Metadata Management Tutorial

Data Mapping Design Using Meta Integration® Metadata Management (MIMM)

Version 9.1.0 (8/25/2017)

TABLE OF CONTENTS

TABLE OF FIGURES

1 Introduction

In previous tutorials we have seen how to harvest, stitch and analyze metadata from sources external to the Meta Integration Metadata Manager (MIMM) repository, including from data stores and logical or conceptual models (e.g., data bases, data modeling tools), business intelligence and reporting environments and data integration environments (e.g., ETL, ELT). We have also seen how one may *author* data store metadata inside of MIMM, either based upon externally sourced metadata or entire from scratch (e.g., documented models and glossaries). Finally, we have seen how to create and edit semantic mappings among these other data stores.

However, it is also important to be able to not only harvest data integration models defining data flow mappings among data stores, but instead it can be necessary to model and/or specify mappings defined only within MIMM. This can be important if:

- There are not sources of metadata which can be harvested (e.g., raw coded specifies the mapping., but one must model this data flow to complete stitching and lineage analysis.
- One is attempting to model a new data flow process with *data mapping specifications* for planning and documentation purposes.
- One wishes to create full *data mapping designs* from these data mapping specifications to produce true data integration processes in an ETL or ELT tool environment.

In this document, mappings will be created both from scratch in the repository as well as from Excel spreadsheet definitions supporting all of the above use cases.

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 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 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® 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 Data Map Modeling

Some data flow processes are not harvestable using the bridges provided as a part of the Meta Integration Model Bridge (MIMB) tool suite. Of course, if these processed are not modeled in MIMM, it will leave gaps in the lineage and impact analysis answers and provide an incomplete picture of the physical architecture of your systems.

In order to address these gaps and produce proper lineage and impact results, MIMM has a Data Mapping Specification editing and management toolset. Data mapping specifications are essentially simply high-level logical (or notional) definitions of the way data "flows" from some number of source models into elements of a target model. These mappings are specified using a simple web based drag and drop type mapping specification editor and are defined using descriptive text and one may also define pseudo operations using an operation editor.

In this chapter we will learn how to take advantage of these tools and capabilities.

This is a continuation of the exercises in Part I (Metadata Management) of the Metadata Management Tutorial. Thus, you should first complete that tutorial before proceeding. If you wish to start directly from this point, without completing the first tutorial, delete or rename the MetaIntegration folder in the repository and execute the script entitled Create Tutorial Level 6 (Completed Part I and Part II) to produce the complete tutorial to that point.

2.1 Using an existing data mapping specification

The data mapping specifications form a content within MIMM which defines logical mappings between source and target data stores. The data stores themselves are modeled separately as model contents within MIMM but external to the data mapping specifications, as shown in this diagram:

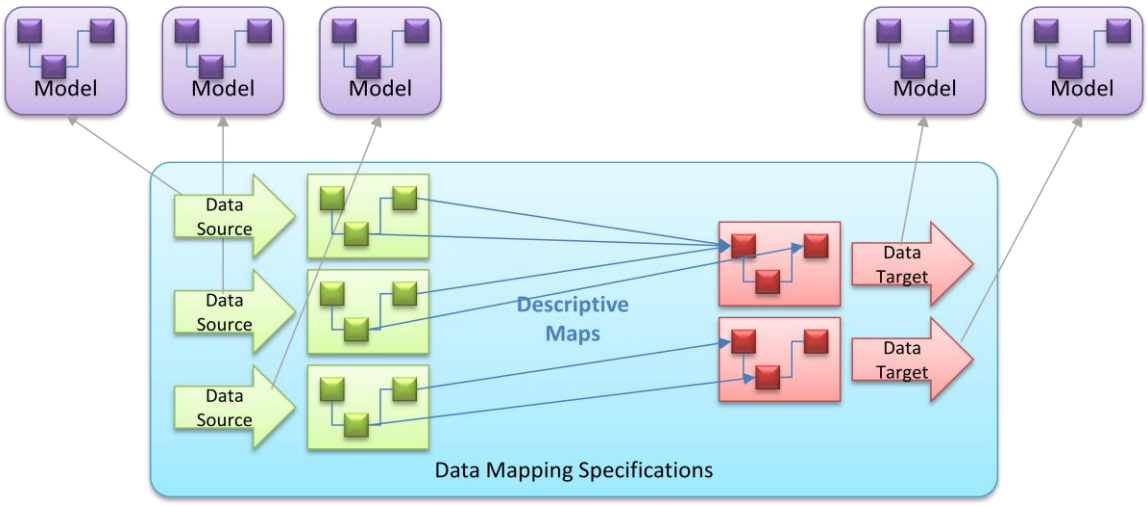


Figure 1 - Mapping Specification Construction

Each data store model content is referenced as either a source, a target, or can be both a source and target. One may then drag and drop objects (e.g., schemas, tables or columns) in a data source onto objects in a data target, thus creating *maps*. These maps may then be documented in *descriptions* or with more explicit *operations*.

Now, we have already completed this activity in the Metadata Management Tutorial. In that case, we saw the example of Adjustments to Staging.

So, let's open that example:

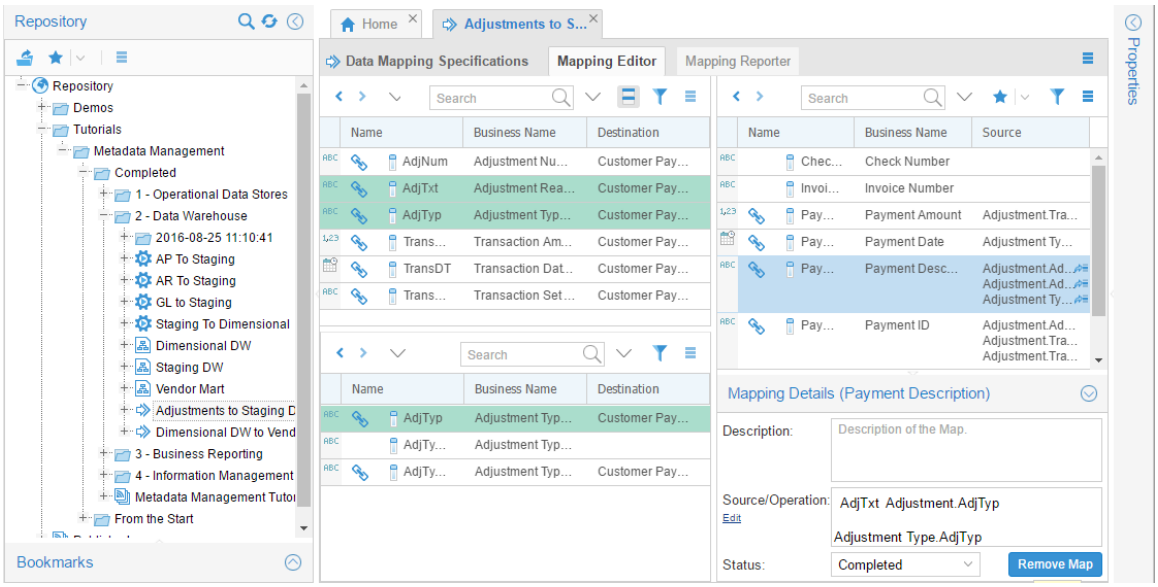


Figure 2 - Existing data mapping specification

Now, let us create a data mapping design from thispecification.

A bit of expanation is in order here. As stated before and as we saw in the Tutorial, the [dataMapping Specification] has target associated links from souce fields, but is not explicit in terms of joins or true transformations. The process of creating a data mapping design is to:

- Identify pipelines in the process, generally one per target table per join requirement
- Compute join conditions based upon existing relationship definitions in the source and targets schemas
- Suggest filtering based upon view definitions

In addition, one may request that aliases be created.

2.1.1 Generating the data mapping designs

So, right-click on the data mapping specification and select Generate Data Mpping Design:

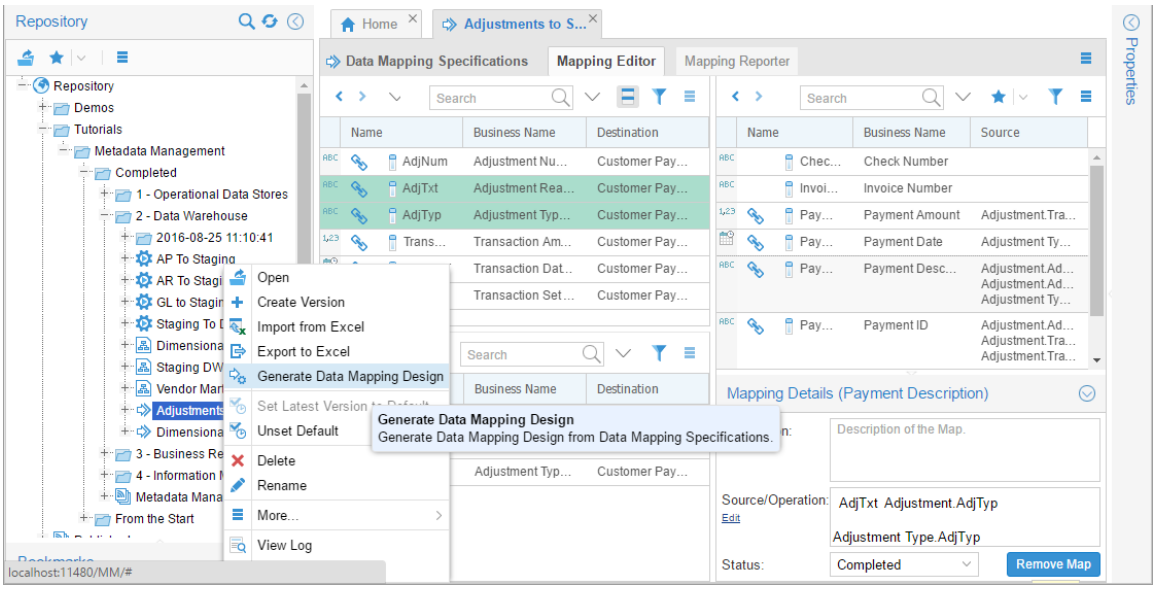


Figure 3 - Generate a data mapping design

Then specify the folder location where each design will be placed (again, one per pipeline) and whether to create aliases:

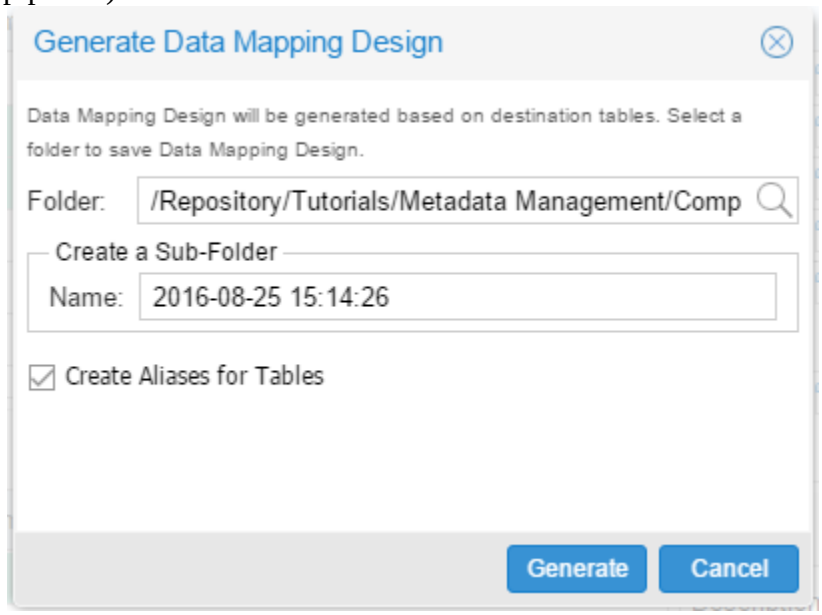


Figure 4 - Options when generating a data mapping design

Click Generate:

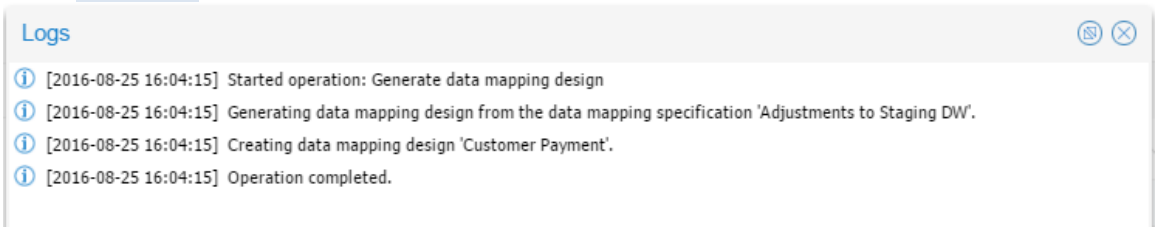


Figure 5 - Generation log

and note the folder with the individual data mapping designs:

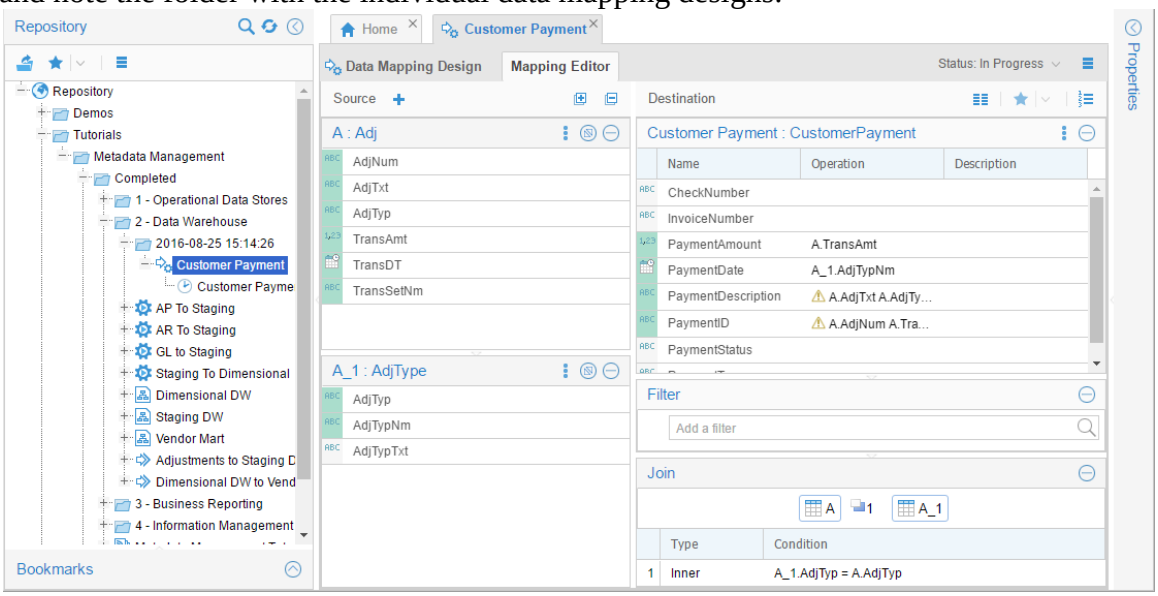


Figure 6 - Generated data mapping design

2.1.2 Completing the data mapping design

2.1.3 Exporting to [DITool]

One may now export to [DITool] using the export bridge:

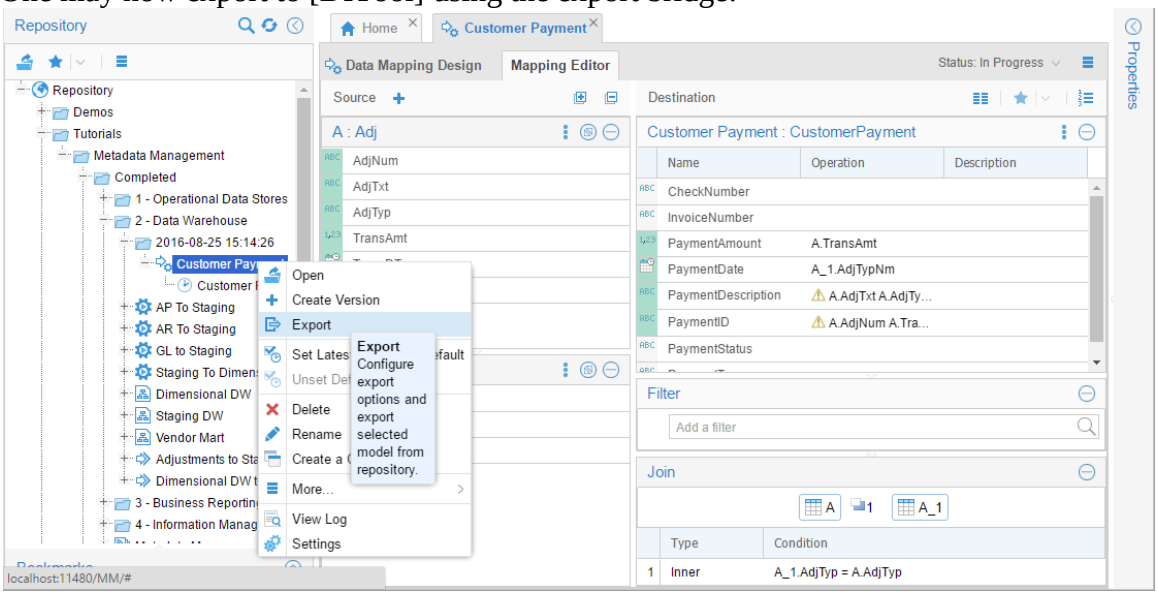


Figure 7 - Exporting to [DITool]

Selecting the appropriate bridge:

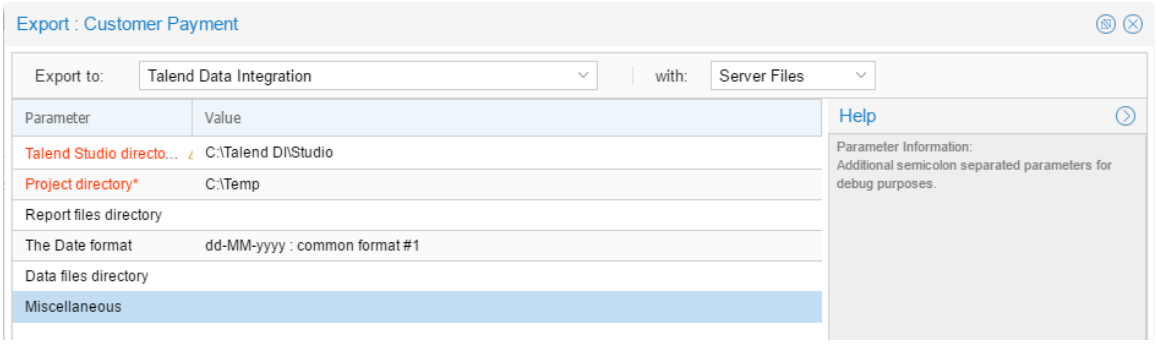


Figure 8 - Specifying the export bridge parameters

Then click on Export:

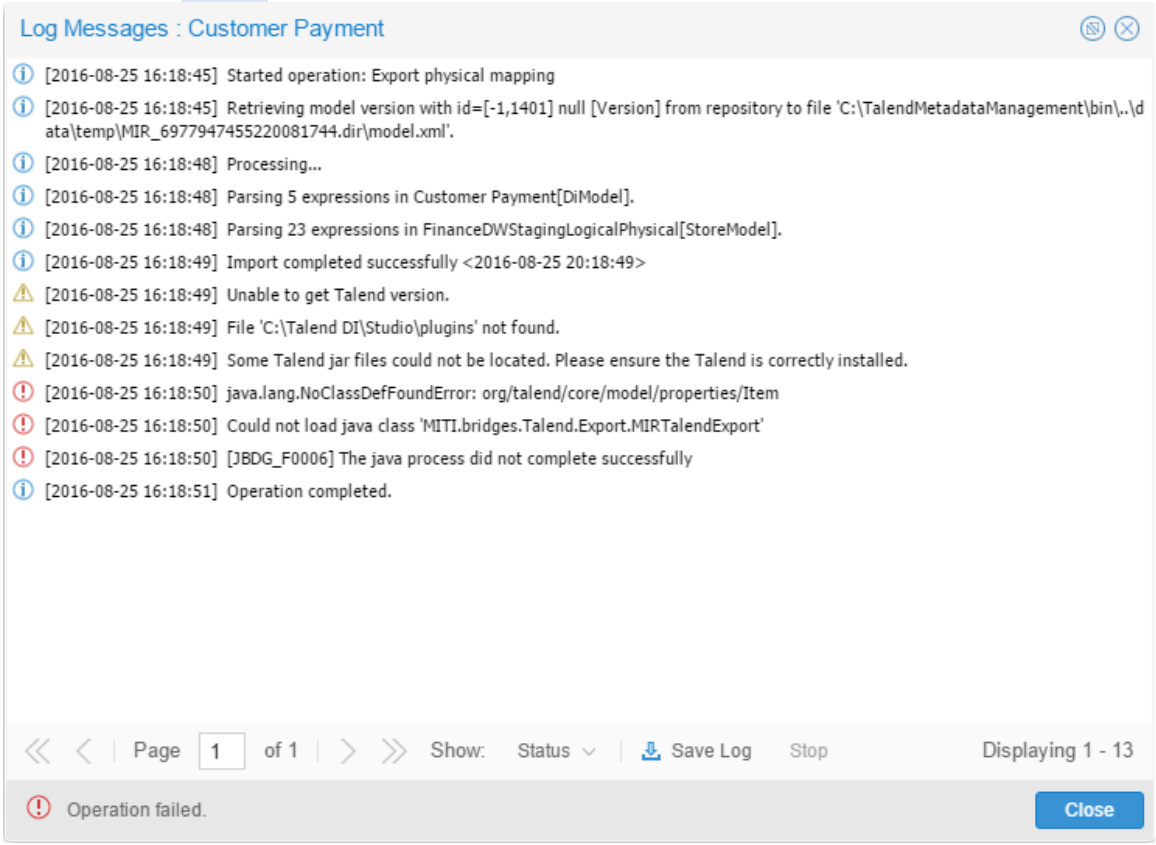


Figure 9 - Export log

And the result may be imported into [DITool]

2.2 Constructing a data mapping specification

Navigate to the repository path:
/Meta Integration/Tutorials/Metadata Management/Finance System/b - Data Warehouse
And right-click on the folder and select New➔ Data Mapping Specification:

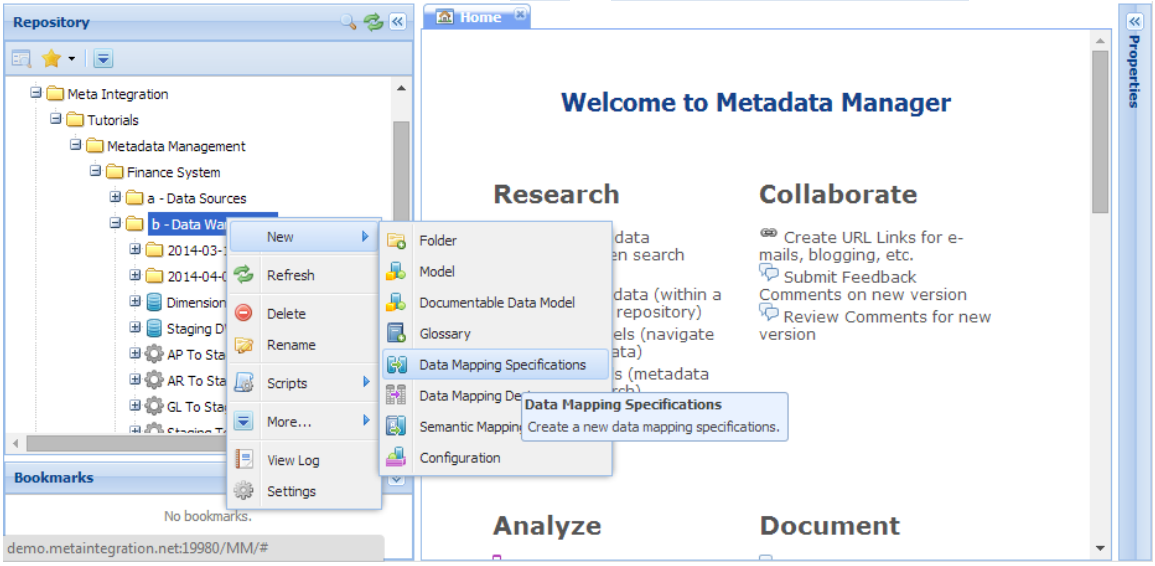


Figure 10 - Create new data mapping specifications

Name the data mapping specifications as "Web Contact to Staging DW" and click on the Create button:

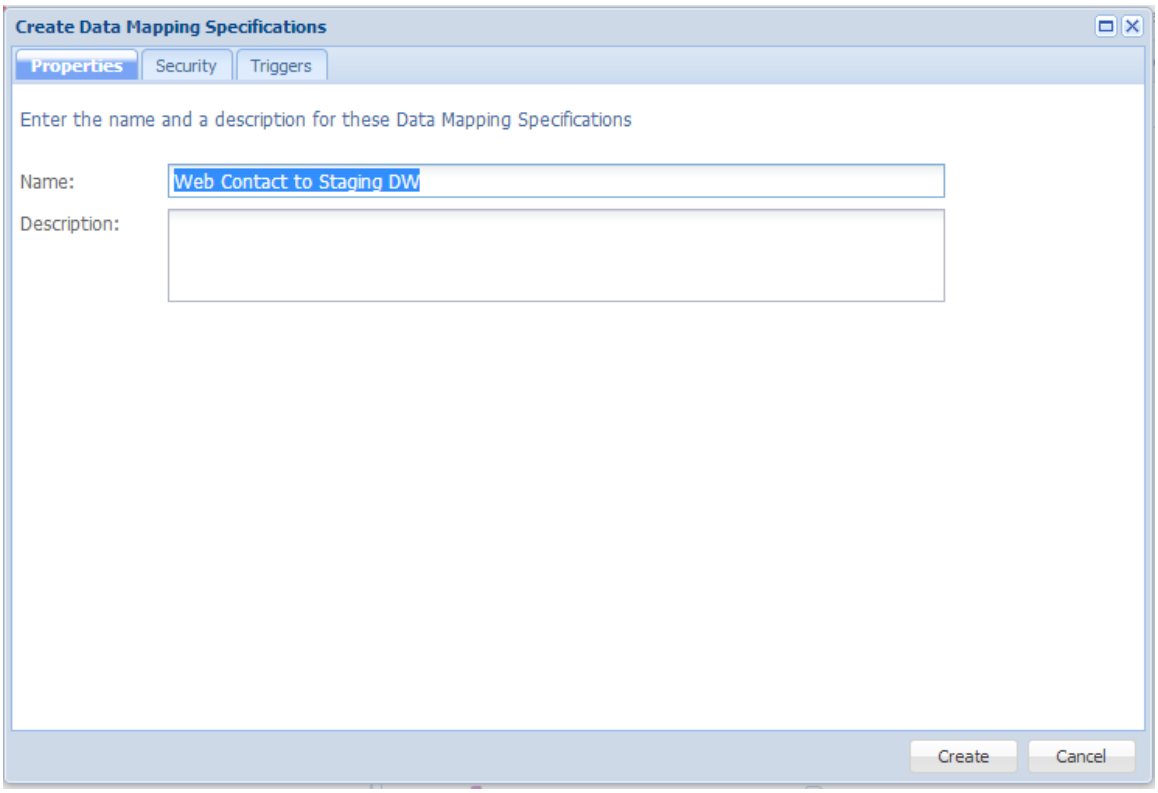


Figure 11 - Naming the data mapping specifications

You now have a data mapping specification opened in the work panel. We begin by dragging the source model into the source side of the data mapping specifications:

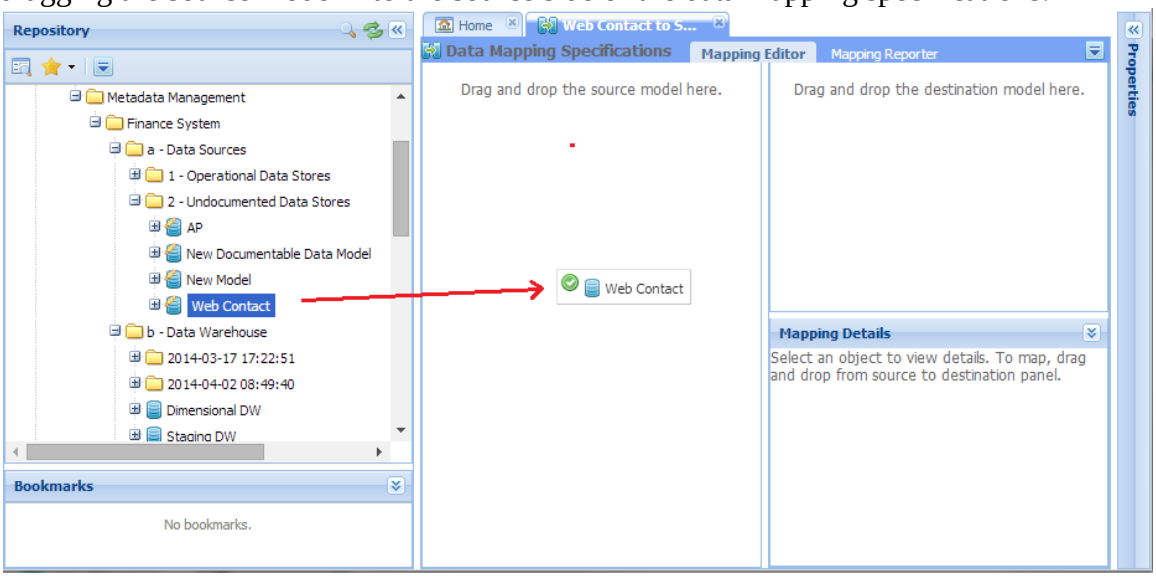


Figure 12 - Drag and drop source model into the data mapping specifications

And now drag the source model into the source side of the data mapping specifications:

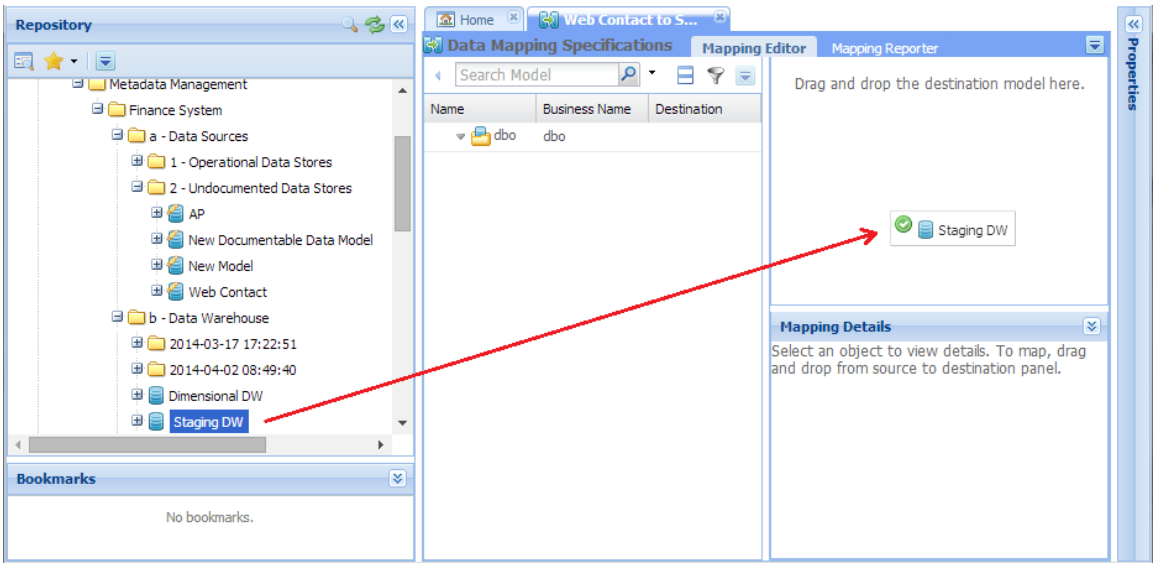


Figure 13 - Drag and drop target model into the data mapping specifications

Now, hide the Repository Panel and begin mappings sources to targets.

2.3 Data mapping specification workflow

Before we do so, however, it would be good to discuss mapping project management. In particular, when defining a task for mapping, it is often important to first define a scope for the mapping effort. In general, this scope is simply a specification of which source and target elements should be included in the resulting mapping.

Secondly, one would like to track progress as the effort progresses. It could take some time to collect the proper information to document and certify all of the mapping definitions (and potentially operations). Thus, one would like to track the status of each mapping and report on this.

MIMM provides the basic features to help manage a data mapping specification workflow process, as show below:

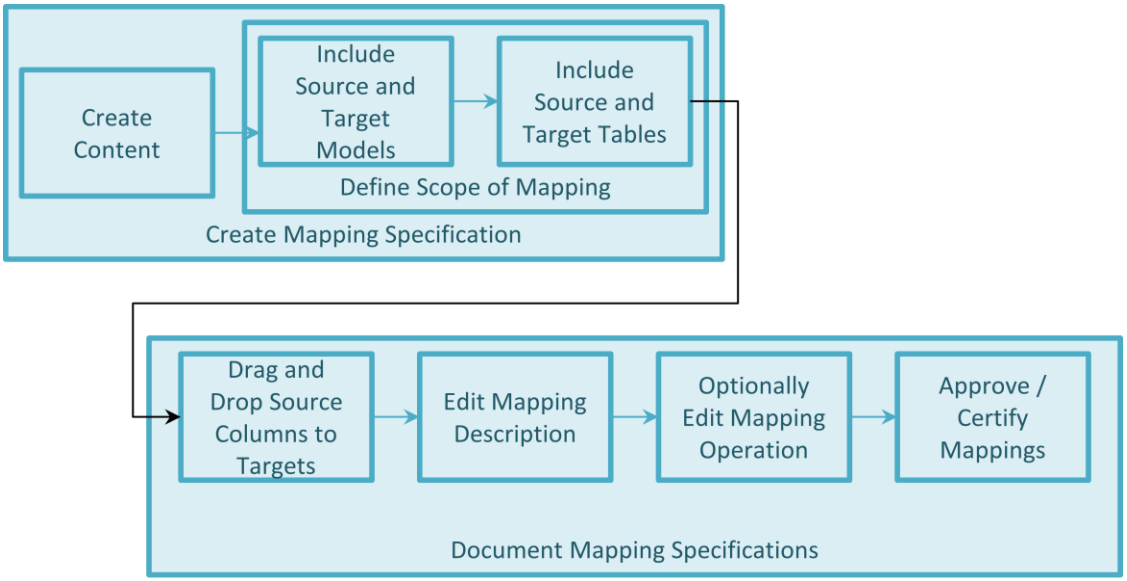


Figure 14 - Data Mapping specification workflow process

2.3.1 Define mapping scope

Let's begin with the example here. Expand the dbo schema in the source side, right-click on the USR (User) table and select Exclude:

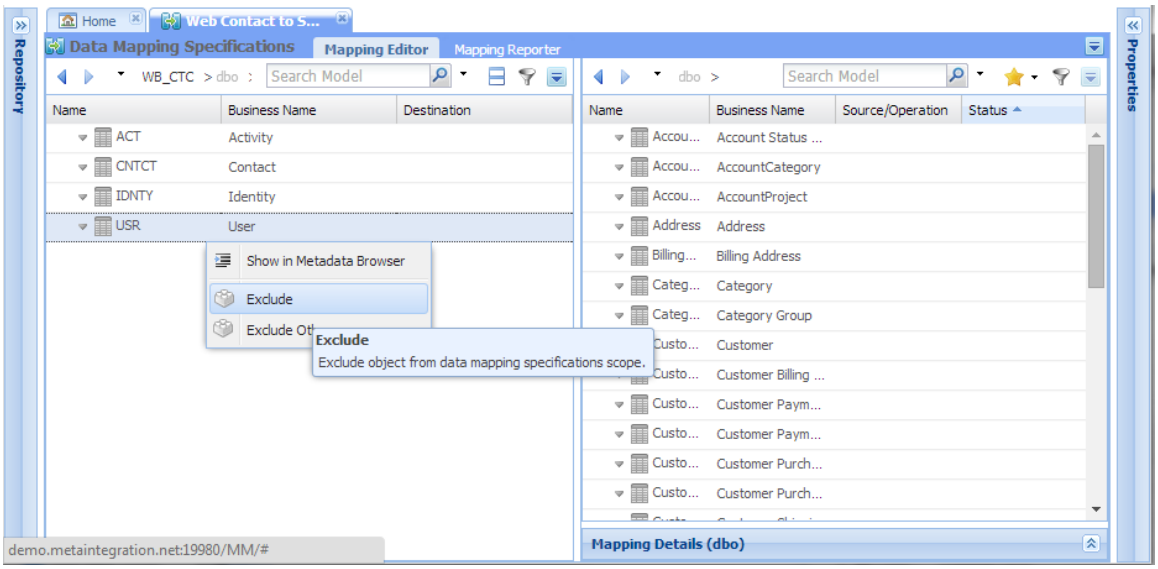


Figure 15 - Exclude the USR table

Note, the name of the USR table is now grayed out, indicating that only the ACT (Activity), CNTCT (Contact) and IDNTY (Identity) tables are to be included in the scope.

Now, for the target side, the requirement is to map only to the following three tables:

- Address
- Customer
- POC

To do so, first right-click on the Address table and select Exclude Others:

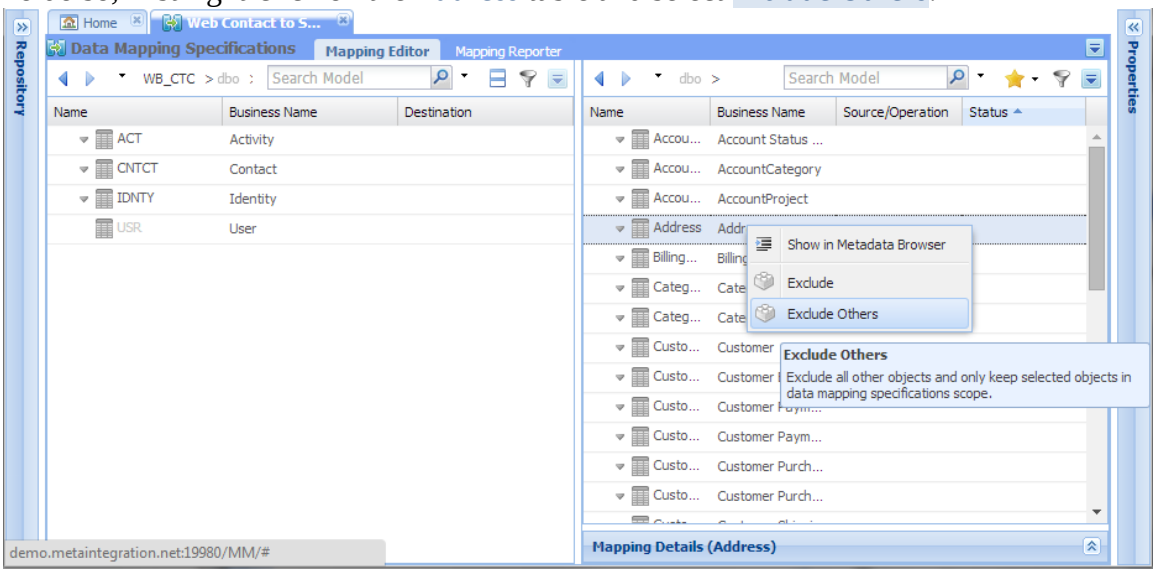


Figure 16 - Exclude Others

Note, MIMM informs the user that this will exclude all other tables. Accept.

Now, to include the other two tables in the requirements, click on Customer and control-click (hold the CTRL and click) the POC tables. Then right-click on one of them and select Include:

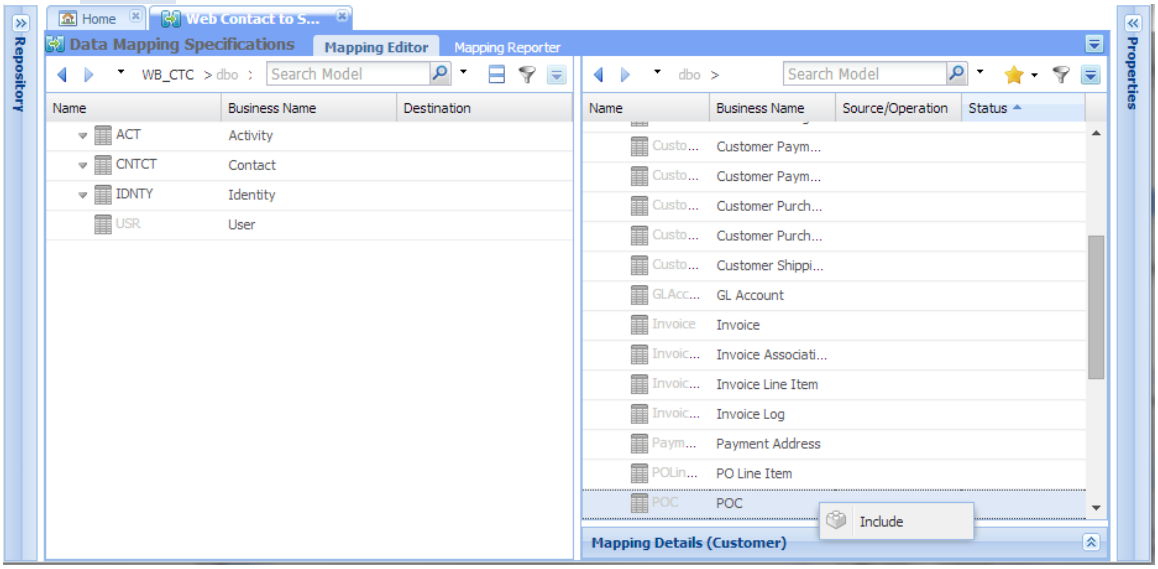


Figure 17 - Include two more tables

As one can see, multi-select (with control-click and shift-click) is enabled in the source and target sides of a data mapping specification.

Thus, we now have only three source and three target tables in our scope. To simplify the presentation, then, use the Filter action icon:

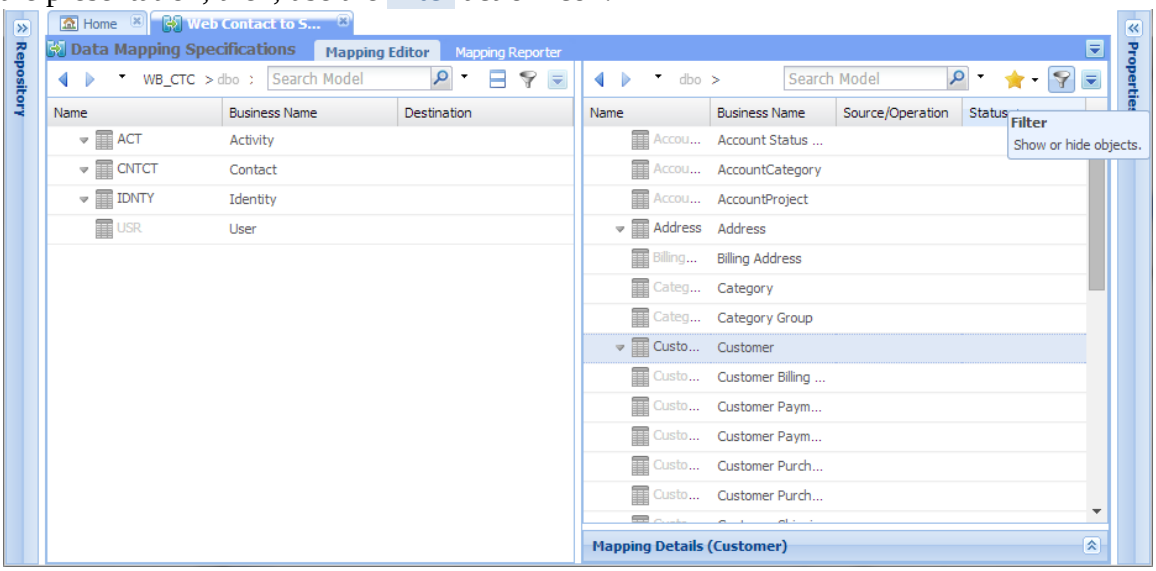


Figure 18 - Filter target objects

Select the Show Unmapped checkbox and click on the Apply Filter button:

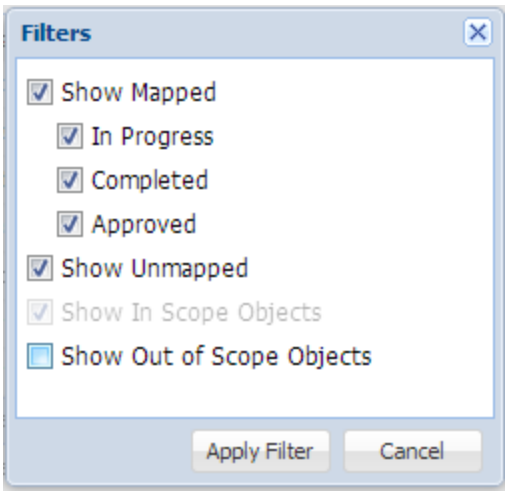


Figure 19 - Filter for in scope objects

And we have a more manageable list:

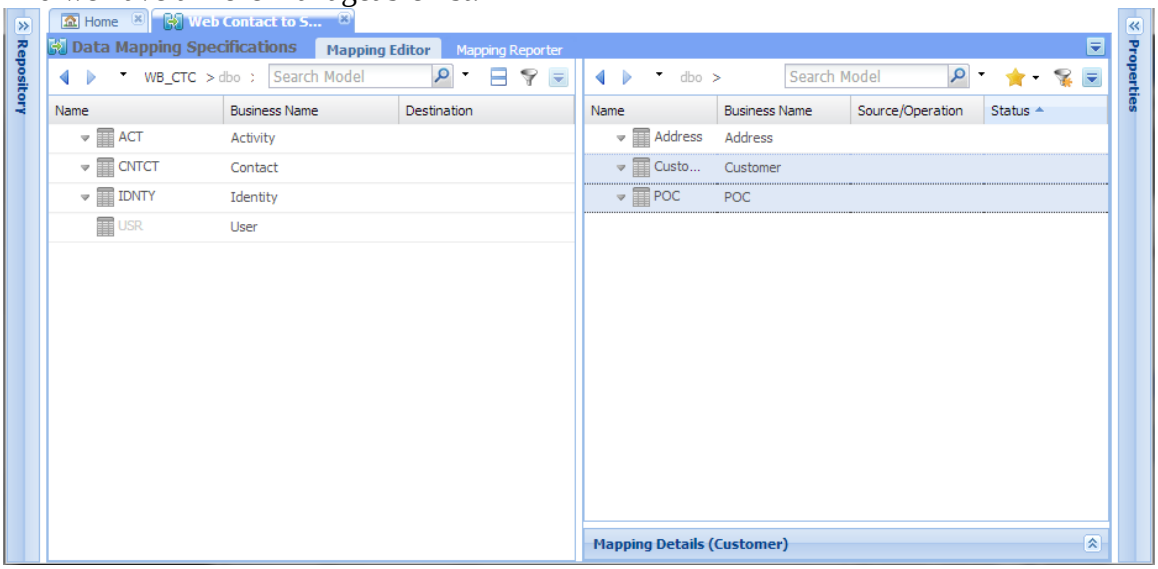


Figure 20 - Filter result

2.3.2 Creating a map

So, let's see if we are lucky and simple table level maps will work. Drag the Contact table on to the Customer table:

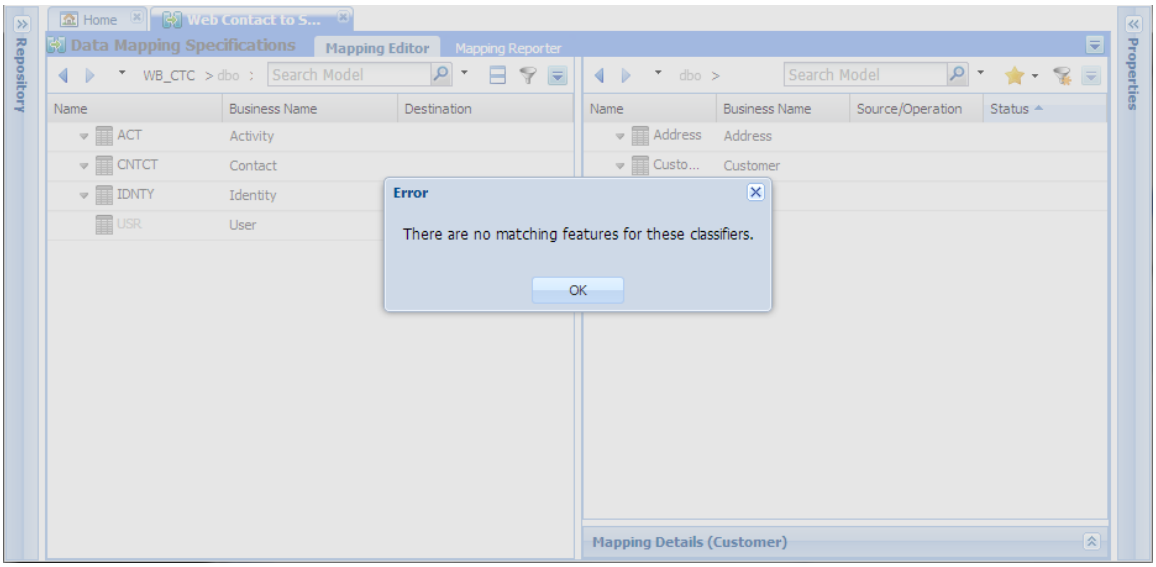


Figure 21 - Table level mapping attempt

Note, none of the columns (generically referred to as *features*) in the target table match any names in the source table. Thus, we will have to map these columns individually. Expand both the Contact table on the source side and the Customer table on the target side. Now, drag the User Name column to the Customer Name column, creating a map:

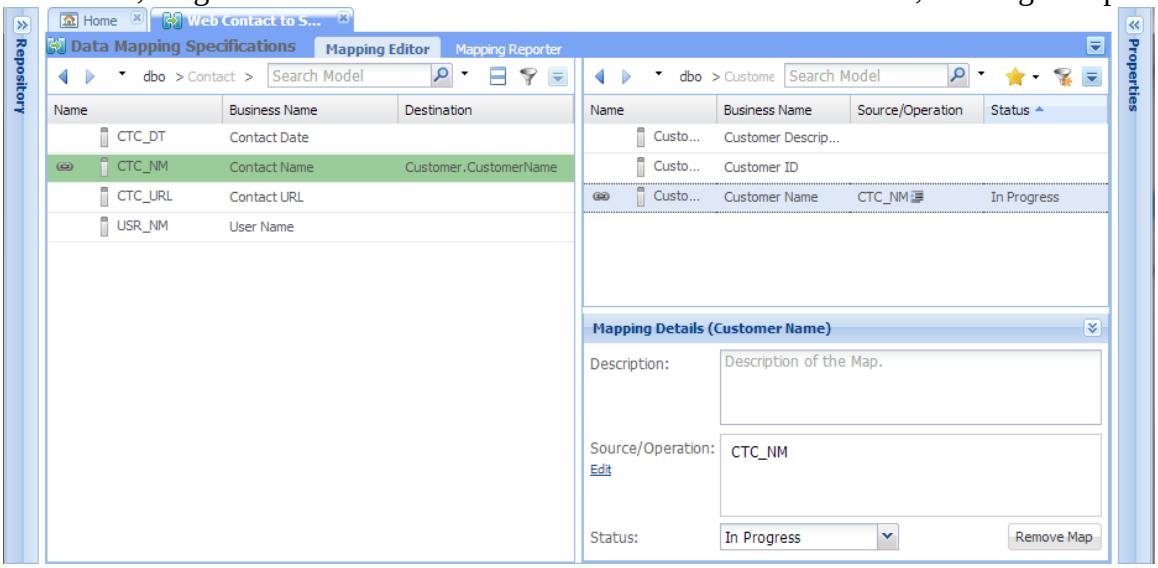


Figure 22 - Map from Contact name to Customer Name

Column level maps are seen as target defined, meaning that each column mapped in the target defines an individual map which may have references to one or more source column. In this way, this map just created is referred to as the Customer.Customer Name map, and any description or operation is thus identified.

Note, the operation is already populated with the source column physical name (CTC_NM). One can simply drag and drop into the Operation to include additional columns. Also, one may populate the text of the Description by drag and drop. Hence, to

place the text "Use the Contact Name where available" in the Description, one may type "Use the", then drag the source column in, and then type "where available":

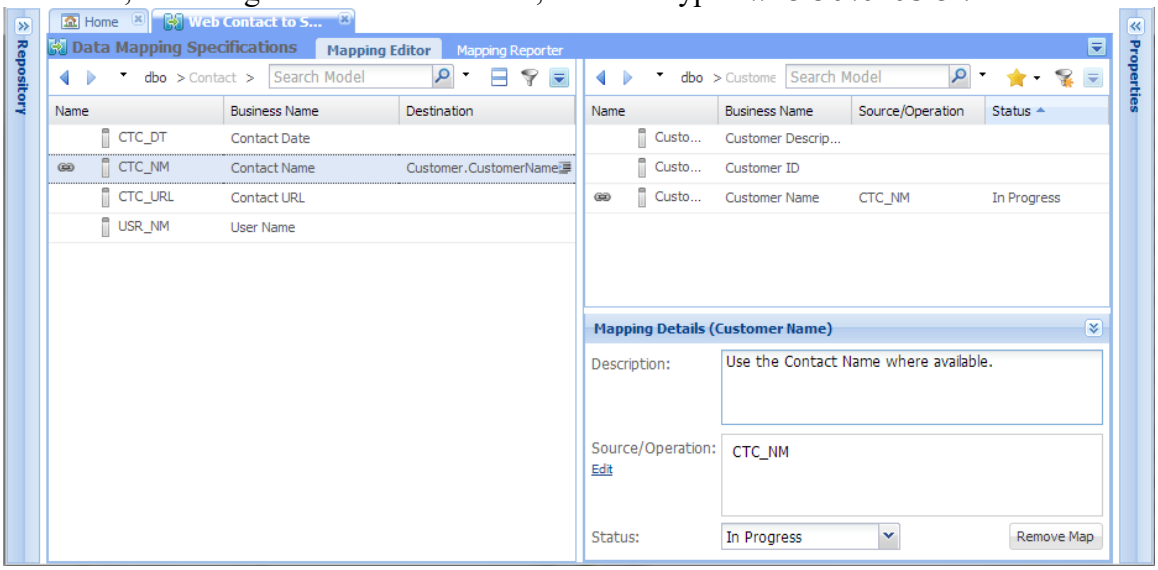


Figure 23 - Entering a map description

Now that we have completed this map, change the status to Completed:

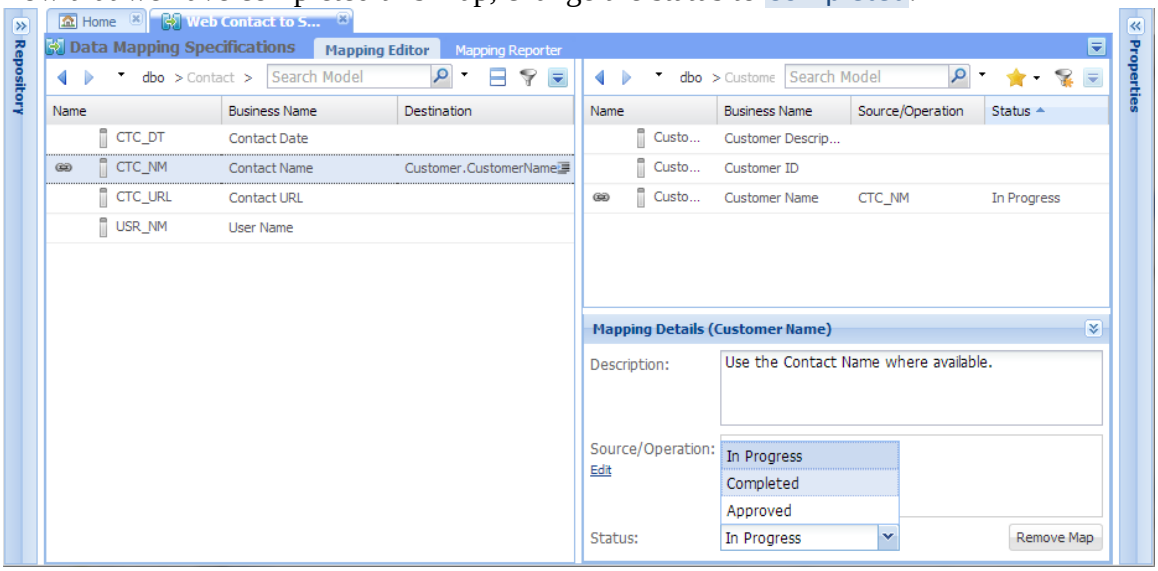


Figure 24 - Setting the status to Completed

With that map completed, drag the following three source columns into the Customer ID target column:

- Contact Date
- Contact URL
- User Name

In this case, the target column is to be composed from the three source columns identified. Enter the text "Based upon Contact Date, Contact URL and User Name" in the Description field, either through typing only or also using the drag and drop to populate the text:

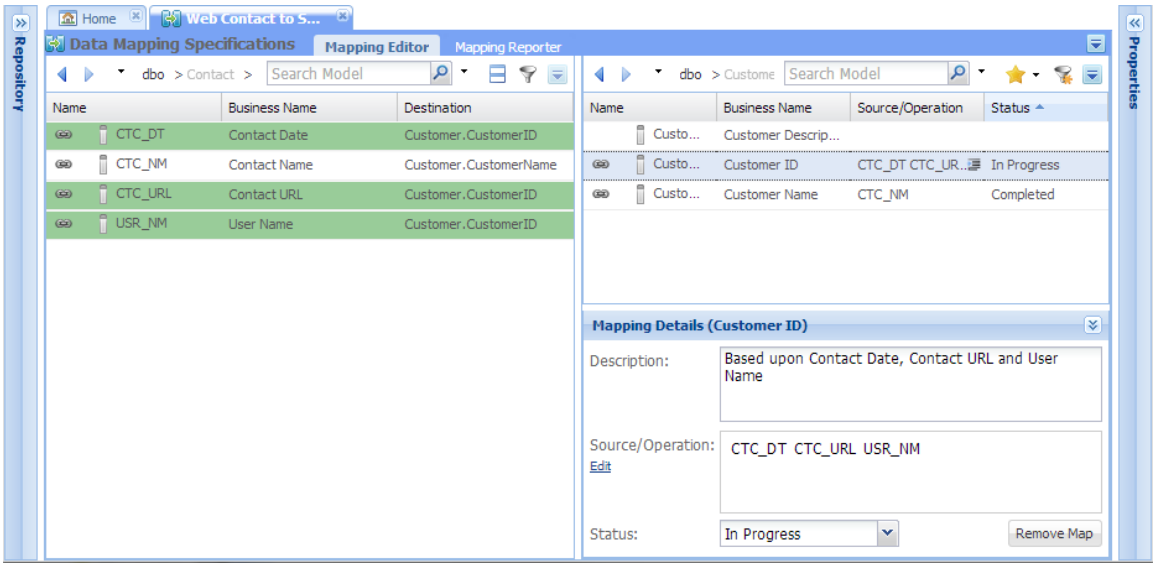


Figure 25 - Column map to Customer ID

With the mapping defined in this way, lineage will be shown but no meaningful operation will be presented. Instead, the description is where the details for the map are presented. Please also set the Status for the map to Completed.

Finally, according to the requirement for the last column map to the Customer Description, the column map should be:

> Text constructed from ACT_CTXT and the decoded value of ACT_TYP depending upon the ACT_TYP:
> - if ACT_TYP < 10 then it is a simple action with little information about customer type and so should not be included here
> Otherwise, the decoded value of ACT_TYP should be included in the description text.

As one can see from the above description, the source data for this map is from the Activity table. Hence, we will need to drag and drop from columns in the Activity table, and thus must have it opened on the source side. One could simply navigate to that table and perform the drag and drop. However, one could also take advantage of the Split feature and expand the Activity table in another panel on the source side:

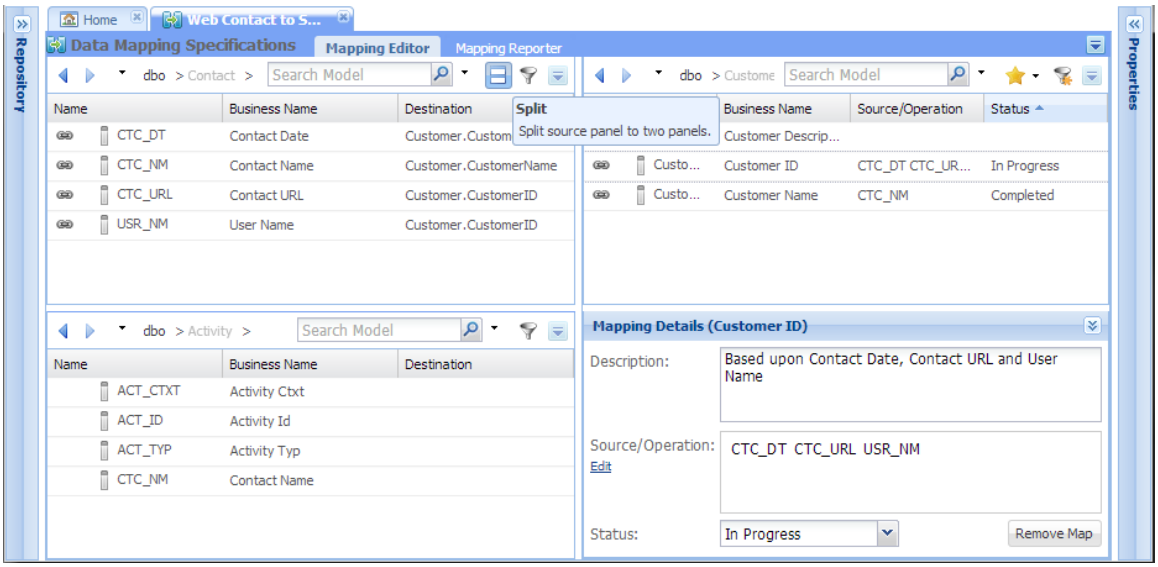


Figure 26 - Source side split

Now, one may drag and drop the two columns into the Customer Description column on the target side and populate the Description:

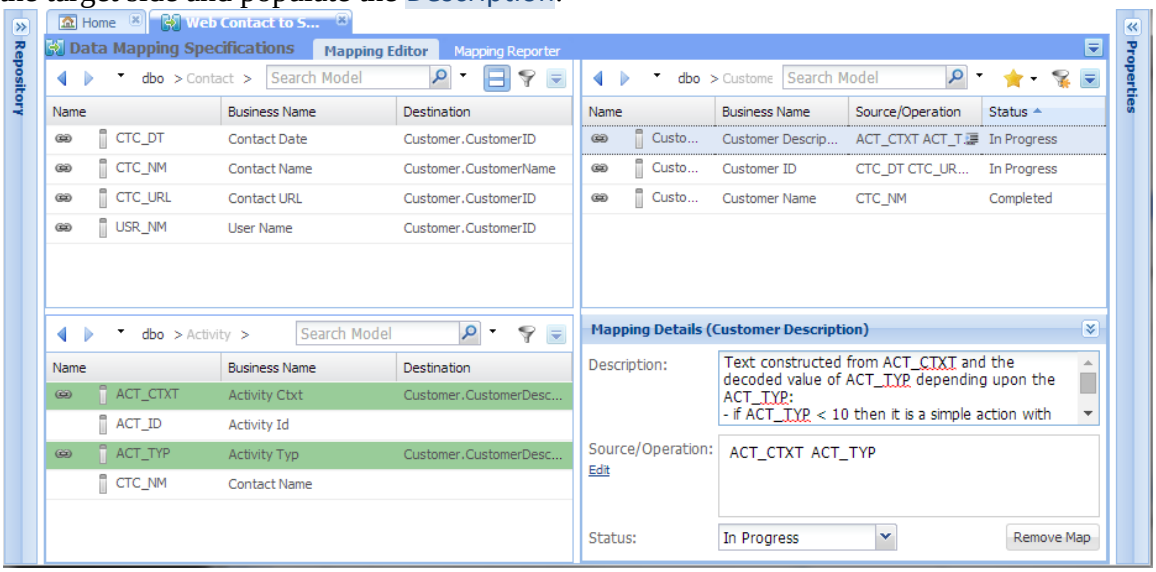


Figure 27 - Customer Description column map

Please also set the Status for the map to Completed.

Click on the Mapping Reporter tab and note the maps as defined is presented:

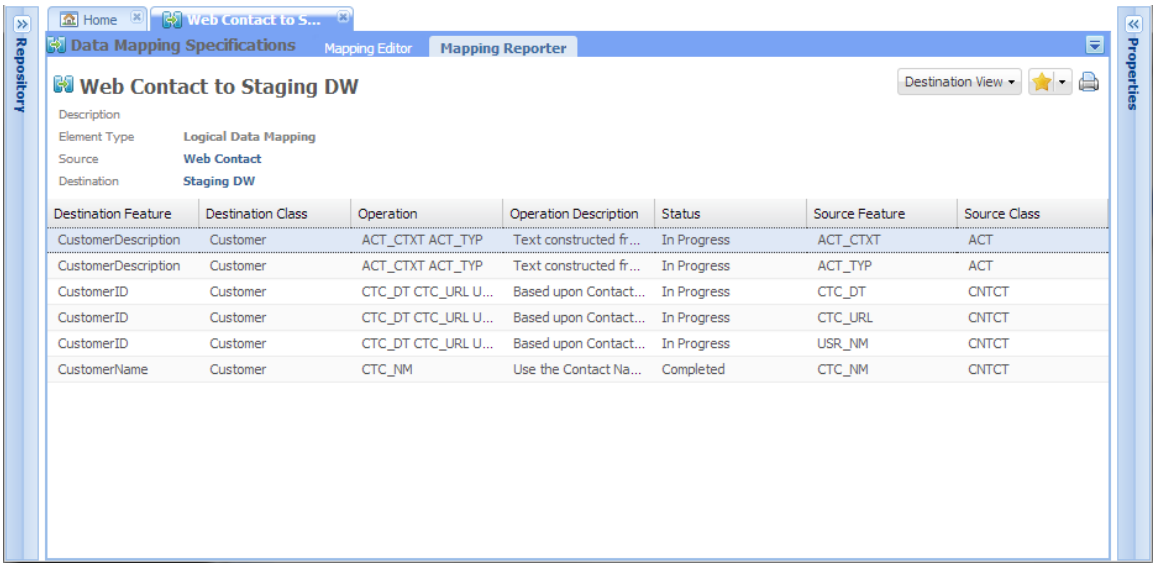


Figure 28 - Mapping Report

2.3.3 Review and approve maps

It is now time to review our work. Click on the Actions icon for the data mapping specifications as a whole (upper right) and select View Column Status:

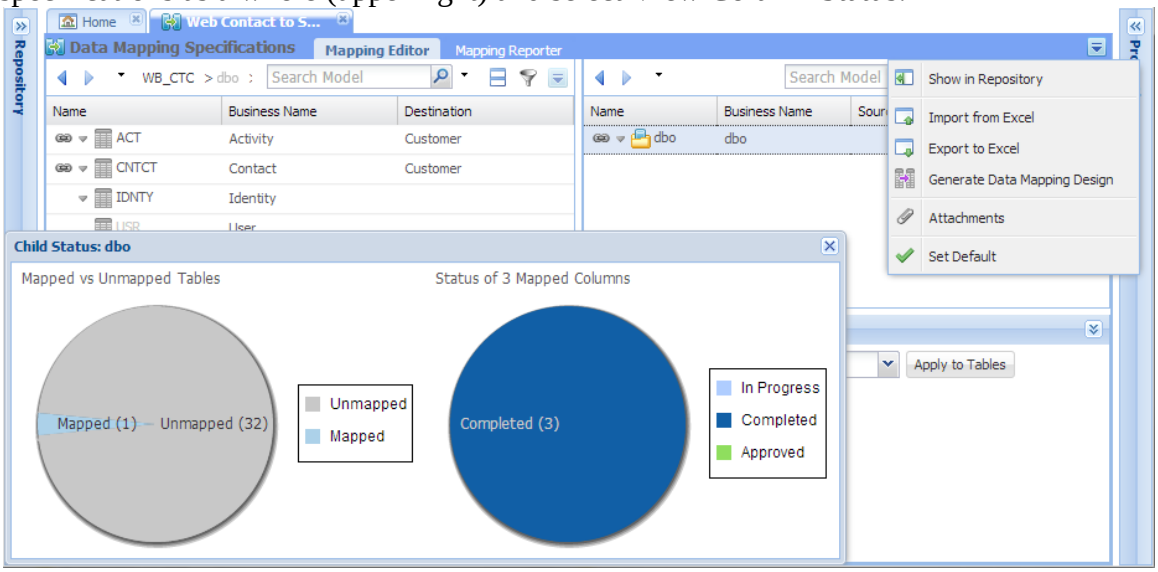


Figure 29 - View Column Status

Here we see that 1 out of 3 total tables (in scope) are mapped and that of the 3 mapped columns all three are completed but not are approved.

Now, someone should go through and review these maps. Based upon that review, the maps to the Customer Description and Customer ID columns within the Customer table are approved, so we should change the Status of those to Approved. Doing so updates the report as follows:

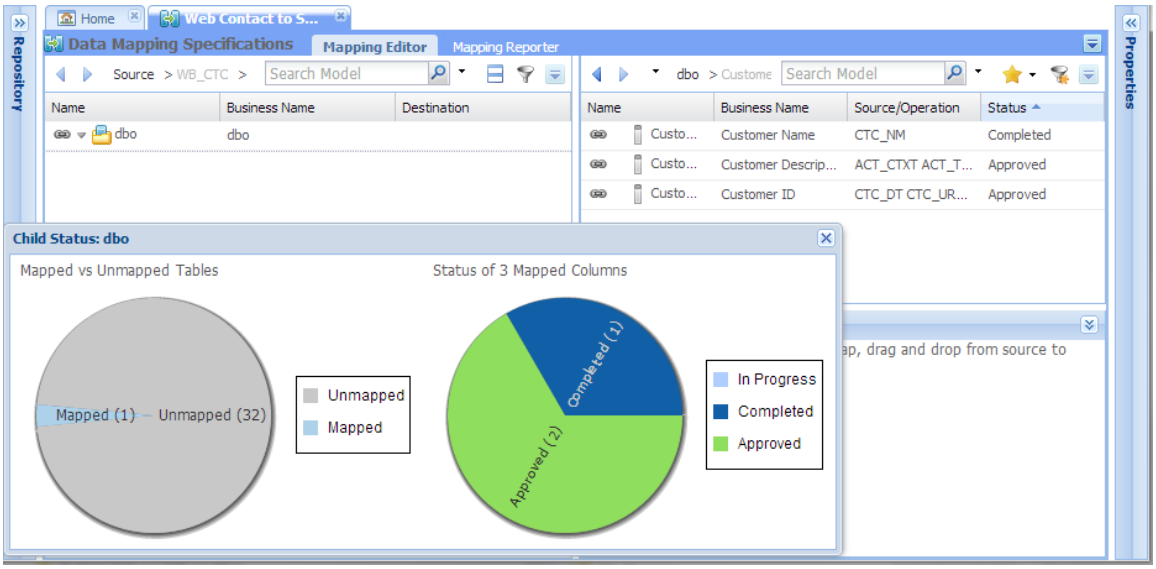


Figure 30 - View Column Status

However, the map for the Customer Name column is not complete. In fact, the requirements can be stated as:

If Name from Identity is empty then use Contact Name

Please enter that text into the Description. Then, drag and drop the Name column in the Identity table into Customer Name as well.

Now, to complete the maps for the other two target columns, please refer to the following table, which is a simply download from the Mapping Reporter. Once done, you have completed this exercise.

Destination Feature	Destination Class	Operation	Operation Description	Status
AddressDescription	Address	ACT_CTXT||ACT_TYP	Description based upon Activity Cbxt and Activity Typ	Completed
AddressDescription	Address	ACT_CTXT||ACT_TYP	Description based upon Activity Cbxt and Activity Typ	Completed
AddressID	Address	CTC_DTUSR_NMCTC_URL	Address ID is computed from Contact, DateContact URL and User Name	In Progress
AddressID	Address	CTC_DTUSR_NMCTC_URL	Address ID is computed from Contact, DateContact URL and User Name	In Progress
AddressID	Address	CTC_DTUSR_NMCTC_URL	Address ID is computed from Contact, DateContact URL and User Name	In Progress
AddressType	Address	"Web Contact"	Apply "Web Contact" as Address Type	Approved
City	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address andContact URL	In Progress
City	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address andContact URL	In Progress
City	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address andContact URL	In Progress
International	Address	ADDRBL_ADDRSP_ADDR		In Progress
International	Address	ADDRBL_ADDRSP_ADDR		In Progress
International	Address	ADDRBL_ADDRSP_ADDR		In Progress
State	Address	ADDR BL_ADDR SP_ADDR		In Progress
State	Address	ADDR BL_ADDR SP_ADDR		In Progress
State	Address	ADDR BL_ADDR SP_ADDR		In Progress
StreetAddress1	Address	ADDR BL_ADDR SP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
StreetAddress1	Address	ADDR BL_ADDR SP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
StreetAddress1	Address	ADDR BL_ADDR SP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
StreetAddress2	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
StreetAddress2	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
StreetAddress2	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
Zip	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
Zip	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
Zip	Address	ADDRBL_ADDRSP_ADDR	Extract State from Address, Billing Address and Shipping Address	In Progress
CustomerDescription	Customer	ACT_CTXTACT_TYP	Text constructed from ACT_CTXT and the decoded value of ACT_TYP depending upo...	Approved
CustomerDescription	Customer	ACT_CTXTACT_TYP	Text constructed from ACT_CTXT and the decoded value of ACT_TYP depending upo...	Approved
CustomerID	Customer	CTC_DTCTC_URLUSR_NM	Based upon Contact Date, Contact URL and User Name	In Progress
CustomerID	Customer	CTC_DTCTC_URLUSR_NM	Based upon Contact Date, Contact URL and User Name	In Progress
CustomerID	Customer	CTC_DTCTC_URLUSR_NM	Based upon Contact Date, Contact URL and User Name	In Progress
CustomerName	Customer	if (NM=="") then (	If Name from Identity is empty then use Contact Name	In Progress
POCDescription	POC	ACT_TYP": "||ACT_CTXT	Concatenation of Activity Typ and Activity Cbxt	Approved
POCDescription	POC	ACT_TYP": "||ACT_CTXT	Concatenation of Activity Typ and Activity Cbxt	Approved
POCEMail	POC	EMAIL		Approved
POCID	POC	USR_NMCTC_URLCTC_DT	Unique combination of User Name, Contact URL and Contact Date	Completed
POCID	POC	USR_NMCTC_URLCTC_DT	Unique combination of User Name, Contact URL and Contact Date	Completed
POCID	POC	USR_NMCTC_URLCTC_DT	Unique combination of User Name, Contact URL and Contact Date	Completed
POCName	POC	NM CTC_NM	If Name from Identity is empty then use Contact Name	In Progress
POCName	POC	NM CTC_NM	If Name from Identity is empty then use Contact Name	In Progress
POCPhone	POC	Phone_Number		Approved

Figure 31 - Mapping Reporter output

3 Data Integration Design

A data mapping design are true data integration process designs containing all the necessary data movement design details, such as lookups, filters, joins and transformation expressions. Thes data mapping designs are complete enough that they may be forward engineered into an Data Integegration (DI) / Extract, Transform and Load (ETL) / Extract, Load and Transform (ELT) design tools using the Meta Integration® Model Bridge (MIMB) export bridges for supported tools (see http://www.metaintegration.net/Products/MIMB/SupportedTools.html).

A data mapping design will reference one or more source models and a target model which have previously been harvested or documented.

One may create a data mapping design from scratch or derive one or more data mapping design(s) from a set of data mapping specifications.

Meta Integration® Metadata Management (MIMM) also provides mapping design project management tools which aide in tracking progress in data mapping design projects, including the ability to:

- Indicate mapping status
- Review project completion statistics
- Prioritize/Report by mapping status.

Now, used the More Actions action icon in the header for the data mapping specification and select Generate Data Mapping Design

3.1.1 Generating the data mapping designs

So, right-click on the data mapping specification and select Generate Data Mpping Design:

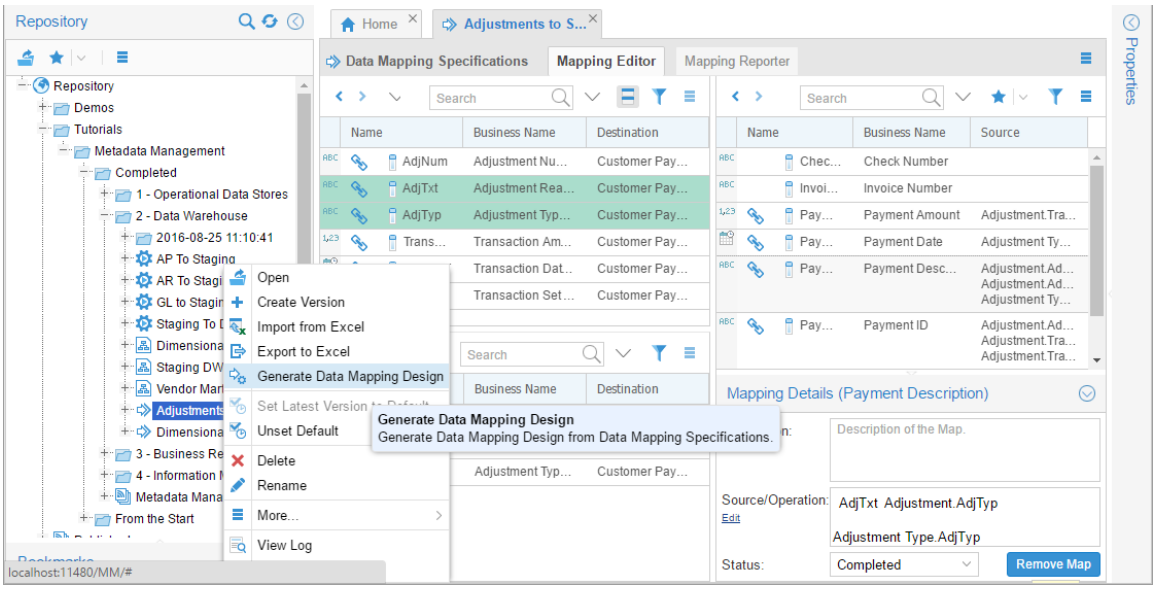


Figure 32 - Generate a data mapping design

Then specify the folder location where each design will be placed (again, one per pipeline) and whether to create aliases:

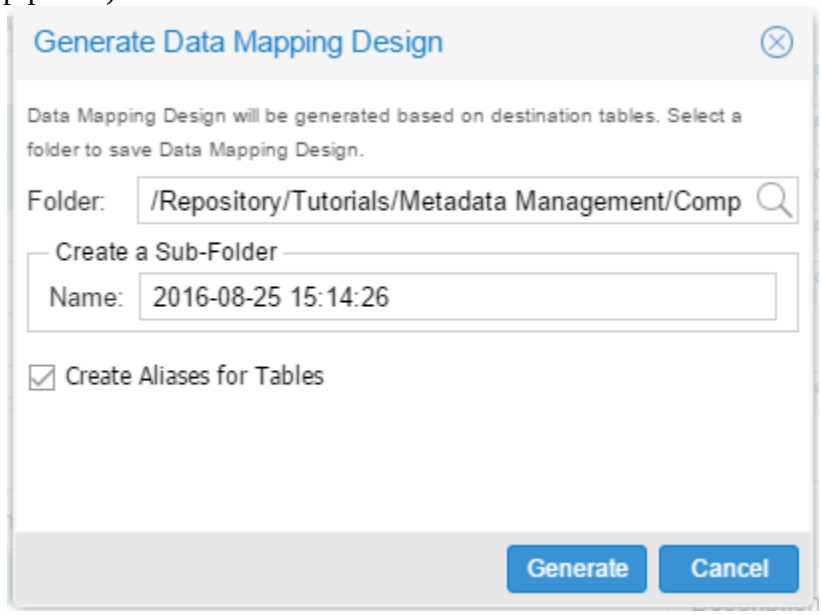


Figure 33 - Options when generating a data mapping design

Click Generate:

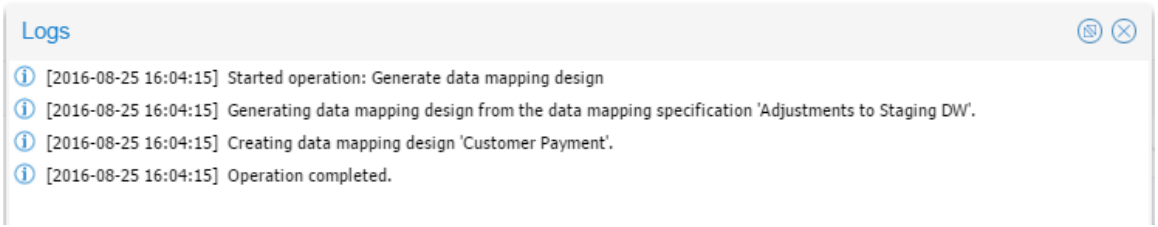


Figure 34 - Generation log

and note the folder with the individual data mapping designs:

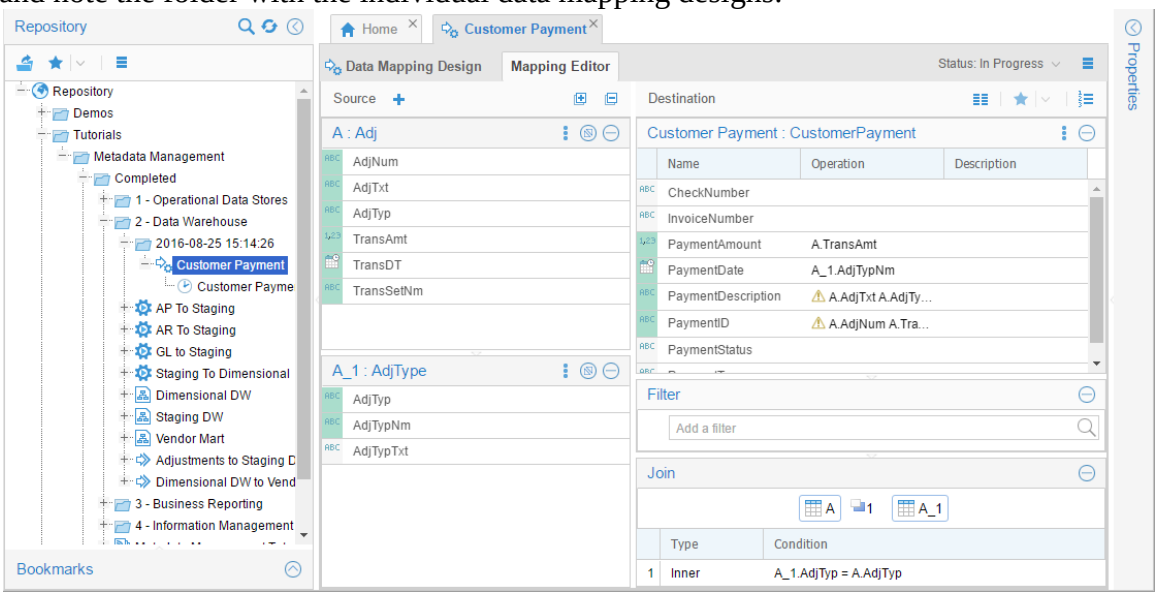


Figure 35 - Generated data mapping design

3.1.2 Completing the data mapping design

3.1.3 Exporting to [DITool]

One may now export to [DITool] using the export bridge:

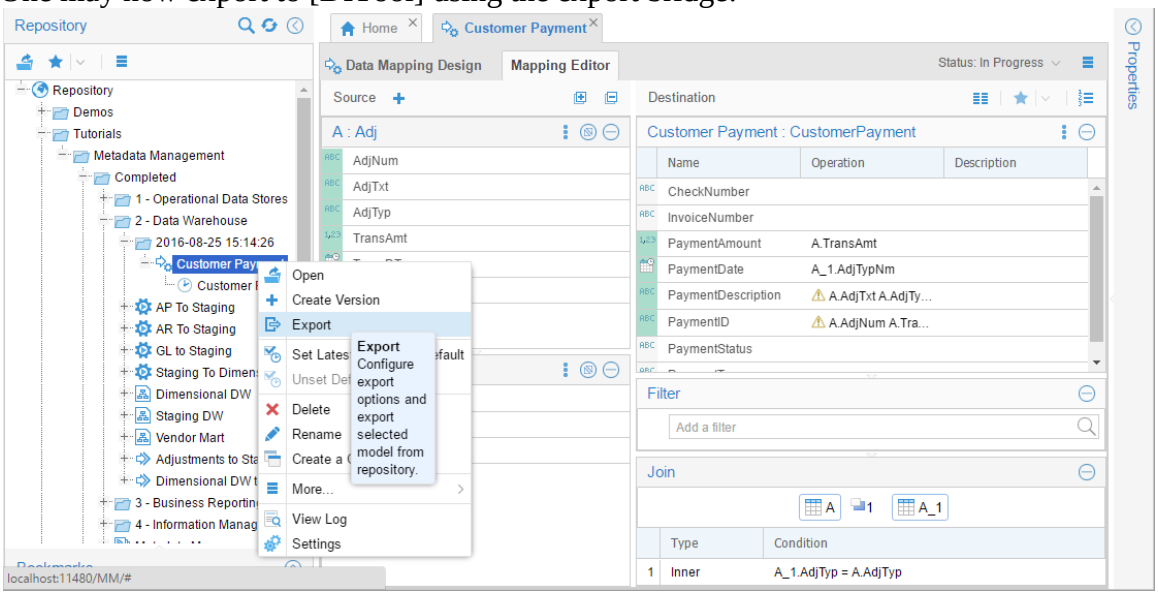


Figure 36 - Exporting to [DITool]

Selecting the appropriate bridge:

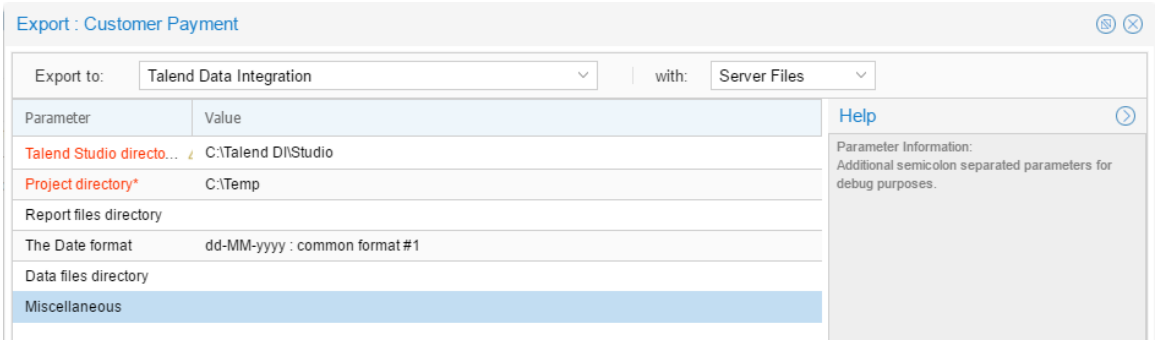


Figure 37 - Specifying the export bridge parameters

Then click on Export:

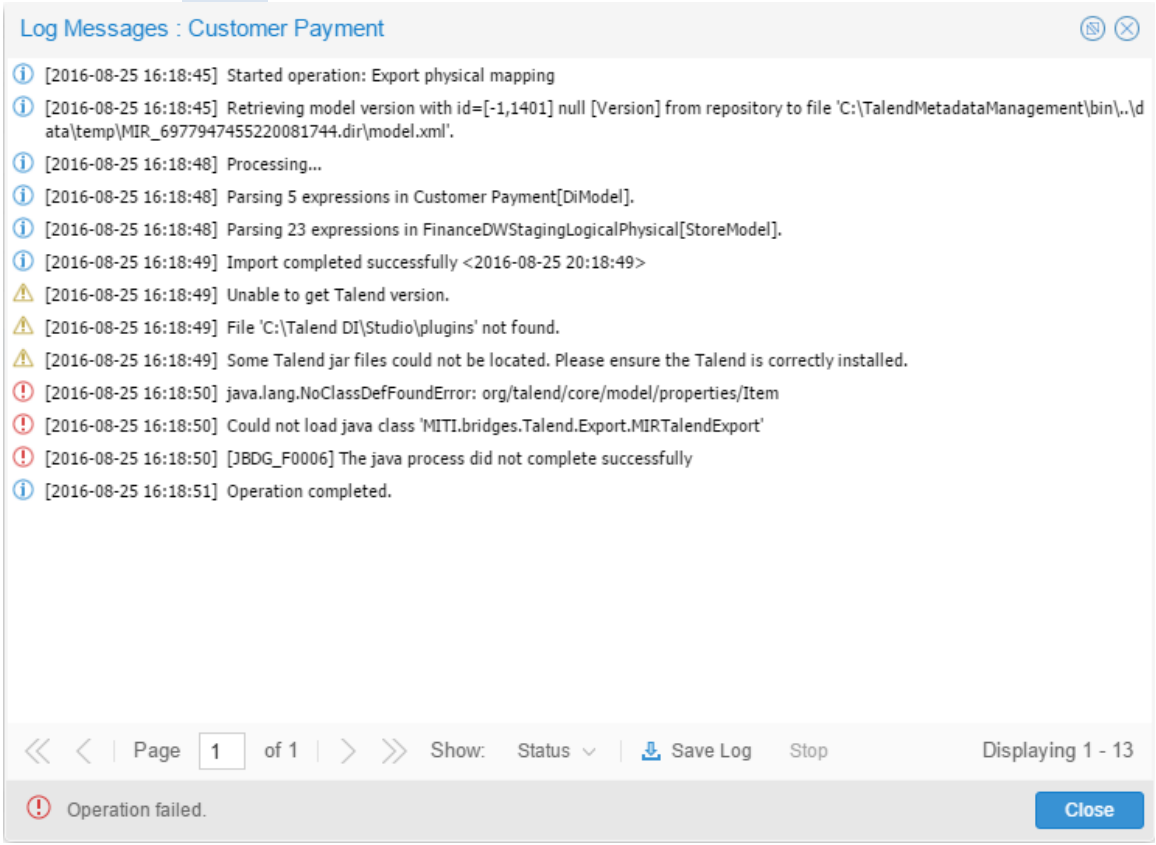
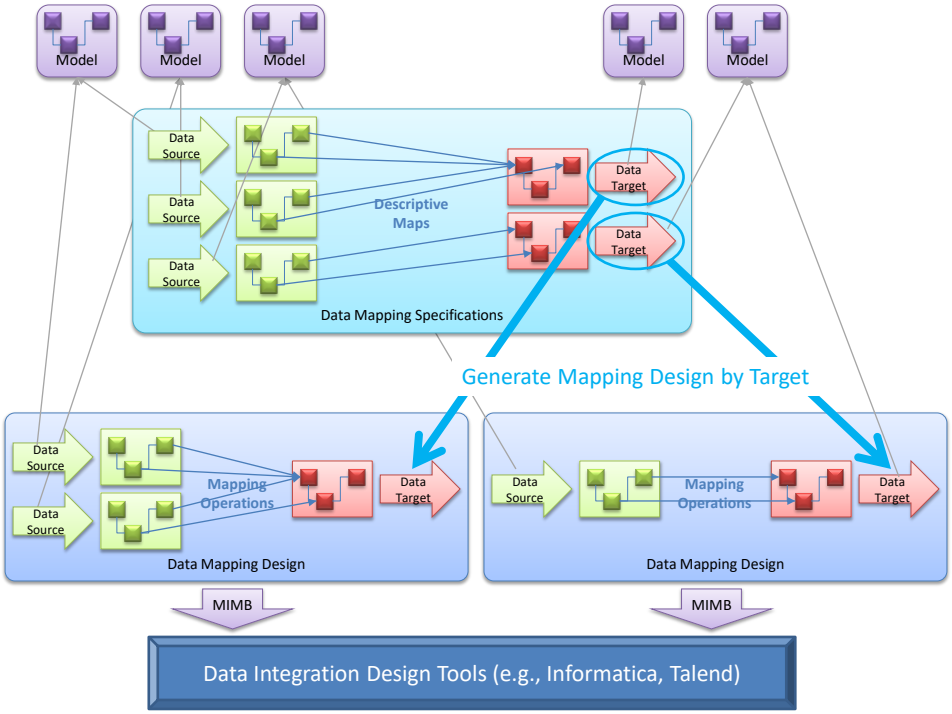


Figure 38 - Export log

And the result may be imported into [DITool]

3.2 Deriving Data Mapping Designs from Data Mapping Specifications



3.3 Data mapping design workflow