in the real world and indicates why it is often necessary to have a knowledgeable person to provide the information about the organization's architecture in order to get these right.

Click on the OK button.

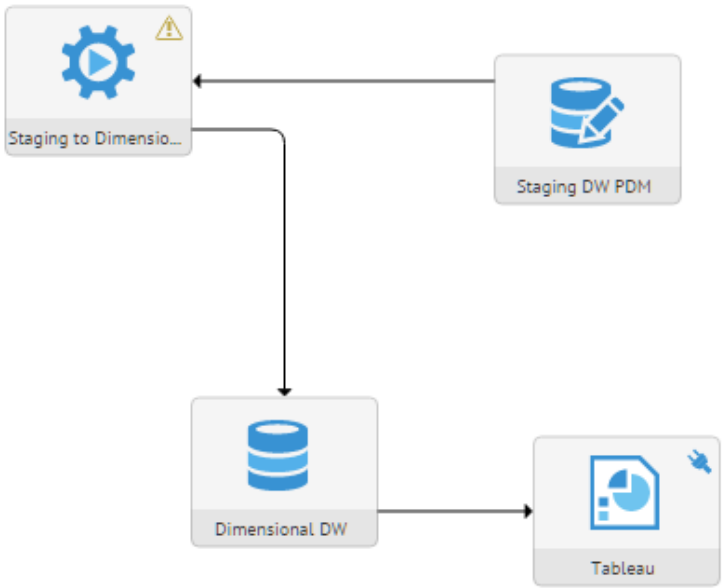


Figure 70 - Architecture with warehouses

Now, click the Build icon again.

Be sure you are still in the Architecture Diagram tab and then click on Edit and Layout

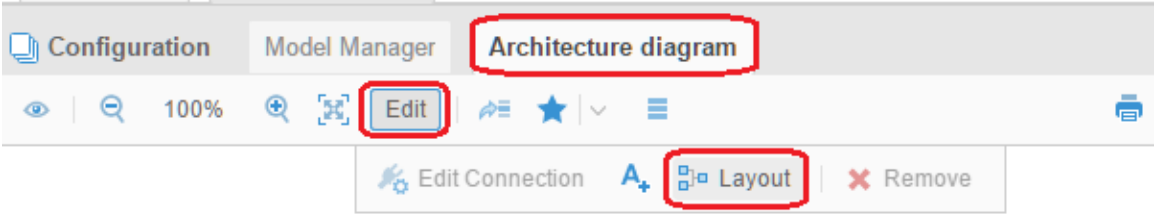


Figure 71 - Layout operation

And we have a nicely rendered architecture from the Staging DW to the BI.

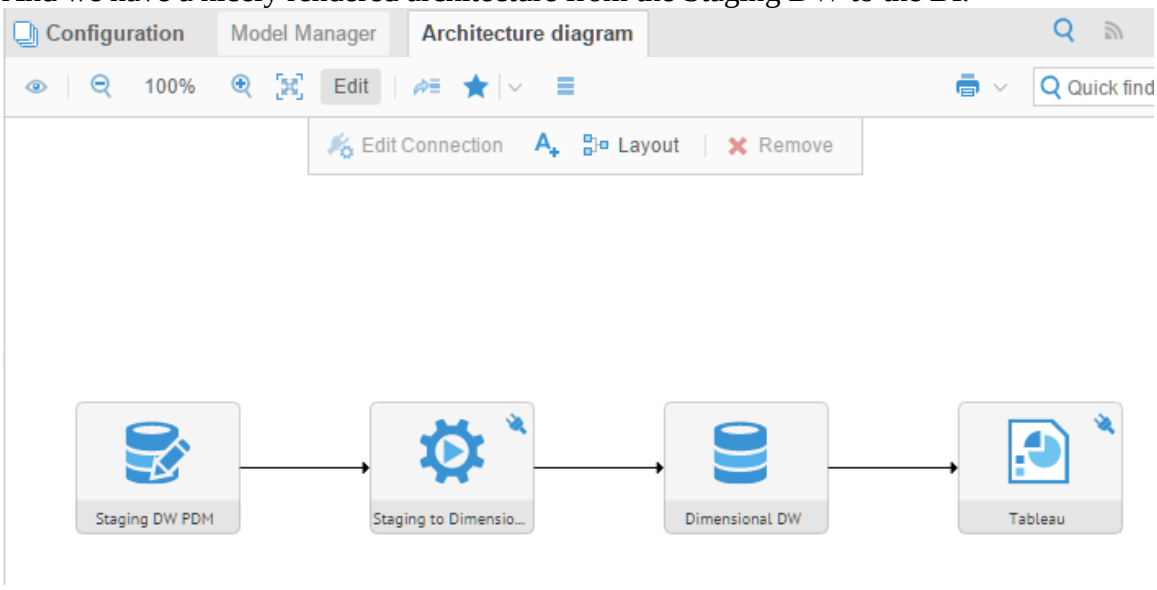


Figure 72 - Architecture presentation

Now, let's move back from right to left ultimately back to the source systems. Bringing in all the rest of the models required (remember, the Staging DW is not needed) in the folder 2 – Data Warehouse:

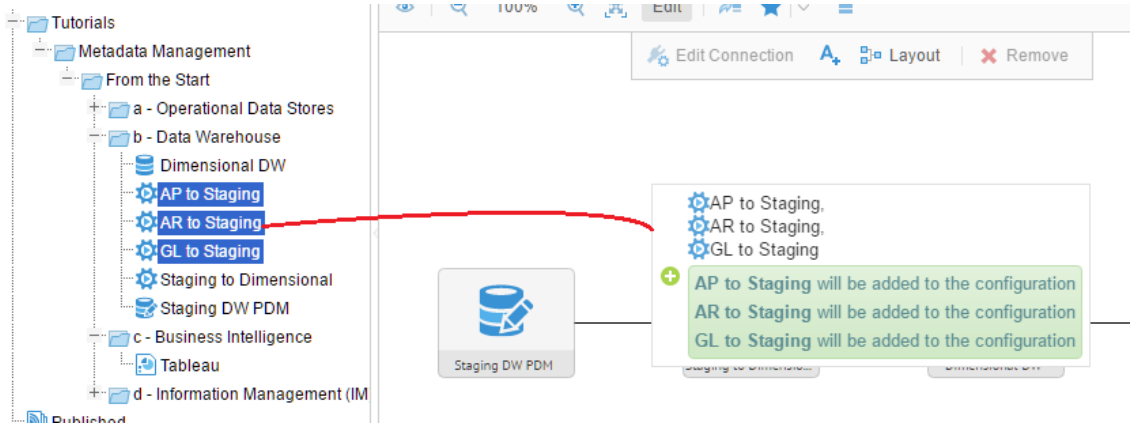


Figure 73 - Adding the rest of the Warehouse models

Note, you may multi-select for drag and drop.

Then, bringing in all the rest of the models required from the folder 1 – Operation Data Stores and any sub folders:

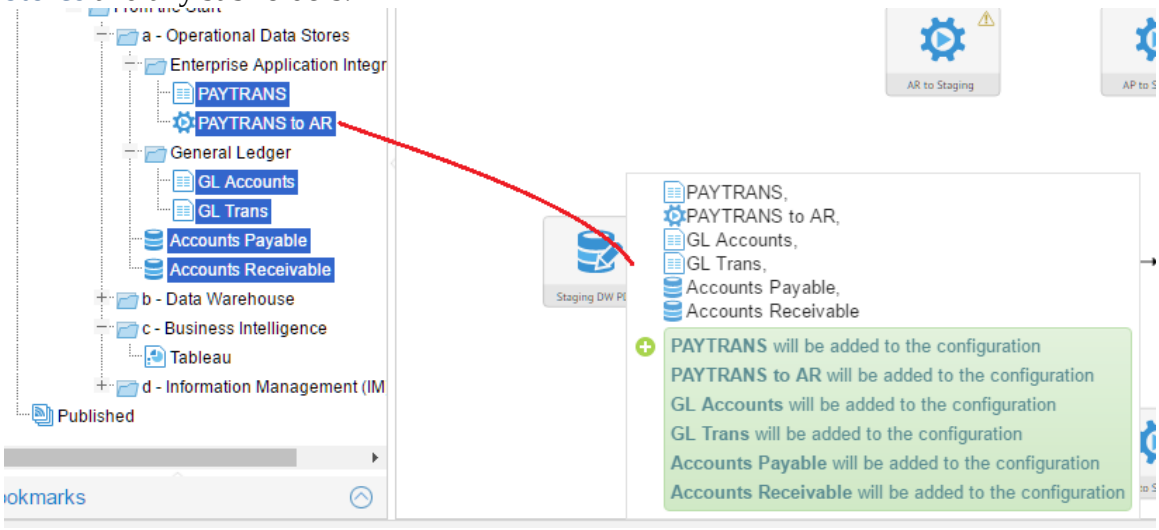
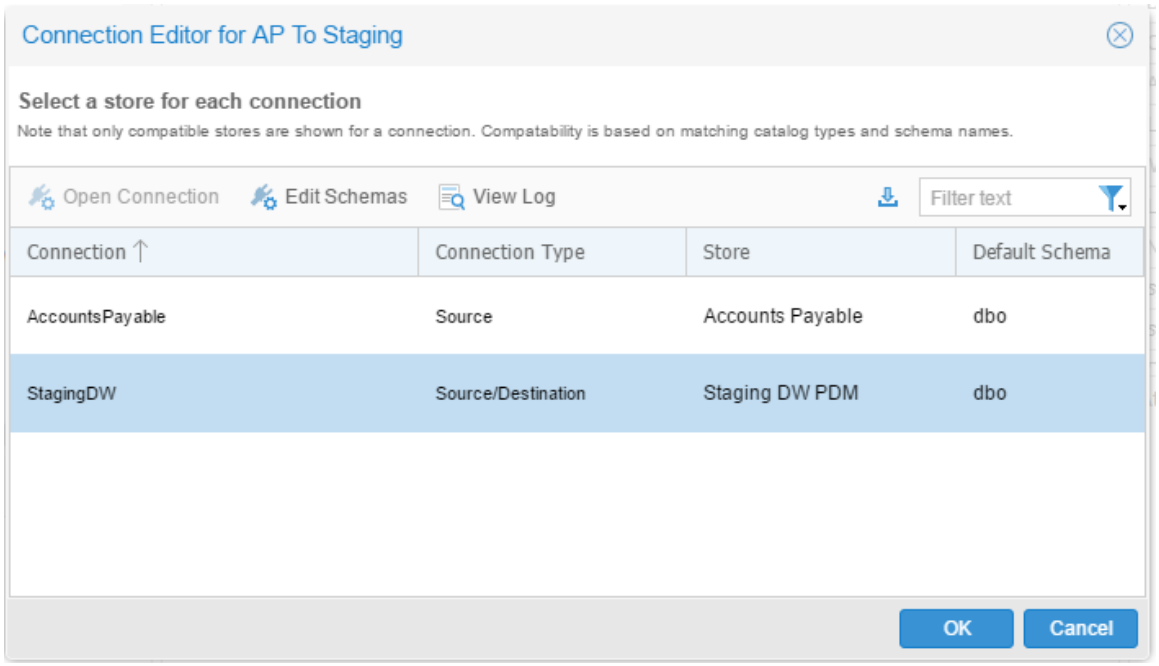
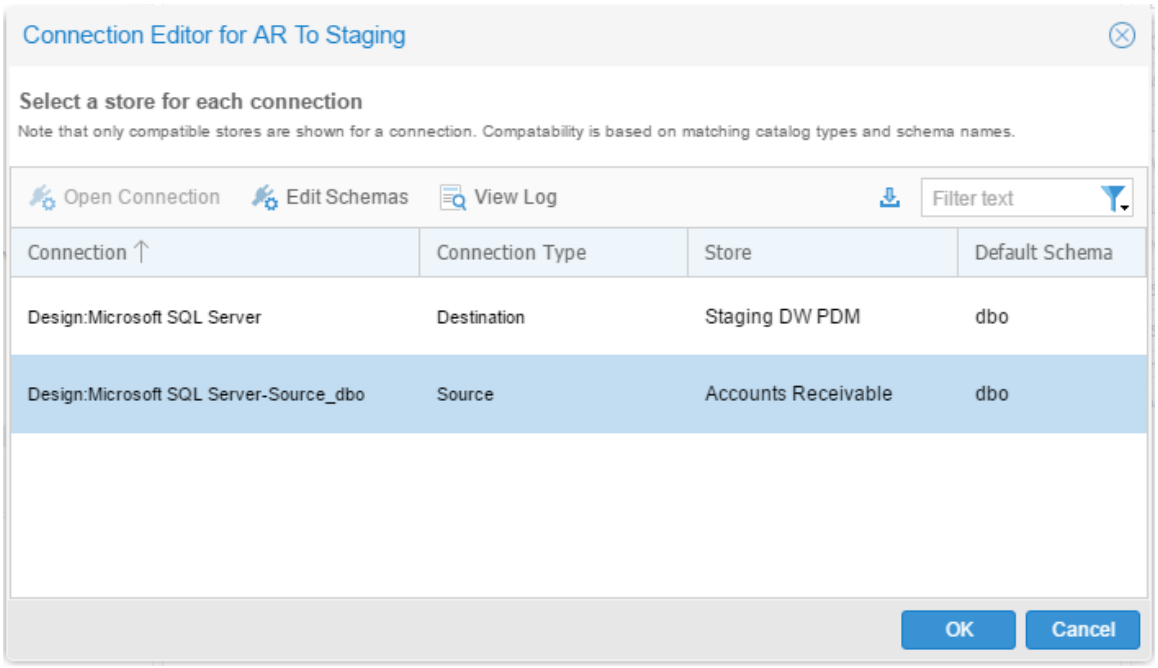
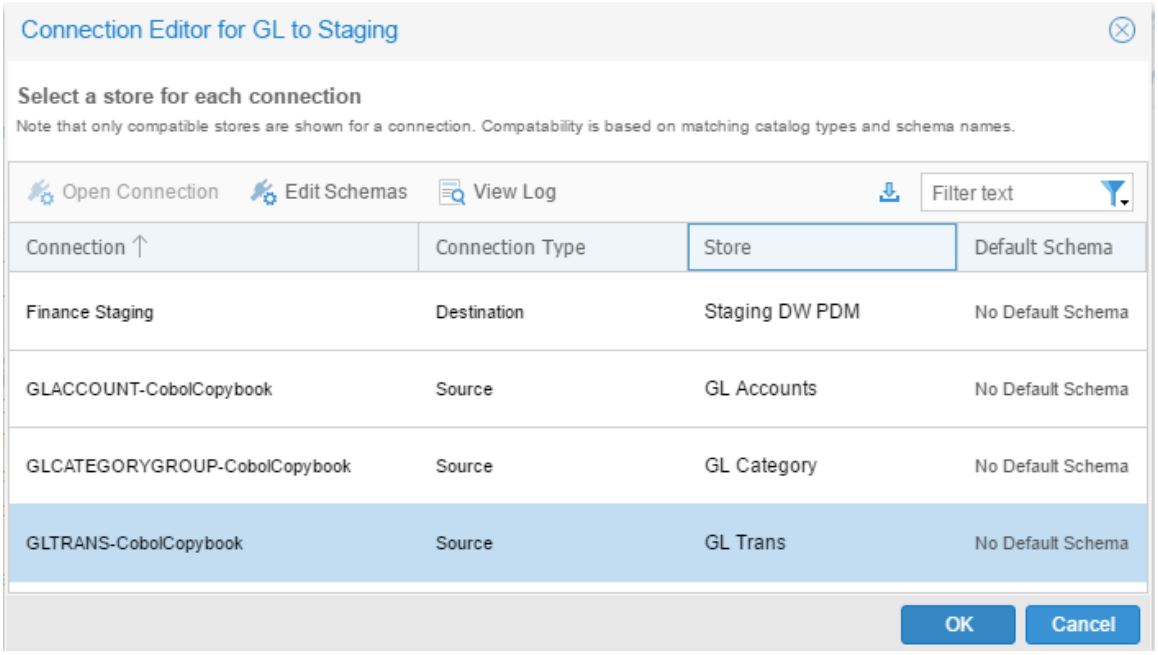


Figure 74 - Operational data stores

Now, we again have unassociated connections in the ETL models we brought in. We need to systematically define all of these connection rules as follows:







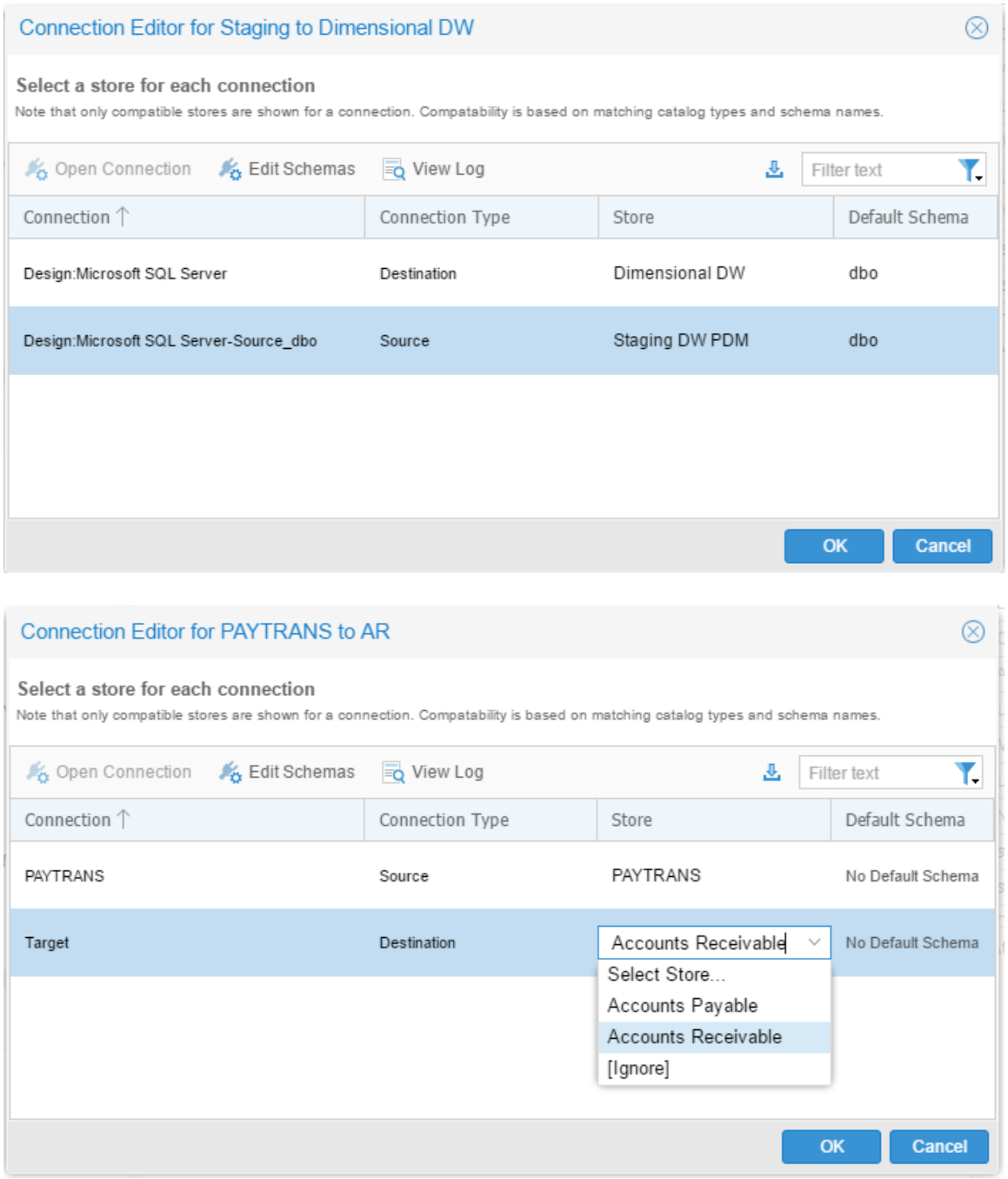


Figure 75 - Remainder of connection rules to complete including the model contents from the b – Data Warehouse [folder] in the Published configuration

Finally, it is time to compute the lineage. Click on click the Build icon:

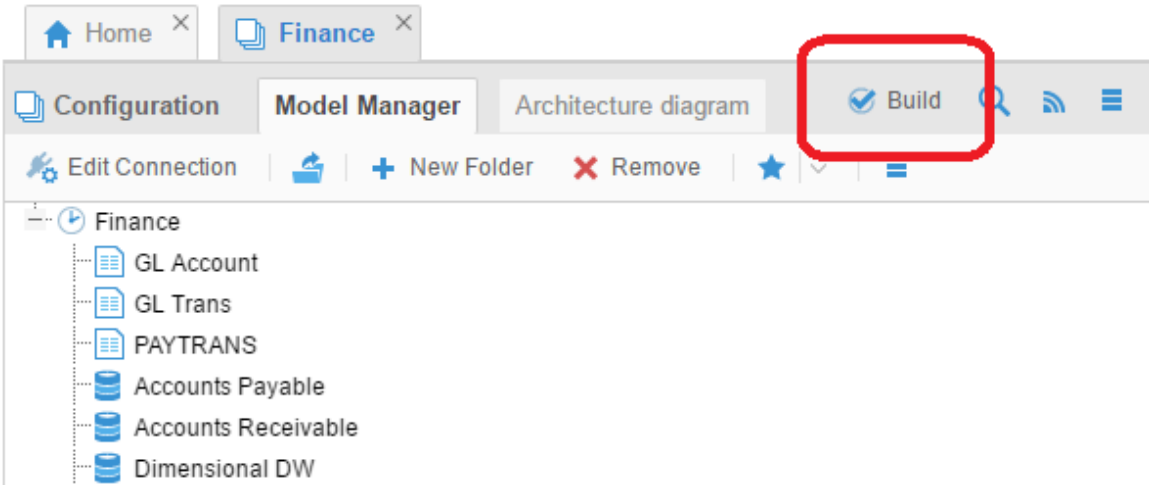


Figure 76 - Build icon

Be sure you are still in the Architecture Diagram tab and then click on Edit and Layout

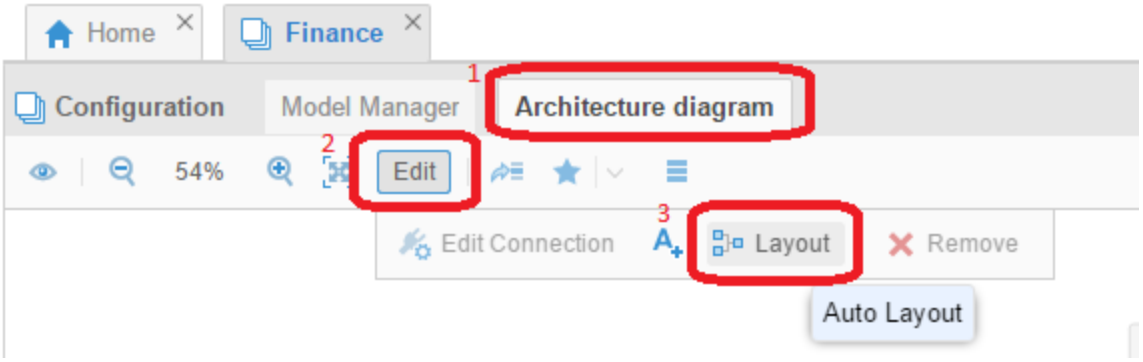


Figure 77 - Layout operation

And we have a nicely rendered architecture from the data flow architecture.

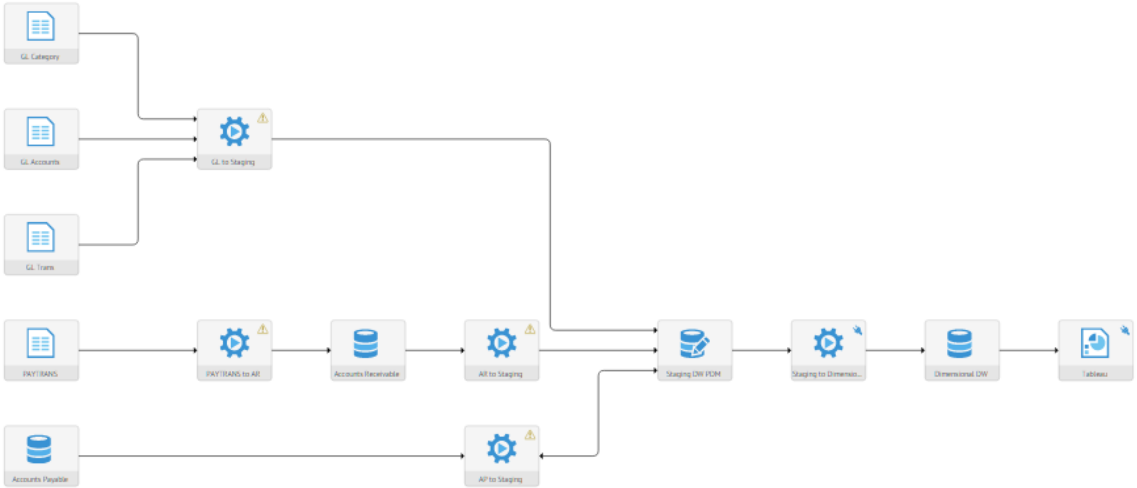


Figure 78 - Data Flow Architecture

4.3 Configuration folder organization

Let's organize, and place the models and mapping in folder, using the plus sign to Add Folder and dragging the models in like so:

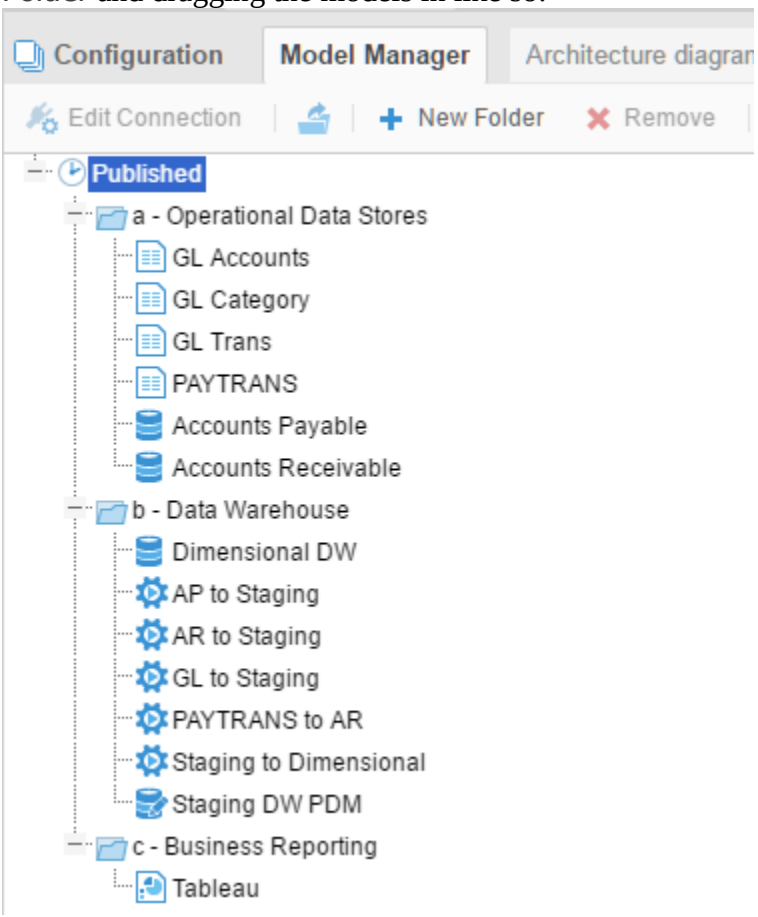


Figure 79 - Folder organization for the Finance configuration

Now, click the Build icon again.

Switch to the Architecture Diagram tab and then click on Edit and Layout

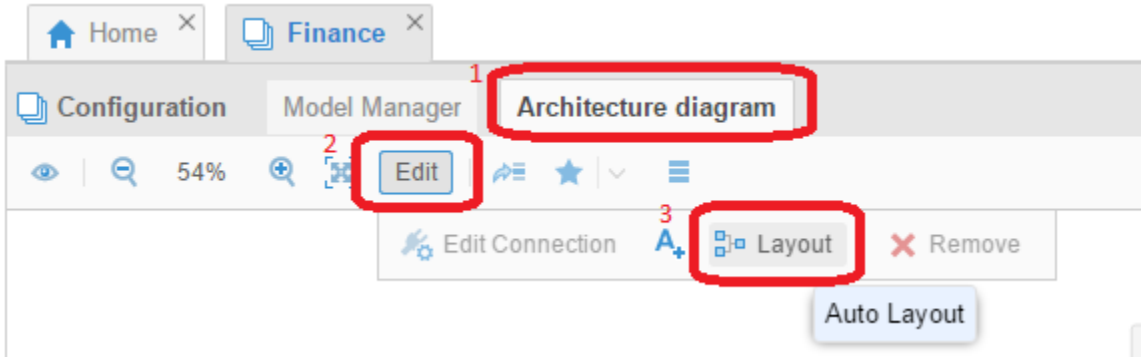


Figure 80 - Layout operation

And the result is:

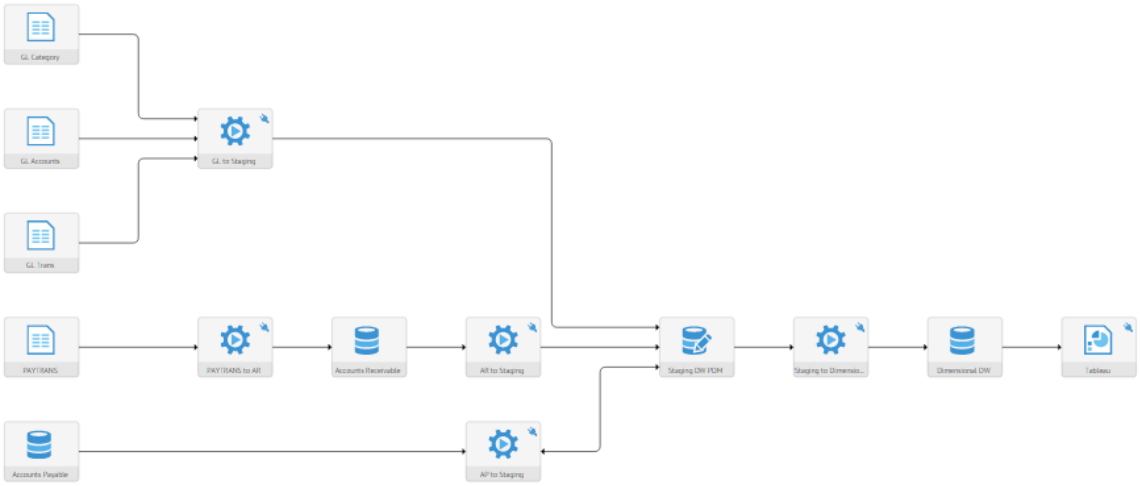


Figure 81 - Published configuration architecture diagram

Click on Edit again to save the layout.

4.4 Configuration Architecture Diagram

Now, let's improve the diagram's appearance.

4.4.1 Shapes

Add annotations to the diagram, using the Shapes icon:

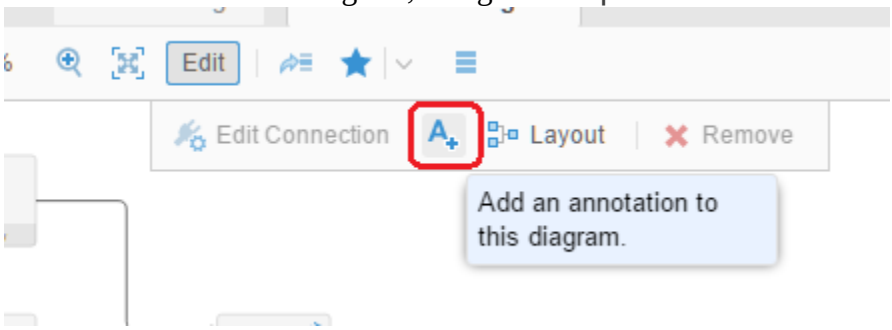


Figure 82 - Add annotation

Then open the Properties panel and edit the text inside:

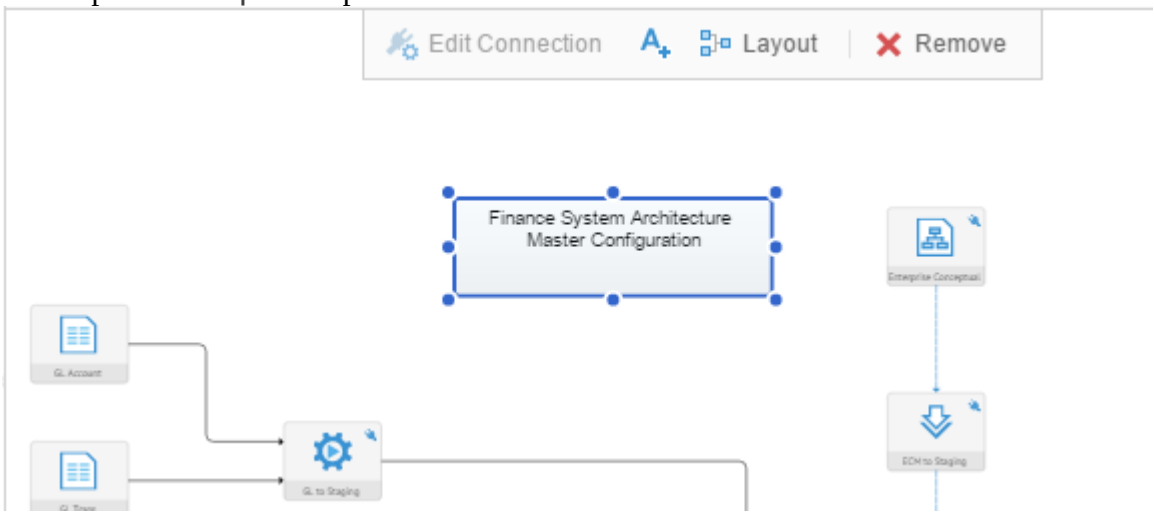


Figure 83 - Add text to annotation

Right-click on the annotation and select Annotation Options and edit the font and colors to match here:

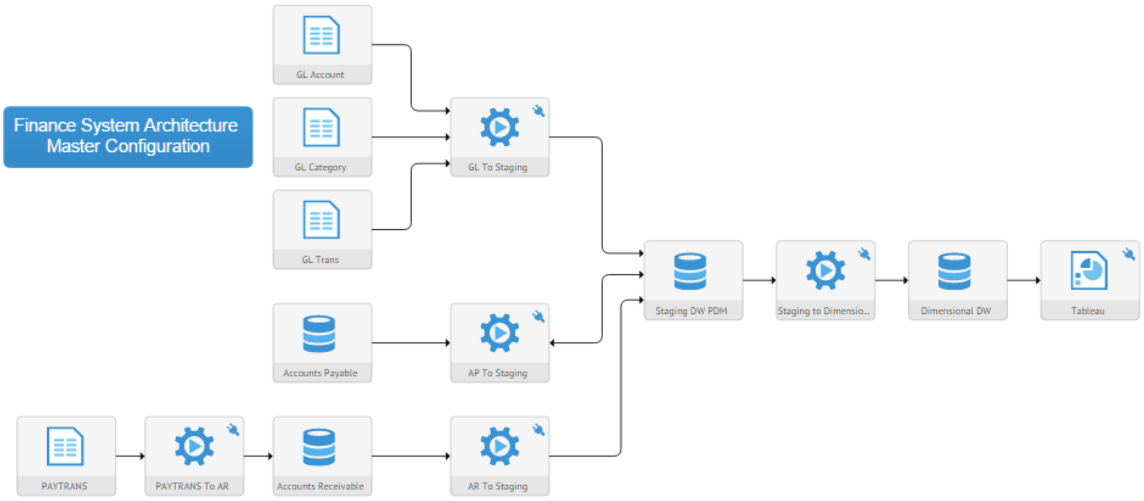


Figure 84 - Annotation font, colors and background

4.5 Documenting a PDM

Recall, we created a physical data model for the Staging database. Now, we have not made use of that model yet. We could, in fact, simply start modeling the database live.

To do so, simply open the PDM and select an element and right-click and select Show in Metadata Explorer. Please do so for the GLAccount table. Now, click on the Name at the top of the page and enter "General Ledger Account", i.e. a more readable business name.

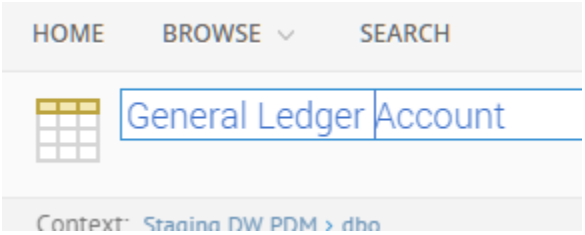


Figure 85 - Name field

Then, click just below this new name and you may enter a Description. Enter "Fund location within the General Ledger".

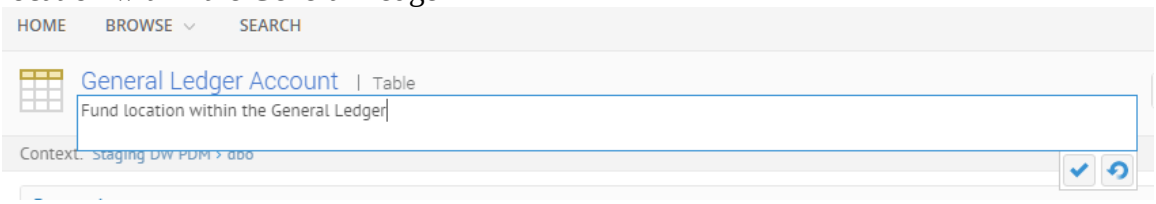


Figure 86 - Description

Now, you may document the rest of the columns like so:

Column Name	(Business) Name	Description
AccountAbbreviation	Account Abbreviation	Abbreviation used to refer to a fund account
AccountAmountAvailable	Account Amount Available	Dollar amount remaining in fund account, calculated as: Account Balance Amount - Account Amount Expended - Account Encumbered Amount
AccountAmountExpended	Account Amount Expended	Dollar amount of this fund account that has been expended
AccountBudgetAmount	Account Budget Amount	Dollar amount budgeted for this fund account
AccountEncumberedAmount	Account Encumbered Amount	Dollar amount of this fund account that has been obligated for expenditure
AccountName	General Ledger Account Name	Name of a fund account

AccountStatus	Account Status	Current Status of this account, including: Closed, Funds Available, Over Encumbered
GLAccountNumber	General Ledger Account Number	GL Account number identifying which fund or account dollar amounts are assigned
VendorInvoice	Vendor Invoice	Document sent from a Vendor in response to a Purchase Order as a request for payment
AddressID	Address Identifier	Unique Identifier for address records
InvoiceAmount	Invoice Amount	Total amount of all line items on the vendor invoice

If you use the grid mode, you may simply double-click on the name grid cell, edit it, then use the Tab key to move to the description, edit that, and then the next line, and so on.

You have now documented the model.

5 Metadata Search, Browsing and Analysis

This section will detail how one uses the Meta Integration® Metadata Management (MIMM) user interface for model analysis. In the last sections we built the repository structure, defined harvesting parameters and imported the metadata from the sources cataloged. We have also connected this information end-to-end using the configuration management capabilities. It is now time to apply the metadata analysis capabilities of Meta Integration® Metadata Management (MIMM) to the information in these configurations.

The configuration is the *scope* of analysis within Meta Integration® Metadata Management (MIMM).

Open the Published configuration in the Metadata Management [folder] and go to the Architecture Diagram tab.

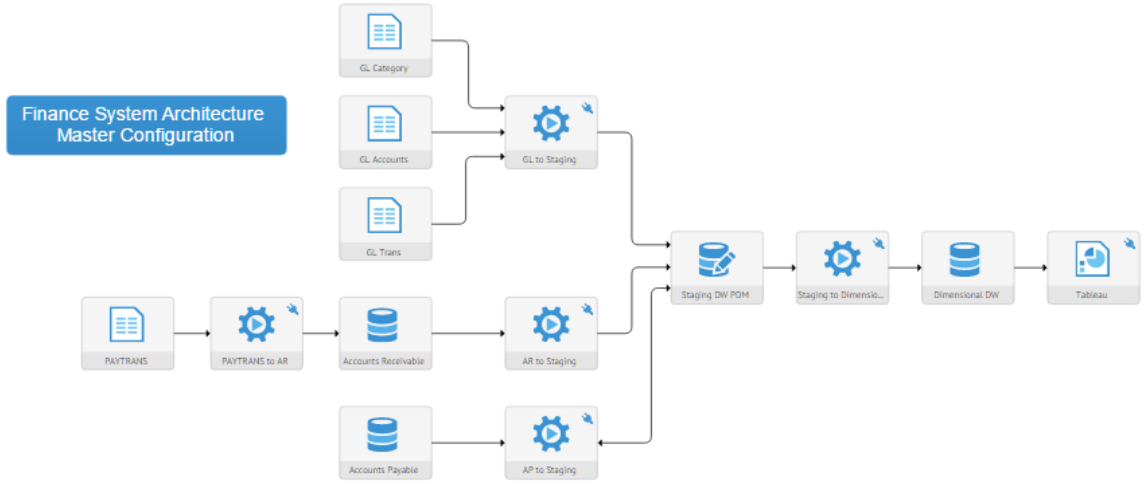


Figure 87 - Published configuration in the Architecture Diagram tab

Note the different types of lineage arrows. The solid black lines are data lineage and the dashed-blue are semantic lineage.

Notice how the top level tabs relate to specific tool which is opened (in the case, the Configuration Manager) while the lower set of tabs relate to the specific user interface features that apply to that tool (e.g., the Model Manager and Archtecture Diagram). Changing the tab at the top, by clicking on it, will change the choices for the UI (lower set of tabs) to match the tool.

5.1 Metadata Explorer UI

The user interface that we have seen to this point in the tutorial has been the *Metadata Manager UI*. This fact is not surprising, as we have been administering (harvesting, managing configurations, etc.) the metadata. There is another, complementary user interface available to users of Meta Integration® Metadata Management (MIMM), the *Metadata Explorer UI*.

5.1.1 Groups, Roles and the Explorer UI

In order to use the Metadata Explorer UI, it is generally necessary to define a *Group* to Use Metadata Explorer. When signing into the system, either via single sign-on, LDAP or using the authentication provided natively with Meta Integration® Metadata Management (MIMM), one is assigned to a group, with an associated UI specification. All permissions (*security roles*) within Meta Integration® Metadata Management (MIMM) may be defined by this group.

This group may also be designated as type "Metadata Explorer UI user". If so, whenever a user who is associated with that group signs in to Meta Integration® Metadata Management (MIMM), the Explorer UI will be presented, instead of the administration UI we have seen so far in this tutorial.

To view roles, go to the Tools › Administration › Groups tab.

This tab shows the groups defined within the system, along with connection statistics. Note that a group may be defined to use the Explorer UI by checking the Use Metadata Explorer checkbox.

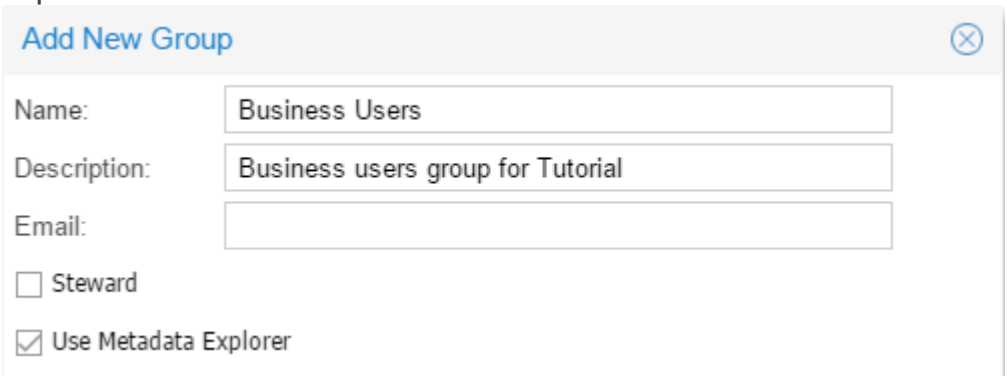


Figure 88 - Add new group

Then click on the Add button.

To create the matching user that we will sign in as, go to the Tools › Administration › Users tab. This tab shows the user id for all of the users who have signed in to Meta Integration® Metadata Management (MIMM), along with connection statistics. Now, we will add a user named "Bob" and assign them to the Business Analysts group.

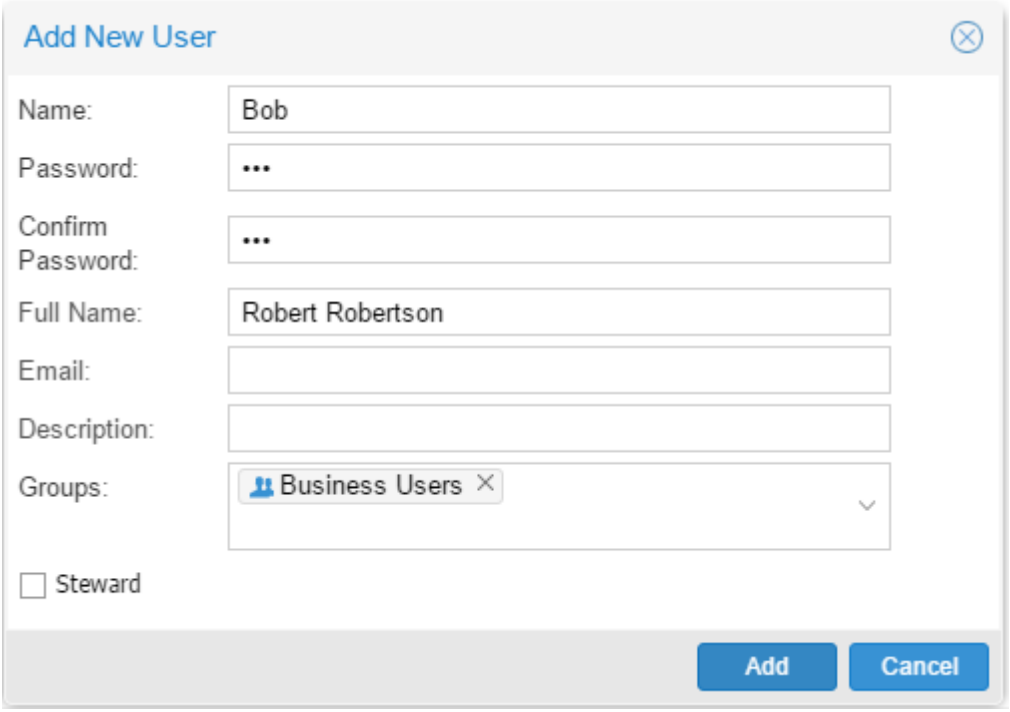


Figure 89 - Add new user

5.2 Business Analyst BI Report Lineage

Returning to the Published configuration, this use case is presented from the point of view of a Business Intelligence Analyst (see the section entitled *Common Repository Roles and Use Cases*). In this case, the question is "Given a BI Report, where does the data come from in terms of data entry and 'external' data feeds?"

In this case the query is based upon identifying the report metadata (field names) for which one wishes to see the ultimate data entry source fields.

5.2.1 Navigation

In this case, it would be convenient if one could simply open the business intelligence tool (in this case SAP Business Objects), navigate to the report, and right-click on the report item (say the Account Amount Available in the Net Vendor Customer Invoices report). This is possible with some BI tools, as they have a flexible "call-back" capability to connect to the MIR API.

5.2.1.1 Sign in

In addition, Meta Integration® Metadata Management (MIMM) has the ability to reproduce this method of locating and analyzing report metadata. To do so, it is not necessary to reproduce the business intelligence system UI. Instead, provides *profiles* with the same "touchstones" such as *report name* and *report item name* that a user of the specific BI tool would see in the BI tool user interface.

Now, start an entirely new browser session (first close your current browser windows) and sign on as Bob. If you see the following:

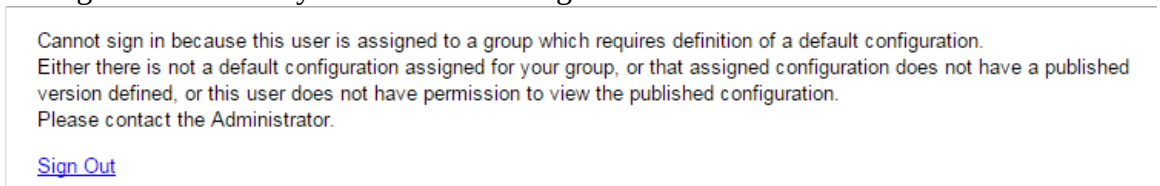
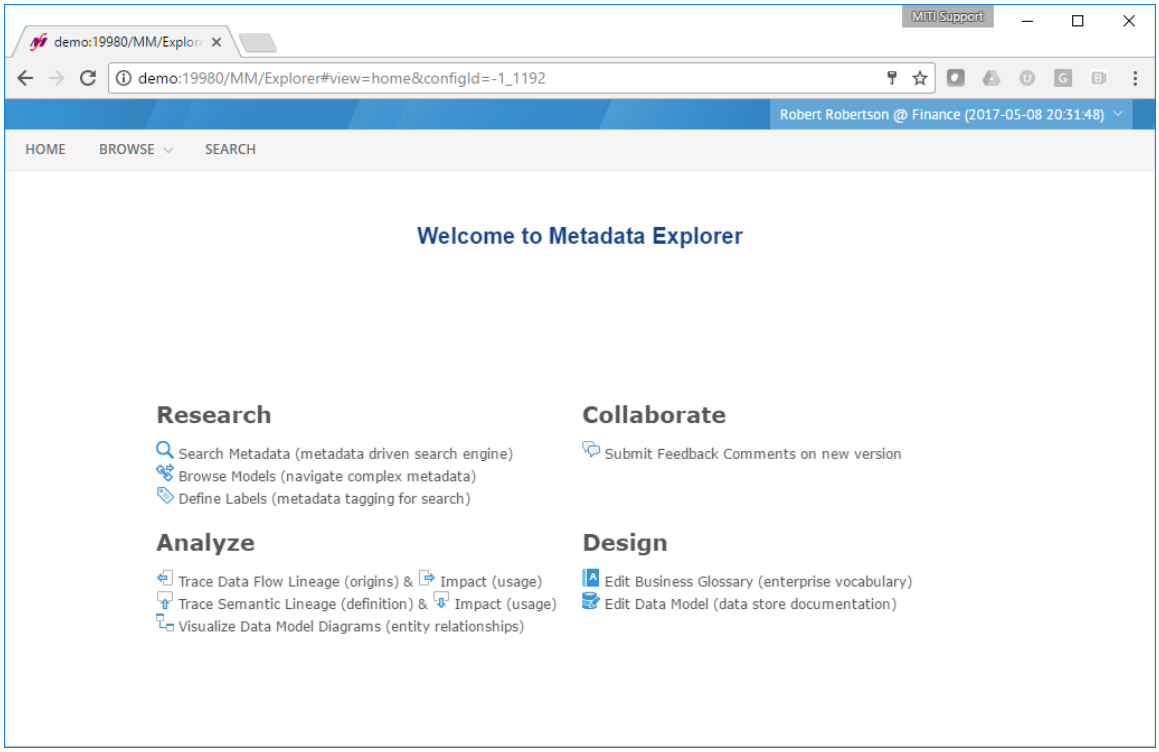


Figure 90 - Sign in error

It turns out that in order to control which version of a configuration is presented to the business users, one must first publish the configuration (and a specific version of it). It was suggested that you do so earlier. However, now do so by returning to the Metadata Manager UI (sign in as Administrator) and then right click on the configuration and select Publish the latest version.

Now sign in again as Bob. Because we are now in the context of a single configuration (Published), we have a simplified view of the Repository:



5.2.1.2 Report item definition

First, we must navigate to the business intelligence system metadata. Go to Search, select Change, select Business Objects for Search In, enter AccountAmountAvailable (all one word as it appears in the report) in the search text, and click the Search button:

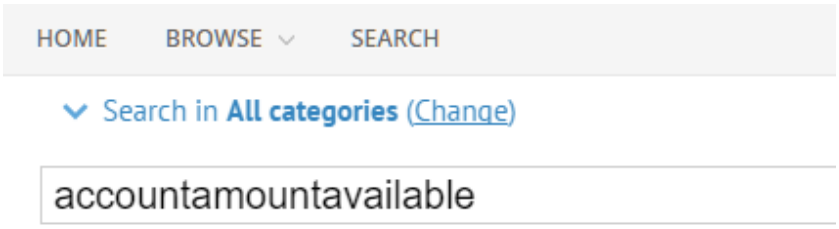


Figure 91 - Explorer UI Navigation

This has taken us directly to the element in the Net Vendor Customer Invoices report. Now, click *on the line* but not the element of the Cell we are looking for and note that action icons appear:

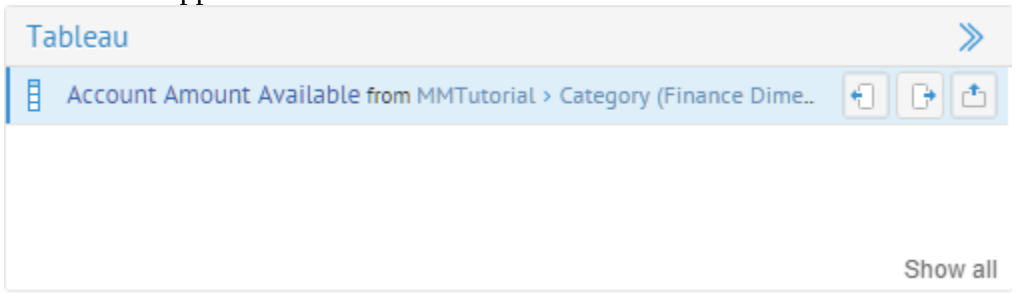


Figure 92 - Action icons appear on search result line

And we have the answer:

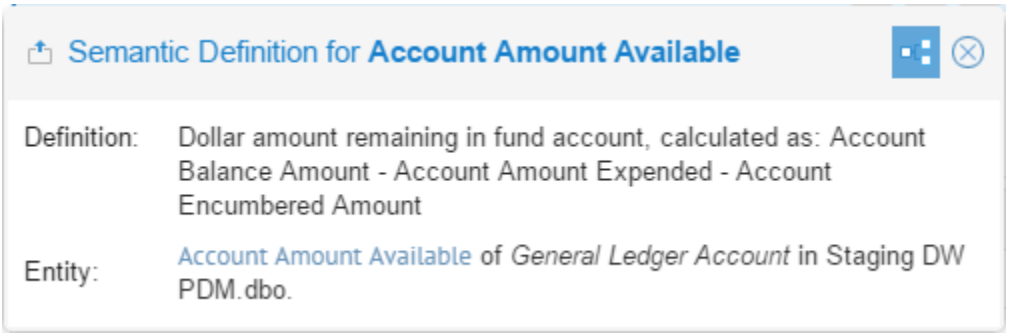


Figure 93 - Semantic definition for Account Amount Available

Note, the definition is coming from the first, best source found when tracing back through the dataflow. In this case, it is in the Staging DW PDM, and in the dbo schema. Remember, we provided that definition when documenting the model. In addition, it is only one of the related elements. You may see this by clicking on the Details ( ) icon:

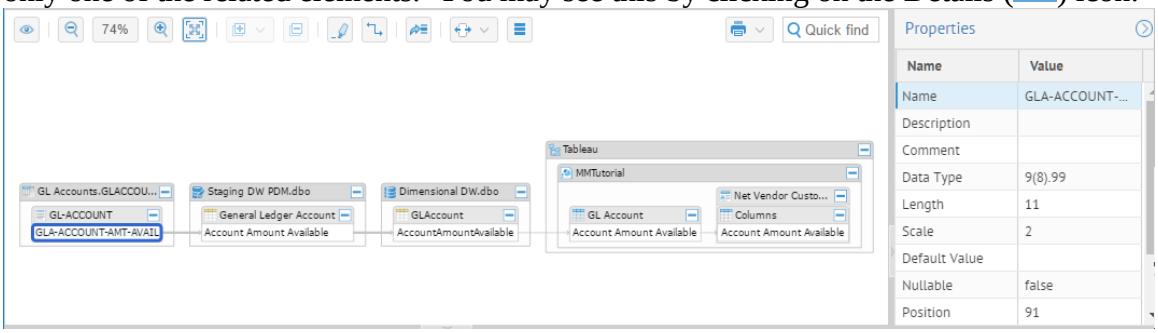


Figure 94 - Trace semantic definition

Now, that was obtained very quickly by using the Explorer UI. Doing the same in the Manager UI takes a bit more work, but may provide a bit more understanding about how the product obtains this information.

5.2.1.3 Metadata Manager UI

So, let's **open a new browser session** and sign back in as Administrator and then open the Tableau model. This means that the entire Tableau environment is then presented:

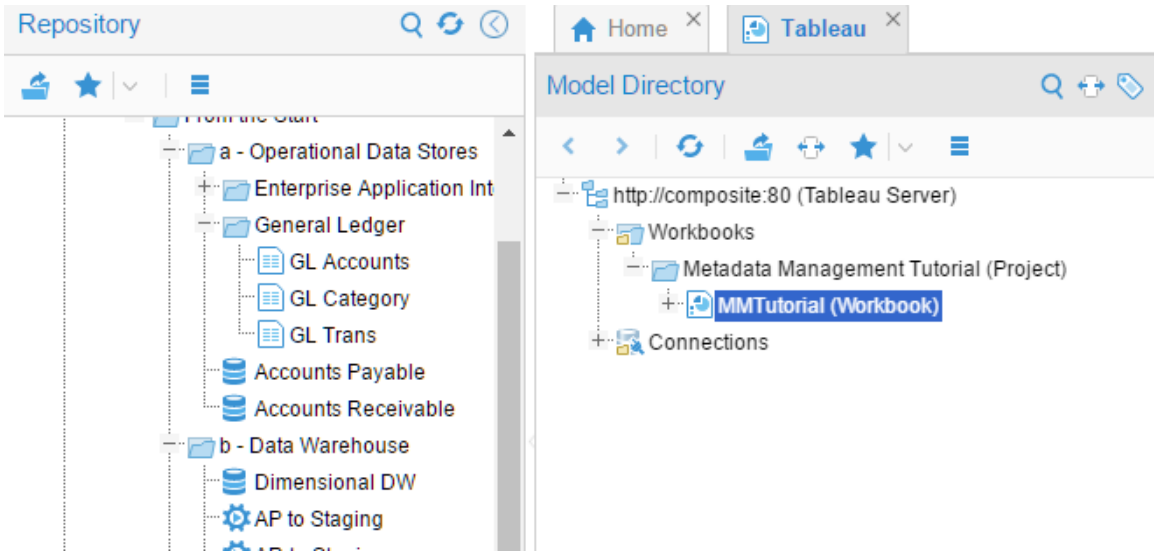


Figure 95 - Tableau metadata captured in Meta Integration® Metadata Management (MIMM)

Note, this action presents another tree structure. This time, instead of a tree representing the contents of a configuration, you have a tree (the Model Directory) representing has been harvested from the Tableau environment. Hide the Configuration Panel using the icon and expand the Properties Panel using the same icon on the right hand side. Note, the Properties Panel will show addition properties describing any tree element in the Model Directory one clicks on.

Now, expand the tree until you arrive at the MM Tutorial (Workbook) element and then double-click to open it.

Then navigate down the tree until you arrive at the following element in the tree of the Metadata Browser:

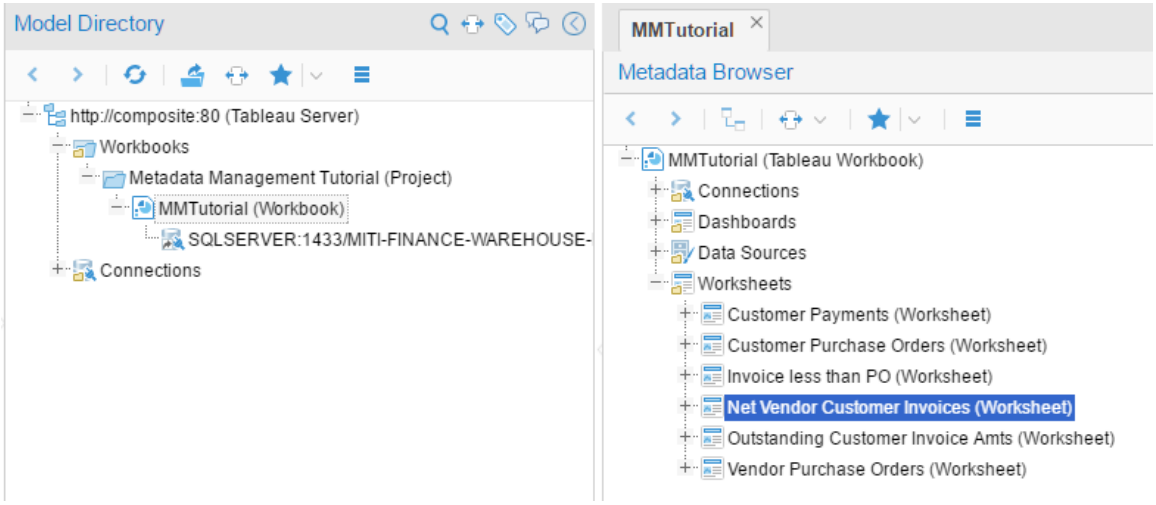


Figure 96 - Net Vendor Customer Invoices report

Hide the Model Directory panel, so we have more room to visualize. Now, you are presented with a tree of the contents of the report as harvested from Tableau:

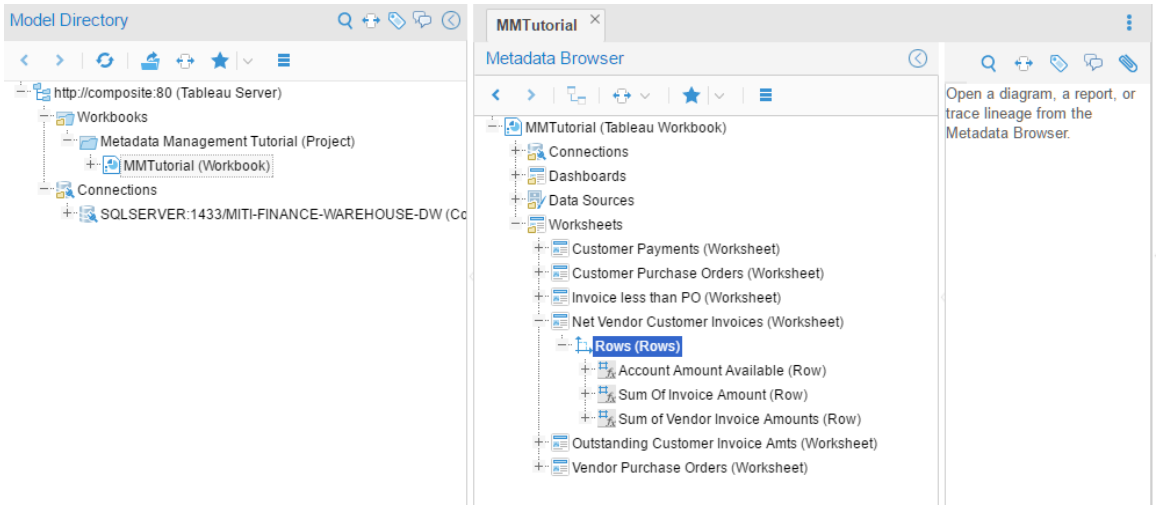


Figure 97 - Net Vendor InvoicessAmounts report

A bit of explanation is necessary. While a user generally sees a report as result data located in various locations and presentations on a report, underneath all of that are specifications of what data may be presented. In Tableau, these are represented as *rows*, *columns* and *marks*.

5.2.1.4 Lineage

Let's trace for the rows:

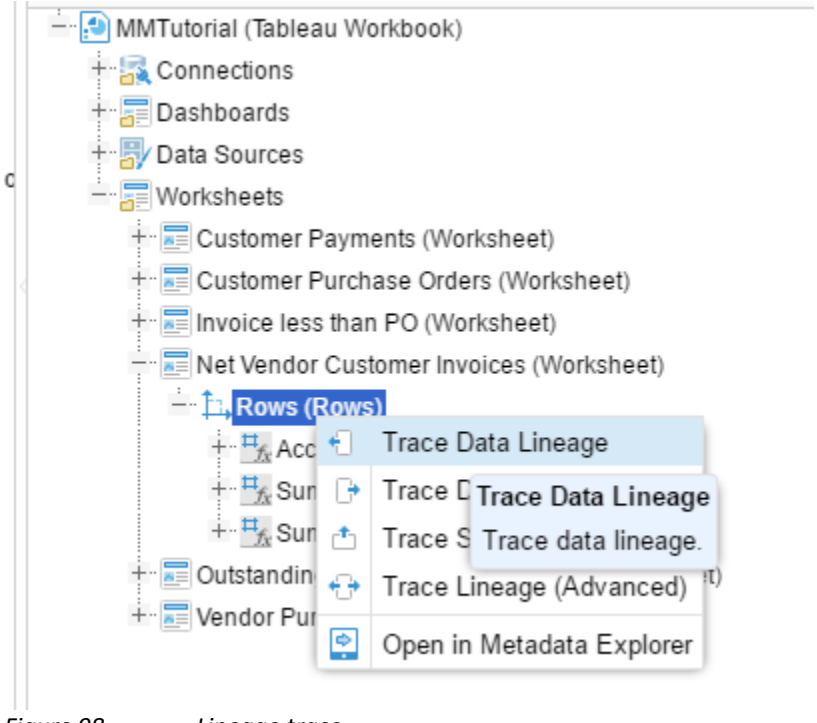


Figure 98 - Lineage trace

If you trace for the entire Published configuration:

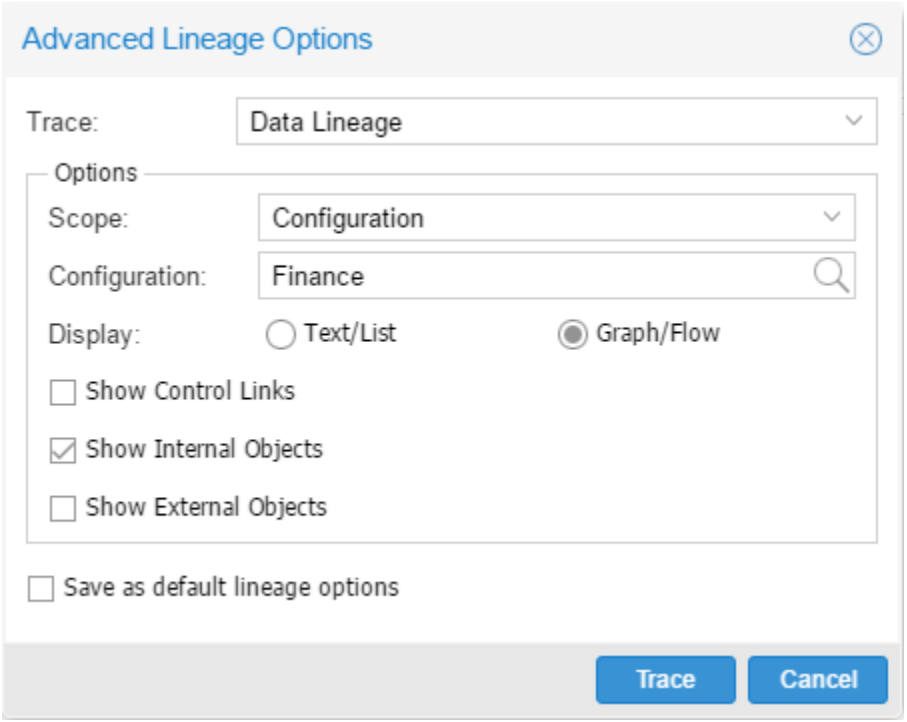


Figure 99 - Configuration wide advance lineage trace options

You will have (if you expand to the Column level):

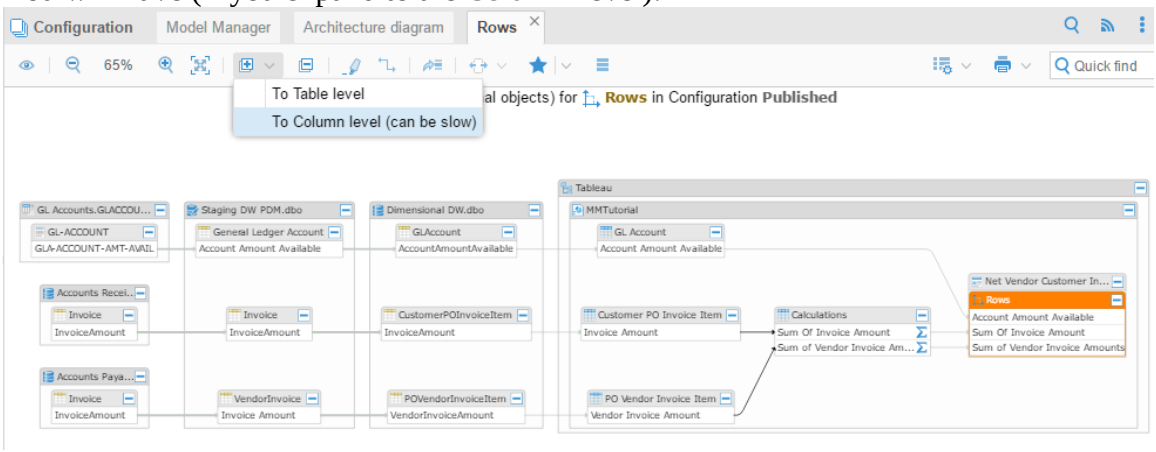


Figure 100 - Configuration wide lineage trace expanded to column level

Now, let's return to the browser window with the Metadata Explorer UI and navigate to the same location in the report, right-click on the Rows heading and select Show in Metadata Browser.

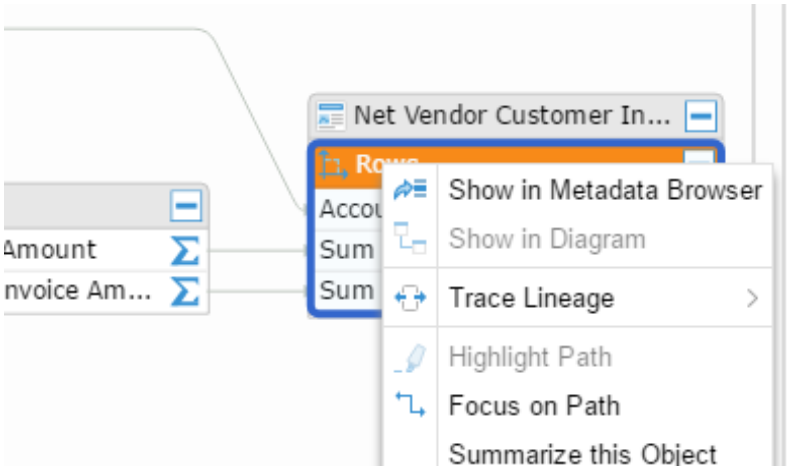


Figure 101 - Show in Metadata Browser

5.2.1.5 Definition lookup details

Then right-click on one of the item, such as Account Amount Available and select Show in Metadata Explorer.

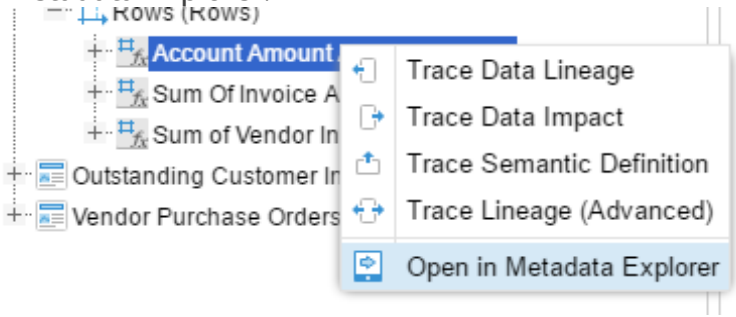


Figure 102 - Show in Metadata Explorer

Then click on the Net Vendor Customer Invoices in Context.

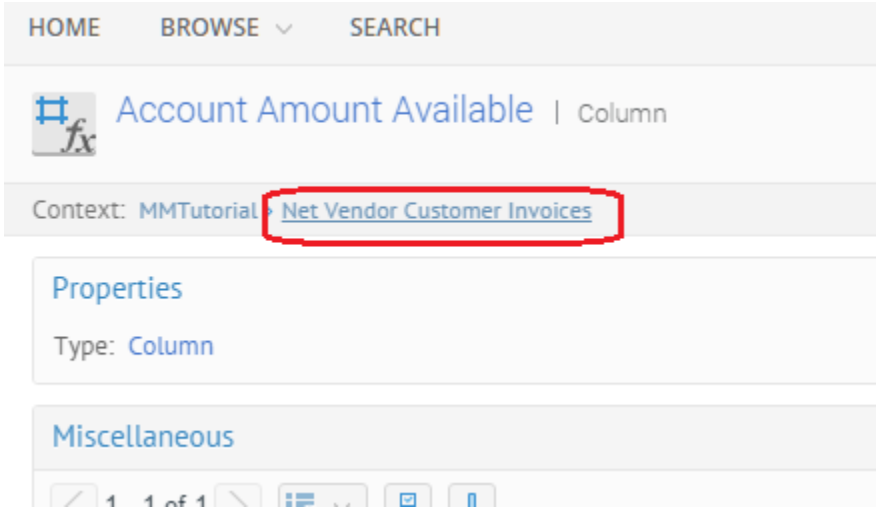


Figure 103 - Click on context

And the Rows are shown in the list for that report.

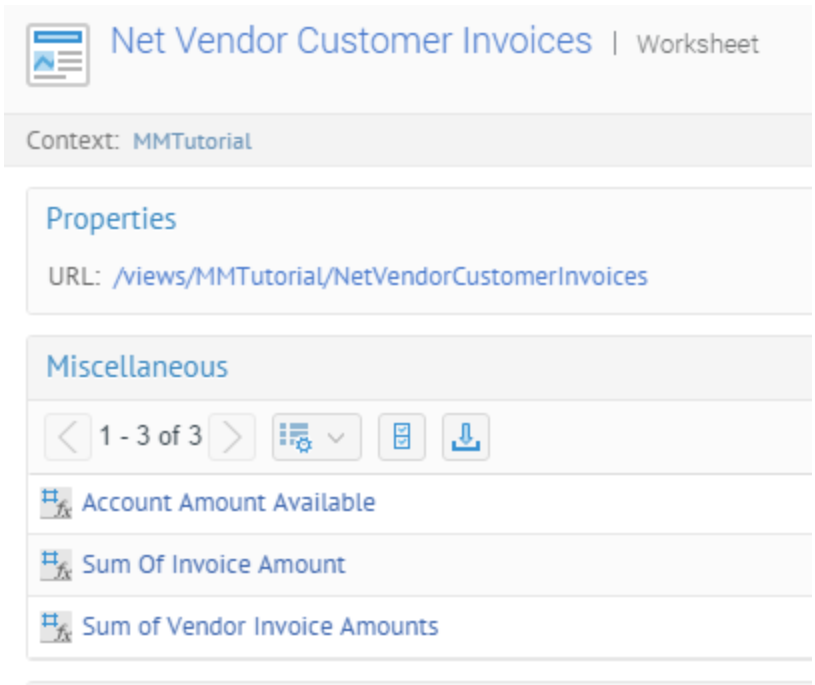
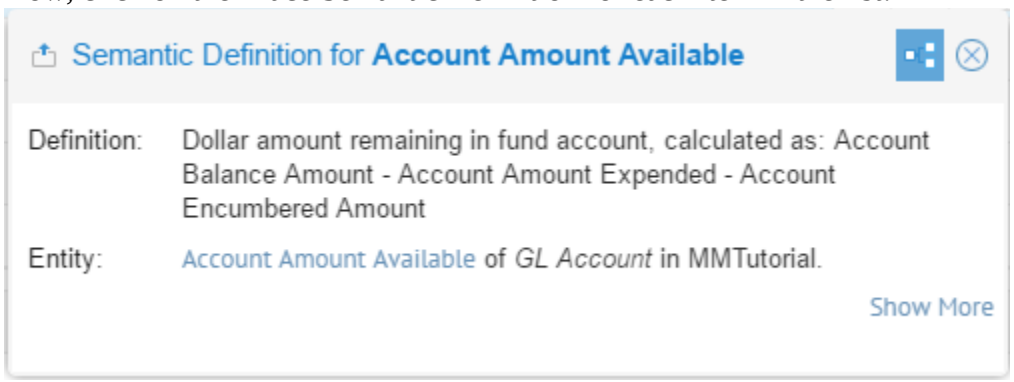
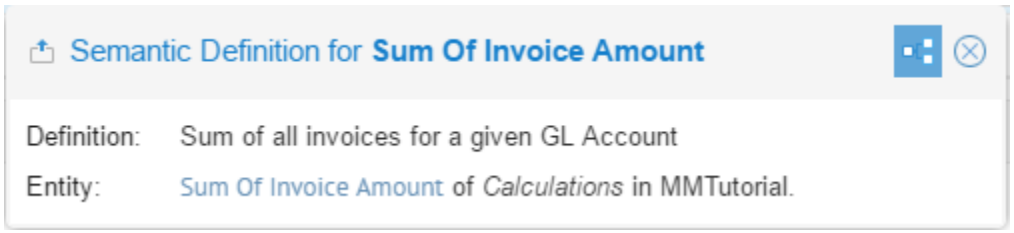


Figure 104 - Columns and Marks

And we have

Now, click on the Trace Semantic Definition for each item in the list:





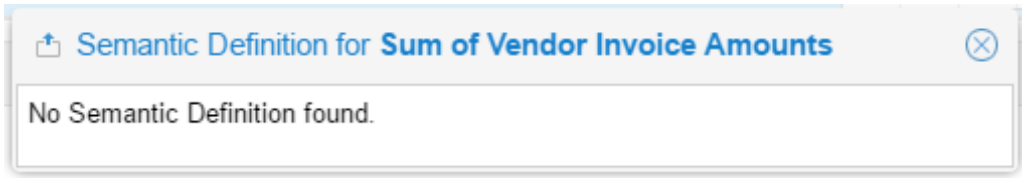


Figure 105 - Definitions

Note, Sum of Invoice Amounts does have a definition (it is defined in the Tableau itself and that definition was harvested). However, Sum of Vendor Invoice Amounts does not have a definition. You can switch back to the Metadata Manager UI and confirm that the MMTutorial workbook does not have a definition.

How about the definition in Staging DW PDM? Tracing back all the way through the data flow lineage, one can check and see that there is indeed a definition in the VendorInvoice.InvoiceAmount column in the Staging DW PDM.

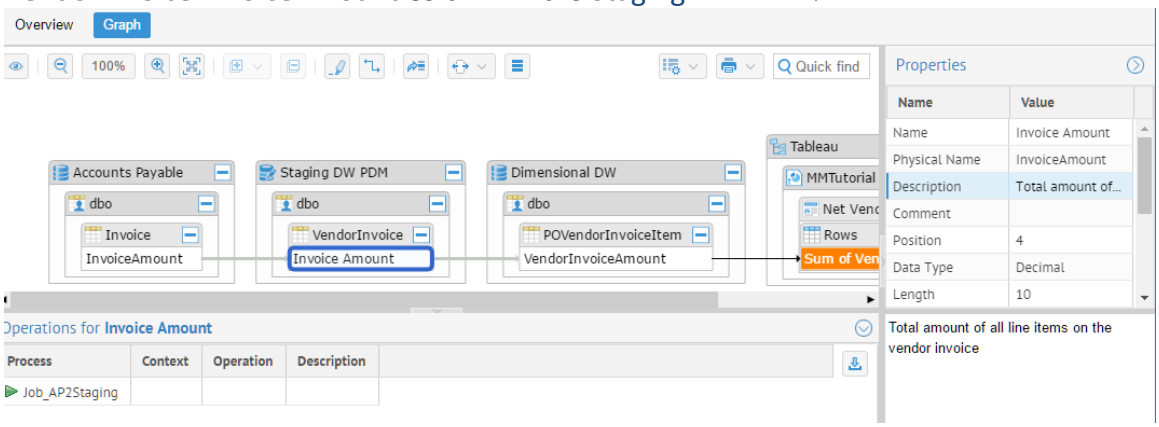


Figure 106 - Trace

So, why is that definition not shown when requesting Trace Semantic Definition?

Remember, for the semantic definition trace, the data flows are only followed if there are NO transformations. However, as one can see in the lineage diagram, there is a transformation (aggregation or summarization) going from the Dimensional DW to the workbook. Thus, any definitions earlier in the data flow are not considered correct and Meta Integration® Metadata Management (MIMM) does not report them as a part of the Trace Semantic Definition.

Now, we cannot simply edit these definitions as they are faithful representations of the source metadata, which apparently was not documented in the original Tableau

5.2.2 Search

5.2.2.1 Context

We may, of course look for this report field in a different manner, i.e., by searching, rather than navigating. It is important to talk about the *context* or scope in which one is operating, though, as the search (and indeed the lineage tracing, analysis, etc.) all depend upon the defined context.

In the previous section, we only looked at the lineage from the perspective of a specific report. Of course, there is a much larger context of:

- The entire business intelligence reporting environment (as harvested) including all reports and OLAP model(s)
- The entire Published configuration, including all the models stitched together in the previous chapter.

In this section we will both search and analyze (especially data flow lineage tracing) the metadata in the context of the entire configuration.

5.2.2.2 Search Context

When searching, then, one must first define the context. We already have the NetInvoicesAmounts report open, and thus when using the Search icon:

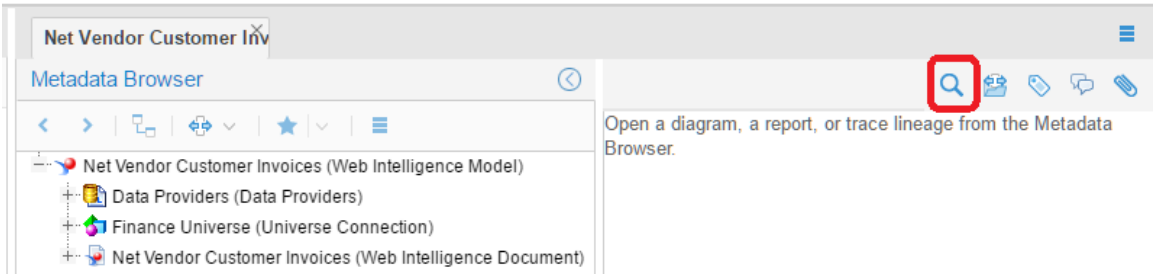


Figure 107 - Search icon

In the header for the central panel in the UI, the context of that search will be the report only. E.g., if the text string entered is "invoice":

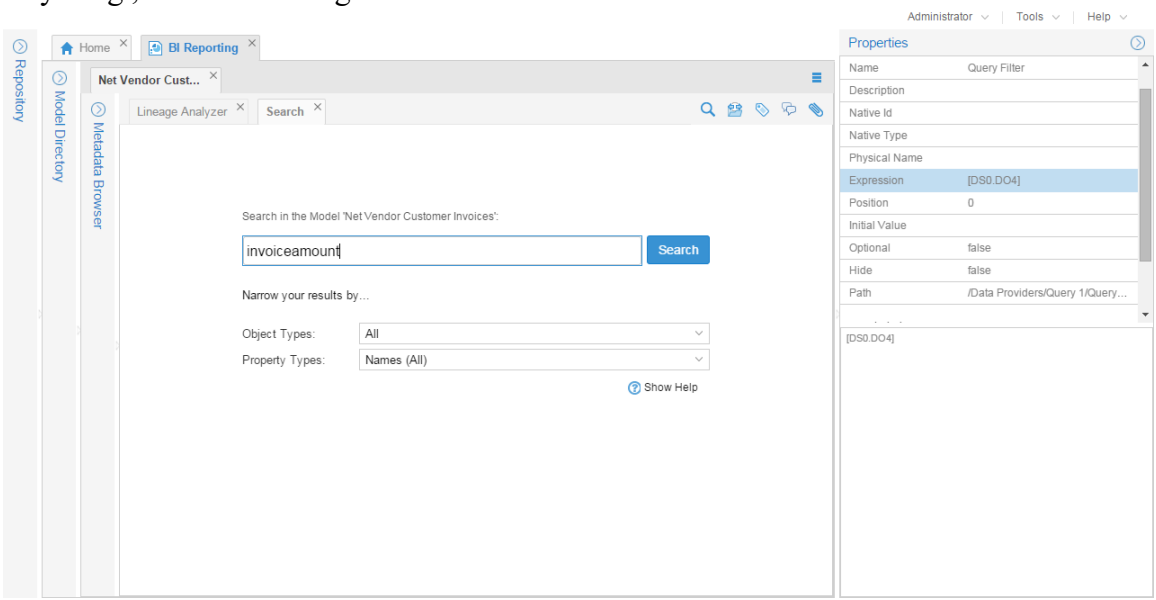


Figure 108 - Report context search criteria

then the results are only within the report:

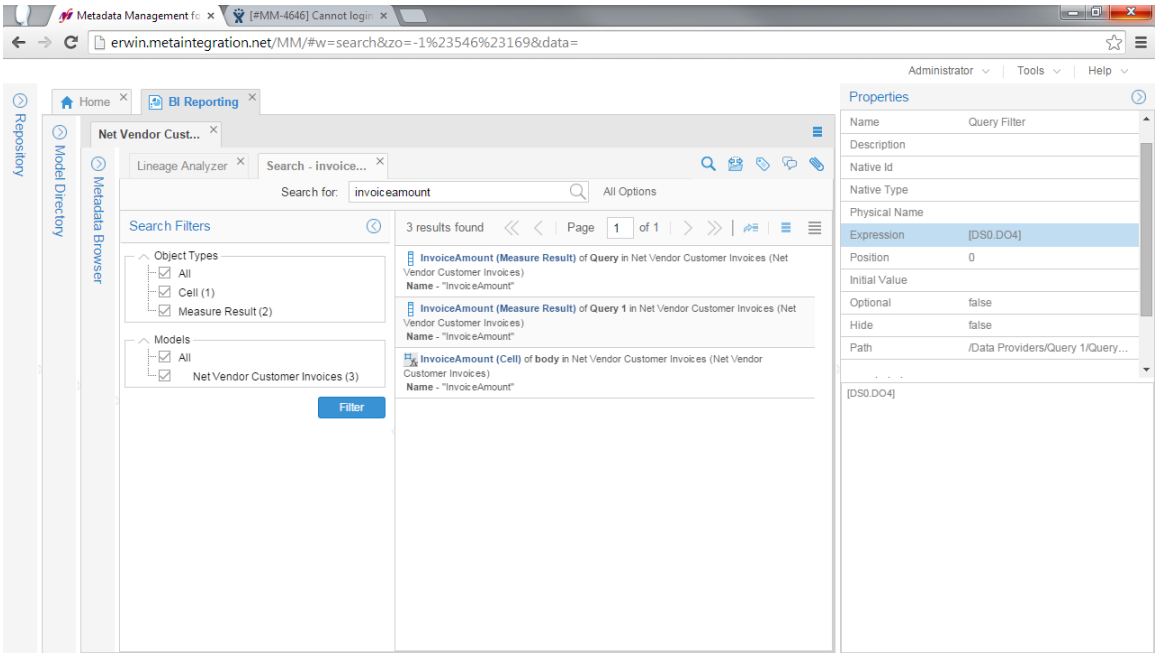


Figure 109 - Report context search result

Now, if one instead expands the Model Directory (entire BI environment) and then uses the Search icon there:

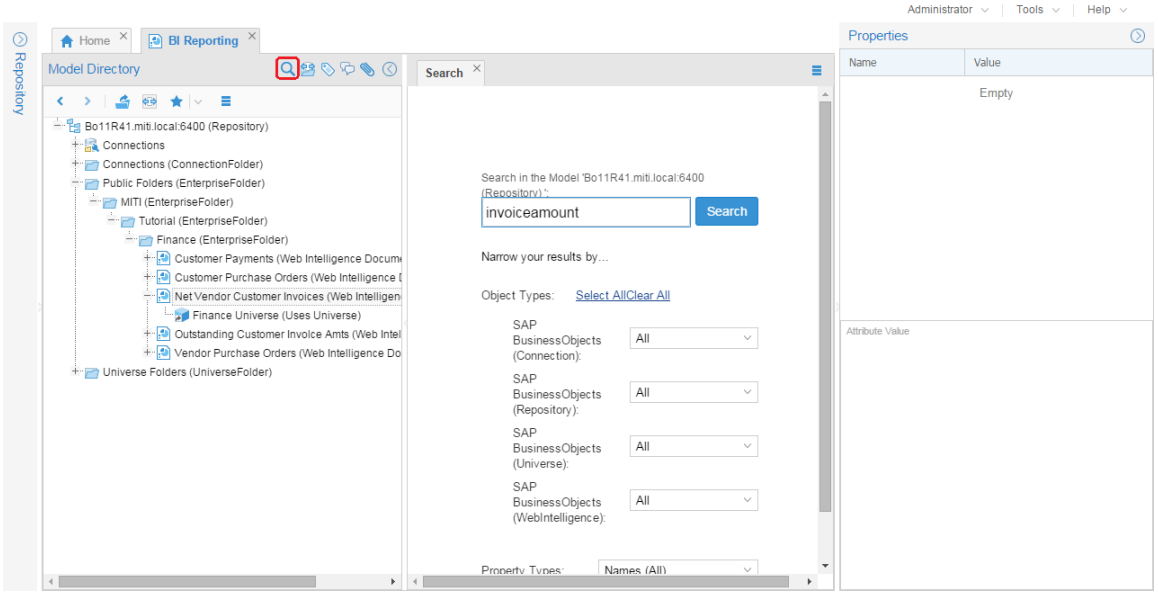


Figure 110 - Model directory level search

the results are much broader:

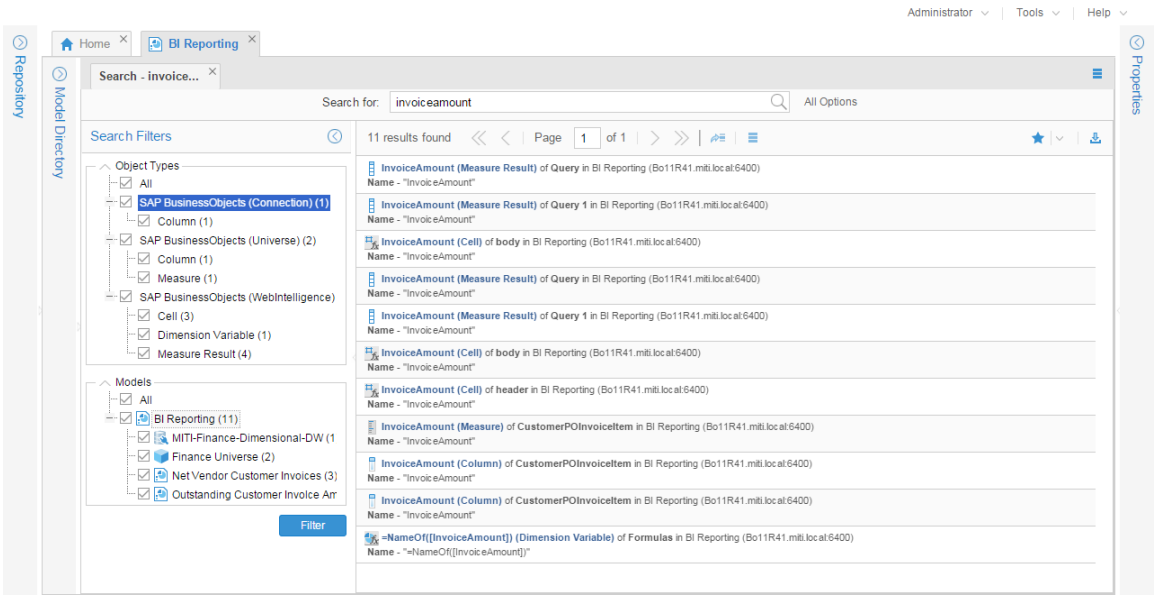


Figure 111 - Model Directory context search result

Finally, if one instead expands the Configuration panel and uses the search icon:

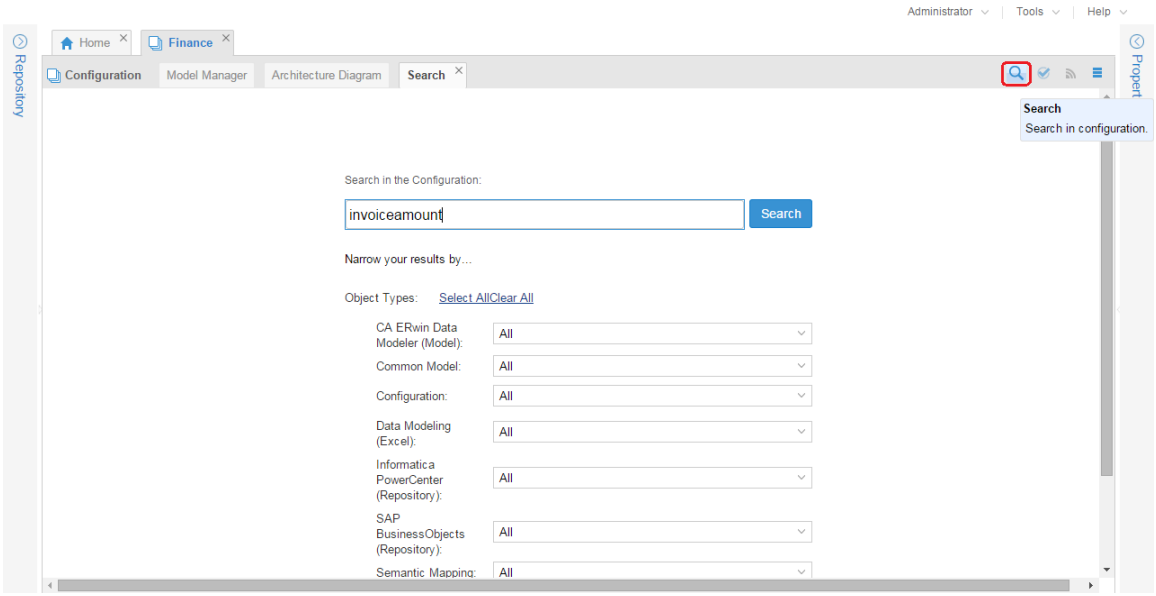


Figure 112 - Configuration level search

the results include the entire configuration:

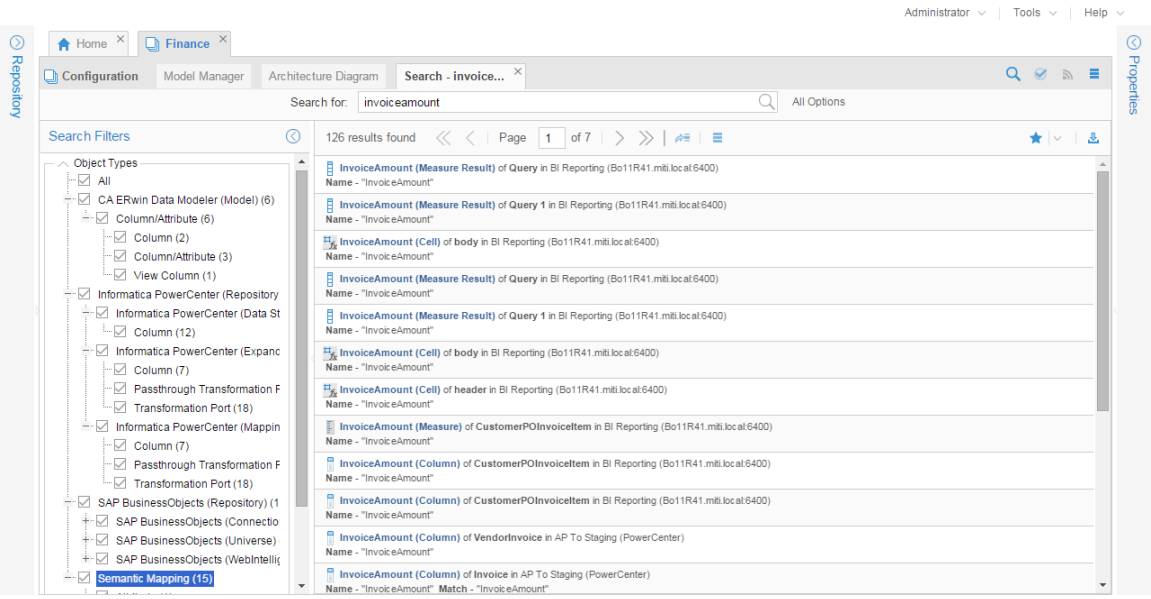


Figure 113 - Configuration context search result

5.2.2.3 Search Filters

Although it appears at first glance that searching in the configuration context will produce far too many results to be useful. Apparently, there are a large number of elements with the name invoiceamount in the Published configuration. This is not a surprising fact. However, as with searching on the web, the amount of information which may be presented generally must be managed, and reduced to what is most pertinent. One could, of course, page through the results using the control panel at the top of the search results. However, there are other methods.

In fact, this is a *metadata based search*, and thus one may use the Search Filters tab on the left of the results to reduce the complexity of the result. One may filter on:

- Matches against the Object Types, e.g. entities (logical) and/or tables (physical) and/or attributes, and/or domains, etc.
- Matches against the Models, e.g., the NetInvoicesAmounts report, the Accounts Payable model, etc.
- Matches by Property Type, e.g., Name, Physical Name, Description, etc.

In this case, we only want elements that are in reports. Specifically, report fields. Thus, one may select only the SAP BusinessObjects (WebIntelligence) checkbox type in the Object Types filter and click on the Filter button:

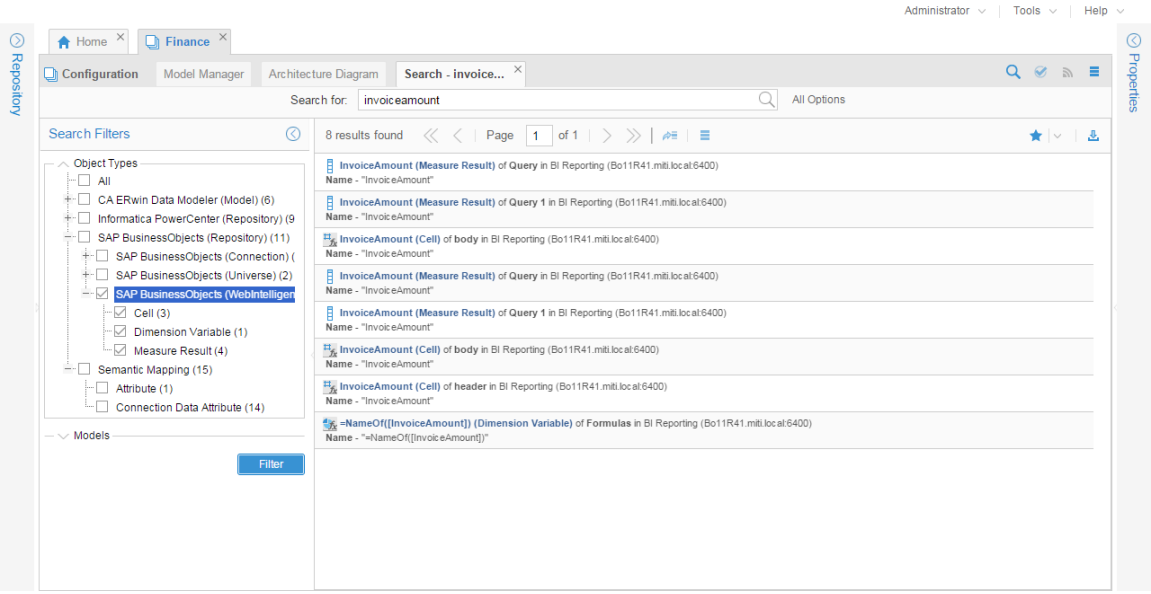


Figure 114 - Search filtered to ReportAttribute

A second way would be to filter the results by the model they are contained within. Return to the original search result by clicking on the search ocon again. Then, expand the Models filter on the left and select the Net Vendor Customer Invoices model (report) and click on the Filter button:

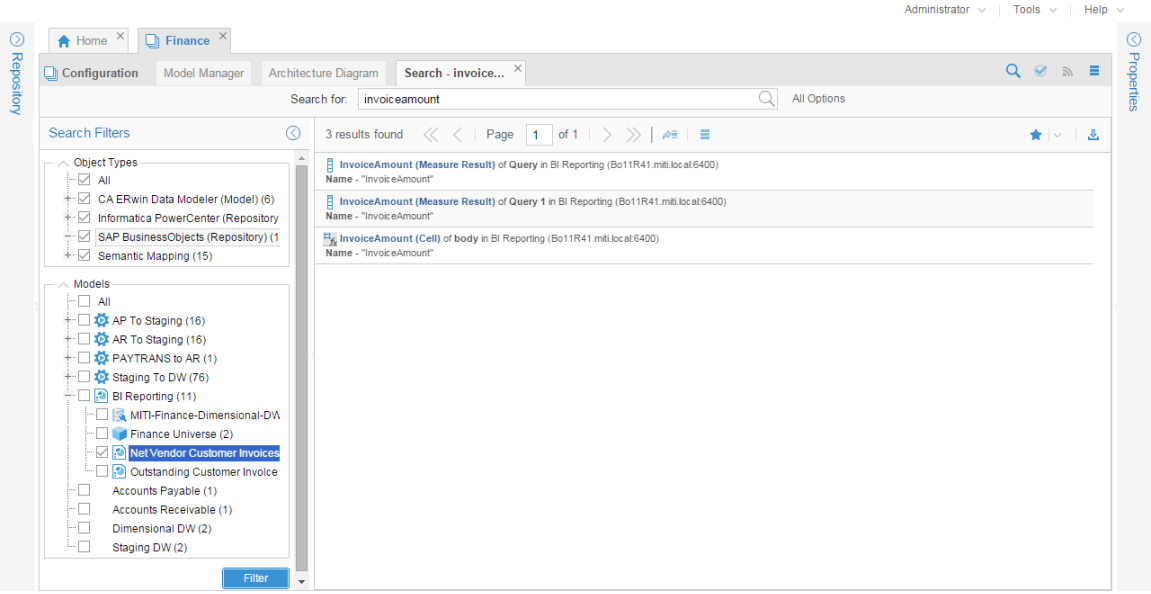


Figure 115 - Search filtered to the NetInvoiceAmounts report

Of course, this would be the same as our original search from the report itself.

One more filter is available, which is by Attribute Type. This filter allows one to filter based upon the type of metadata property (or attribute) that the search matched on for a given element (e.g., *Name*, *Description*, *Data Type Name*, etc.). Even custom attributes are included in this filter. The best choice here would be by Name. However, as there

are a large number of matches on *Name*, it will not be as effective as the first filter by Object Type.

5.2.2.4 Search Language

In addition to the pre and post filtering selections, one may search for very precise results by specifying a search using the built-in query language. This is quite similar to most other web based search engines.

Specific searches may be made for the following type of criteria:

To search for	Example	Result
Any words	invoice amount	Any result containing the word "invoice" or the word "amount"
Exact phrase	"invoice amount"	Any result containing the exact phrase "invoice amount"
All words	+invoice +amount	Any result containing BOTH the words "invoice" and "amount"
Exclude words	invoice –amount	Any result containing the word "invoice" but cannot contain the word "amount"
Wild card end	sale*	Any result containing part of a word beginning with "sale"
Parent and child	invoice.amount	Any result where the parent is named "invoice" and the child is named "amount". E.g., the attribute "amount" contained within the entity "invoice".
Exact name	.amount	Any result only containing the word "amount"
Object Type	type:Column	Any result which is of type "Column"
Property Type	property:name	Any result where the search criteria matches the name

One can, of course, append these together into a search string. E.g.:
Invoice +amount -vendor property:physicalname type:column

5.2.2.5 Lineage Following Search

Now, search for AccountAmountAvailable (another field in our report of interest) in the Published configuration and filter the results to only the Net Vendor Customer Invoices report. Right-click on the *(cell)* result in the Report body and select Trace data lineage and you are presented with the Advanced Lineage Options dialog:

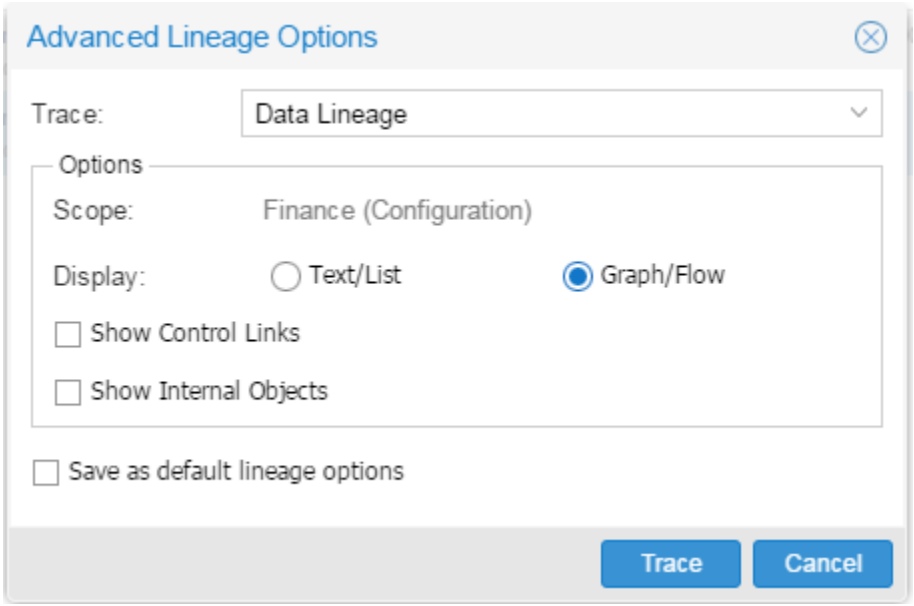


Figure 116 - Advanced Lineage Options

The lineage trace is a fully detailed trace of semantic and/or data flow lineage for detailed analysis. Unlike the Lineage Analyzer icon for a model, this presentation allows one to narrow in on specific elements (e.g., a query element) and how they relate in terms of data flow and/or semantic lineage within a model, a model directory, or a configuration.

In this case based upon our context menu selection, we chose Data Lineage, the scope is by default the Configuration containing the report and the only configuration is Published:

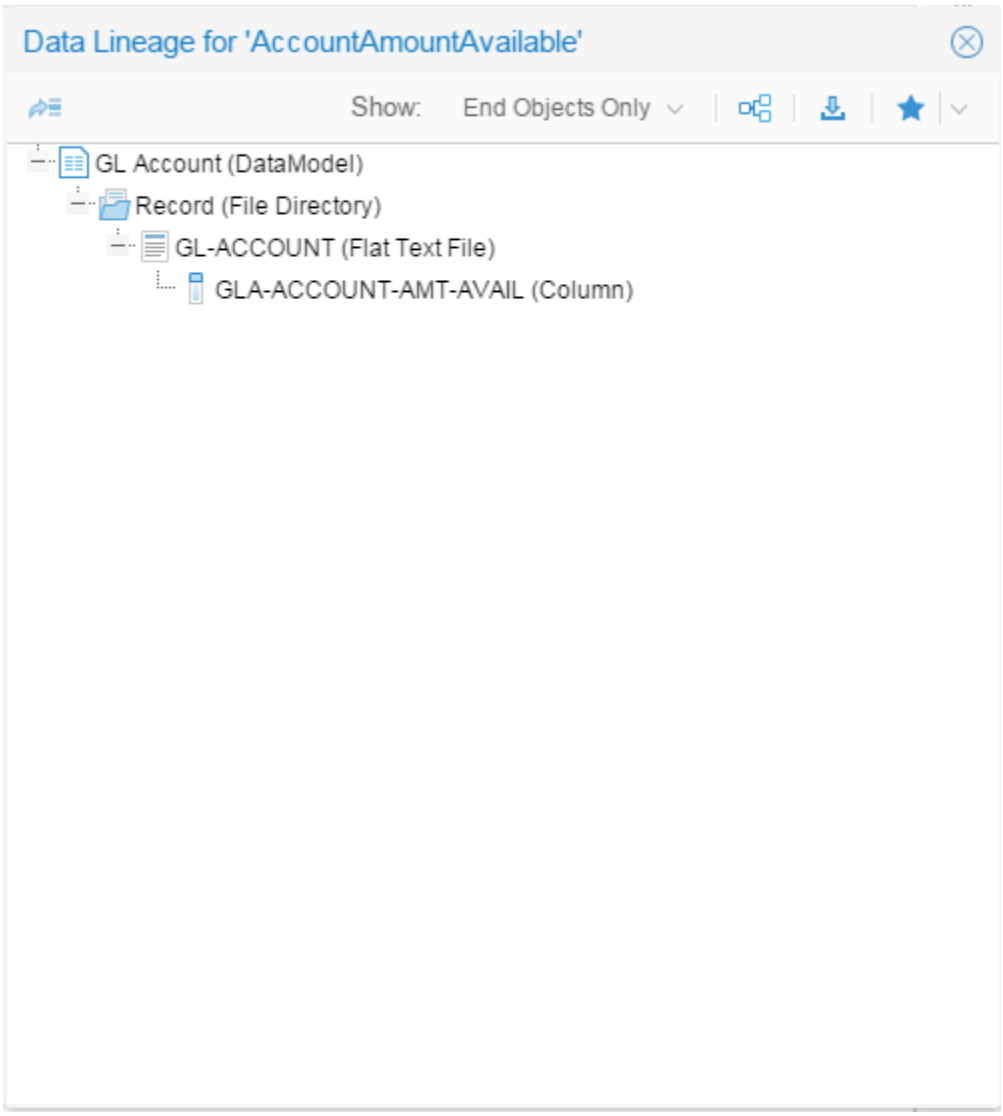


Figure 117 - Textual ultimate source trace for AccountAmountAvailable

Note, the result is a dialog presenting where this information actually originates in the entire configuration. In this case, it is a Flat file where the General Ledger system is still maintained on the mainframe. Thus, what is reported here for this field in the report is actually obtained through update of the old CICS based General Ledger Account. Meta Integration® Metadata Management (MIMM) knows this because we stitched together all those separately harvested models in an earlier chapter.

Of course, this is simply the short answer, as there is a great deal of detail in between, but it is the most pertinent answer if the goal, as it was here, was to determine where to update the information in the system which will update this field on the report. To see the additional detail, one may change the pull down in the header to All Objects and see the following:

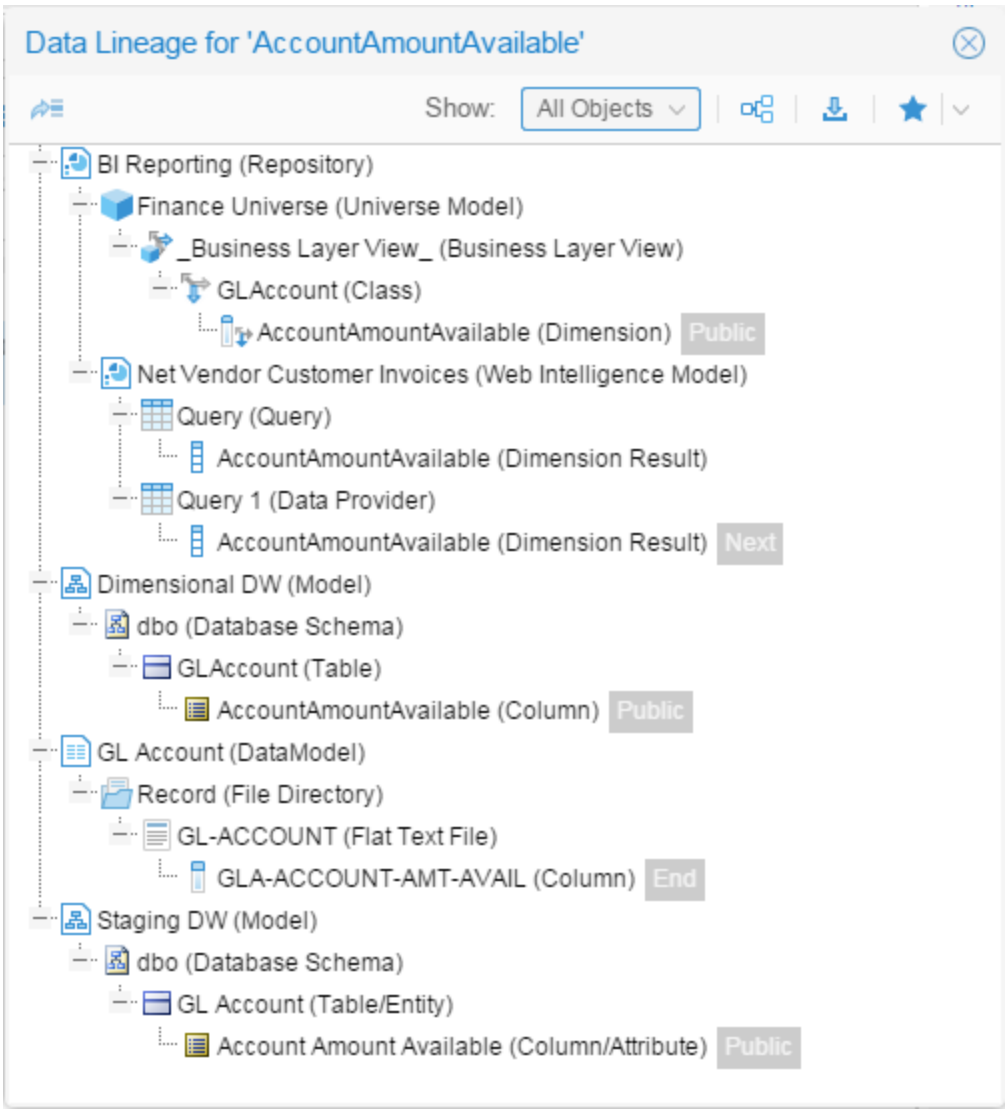


Figure 118 - Textual data flow lineage trace for AccountAmountAvailable

Note, the report now shows ALL of the interim data stores used to pass the data for AccountAmountAvailable from the General Ledger through the Staging Warehouse through the Dimensional Warehouse and through the universe (OLAP Model) in the business intelligence system.

We also can use the Show Lineage Graph icon to view a complete graphical trace of the same lineage:

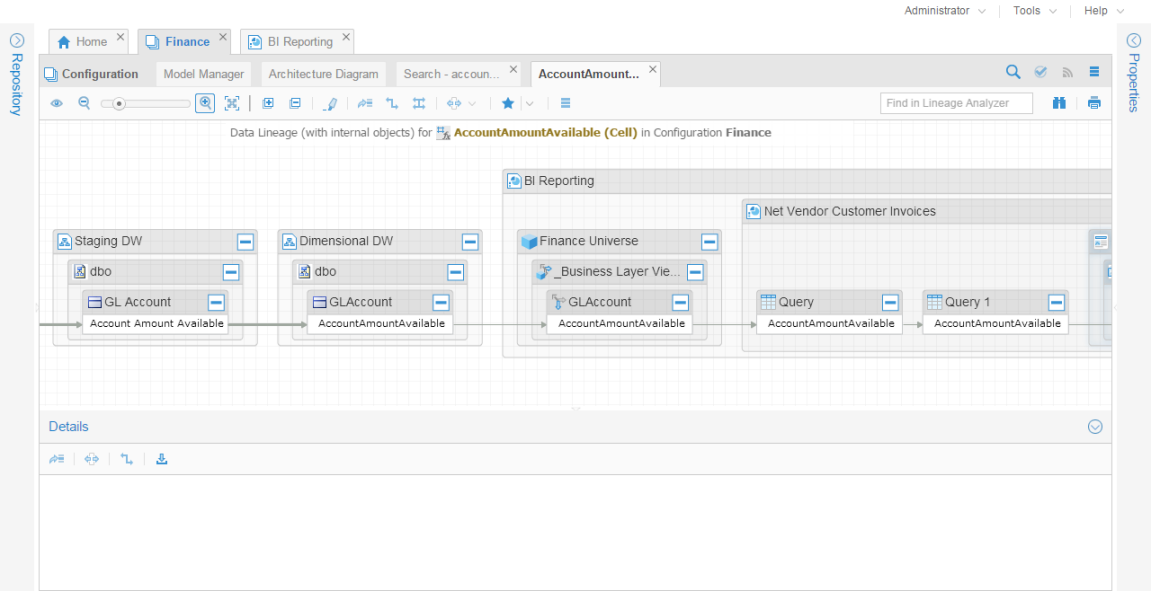


Figure 119 - Graphical data flow lineage trace for AccountAmountAvailable

And click on individual mappings, showing any operations, etc.:

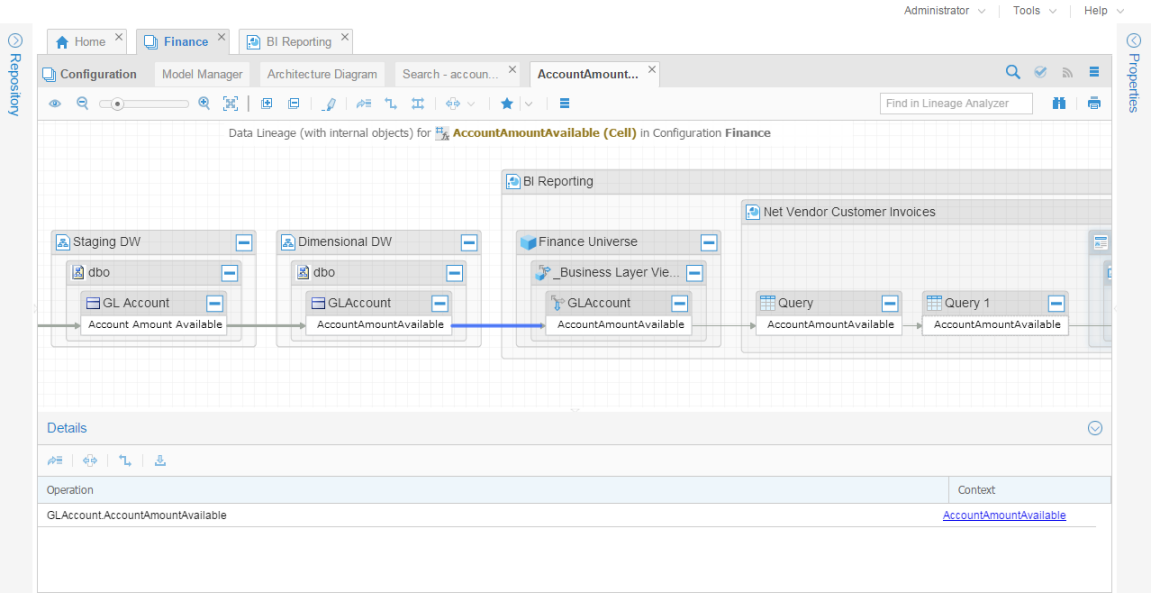


Figure 120 - Operation signature in data flow lineage trace

Again, this picture represents the same data flow lineage trace as the textual report.

Now, let's send this information off to the interested part via an e-mail. To do so rapidly, simply click on the Bookmark icon and select Get Link:

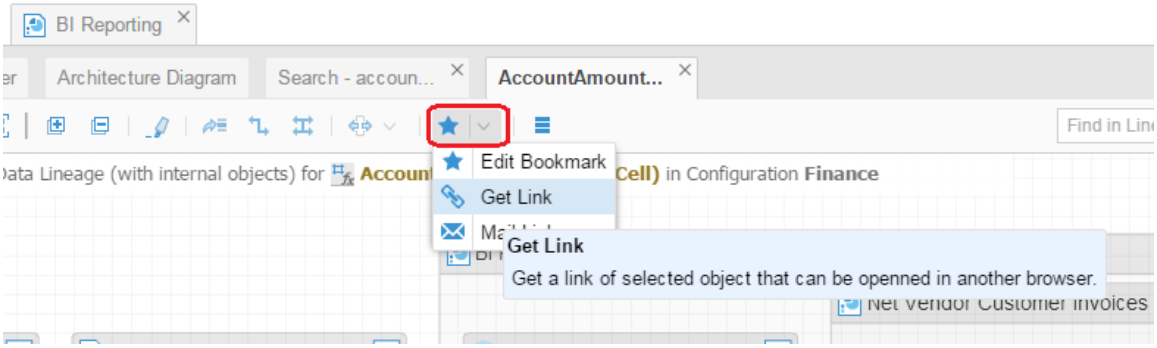


Figure 121 - Bookmarks

Copy and paste the URL into an e-mail, and when the recipient opens the link they will immediately be presented with this exact same lineage trace.

5.2.3 Browse

In addition to finding elements using the search, one can instead browse for information according to the type of content (e.g., Data Model, Database, Business Intelligence, etc.) and then the object metadata types contained within (e.g., tables, coloumns, etc.)

5.2.3.1 List view

In this case click on the Browse pulldown in the header, select Data Modeling > Columns:

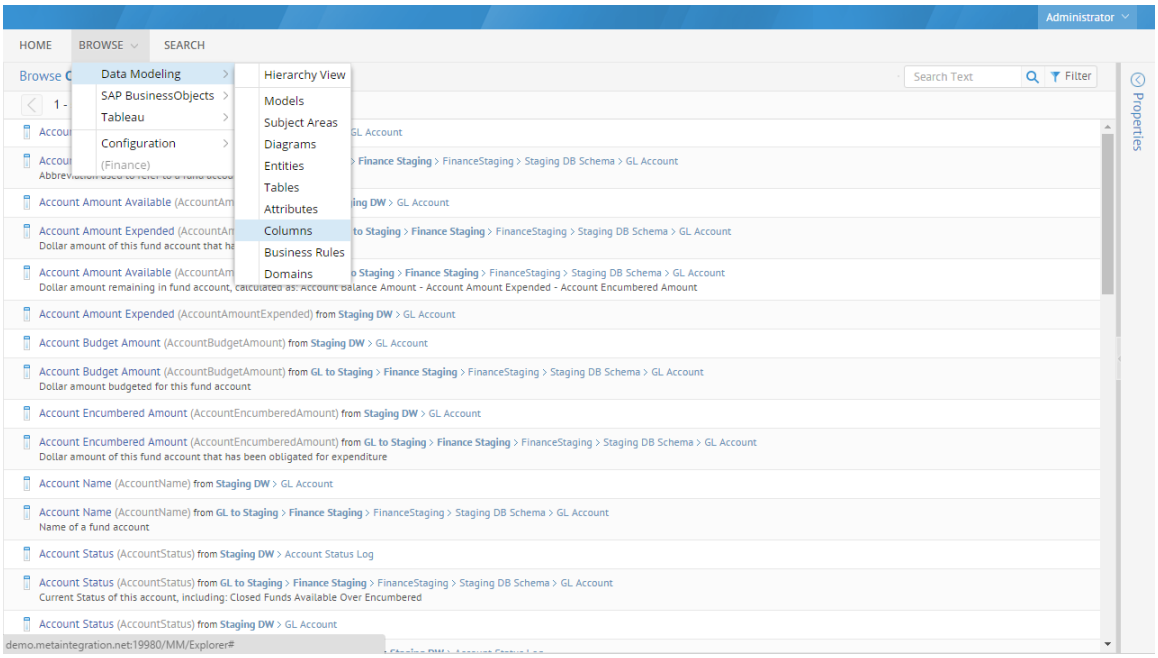


Figure 122 - Browsing for metadata

This result displays all of the columns in all of the data models in the tutorial (all 728 of them). Right now, you only see 50 results at a time, but that may be changed using the options pull-down:

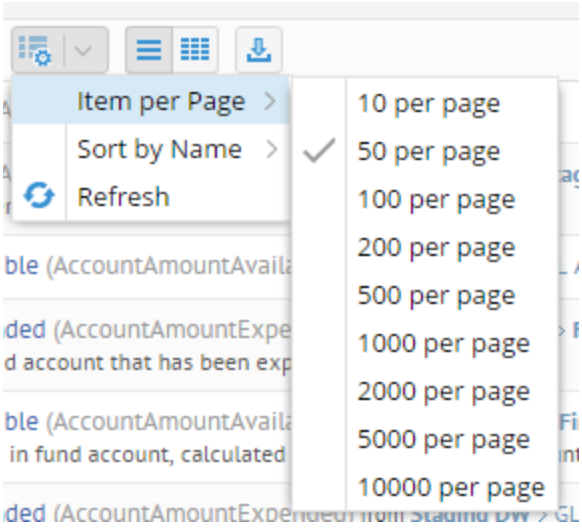


Figure 123 - Number of items per page

One may page through them using the left and right caret icons, and also sort by name.

In addition, on the upper right, there is a search option, where one can specify text to filter on, e.g. on may enter "Account" and have:

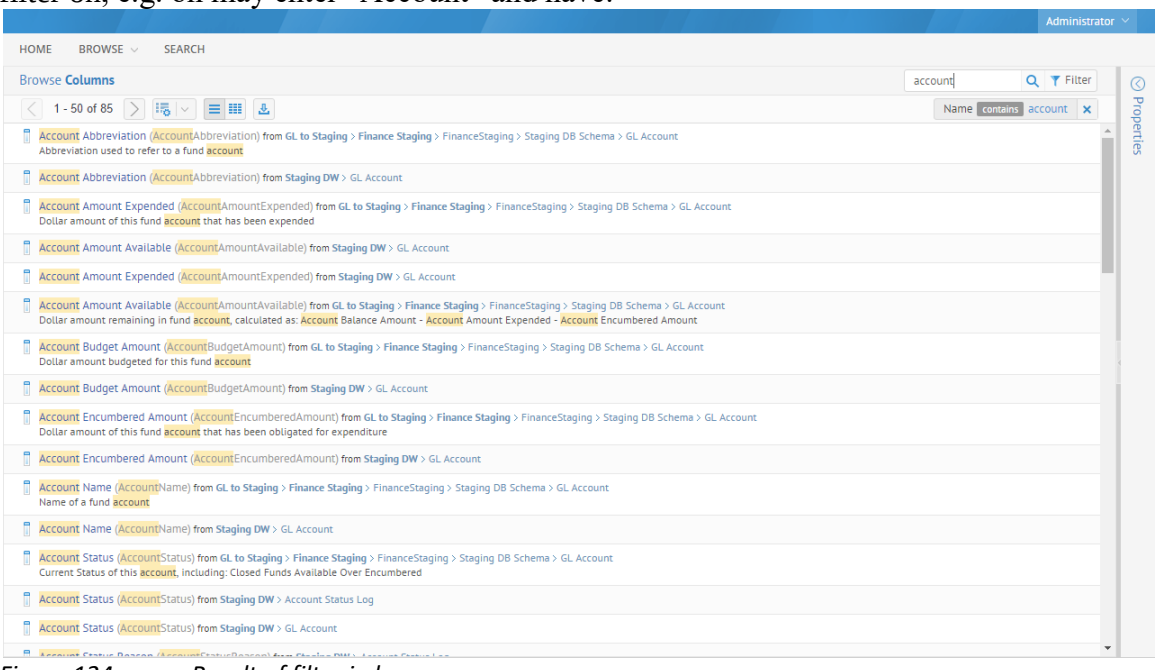


Figure 124 - Result of filter in browse

Note, the current filter is displayed in the upper right. Click on the close (X) next to the filter will clear all current filters:

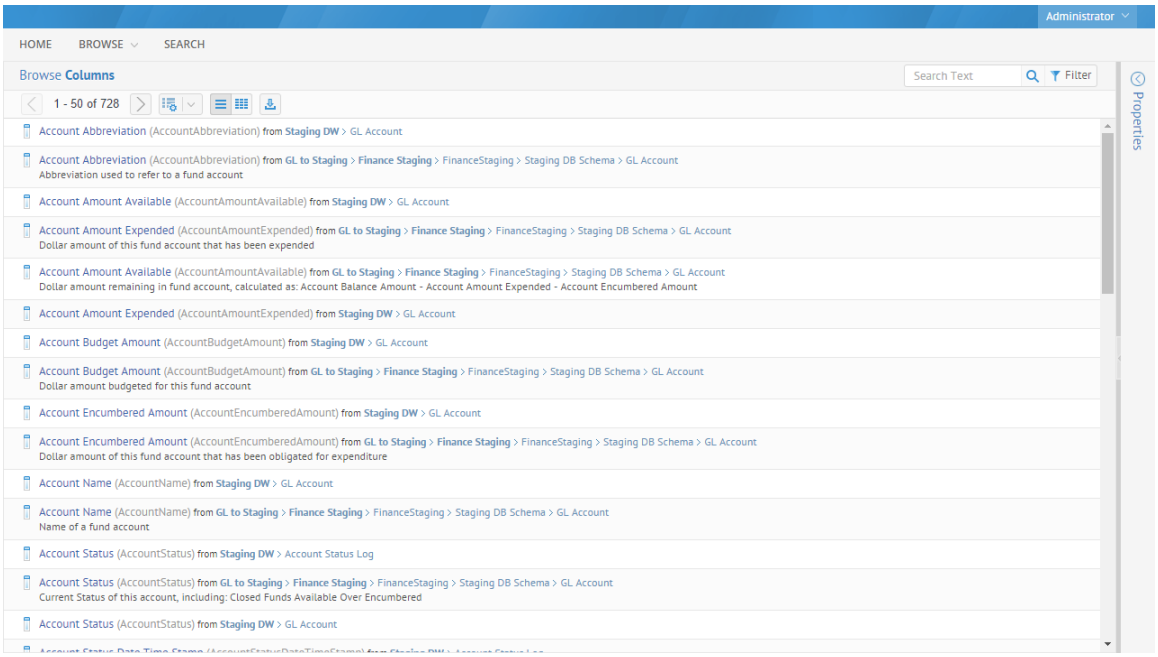


Figure 125 - Browsing without filter

And the results are again all columns.

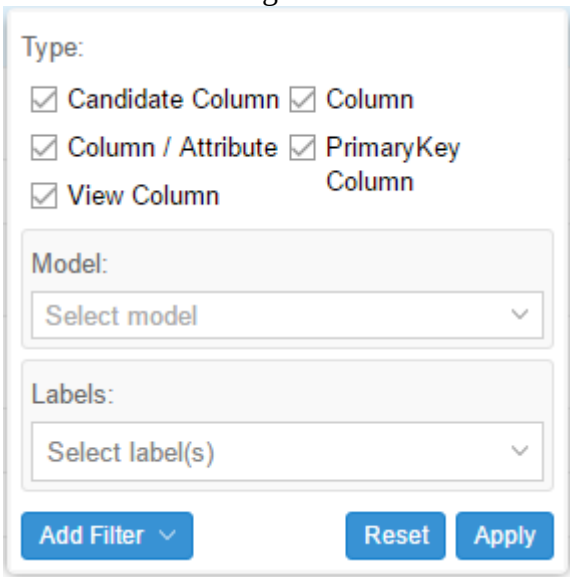


Figure 126 - Filtering options

Now, expand the Filter panel on the right and other criteria are displayed upon which to filter.

- The first area shows filtering by metadata type. In this case, it would be different types of columns that could be shown.
- The second filter is by model. Click on the pull-down and select the Staging DW model. Then click on Apply. Note the results are limited to only those columns in that model.
- The thrid option is to filter by parent object
- One may filter by Stewardship assignment

- One may also filter by label assignments.
- Finally, using the Add Filter button one may filter by any number of properties defined for the object types in question.

Figure 127 - Property types to be filtered by

5.2.3.2 Grid View

One may also use the results of a search or browse action and use the results as a report on the metadata in Meta Integration® Metadata Management (MIMM). In this case, the most common presentation is the grid view. To do this, click on the Display grid icon:

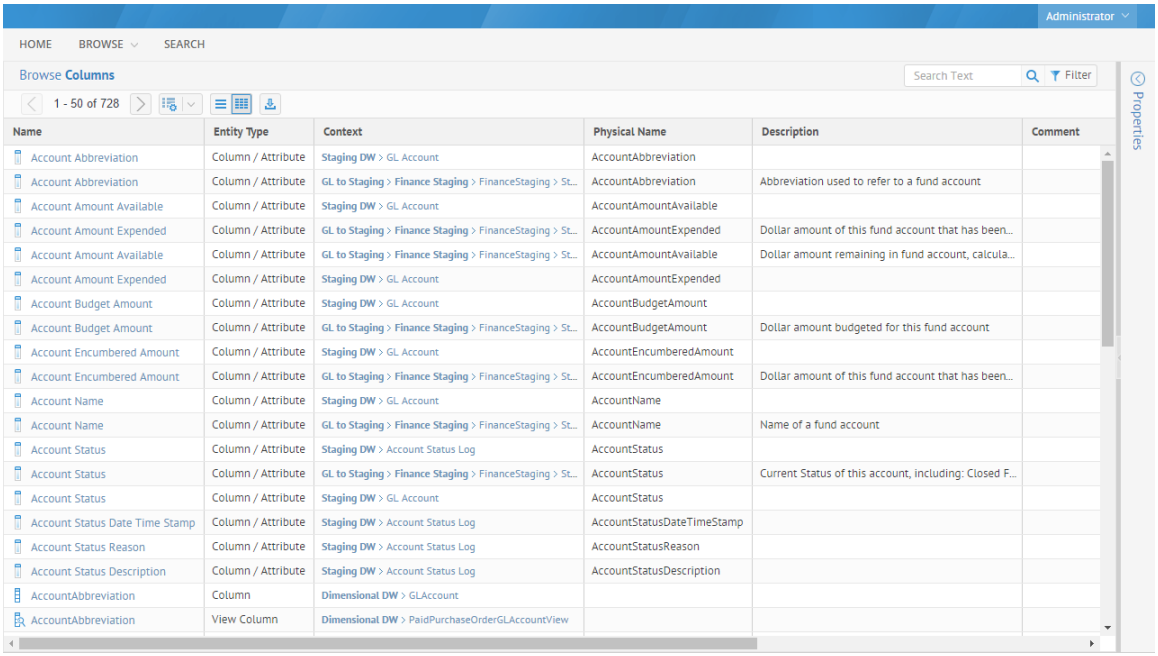


Figure 128 - Display grid

From here, let us show more columns. Click on the pull-down next to any column header and select the Type length, Type scale and Nullable columns:

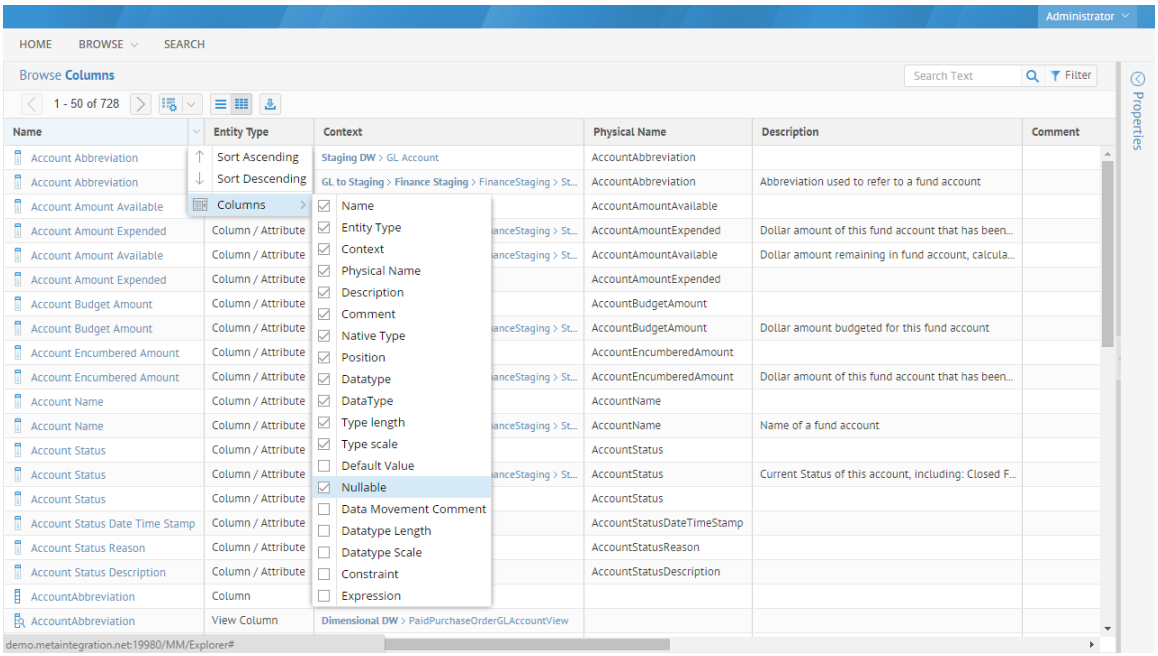


Figure 129 - Adding columns to the grid view

Now, let's sort the list by DataType scrolling to the right until we see the DataType column, then clicking on the DataType column header, or by selecting the pull-down next to the DataType column header and selection sort ascending:

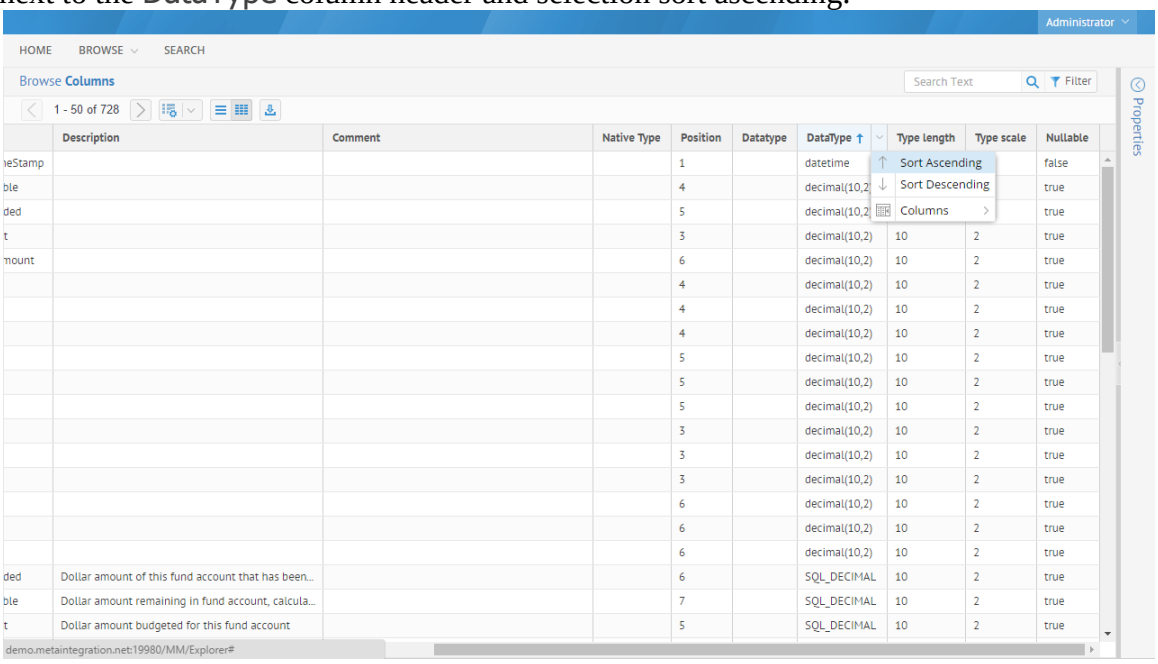


Figure 130 - Sorting columns

Now, scroll to the bottom of the list. Recall that there are only 50 rows displayed and also note that when sorting using the column headers, only the current page is sorted. That is why there are varchar(255) entries on this first page, as the master sort for the

page breaks was alphabetical by Name. We could, of course, increase the number of rows to, say 1000, this way the sorting would include everything in the results.

Of course, we have not yet filtered our results. Click on the Filter icon and specify only Column under Type:

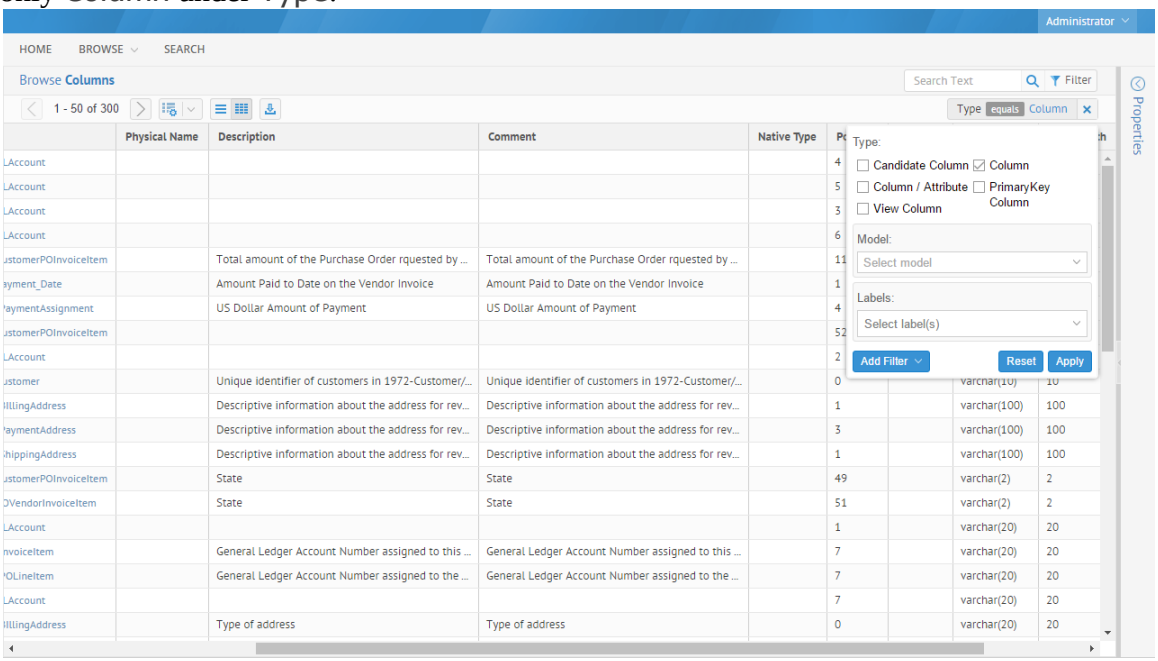


Figure 131 - Filtered to columns

This way, we do not see all the other types of columns.

One may also simply right-click on a cell, and filter criteria shortcuts will be presented based upon the value in the cell and the column you selected.

Now, let's add a filter to only show those items that DO NOT have a DataType of varchar(20) by right-clicking on the cell with that Data Type, and specifying the Not Equals button.

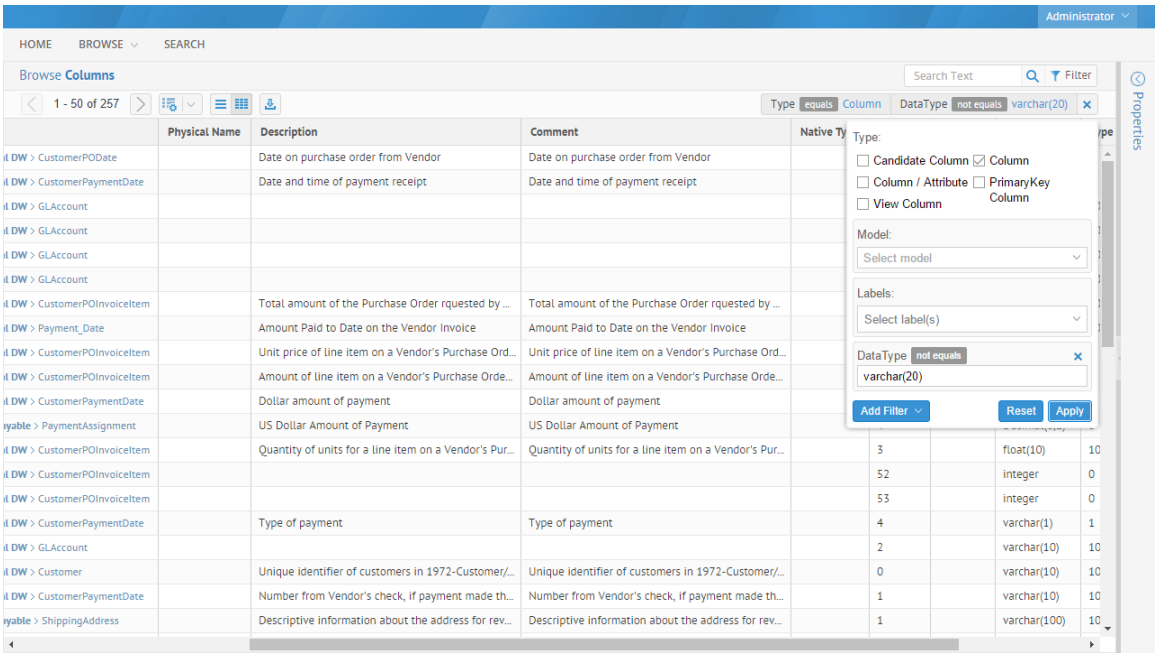


Figure 132 - Using custom filters

Alternatively, one could right-click on a row that has a DataType of varchar(20) and select Filter on those rows not equal to varchar(20).

You may also move columns around with drag and drop. In addtion, you may use the  (download) your filtered results to CSV format.

5.3 Explorer UI from Administrative Users

One may also access the Explorer UI even though signed in as a user associated with a rolename which is not defined as a business user. In this case, one simply goes to Tools › Explorer UI and then sign in again. You are then prompted to indicate your configuration, as the Metadata Explorer UI always zeros in on a specific configuration for analysis.

Finally, when one opens a configuration, one may choose the Open in Metadata Explorer option and go the the Metadata Explorer UI for that particular configuration.

5.4 Attribution of Elements in the Repository Tree

In most cases, it is important to capture information describing the [folder]s, models, and other objects captured within the repository. Information, such as Point of Contact (POC) relate to [folder]s as a whole. Deployment and lifecycle status information may need to be recorded as models are harvested into the repository. In all these instances, the Meta Integration® Metadata Management (MIMM) allows one to define and maintain custom attributes, in order to capture these descriptive attributes.

These custom attributes are created from the Metadata Manager UI using the Tools › Administration › Attributes tab.

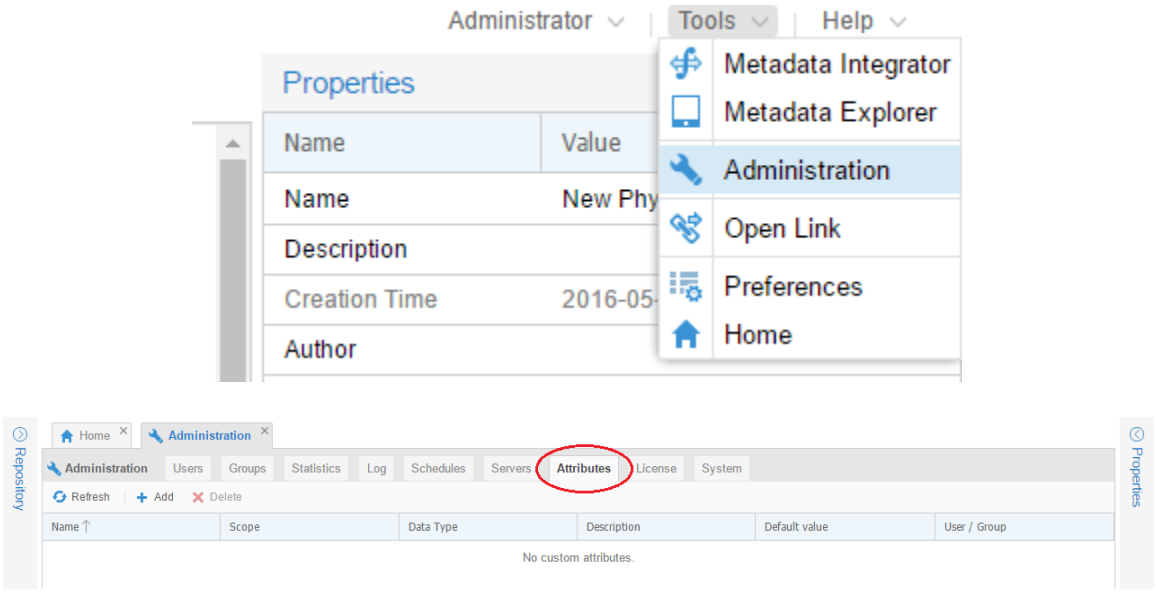


Figure 133 - Navigating to the Attributes tab

Once created there, these custom attributes then appear in the Properties Panel and are accessible via all navigation, viewing and search activities.

5.4.1 Repository level custom attributes

E.g., Use the + Add icon to add a custom attribute named Havesting PoC.

Note, you may click on the Advanced button in order to assign the new custom attribute to specific objects in the repository. In this case, please associate it with these types:

- Model
- Physical Data Model

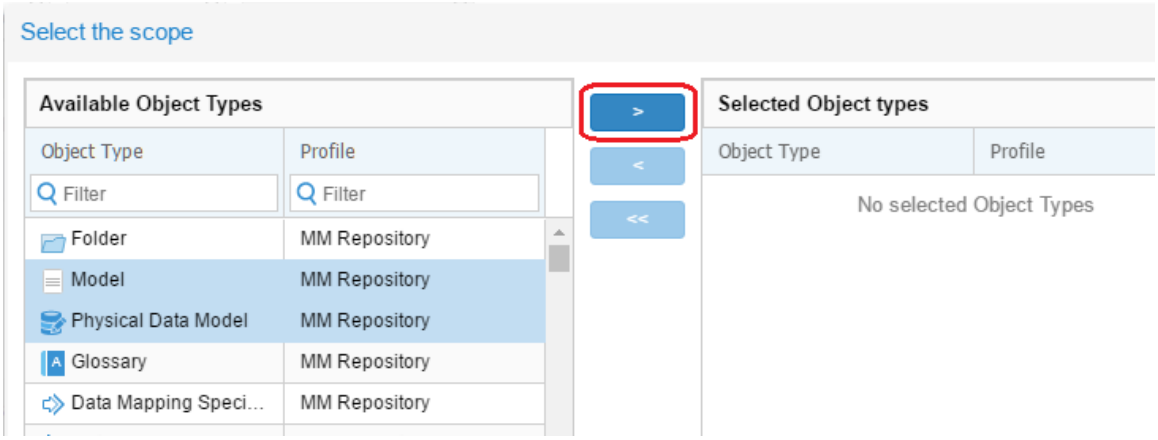


Figure 134 - Add Models (External Content)

Then click on the OK button.

Then finish entering the specifications and click on the Create button.

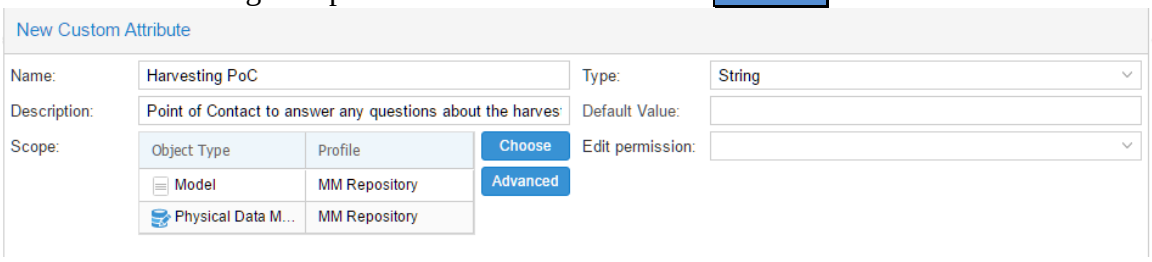


Figure 135 - Created custom attribute

Then, in the case of the PAYTRANS model, go to the Repository Panel, click on the PAYTRANS model and type "Harriet Harvester" as the Harvesting PoC in the Properties Panel.

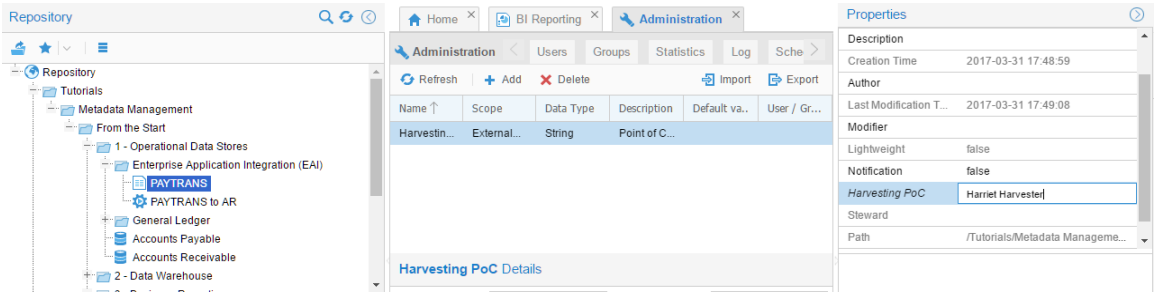
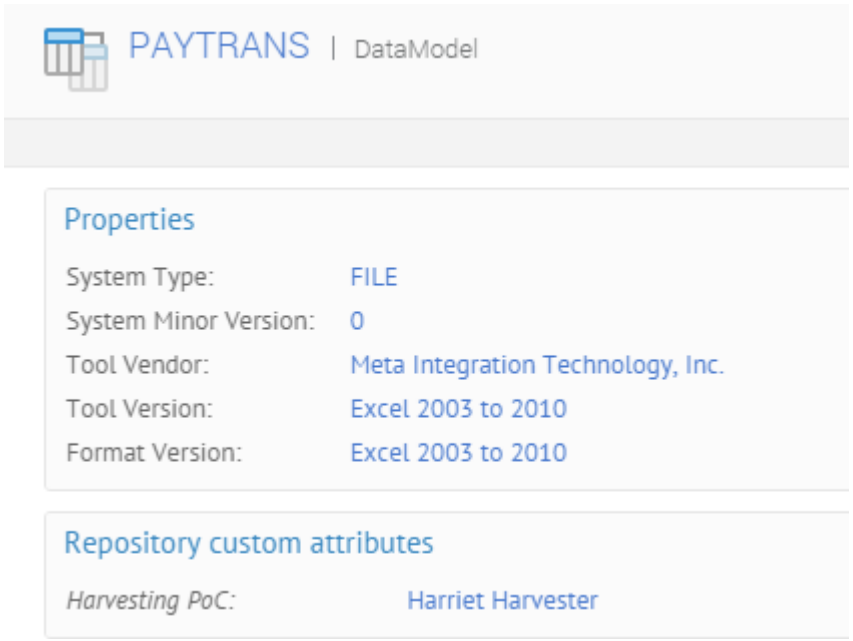


Figure 136 - Custom attributes in the Properties panel

Now, open the PAYTRANS model in the Metadata Explorer UI (open the model, right-click on the top object of the Metadata Browser panel and select Open in Metadata Explorer.



Note, because you added the custom attribute to the repository level object (model or external content), not to some object inside of the model, the customer attribute is not a part of the Properties inside the model (and thus in the Metadata Explorer UI). Instead, it appears in the Repository custom attributes section of the details page.

5.4.2 Content level custom attributes

E.g., Return to the Attributes tab in Tools ➔ Administration, and use the + Add icon to add a custom attribute named Security Classification Status with the attributes as follows:

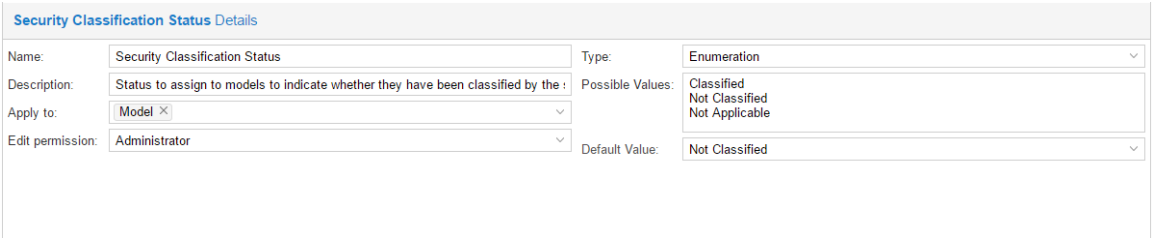


Figure 137 - Custom attributes in the Attributes tab

The Description is Status to assign to models to indicate whether they have been classified by the classification authority.

Note, this time we will click on the Choose button in order to assign the new custom attribute to specific objects in the repository. In this case, please associate it with these types:

- Database
  - Schema
  - Table
  - Column

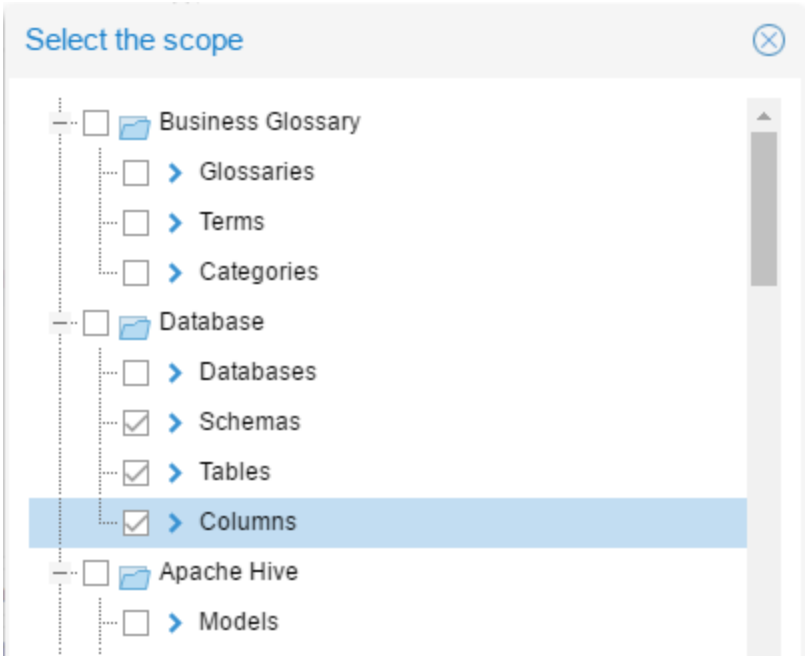


Figure 138 - Add Physical Data Models

Then click on the OK button, and finish the specifications:

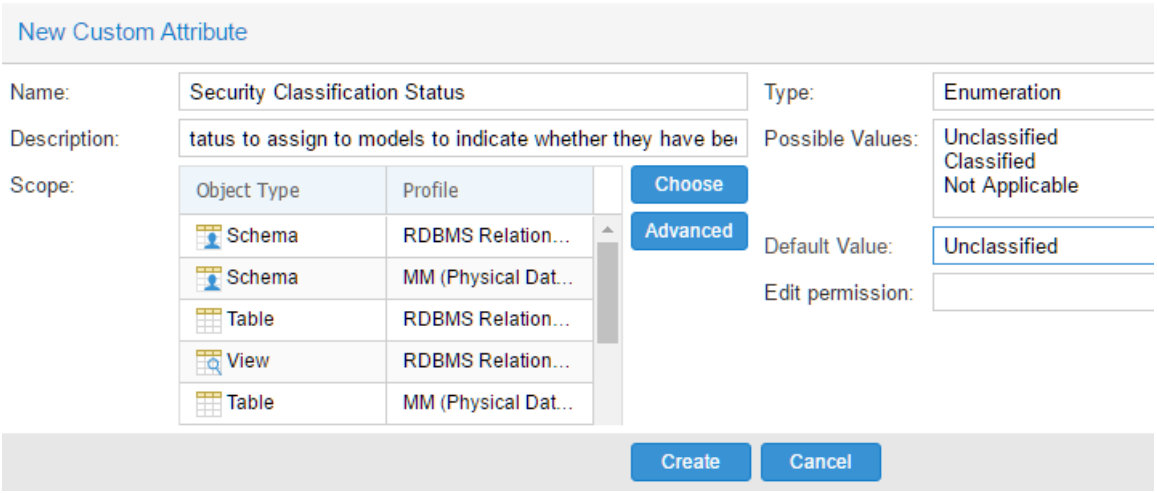


Figure 139 - Security classification status

Then click on the Create button.

Note, by using the Choose button, instead of the Advanced button, more items than just those for PDMs were chosen. One can still clean that up, but it is good enough for now.

And now, add one more custom attribute to set the actual security classification on columns and tables, rather than just the status:

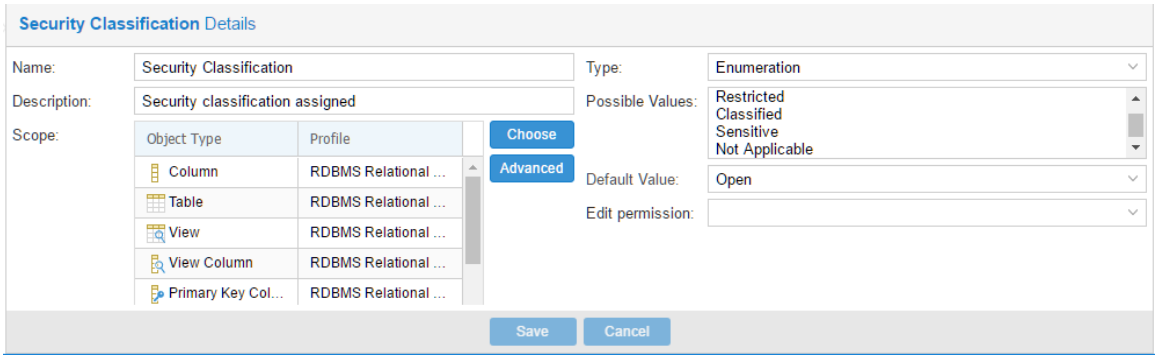


Figure 140 - Security Classification

Now, to use the new attribute, in the case of the Staging DW PDM, go to the Repository Panel, click on the Staging DW PDM model and note there is no Security Classification Status in the Properties Panel. Instead, this custom attribute was assigned internally to the model, not to the repository level content object.

Thus, to see this custom attribute you would open the Staging DW PDM and expand the Schemas level and click on dbo (Schema) in the Metadata Browser panel and open the Properties panel:

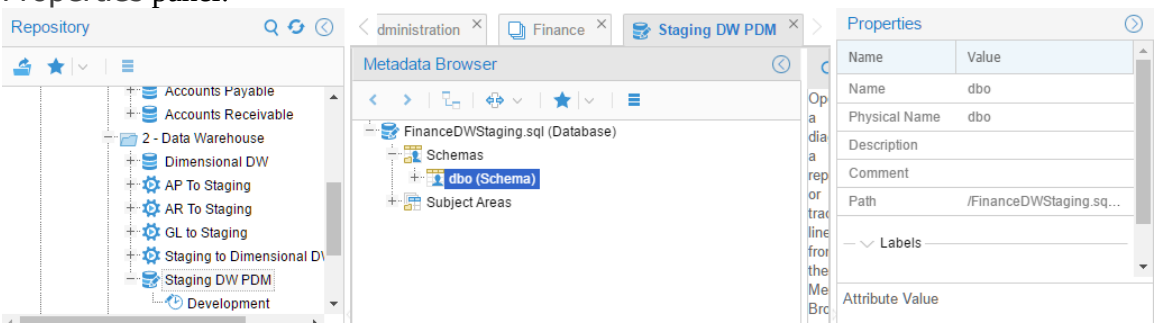


Figure 141 - PDM top level object properties

Note, the new custom attribute is not available. This is because one has not yet been attached to this specific PDM Database. Also, please note that the Metadata Manager UI does not allow editing inside of models. It is for management of models as a whole. Instead, one should use the Metadata Explorer UI.

Now, open the PDM model in the Metadata Explorer UI (right-click on the top object of the Metadata Browser panel and select Open in Metadata Explorer.

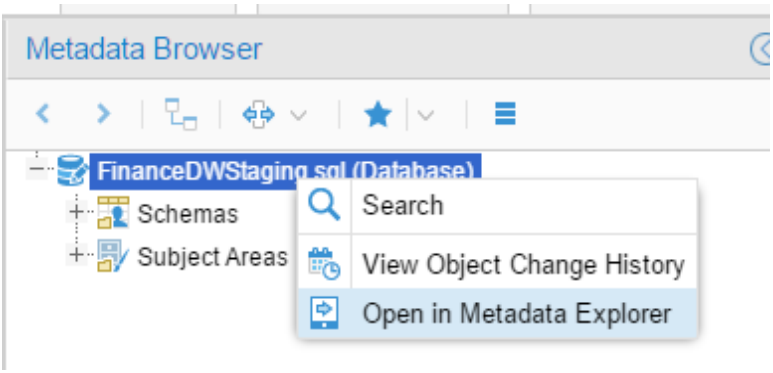


Figure 142 - Metadata Explorer

Click on the dbo schema:

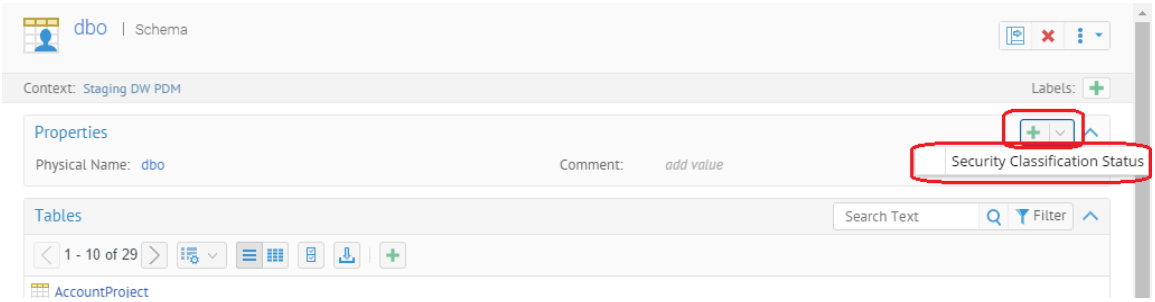


Figure 143 - Staging DW PDM in Explorer UI

Note, custom attribute is not a part of the Properties. In order to add it to this PDM, click on the plus sign (Add custom attribute) in the upper right of the Properties panel. Select Security Classification Status and provide the value UnClassified.

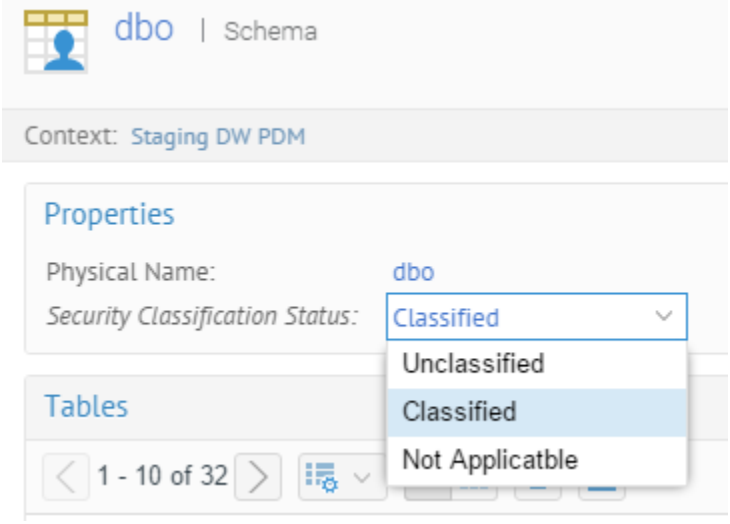


Figure 144 - Assigning the Security Classification Status

Note, that for other schemas, tables and columns, there is no default assignment of Unclassified.

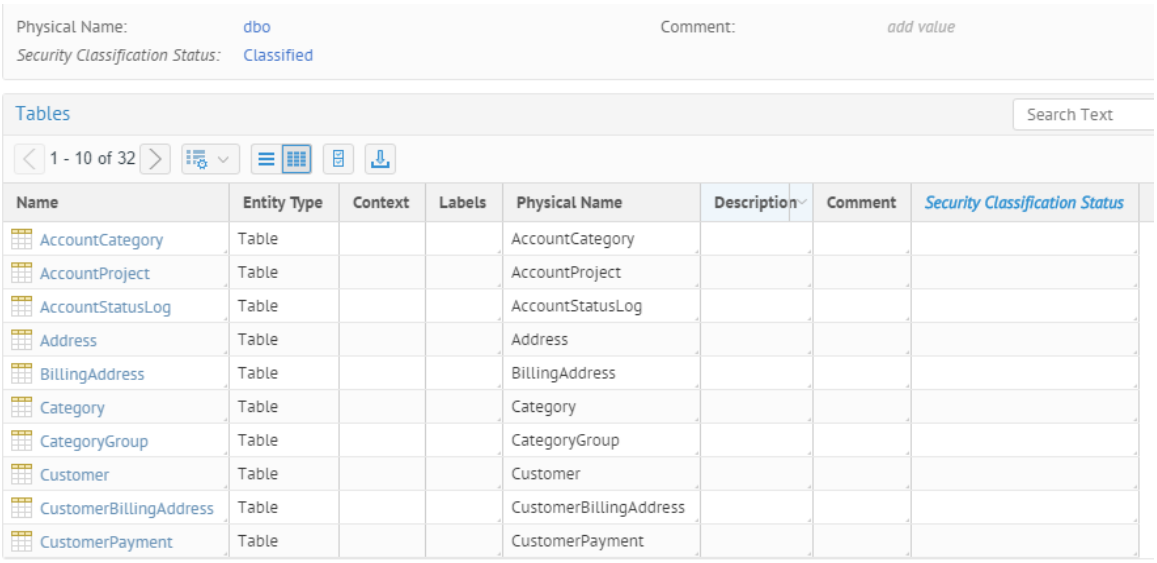


Figure 145 - Status

This is because custom attributes are only given a value (even default) when assigned to an object. I.e., when you assign the attribute, the default is suggested. If one has never assigned the attribute to a particular object (Model in this case) then it does not exist for that object.

So, let's classify all these tables. Switch to grid view for the tables. There are 32 of them, so increase the number of rows:

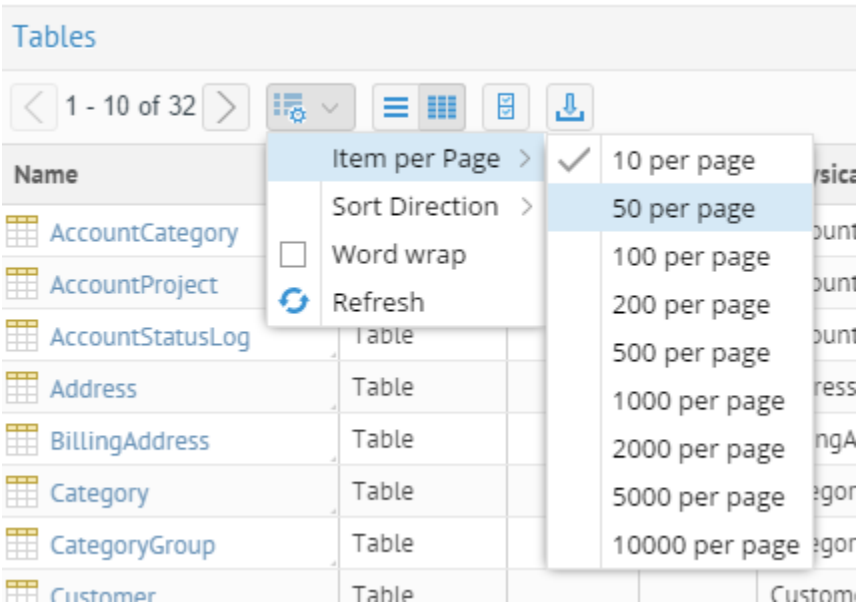


Figure 146 - Number of rows

Select all of the rows and use the pencil icon to edit the values for all and specify "Classified" for the Security Classification Status and "Unrestricted" to the Security Classification.

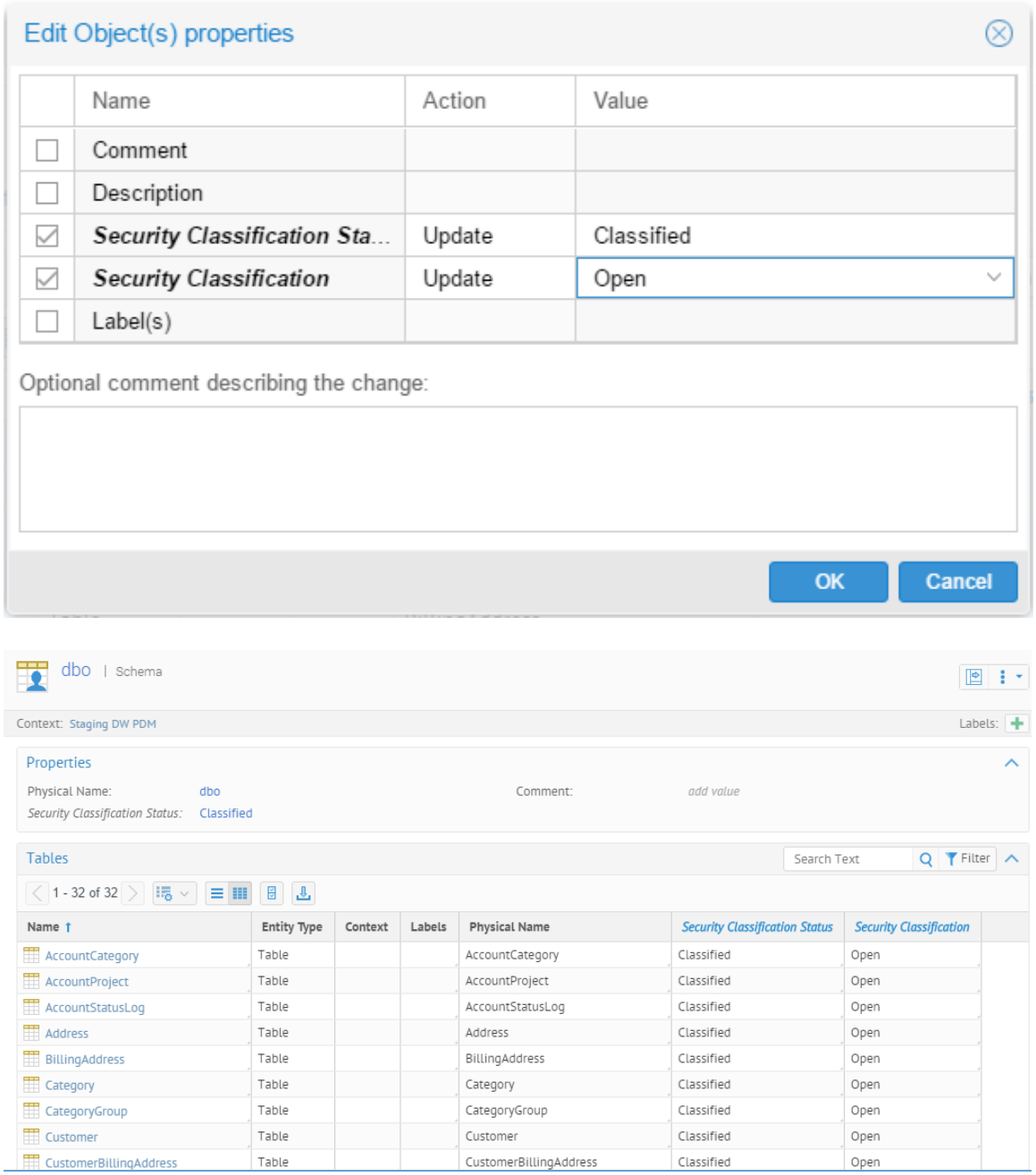


Figure 147 - Custom attributes

Now, if you return to the tab with the Metadata Manager UI, click on Refresh and look for this custom attribute, open the Staging DW PDM and click on the top level object (Database) in the Metadata Browser panel and open the Properties panel:

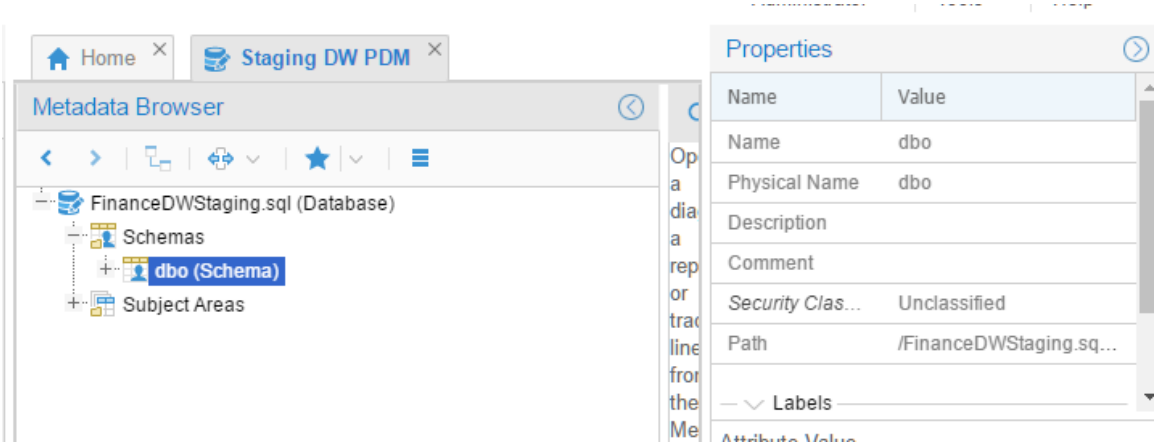


Figure 148 - Classification status in Manager UI

5.5 Attachments

The Meta Integration® Metadata Management (MIMM) allows one to attach any document to model or configuration in the repository. To do so, simply open the model or configuration and click on the (paperclip) Attachments icon or select the More Actions menu and select Attachments.

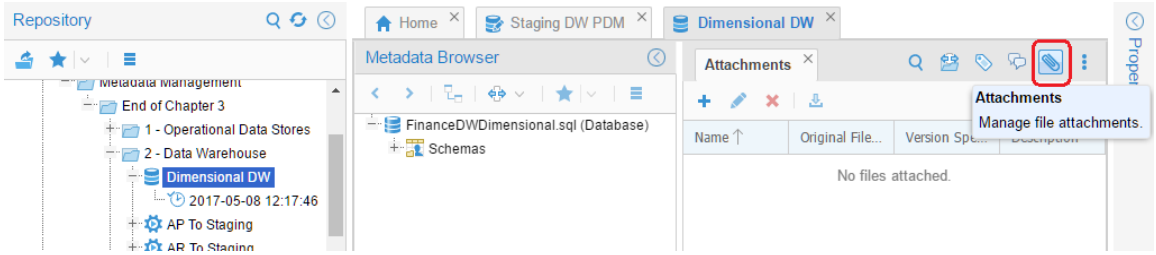


Figure 149 - Attachments

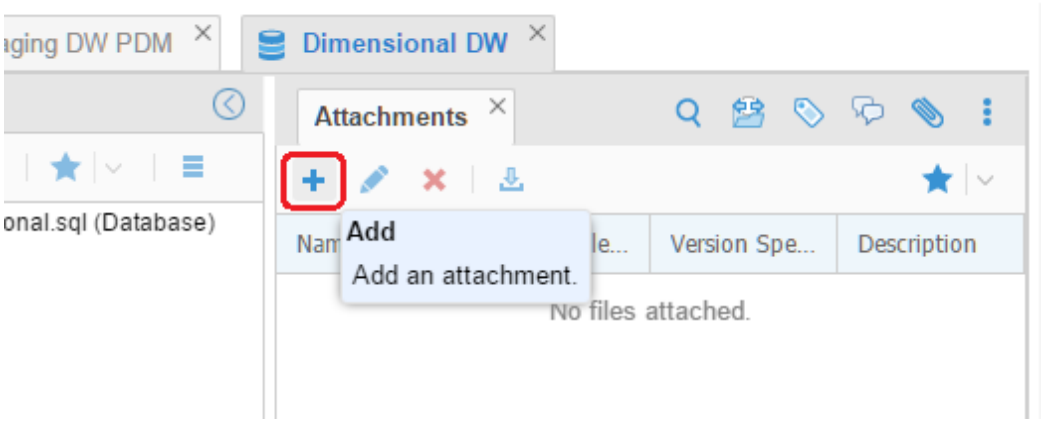


Figure 150 - Add Attachment

Then, simply use the Add, Remove or Download icons to manage your attachments

Note, in this case, we specified Version Specific, which means that this attachment will only be associated with the specific version of the model. Otherwise, it would be the same attachment for ALL versions of the model.

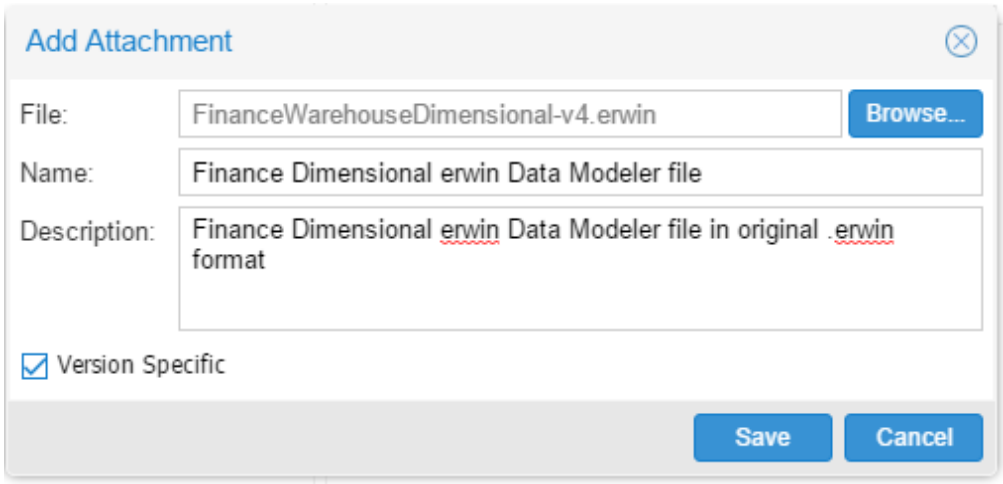


Figure 151 - Attachments panel