Metadata Management Tutorial

Repository Management Best Practices 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 expanded set of capabilities in many facets of metadata management, including:

- Business Glossary
- Data Governance
- Metadata comparison, integration, and mapping
- Version and configuration management
- Data life cycle related metadata management
- Lineage and impact analysis
- Enterprise architecture development, management and deployment.

This document is the culmination of seven years of experience in supporting the enterprise metadata management and integration requirements of numerous clients. It presents in detail and with supporting tutorials our vision, metadata management process and methods, best practices, as well as, many strategic scenarios that leverage Meta Integration® Metadata Management (MIMM) suite. As this section is referred to as the Administration topic it is expected that the reader will have completed the other tutorials (Please see the ReadMeFirst document for a complete outline of the tutorials) so as to have a thorough understanding of topics. The considerations here 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 Meta Integration® Metadata Management (MIMM) 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 the Meta Integration® Metadata Management (MIMM) suite 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 the Meta Integration® Metadata Management (MIMM) environment itself is capable of working with over 100 different versions of third-party tools, 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) 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

1. within a folder.

2 Naming and Organization Best Practices

This section provides general recommendations and best practices when naming and organizing metadata contents within the repository and configurations.

2.1 Repository Structure Organization

The repository panel in the Metadata Management user interface represents the *structural* organization within MIMM. It is useful for:

- Uniquely identifying contents (through a pathname) within MIMM
- Browsing for specific contents
- Maintenance of existing and adding of new contents
- Ease of application of Security Role Assignements
- Ease of execution of scripts.

Thus, the organization should consider these requirements.

The following sections identify several methods of organization which one may consider.

2.1.1 Subject Matter

Organizing the repository by subject matter means creating a folder structure reflective of the high-level subject matter covered by the metadata to be harvested and authored within MIMM. The advantages of organizing the repository by subject matter include:

- High-level organization that will likely be fairly adaptable as the repository grows into new sources of metadata and new business areas
- Aligns well with an overall enterprise information management perspective, especially glossaries and vocabularies.

2.1.2 Enterprise Organization

Organizing the repository to match the enterprise organization means creating a folder structure reflective of the business organization already defined for the enterprise. The advantages of organizing the repository by subject matter include:

- High-level organization that will likely be fairly adaptable as the repository grows into new business units as this is already defined
- Aligns well with business user's view of the world
- Simpler administration when incorporating new enterprise divisions..

2.1.2.1 Business

Organizing the repository to match the actual business areas means creating a folder structure reflective of the way the already defined for the enterprise. The advantages of organizing the repository by subject matter include:

- High-level organization that will likely be fairly adaptable as the repository grows into new business units as this is already defined
- Aligns well with business user's view of the world
- Simpler administration when incorporating new business organizations.

2.1.2.2 Development

Organizing the repository to match the development of systems is quite common for subsets of the repository, in particular for sand-box type areas.

2.1.2.3 Deployment

Organizing the repository by a Dev-Test-Prod type of structure can be quite effective. In fact, oftentimes this gets overlaid on top of one or more of the other types of organizational structures above. The Versioning and Configuration Management Tutorial uses this structure and has a detailed discussion, summarized here:

- In metadata, the Prod environment IS NOT "derived" from the Test environment. Neither is the Test environment derived from the Prod.
- Instead, each environment with have one or more configurations associated.
- In fact, each folder (Dev, Test, Prod) will likely have very similar artifacts (models, etc.) but simply with different bridge parameter information, different connection names, etc., but the ultimate architecture diagrams will be quite similar.

2.1.3 Source of Metadata

Many organizations assign out responsibilities by the tool being used. E.g., the data integration group which uses Talend is one unit of responsibility and the BI group that uses Tableau is another. If an organization works this way, it is likely valuable to organize the repository accordingly.

Note, this does not mean that one will organized all configurations in the same manner. Configurations would need to cut across these tool specific groups.

2.2 Configuration Organization

A configuration is the largest unit of analysis and publication. It is also the only (generally limited) view into the repository for the majority of users, specifically those using the Metadata Explorer user interface. As such, it is critical that the presentation is sensible and convenient for the specific users of each configuration

The contents within a configuration may be organized using a folder structure, presented anytime the configuration is opened and given as the default presentation for Explorer users. Thus, it is possible to tailor each configuration with a different organizational structure, even though it may contain the same basic contents as another configuration with completely different structure.

Many the same considerations apply for configuration organization as do in repository organization and will not be repeated here. One thing to keep in mind though is that the configuration organization should be considered only from the read perspective rather than the update perspective, as nearly all users of it will be explorer (read only) users.

2.3 Content Naming Guidelines

The tutorials demonstrate naming conventions that work fairly well for most organizations. However, they are also designed to educate, rather than help identify and differentiate content in the repository. Thus, e.g., the word "Glossary" is found for each glossary content. This is generally unnecessary.

In fact, the real concerns in terms of good naming are:

- Use names that your audience will be familiar with. For some organizations, an Oracle DB name of FINEDW may be what people use in ordinary parlance. Then, it is a good name as it is short and identifiable. However, if this is not the common use name, try using that. Thus, one could instead use "Finance EDW" or "Finance Warehouse" or "Finance Dimensional"
- When using longer names, keep in mind that more than about 20 characters may be difficult to present in fairly busy lineage diagrams. Thus, one should try to emphasize the prominent and identifying elements of a name and eliminate the common.
- Avoid using the object type in the name. Finance Warehouse Physical Data Model is really not a good choice. Finance Warehouse or Finance Warehouse DB is much better. In some cases, the object type is critical to the meaning. E.g., Finance Glossary could be a good name, or one may refer to the Finance Dictionary.

3 Metadata Version and Configuration Management Theory

When approaching the concept of configuration management and looking at parallels with software CM, a configuration is a set of versions of source and object modules, most likely associated with a build of the system. The metadata example above produced two *configuration* only. In reality, there are several dimensions of possible versions to consider, and thus a multitude of configurations. These dimensions include:

- Multiple deployed versions of each of the source systems, ETL schemas and transformations, warehouse and staging (distributed warehouse), and BI reports and design
- Multiple design, developmental, beta, etc., versions of all of the above
- Multiple versions of standards and/or reference models, standard code sets, code set mappings, look-up and reference tables, etc.
- Multiple versions of data migration transformations for new versions of data systems (i.e., data migration for the accounts receivable source system from version 4.1 to 4.2).
- Other more subtle ones which are not within the scope of this discussion.

All of these dimensions may be represented by different configurations of version within the repository. The following looks at some of the implications of building and maintaining these configurations.

3.1 Impact analysis

Impacts can then cascade for any of the following reasons:

- Data-flow impacts down-stream
- Additional data inputs required for the change to be successful (lineage impacts)
- Standards impacted by change.

Each of these changes creates another version of the mapping in the metadata management environment.

Note, that the software CM approaches are *not* a sufficient analogy (do not address) to the issue of mappings (e.g., the stitching between the CASE tool models and the source ETL schema). In order to answer any of the impact analysis questions commonly asked, the metadata CM environment must be able to identify what version of each of the *mappings* among the metadata, not just simply what versions exist of the schemas and transformations themselves.

To better understand this metadata specific requirement, the following diagram gives the configurable items, versions of those and specific file formats required for managing the metadata for an RDBMS to Informatica ETL to ERwin designed warehouse DB data flow. In this example there are two versions of a source system RDBMS, and thus two versions of each of the source schemas in Informatica and the ETL in Informatica to the warehouse. The existence of these two versions then requires two versions of the mappings from the RDBMS schemas and the Informatica source schemas.

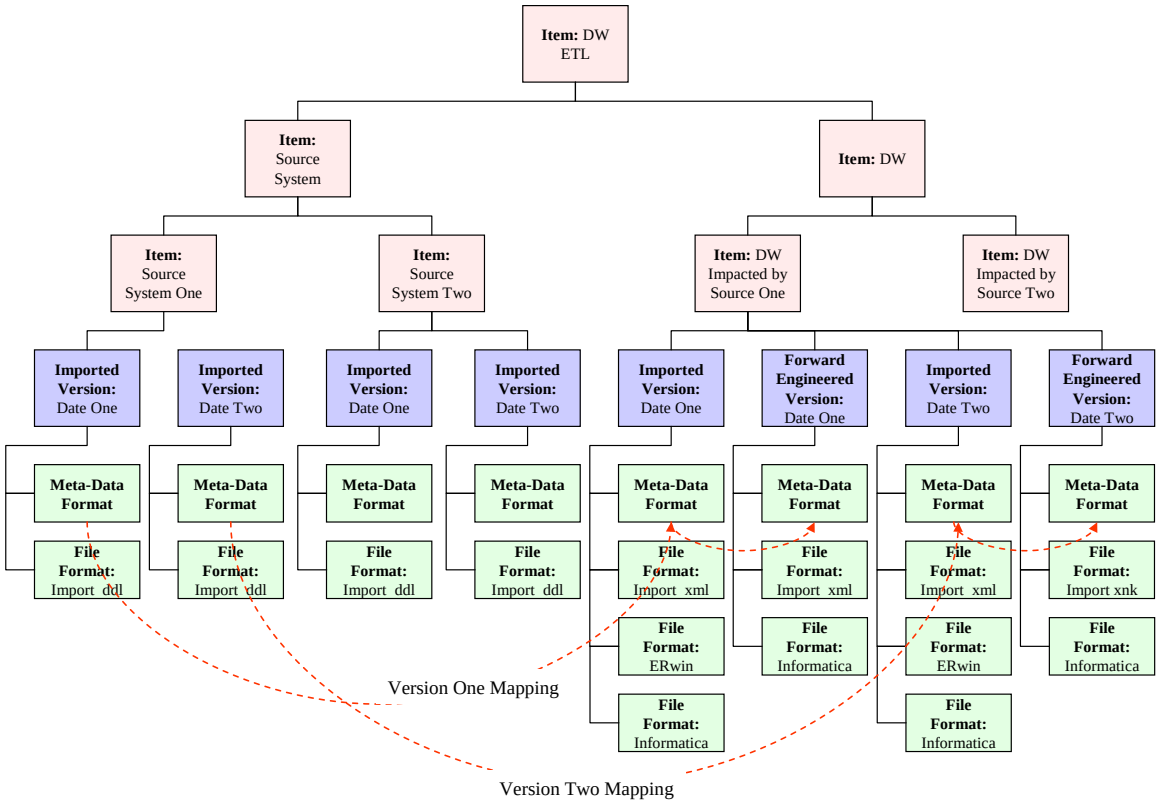


Figure 1 - Multiple Versions within the Repository

3.2 Lineage Analysis

The Sarbanes-Oaxley question in the earlier section represents a type lineage analysis question. These can be divided into the following types of analysis:

- Based upon the current operational configuration
- Based upon a configuration of historically operational systems
- Based upon proposed or developmental configuration.

As one may conclude from this list, it is often necessary to perform lineage analysis on *historical* and *proposed configurations* of versions of the metadata and mappings (including stitched mappings). In this way, lineage analysis, in a similar fashion to impact analysis, requires a sophisticated version and configuration management environment.

3.3 Understanding and analyzing traceability through the life-cycle

Here the word lifecycle refers to the metadata management lifecycle, analogous to the software engineering lifecycle. In this way, not only is it important to be able to identify operational configurations of versions of the metadata, it is also necessary to track the *version history*, i.e., what versions lead to other versions. A good version nomenclature (numbering) system is critical. Valuable also is the ability to notate version history relationships among the versions.

Not so analogous to the software engineering CM principles is the need to provide traceability and *automated generation* of versions of mappings, whether stitching, transforms or standardization relationships. A very common reason for the need for a new version is the change to a data type or data schema. This change will "cascade" down the data stream, impacting many mappings among metadata elements. Each of these changes creates another version of the mapping in the metadata management environment. In order to be manageable, these mappings must be created in as automated a fashion as possible.

3.4 Managing concurrent engineering activities

The analogy with software engineering CM is very strong for currency control of metadata design and engineering. The metadata CM environment will interact with the design tools:

- CASE for RDBMS's, messages and software interfaces
- The ETL designer and/or repository
- The DW designer or CASE tools
- The BI designer and/or repository.

In the software engineering analogy, the source code is completely maintained within the CM tool. This is not always the case for metadata, as the metadata that may be extracted from a particular tool (ETL, BI, DW builder, etc.) could easily be less than what is necessary to re-create the functionality of the tool. Thus, concurrency issues must also be managed, and in fact should be managed, in the ETL/BI/DW/CASE design tool wherever possible. Multiple versions in the ETL repository may exist before a new version is placed in the enterprise metadata management environment. Again, this consideration also impacts the metadata CM process, ensuring that the appropriate milestones are defined for new versions to be placed in the enterprise metadata management environment.

3.5 Ensure proper distribution, assembly and deployment/reuse of assets

In the software engineering analogy, make files and other specifications allow for the direct and automated generation of an *operational* configuration of the software. I.e., the source code is compiled into object code and is then assembled with other libraries and components to create a functioning system. Unfortunately, while CWM and the efforts of ETL/BI/DW/CASE/Repository vendors have made metadata movement and reuse more effective, this is not equivalent to having source code generate object code. Instead, these tools refer to *forward-engineering*.

It is important to note this very strong distinction between the version and configuration management of software and that of metadata. Metadata, is very likely to be forward-engineered into, that is reused by, many different data management tools. However, this type of reuse is often the result of some for of metadata translation or migration, oftentimes across different methodologies (e.g., UML and Relational). A common example would be:

1. Develop the data warehouse schema in ERwin
2. Forward engineer that into an RDBMS
3. Migrate the model to an ETL design tool, say Informatica Designer, as a target schema
4. Forward engineer that same model into the ETL engine
5. Migrate the updated ETL target model to the BI design tool, say BO Designer
6. Forward engineer into a reporting tool, such as Crystal Reports.

In other words, metadata reuse is a critical way in which metadata is leveraged. It is also an example of why metadata management is so critical to the enterprise, as its quality may impact many different systems.

This view of metadata as a reusable asset also means that impact analysis is not simply an answer. I.e., it is not simply a report that is used as a reference to make changes from. Instead it is an integral part of the CM process to actually migrate the changes to the appropriate tool. Again, using the data warehouse example, changes to the design of the ODS in a CASE tool are moved to the ETL source schema and forward engineered. These changes are also migrated to the standard reference model(s) and this same information can be migrated to the target side of the ETL tool as well as the BI designer., i.e., wherever the data flow takes one.

In order to manage this metadata properly, these steps all must be captured as versions of metadata for the given tools, including the related development, testing, deployment and operational configurations of those versions. In practice, this level of CM makes extensive use of the automated metadata comparison, mapping comparison, mapping generation and mapping update functions, which comprise a complete metadata CM environment.

4 Common Repository Roles and Use Cases

4.1 Business End Users

4.1.1 Business Analyst

The business analysts is generally an individual who works directly with the consumers of BI reports and users or managers of the data collection and entry systems. These users are generally assigned to roles associated with the *Explorer UI*.

4.1.1.1 Business Report Lineage (View Ultimate Sources)

Given a BI Report, where does the data come from in terms of data entry and "external" data feeds?

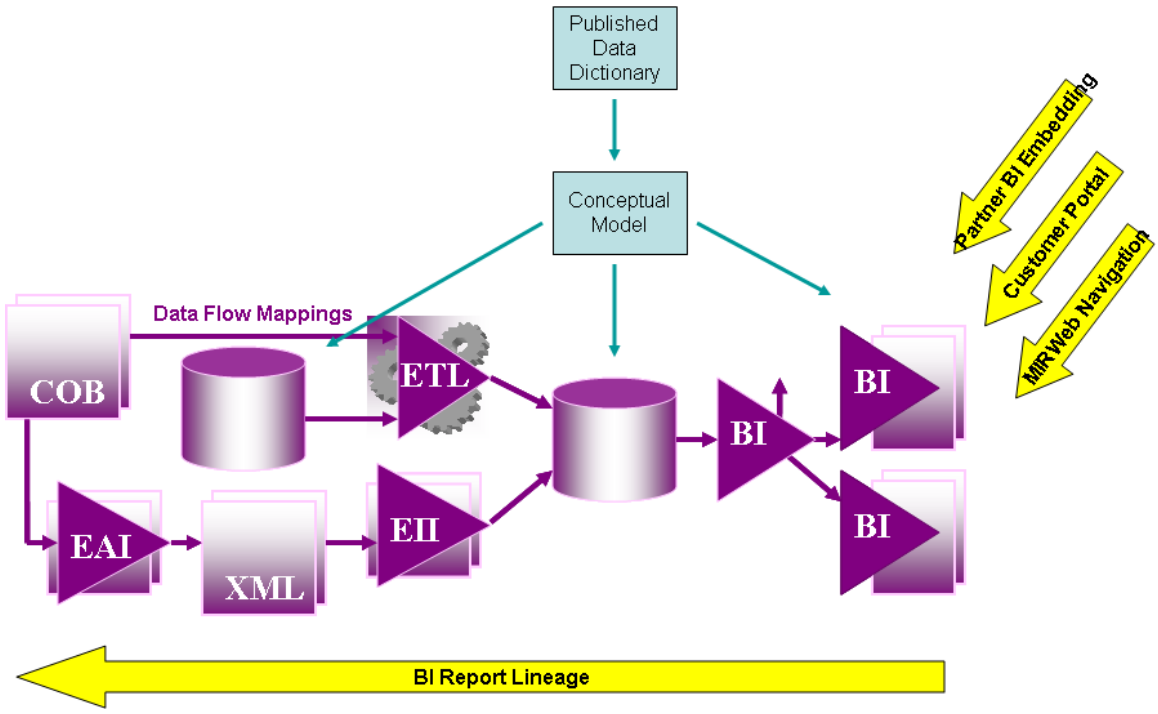


Figure 2 - Business Report Lineage (View Ultimate Sources)

4.1.1.2 Data Entry and Data Feed Impact

Given a data entry field or "external" data feeds, what BI reports and marts are impacted?

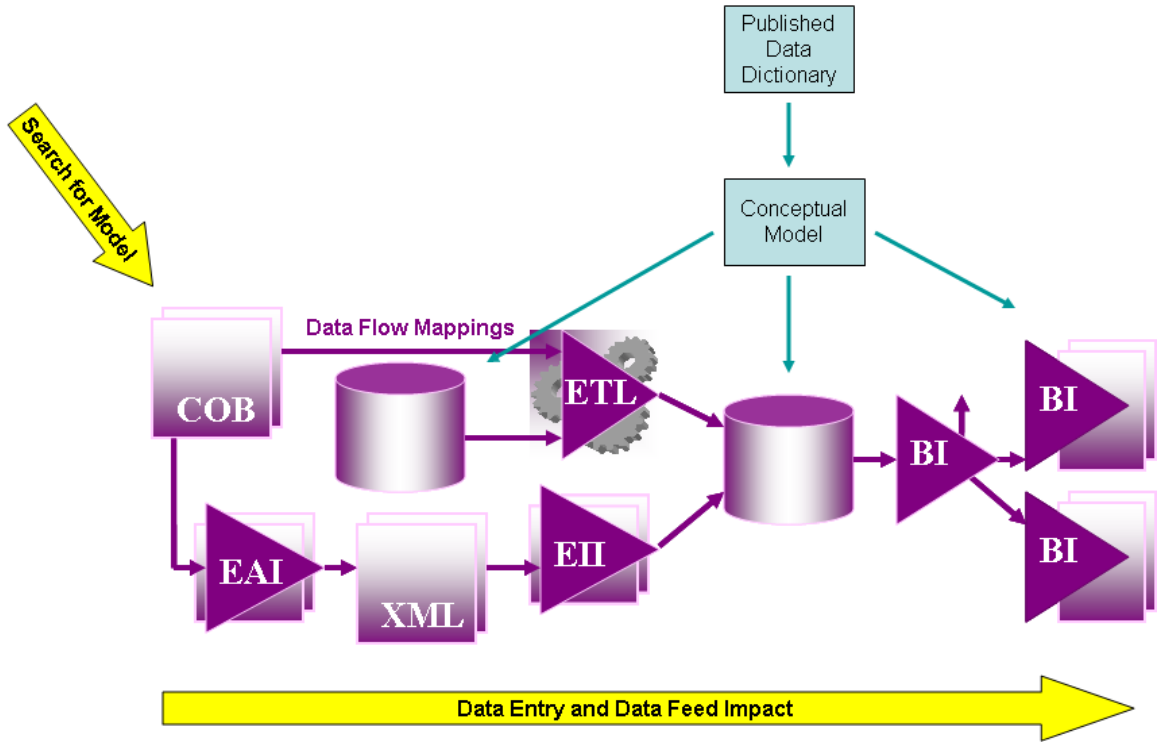


Figure 3 - Data Entry and Data Feed Impact

4.1.1.3 Business Analyst Definition Lookup

Given either of the above, what are the preferred business and data standard definitions or the associated business processes?

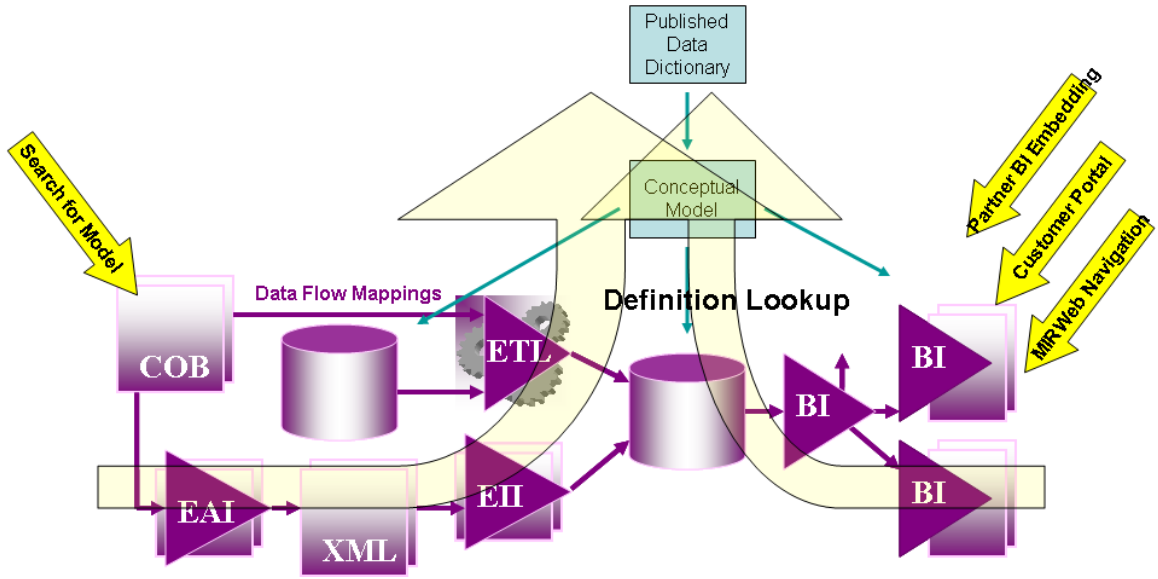


Figure 4 - Business Analyst Definition Lookup

4.1.1.4 Business Analyst Business Process Lookup

Given standard business terminology or processes / activities, what are the data entry and "external" data feeds and/or BI reports and marts which are involved?

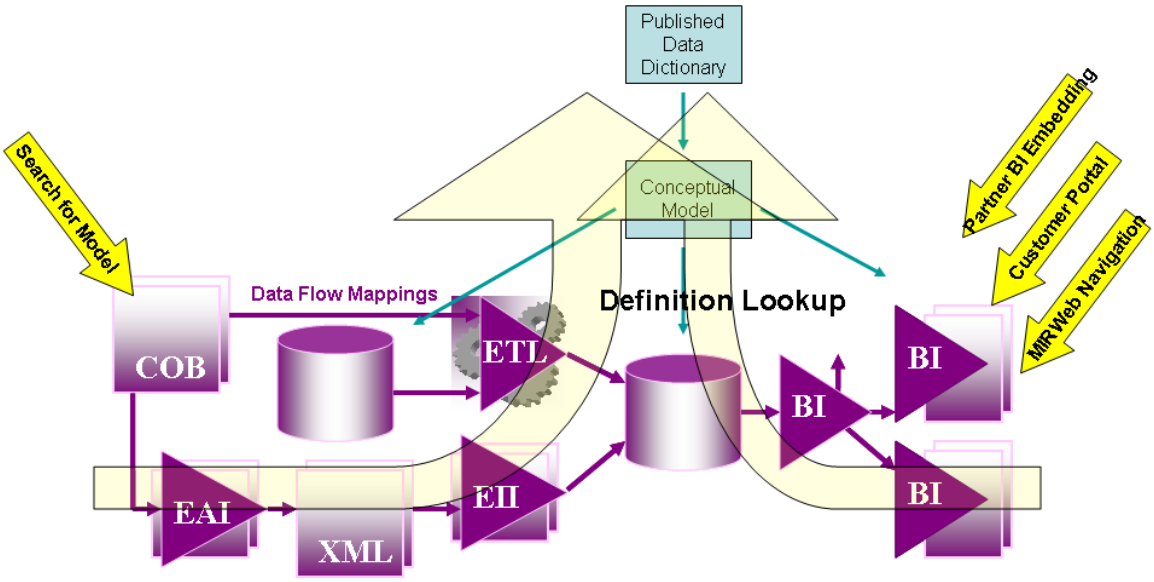


Figure 5 - Business Analyst Business Process Lookup

4.1.1.5 Business Analyst Architecture Visualization

Given a particular business objective, show an architecture diagram.

Figure 6 - Business Analyst Architecture Visualization

4.1.2 Business Data Entry System Users and Managers

The Business Data Entry System Users and Managers are generally users or managers of the data collection and entry systems.

The Explorer UI is always limited by configuration, which is generally the published "official" one as maintained by the Repository Administrator role Repository Publisher/Editor and their particular subject are of interest

Use cases are a subset of those for the Business Analyst, specifically the

- Data Entry and Data Feed Impact
- Business Analyst Definition Lookup

4.1.3 Business Reporting System Users and Managers

The Business Data Entry System Users and Managers are generally users or managers of the data collection and entry systems.

The Explorer UI is always limited by configuration, which is generally the published "official" one as maintained by the Repository Administrator role Repository Publisher/Editor
Use cases are a subset of those for the Business Analyst, specifically the

- Data Entry and Data Feed Impact
- Business Analyst Definition Lookup

4.2 Information Management Analysts

4.2.1 Data Administrator

Data Administrators are generally responsible for good management of the data element standards, enterprise and/or conceptual model design, conformance, data dictionary administration, and publication of standards and data administration assets. "System level" DA's will be generally limited to a particular configuration. Enterprise level DA's will "cut across" the entire organization (and thus what most system level individuals see as configurations.

4.2.2 Data Steward

Data Stewards are generally responsible for a specific set of data standards or elements within an organization. Oftentimes, they are people with an IT background, as well as some understanding of the business area. As such, their role with the repository is limited except as it relates to specific data elements in the Data Dictionary or conceptual models and how those elements are reflected in actual systems. Governance, conformance and element standardization and publication/dissemination are this person's primary responsibility.

Repository scope is:

- Generally limited to a proposed and current configuration as administered by a DA.
- For some enterprise level data elements, the analysis will "cut across" the entire organization (and thus what most system level individuals see as configurations)

4.2.2.1 Use Cases

- Data Dictionary Use Cases
- Data Governance Use Cases
- Standards Publication and Dissemination
- Standards Conformance
- Version history by element

4.2.3 Data Proponent

Data Proponents are generally responsible for the business community liaison activities for a specific set of data standards or elements within an organization. Oftentimes, they are people with an little IT background, but a very high level of understanding of the business area. As such, their role with the repository is limited except as it relates to specific data elements in the Data Dictionary or conceptual models and how those elements must be defined, rationalized, as well as analyzing the data processes used to reflect business processes.. Governance, business process and IT process fidelity and element standardization are this person's primary responsibility. Generally, the will depend upon a data steward and/or data administrator to ensure conformance, publication, etc.

Repository scope is:

- Generally limited to a proposed and current configuration as administered by a DA.
- For some enterprise level data elements, the analysis will "cut across" the entire

4.2.3.1 Use Cases

- Data Dictionary Use Cases
- Data Governance Use Cases
- Standards Publication and Dissemination
- Standards Conformance

4.3 Information Technology Developers

4.3.1 Repository Point of Contact

These are IT Development staff members, usually related to a particular source tool (e.g., Infa or BO) and particular projects of programs, who are the point-of-contact for that effort and the repository. They are generally responsible for the posting of new versions, coordination with the larger metadata life cycle process (milestones, uploads, impact requests, etc.), and monitoring of the status of their model contents and the configurations they are contained within.

Repository scope is:

- For "system level" PoC's analysis will be generally limited to a particular configuration and update to a particular set of model contents (related to the development activity) and their related configurations
- For PoC's, the analysis will "cut across" multiple configurations. Again update is to a particular set of model contents and their related configurations

4.3.1.1 Use Cases

- Upload unexpected new version of a model and respond to change impact notification as a part of Assess impacts from model or mapping change
- Upload new version of a model as a part of Collect models and mappings for planned configuration
- Create Data Flow Mapping
- Migrate Data Flow Mapping
- Monitor Model/Mapping Containing Configurations Statuses

4.3.2 System Analyst

These are IT Development staff members, usually related to a particular source tool (e.g., Infa or BO) or particular projects of programs (e.g., Finance), who are responsible for analyzing the impact of changes. They work in concert with the Repository PoCs to assess and respond to requests of impact costing and in developing new models to be posted as new versions.

Repository scope is:

- For "system level" SAs analysis will be generally limited to a particular configuration
- For some cross-configuration SAs the analysis will "cut across" multiple configurations.

4.3.2.1 Use Cases

- Create Data Flow Mapping
- Migrate Data Flow Mapping

- Version history by element

4.3.3 Program/Project Manager

These are IT Development members who are responsible for a particular project or program. They will generally be managing the development as well as reviewing or managing the repository configuration management process.

Repository scope is:

- For "system level" Managers, analysis will be generally limited to a particular configuration and update to a particular set of model contents (related to the development activity) and their related configurations. Administration may include configurations where it is the Manager's responsibility.

- For some Managers, the analysis will "cut across" multiple configurations. Again update is to a particular set of model contents and their related configurations, Administration may include configurations where it is the Manager's responsibility.

4.3.3.1 Use Cases

- Configuration Management Process Driven by Physical Model Change
- Fully Automated Configuration Management
- Configuration Management of Architecture Change
- Build Initial Configuration
- Configuration Management Process with Multi-Models

4.3.4 IT Architect

These are IT Development staff members who are involved in architecture issues for a given set of systems and how they are represented in the repository. They will generally be using the repository as an analysis tool, especially what-if type analysis when true architecture changes are being proposed and developed.

Repository scope is:

- For "system level" architects analysis will be generally limited to a particular configuration.
- For other Architects, the analysis will "cut across" multiple configurations.

4.3.4.1 Use Cases

- Configuration Management Process Driven by Physical Model Change
- Fully Automated Configuration Management
- Configuration Management of Architecture Change
- Build Initial Configuration
- Configuration Management Process with Multi-Models

4.4 Repository Administrators

4.4.1 Repository Role and Security Administrator

This is a Repository Project staff member, usually having responsibility for adminiering Role Groups and Permission Groups. They coordinate with the enterprise security administrators to ensure that users of the repository are assigned to appropriate Role Groups and Permissions Groups. This administrator should have Administrator privileges for Repository.

4.4.2 Use Cases

- Authentication via LDAP
- Authentication via MIR
- How system determines permissions
- How system determines profiling
- How administrator modifies permissions
- How administrator modifies profiling
- Server side personalization user object based

4.4.3 Repository Structure Administrator

This is a Repository Project staff member, usually having responsibility for adminiering the hierarchical organization and link references within the folder and content structure of the repository. They coordinate with the other Repository Administrators, Project Managers, Repository PoCs and others to ensure that the way the repository is organized is efficient, effective and manageable. This administrator should have Administrator privileges for Repository.

4.4.4 Repository Configuration Administrator

This is a Repository Project staff member, usually having responsibility for administering or coordinating the administration of configurations within the repository. They coordinate with the other Repository Administrators, Project Managers, Repository PoCs and others to ensure that the way the configurations are organized and maintained is efficient, effective and accurate. This administrator should have Administrator privileges for Repository.

4.4.5 Use Cases

- Configuration Management Process Driven by Physical Model Change
- Fully Automated Configuration Management
- Configuration Management of Architecture Change
- Build Initial Configuration
- Configuration Management Process with Multi-Models

4.4.6 Repository Analyst

This is a Repository Project staff member, having responsibility for analysis of repository useage, organization, efficiency, etc. They coordinate with the other Repository

Administrators, Project Managers, Repository PoCs and others to ensure that the repository is operating in an efficient, effective and usable manner. This administrator should have Administrator privileges for Repository.

4.4.7 Repository Publisher/Editor

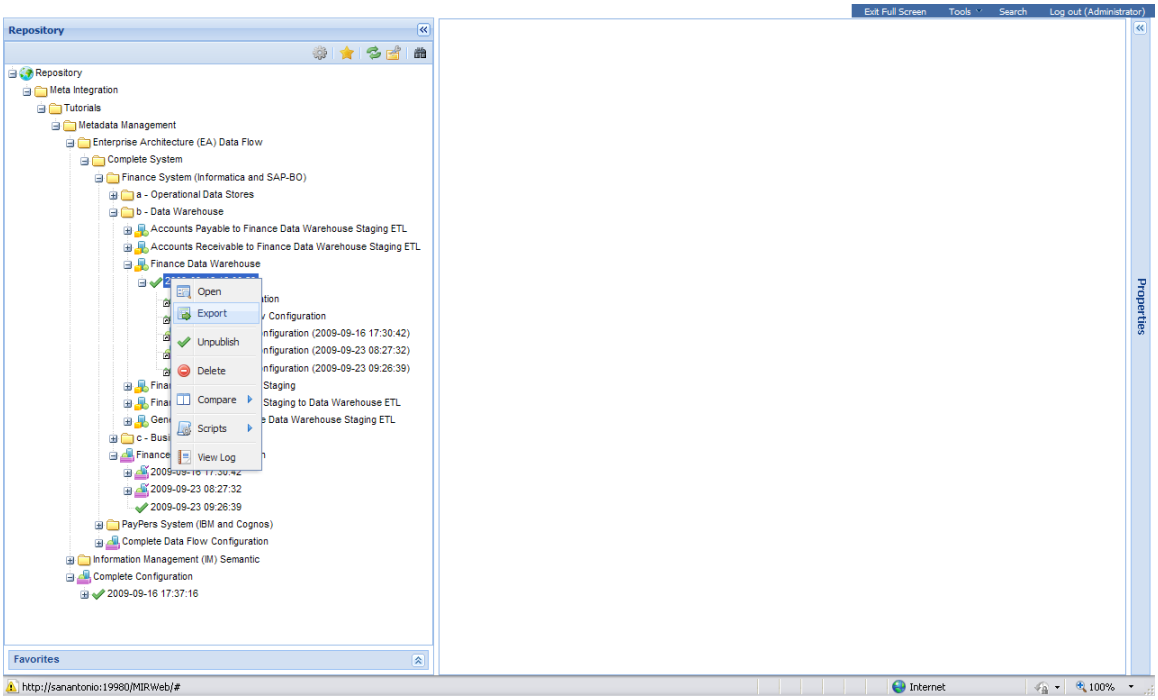
This is a Repository Project staff member, having responsibility for ensuring that the correct versions and configurations information is available to the right customers in the most usable and valuable format or manner. They coordinate with the other Repository Administrators, IT Developers, IM Analysts and Business End Users to actually make changes. They mostly analyze, review and recommend changes to porcesses that affect publication of information in the repository. This administrator should have Administrator privileges for Repository.

5 Metadata Forward Engineering

When adding new components to your information architecture, e.g., a new Business Intelligence design and reports, it is often advantageous to reuse the metadata already collected in the repository.

In this scenario, a new BI solution using IBM Cognos ReportNet Framework Manager and Report Studio will be used to report on a subset of the metadata within the data warehouse. Since this is a new Framework Manager file (not an existing one), we will help the developers by exporting the data warehouse model that we have extracted from CA ERwin Data Modeler. We know that it is a valid model, as it stitched cleanly to the Target side of the ETL from Staging to Warehouse, and also cleanly to the existing BI design models.

First, export the data warehouse model into the IBM Cognos Framework Manager Format by right-clicking on the published version of the Financial Data Warehouse model and selecting Export.



Exporting a model from the repository

Then pick the correct tool format.

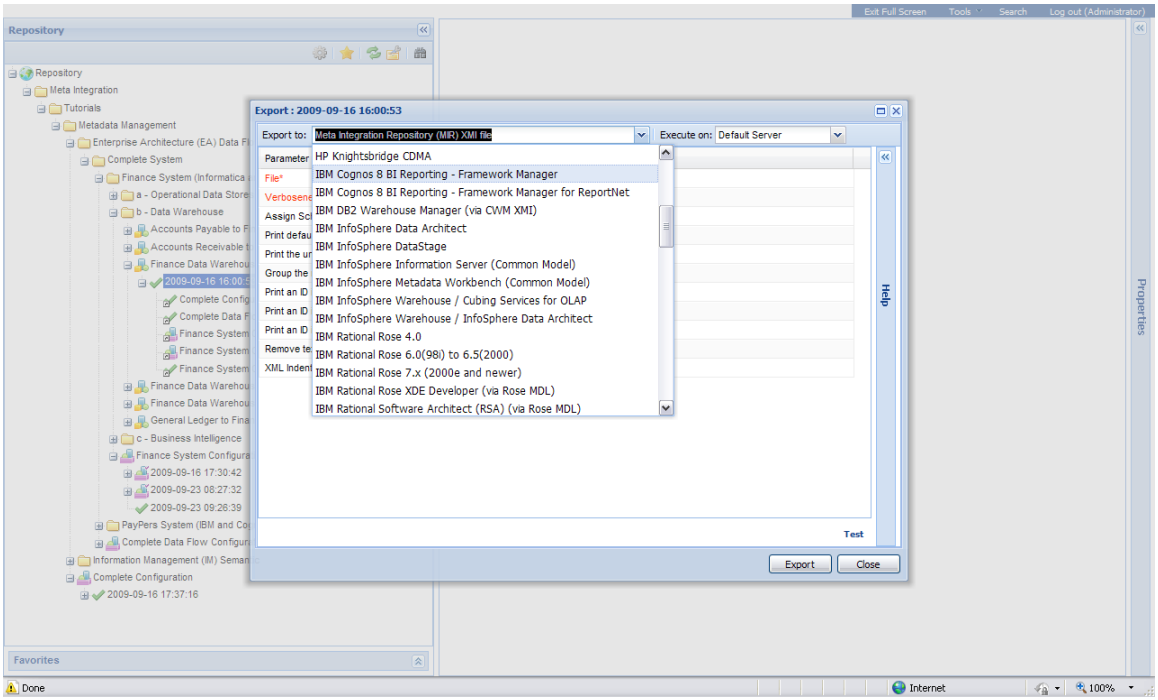


Figure 7 - Selecting the export tool format (bridge)

Now enter the proper bridge parameters, including the file to output the result to. Refer to the tool tips (in the Help panel at the right) for assistance.

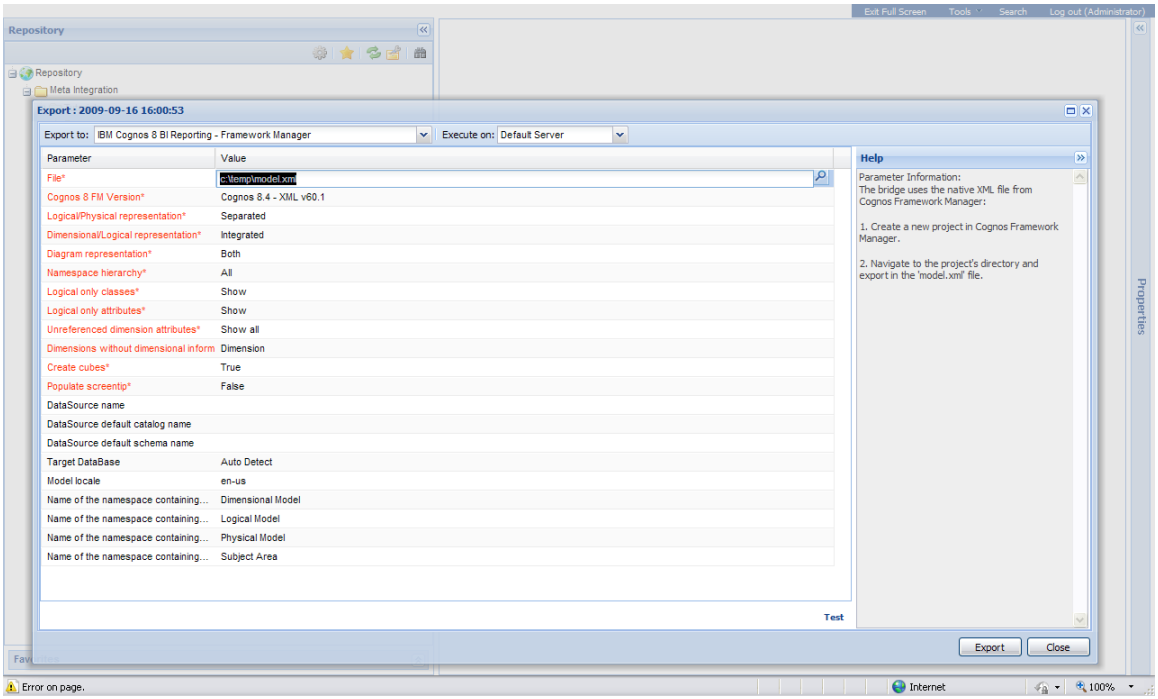


Figure 8 - Populating export parameters

and a location to save the file. Now press the Export button.

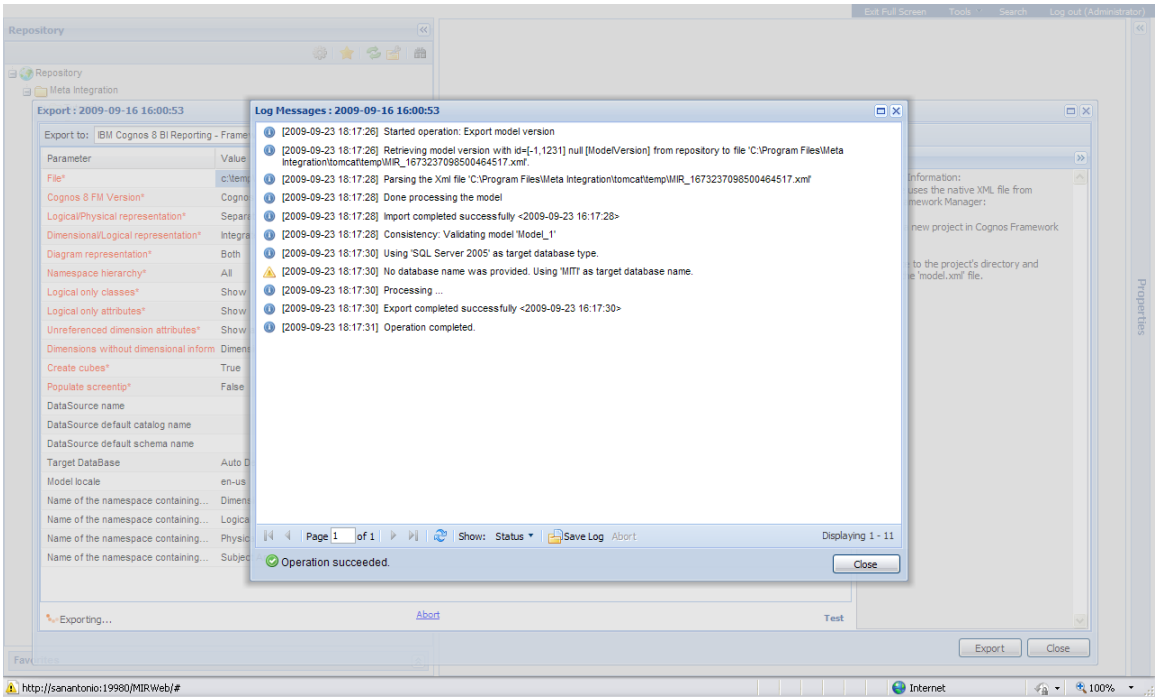


Figure 9 - Exporting Framework Manager Model for new BI Design

One may now take this files and open it in the target tool according to the instructions in the tool tips.