IDERA® | Database Development Solution Brief

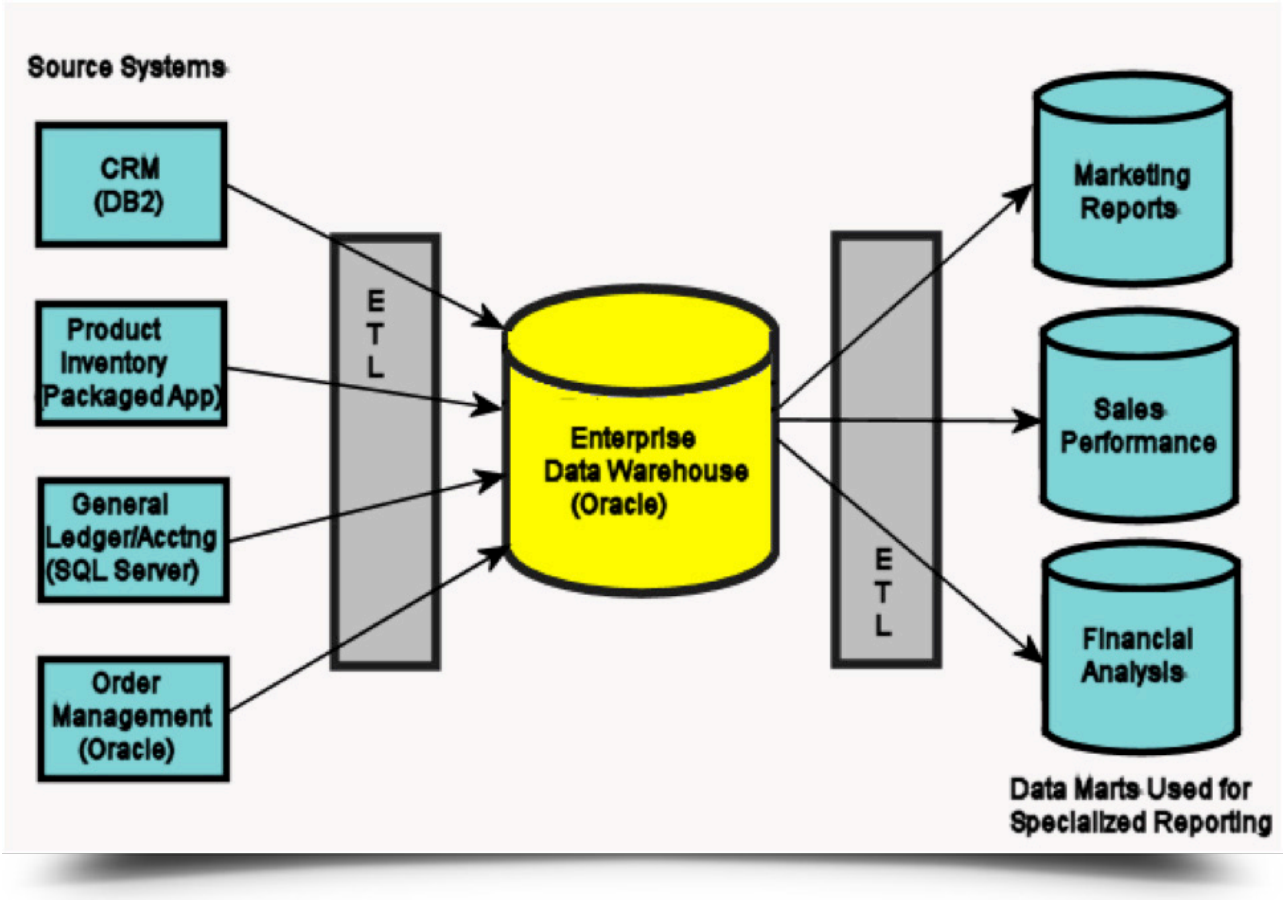
MANAGING THIRD-PARTY DATABASES AND BUILDING YOUR DATA WAREHOUSE

INTRODUCTION

It's a recurring theme. Companies are continually faced with managing a growing and sometimes disparate suite of third-party applications and tools and the underlying databases that support them. How many times have you tried to get a piece of information changed with some service provider and ultimately realized they store your information in multiple databases? As much as this is a problem when trying to get your address changed, it's an even bigger problem for the company as they try to manage their data. The only thing worse than no data is bad data – good data is especially important if you've been tasked with building out and maintaining your organization's data warehouse.

IDERA has tools that can help. Suppose you have multiple databases from third parties that all contain information about your business and that none of these databases and applications can communicate with each other. One solution is to build a data warehouse. It sounds simple enough but we all know the answer to that one. When done correctly and with some forethought, the warehouse can be a valuable asset and provide insights into the business, customers, or both. But what happens when the vendor for some aspect of your business provides an upgrade that includes database changes? How does this affect your data warehouse?

Some of the bigger challenges in building a data warehouse from third-party application databases are defining the data models from those applications (the source), knowing when the model has changed in the source, and incorporating the changes into your data warehouse models. Changes could be as subtle as a change in a data type on a column or much more involved with entirely new entities and relationships. Obviously any of these changes can impact the data that are being used to create the data warehouse. At best, you could end up simply with no data or bad data, but left unmanaged, it could bring your data warehouse to its knees.



Example of an Enterprise Data Warehouse Configuration

IDERA Solutions

Using DB PowerStudio (DBPS) products and ER/Studio Enterprise Team Edition with Team Server from IDERA, you can develop a workflow that allows for building out data warehouse models as well as managing the third-party databases, including database configuration, space and performance, reverse-engineering the source databases, and detecting changes to the configurations and schemas.

For the data warehouse design, we will utilize ER/Studio Data Architect in our workflow to model the data warehouse and source databases, and ER/Studio Team Server to publish the data dictionaries, glossaries, and terms for the business. Finally, we'll cover data lineage so you'll know from whence the data comes.

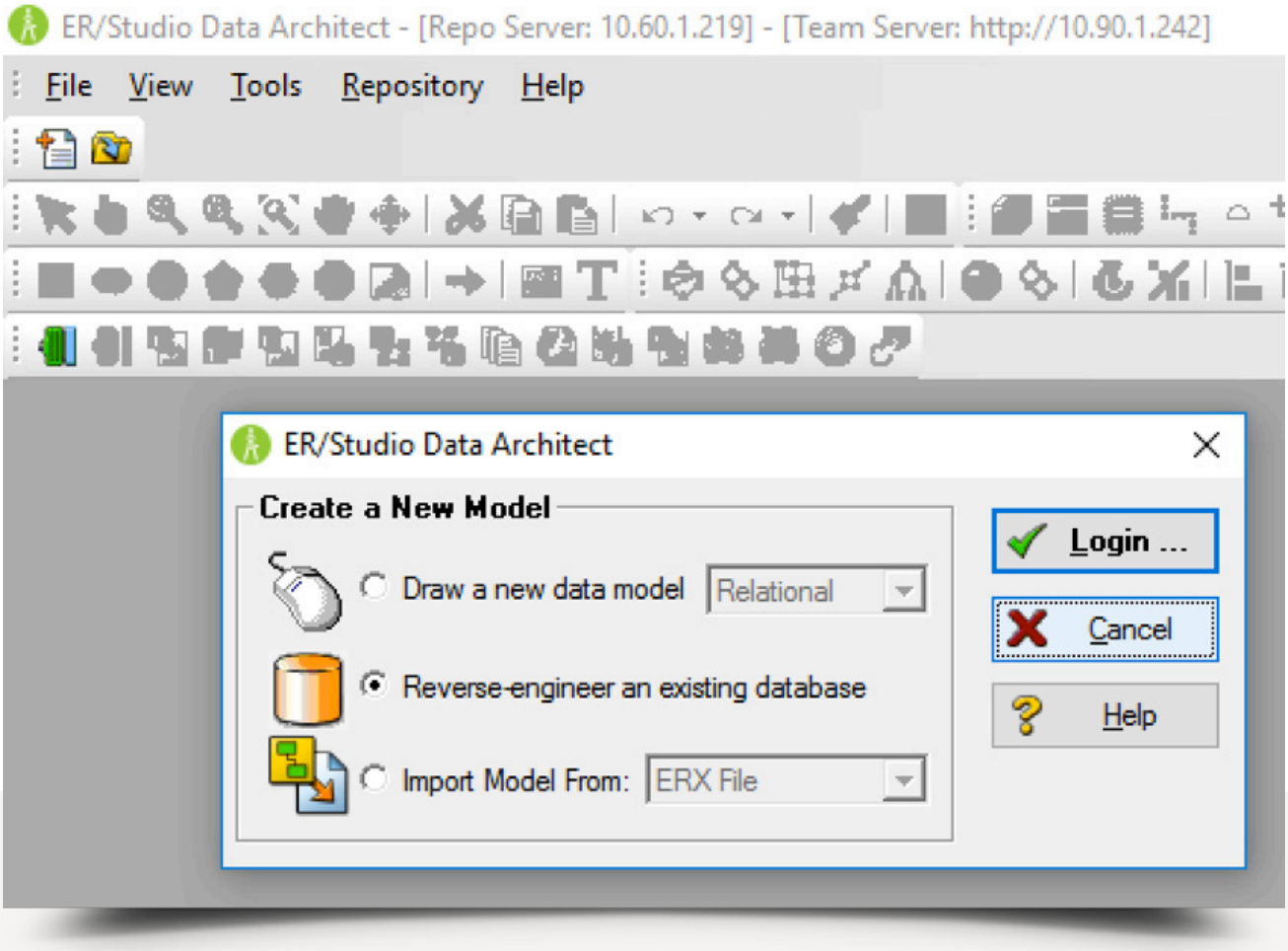
DB PowerStudio consists of four modules that provide database administration and management, SQL development, automated tuning and profiling, and schema and data synchronization. All of these tools can be useful in managing and monitoring a data warehouse, but to get started with a workflow we'll focus on database management and schema management from the DBPS suite.

BUILDING YOUR DATA WAREHOUSE

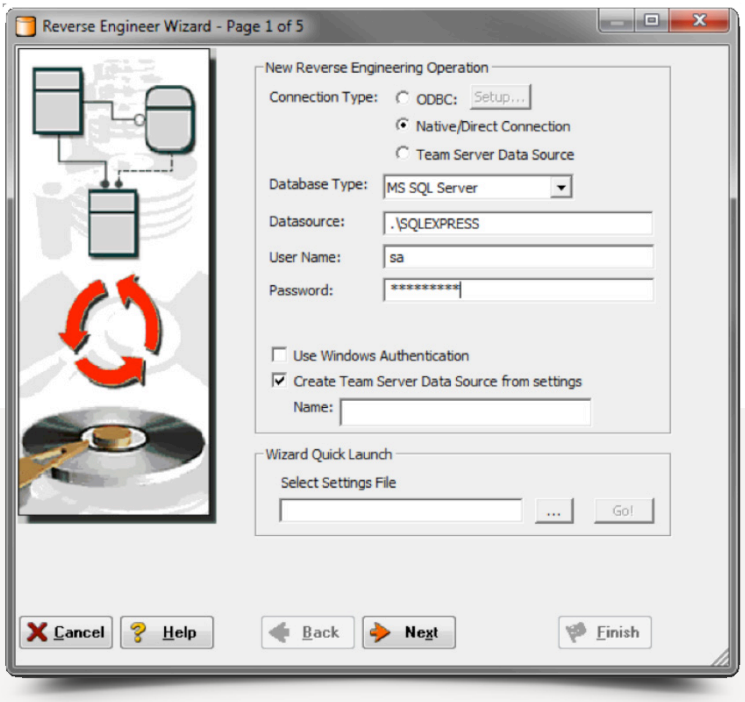
Reverse-Engineering Databases

ER/Studio Data Architect has the power to easily reverse-engineer, compare and merge, and visually document data assets residing in diverse locations from data centers to mobile platforms to databases provided as the backend to third-party applications. Let's fire it up!

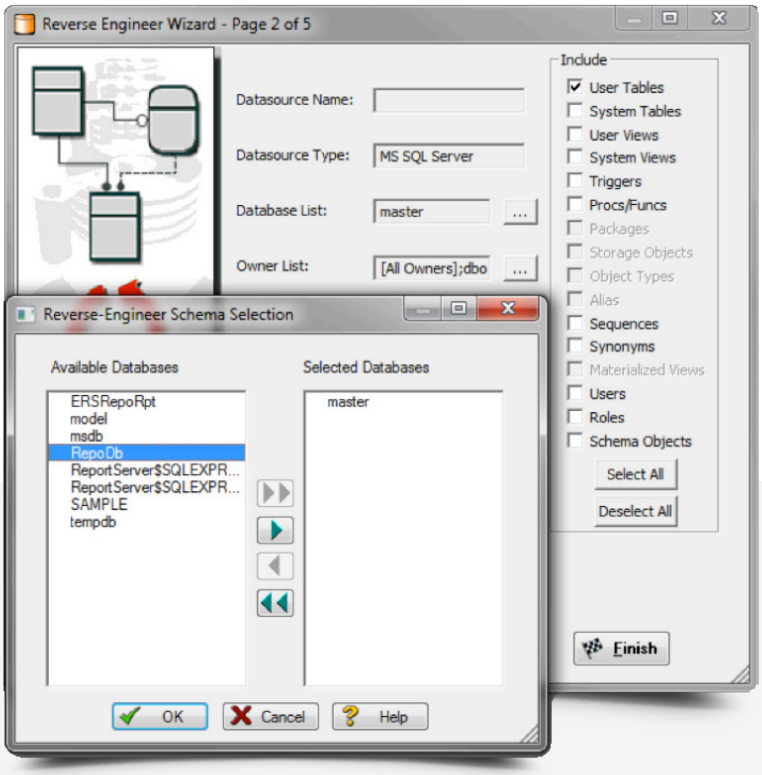
Data Architect provides a wizard-driven approach to streamline getting things done. We'll begin the workflow by reverse-engineering a database. Just select File > New then Reverse-engineer an existing database and Login.



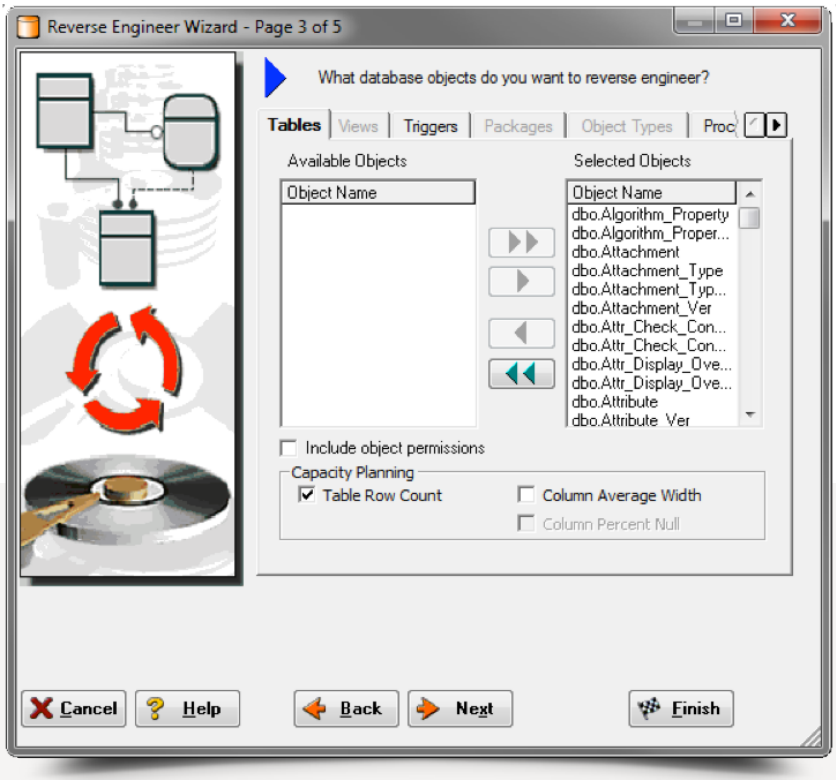
Next you'll be prompted to connect to a data source. It can be via ODBC, a native / direct connection or a Team Server data source (more on Team Server later). Native connections supported are Hive, MongoDB, Oracle, MS SQL Server, Azure SQL Database, Sybase ASE, IBM DB2 UDB and OS/390. This example uses MS SQL Server.



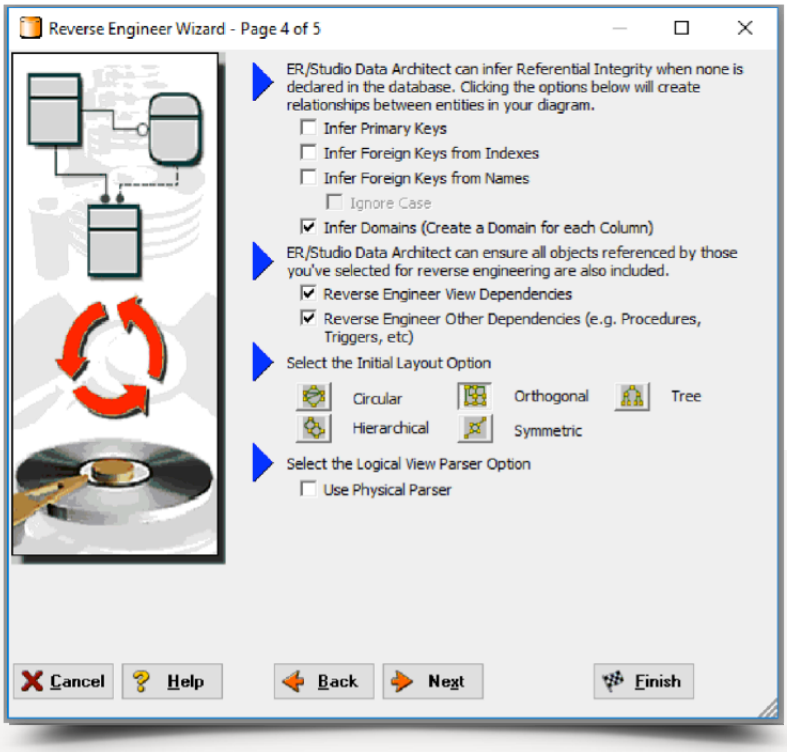
Select from the list of available databases and include the desired information using the checkboxes below.



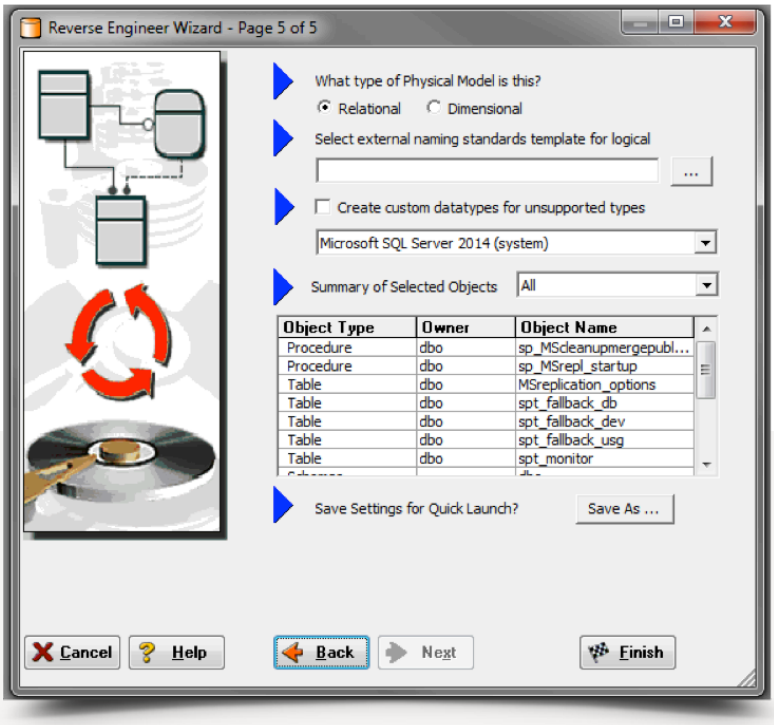
Remove any specific objects you don't want to include. All tables, triggers, procedures, etc. are included by default if they were selected on page 2 to be included.



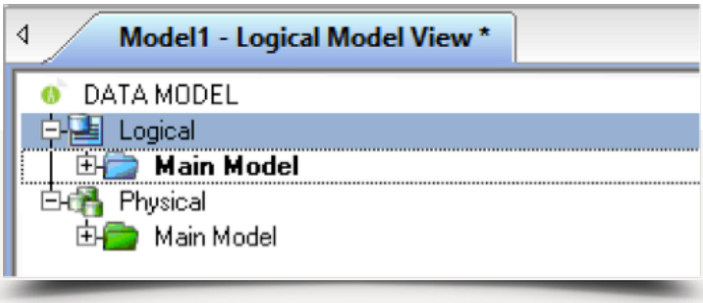
Select from the options listed in order to have the proper relationships and dependencies included in the model.
We recommend that you choose either Circular or Orthogonal as the initial layout for reverse engineering.
Other layout choices can take significantly longer to process, depending on the number of entities that will be reverse engineered. We suggest that you always choose Infer Domains unless you have a compelling reason to leave it unchecked.

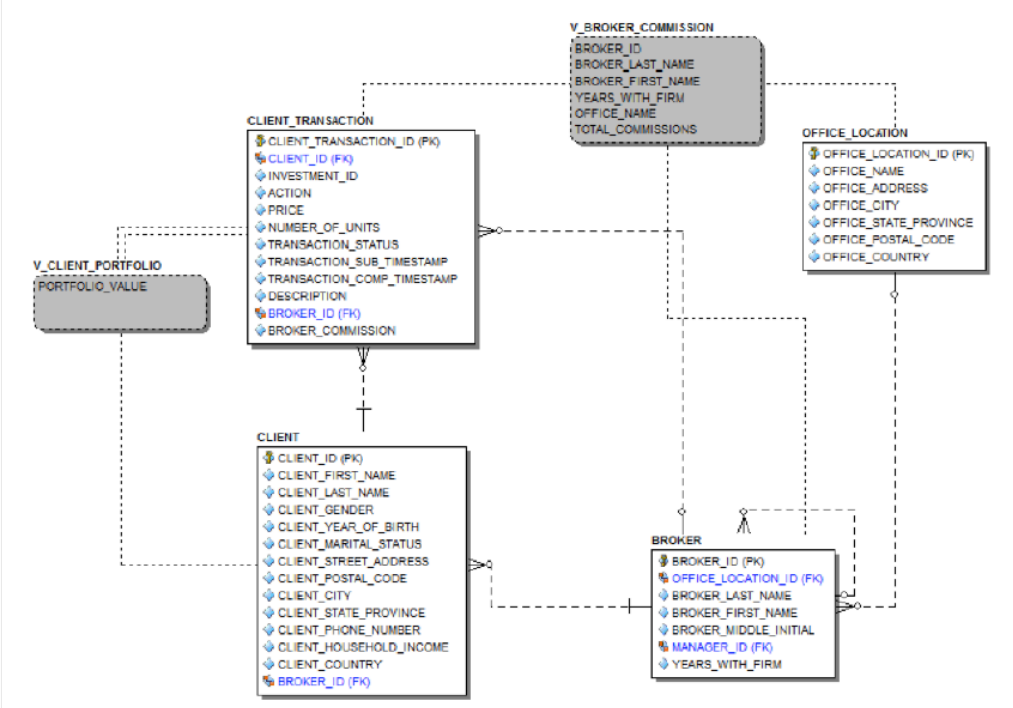


Finally select the model type, apply any previously defined naming standards templates, and click Finish.



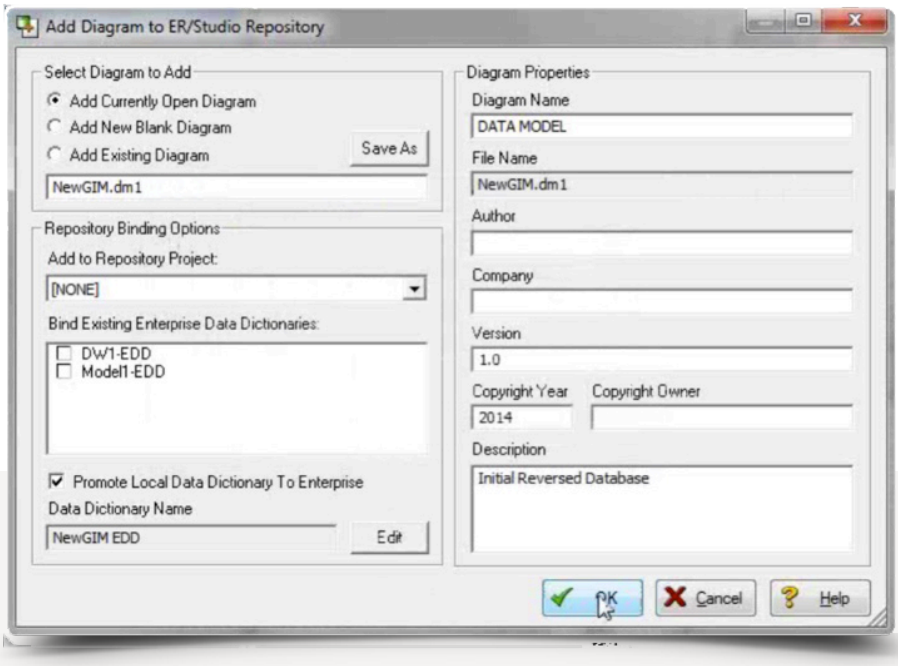
When reverse engineering from a live database, both a logical and a physical model are created along with other entities selected during the reverse engineering process.



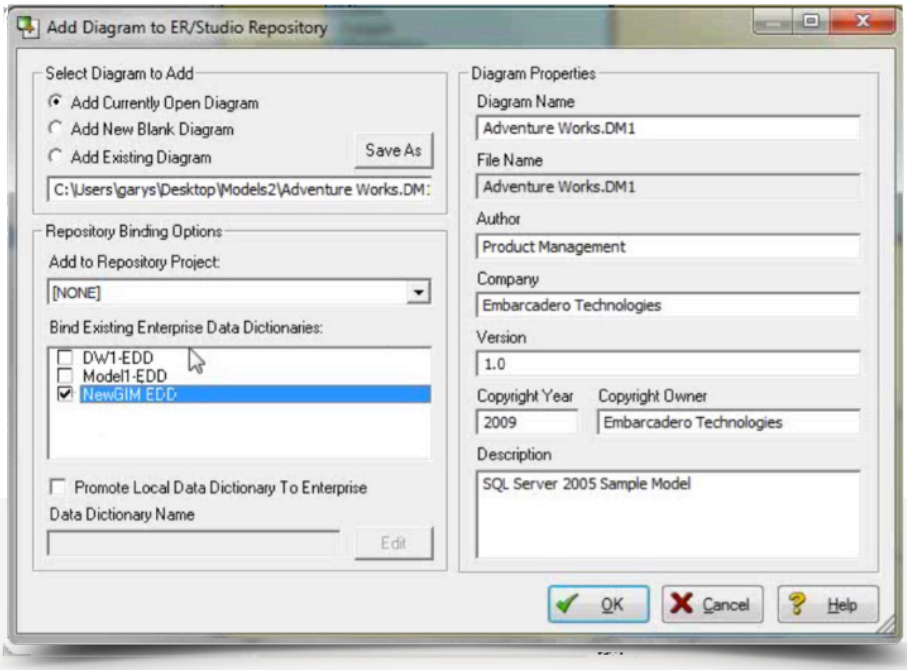


Build the Enterprise Data Dictionary

Once the models are generated, we look next to storing that model in the repository and building an enterprise data dictionary. Add the diagram to the repository and build the enterprise data dictionary in one step. From the Repository menu item, select Diagrams > Add Diagram. This will open the following dialog and allow you to add the model to the repository and promote the local data dictionary to an Enterprise Data Dictionary (EDD). It is important to select the checkbox for Promote Local Data Dictionary to Enterprise. If you do not promote the local version in this step then it is rather difficult to do so after the model has been added to the repository.

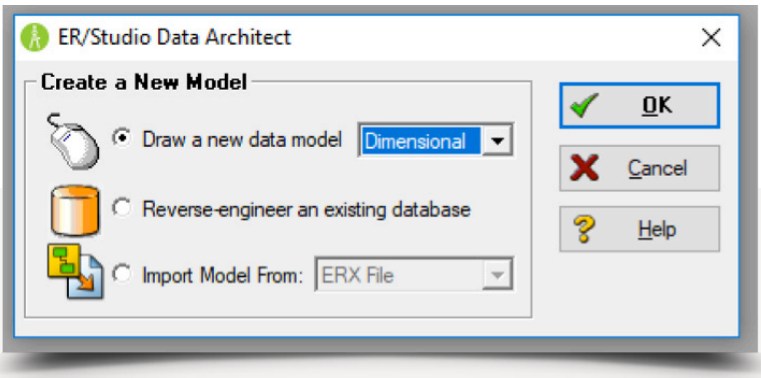


To create an EDD for the first model and for subsequent models that have been reverse engineered, select the option to bind to an existing EDD when you add the diagram to the repository. It is desirable to ensure that the EDD elements are bound to the proper domain. In this case you'll leave the Promote Local Data Dictionary to Enterprise unselected.

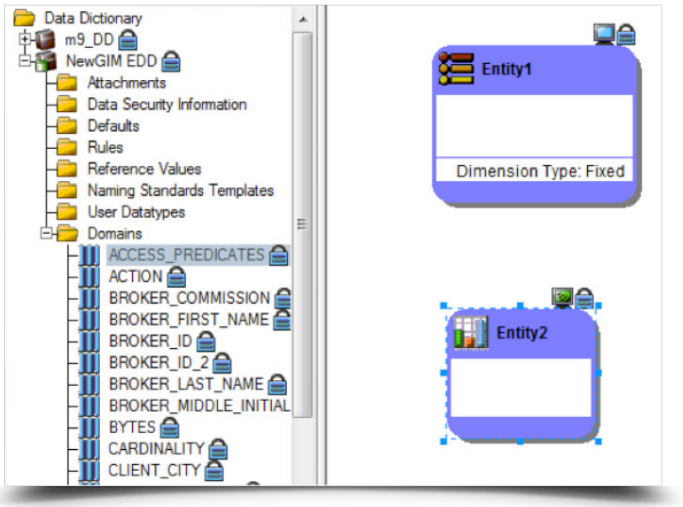


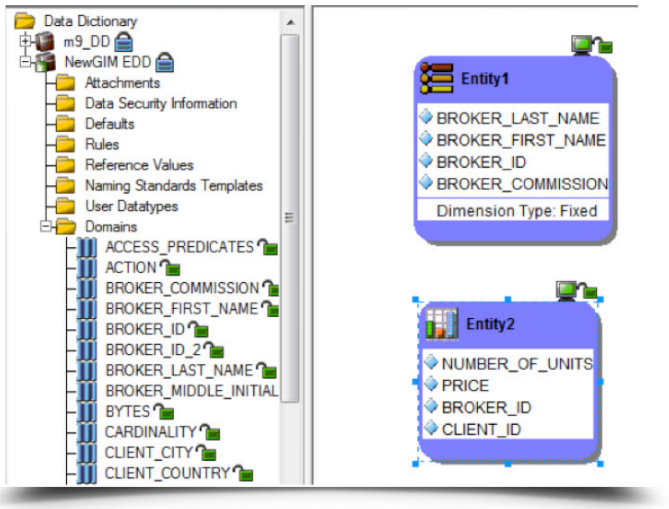
Model the Data Warehouse

Assuming there is no pre-existing data warehouse (DW), you'll want to create a new model and select Dimensional from the dropdown menu.



To get started, it's easiest to drop a couple of entities into the model and then add it to the Repository. Just as when a reverse engineered model was added, you'll bind the model to the existing EDD even though it's being built from scratch. From the Data Dictionary tab you'll now have access to the domains in the EDD. From this tab you can check out the domains and add them to the entity of your choosing. This is how you'll build out the data warehouse model attributes and maintain consistency with the data coming from the physical databases that were reverse engineered. Of course you'll need to keep a watchful eye for changes to the physical databases. We'll cover how to monitor for changes in the Change Manager section of this document.

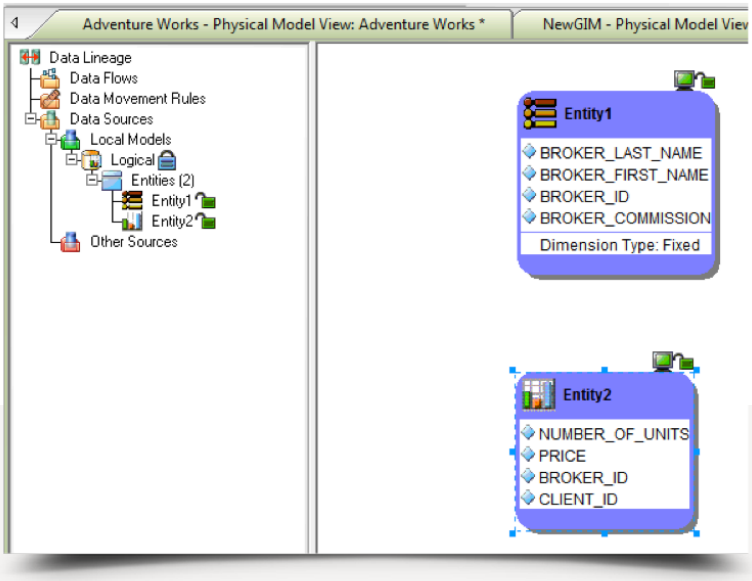




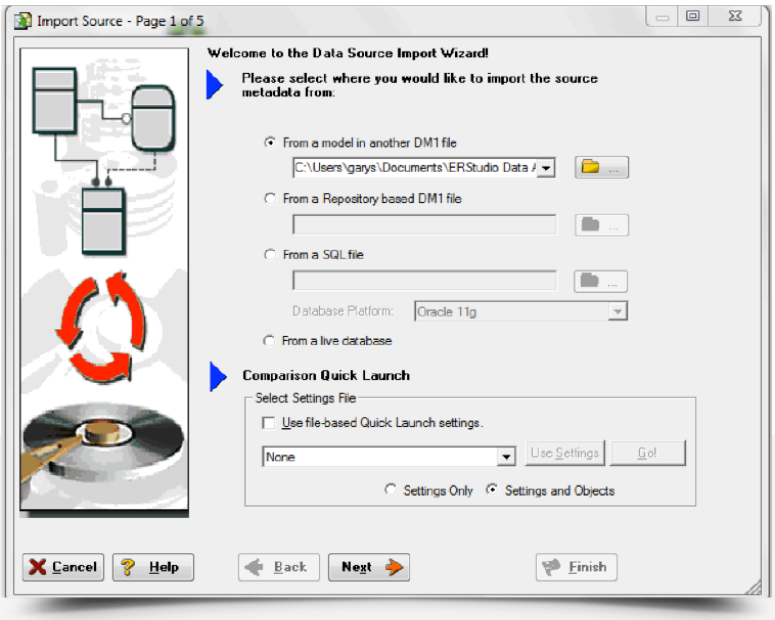
Once the entities have been populated from the domains in the EDD you can move on to the Data Lineage tab.

Data Lineage

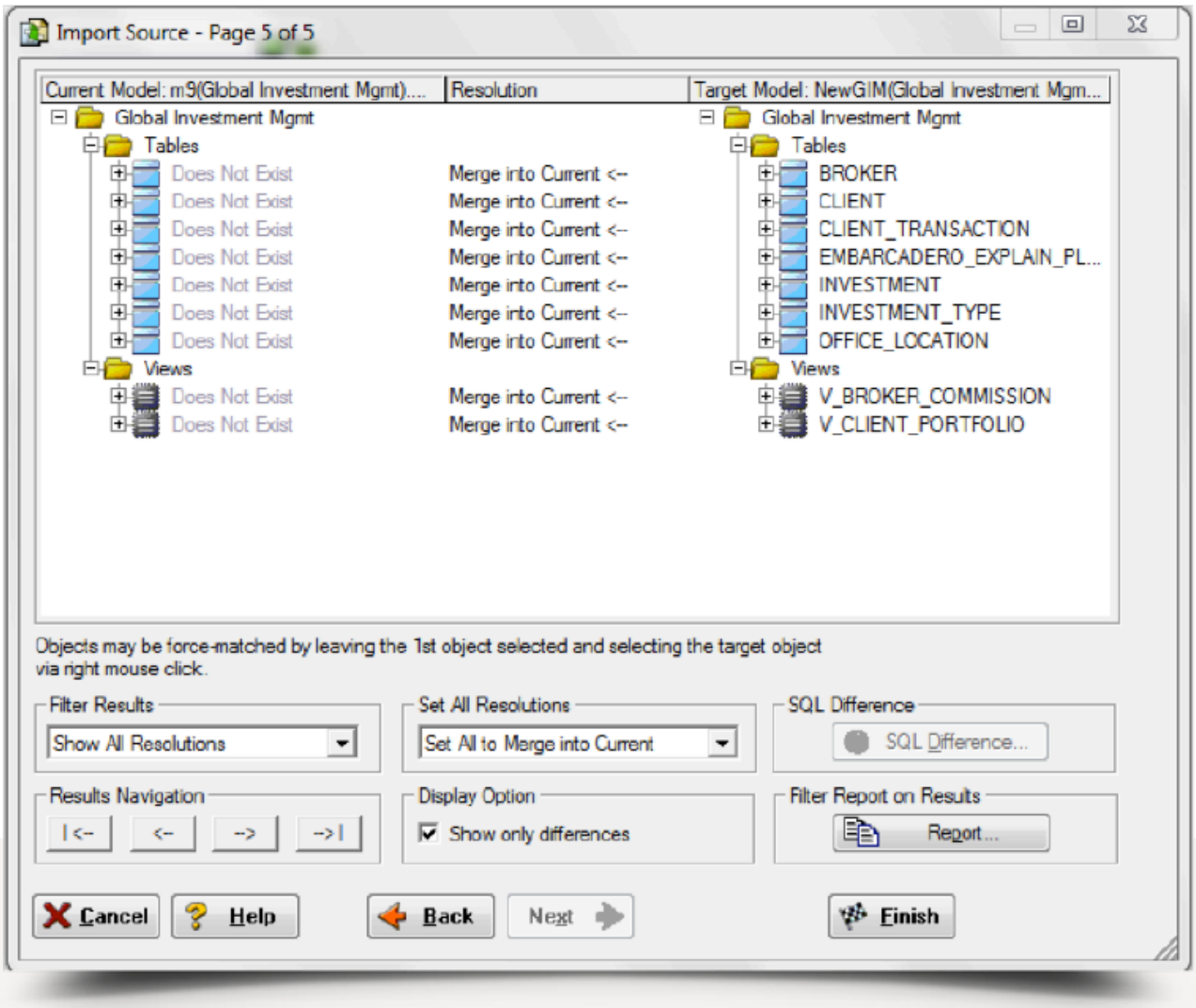
With the entities defined, we use the Data Lineage tab to document the data movement. This movement is referred to as Extraction, Transformation, and Load (ETL). Points A and B can be any source such as flat files, high-end databases like Oracle and SQL Server, XML files, and Excel worksheets. This is sometimes referred to as source and target mapping. A model produced in ER/Studio can represent any point along the way. Data architects need the ability to specify the "source" or "target" of data, down to the column/attribute level. Along with the metadata that defines the source and target mapping are rules for how the data is manipulated along the way.



Right-click on Other Sources in the data lineage tab to start the data source import wizard. You will select models from other DM1 files (the databases you've reverse engineered are each stored in a separate DM1 file).



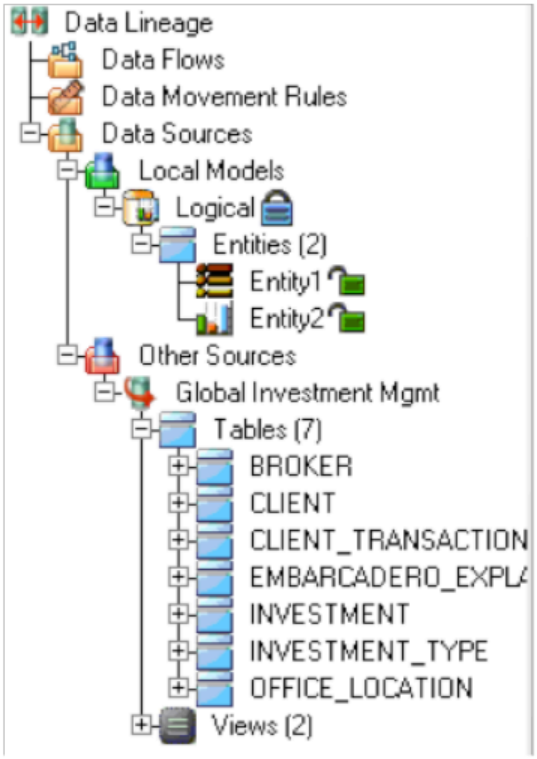
Clicking Next will prompt you to select the specific model (both logical and physical models will be shown) and specific model objects to import, and the last step (page 5) in the wizard is where you'll merge the objects from the model you have chosen. Generally, you'll set all resolutions in the dropdown to "Set All to Merge into Current".

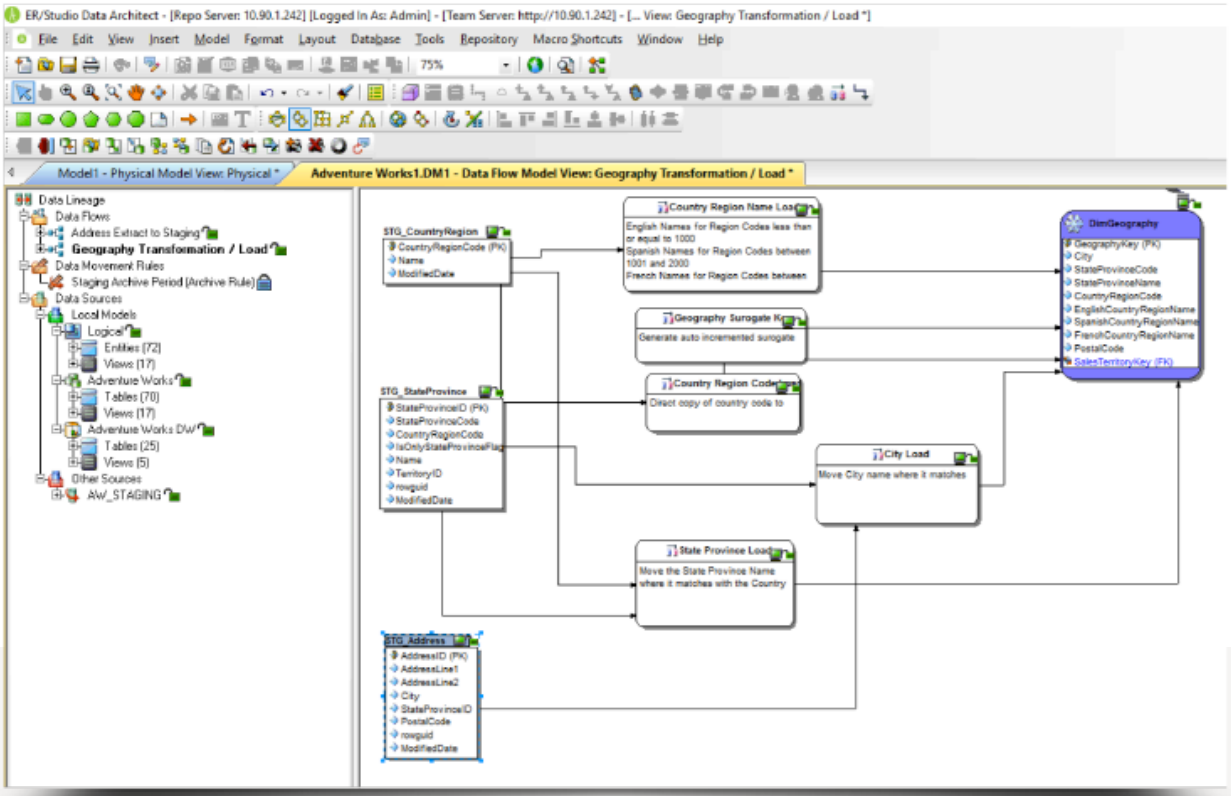


Using your entities from other model sources, you're ready to begin documenting the ETL job. On the Data Lineage tab you'll drag your source entities to the right and drop them on the palette.

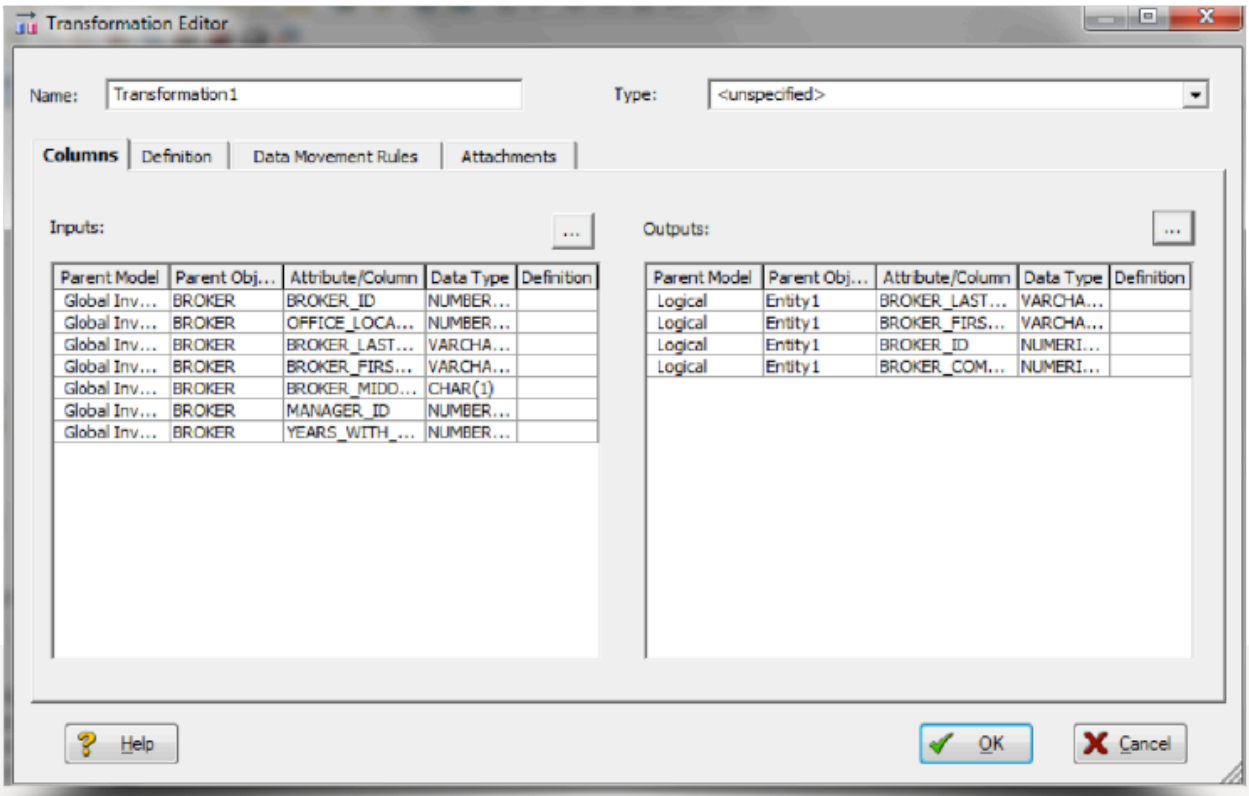
Next you'll select from the objects you want under Other Sources. In the following steps you'll add a transformation block and add the data stream (inputs and outputs).

If you want to create a column-level mapping, only define one source and target column per transformation. A macro is available to export the mapping.

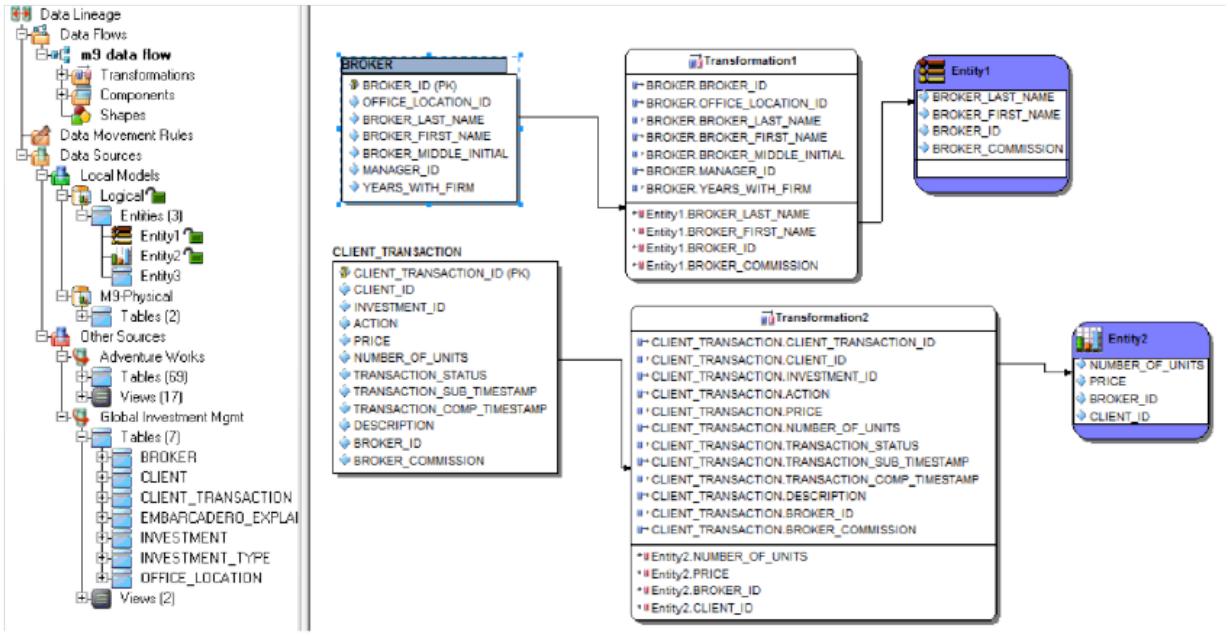




Once the transformation block and data streams are added to the data lineage diagram, double-click on the transformation block to open the transformation editor and define the inputs and outputs to the transformation process.



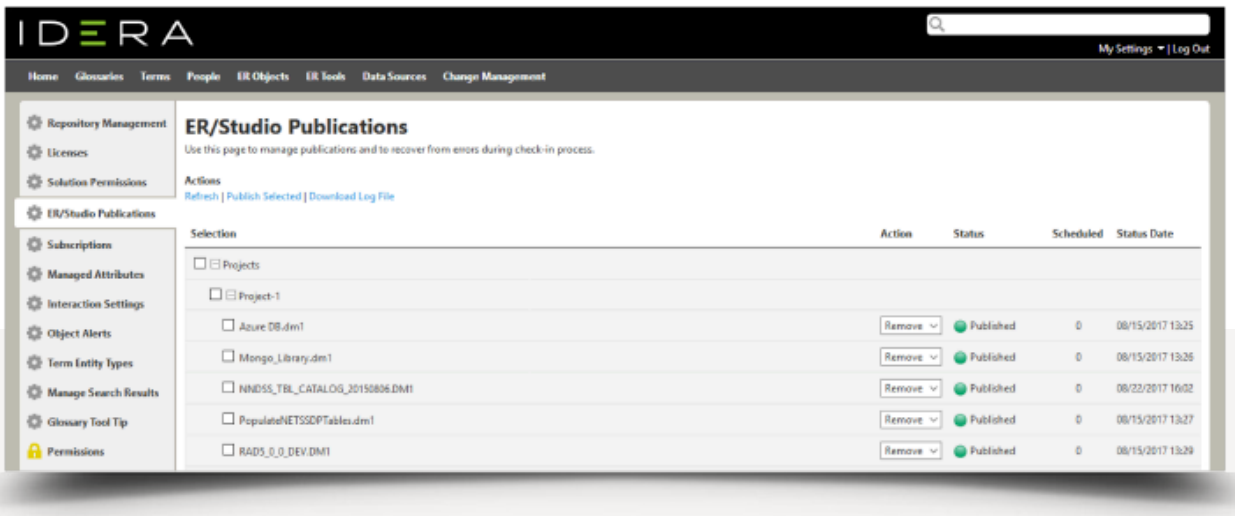
At this point you'll simply repeat this process until your ETL jobs are all fully documented. ER/Studio does not provide full ETL functionality, but it does provide the ability to document the ETL process.



COLLABORATING WITH TEAM SERVER

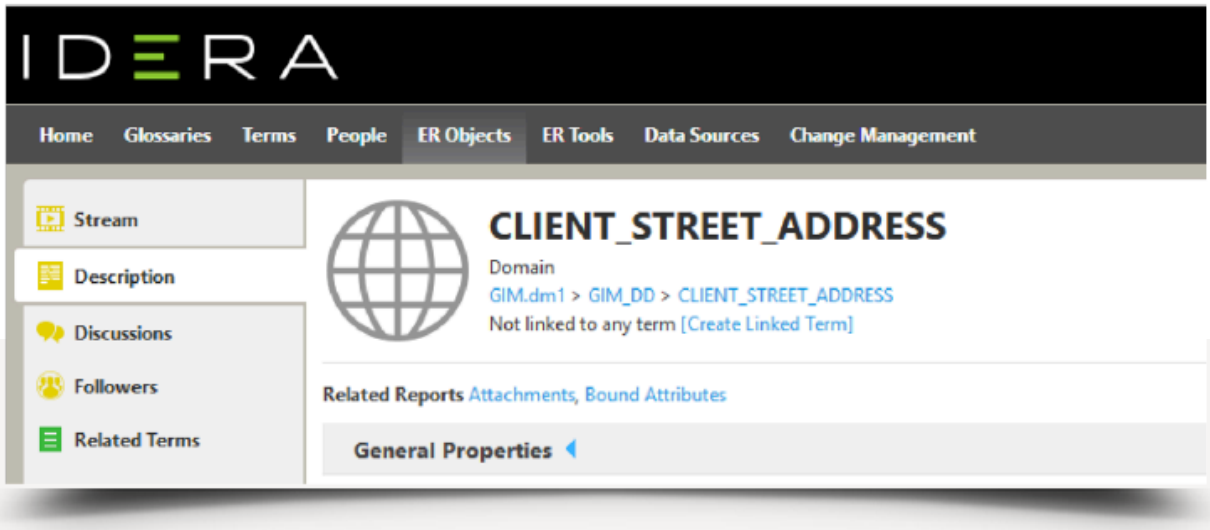
ER/Studio Team Server is a model and metadata collaboration platform that provides greater meaning, understanding and context to enterprise data. Data professionals, developers, and business analysts gain better comprehension and compliance using integrated model, metadata and collaboration tools. Team Server's collaborative enterprise glossary brings together the entire organization to foster improved metadata, business definitions, and security policies to create a foundation for governance and compliance initiatives. Online models showcase data relationships, while powerful search capabilities help users locate enterprise data with ease.

The data models and enterprise data dictionary that have been added to the repository would then be published by the Admin user. Select My Settings > Admin to navigate to the publications page, select the model to publish and click Publish Selected from the Actions list. Once the model has been successfully published, the Status will indicate "Published". The model, diagram, and metadata are now being shared throughout the organization.

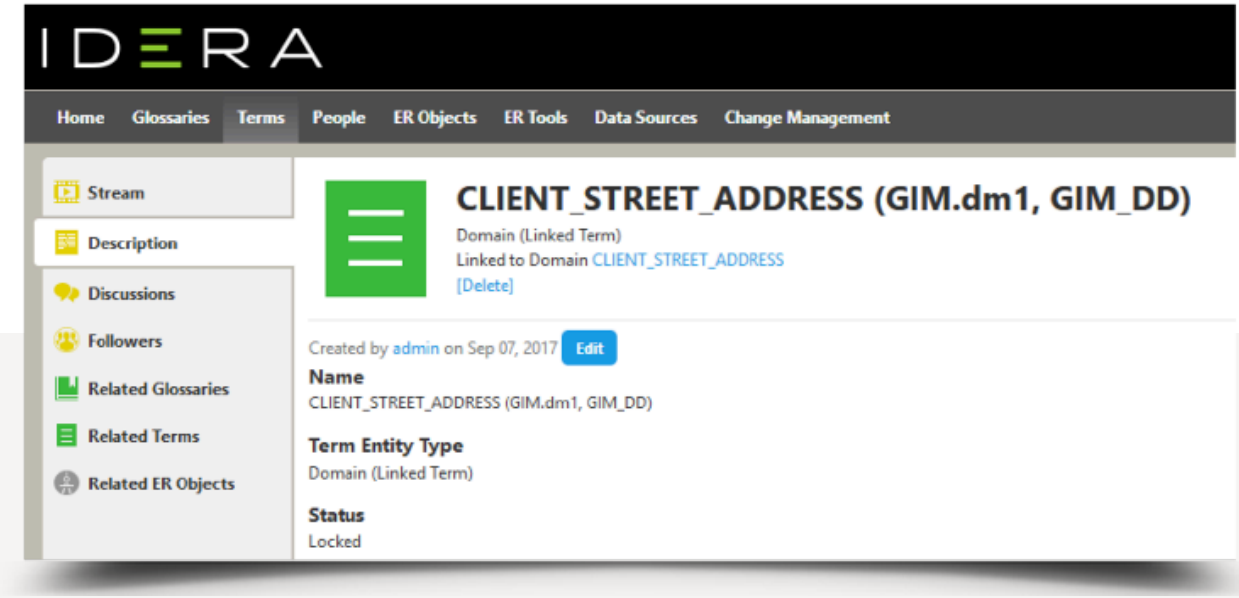


Manage a Single Source of Business Definitions

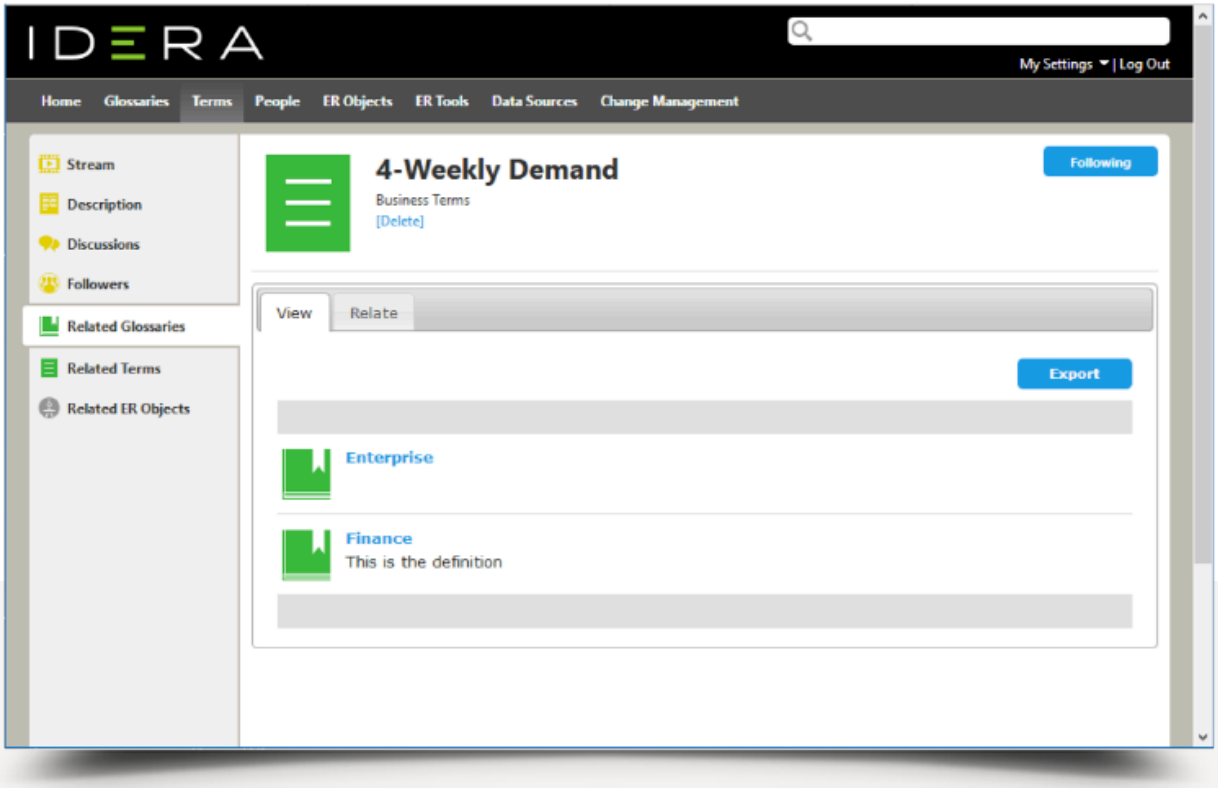
Using search or direct navigation from ER Objects, a "term" can be created and linked. In this case, CLIENT_STREET_ADDRESS, a domain attribute, has been selected from a local Data Dictionary. Click Create Linked Term to create a CLIENT_STREET_ADDRESS Domain Linked Term.



The page now shows that the Domain Linked Term has been created.

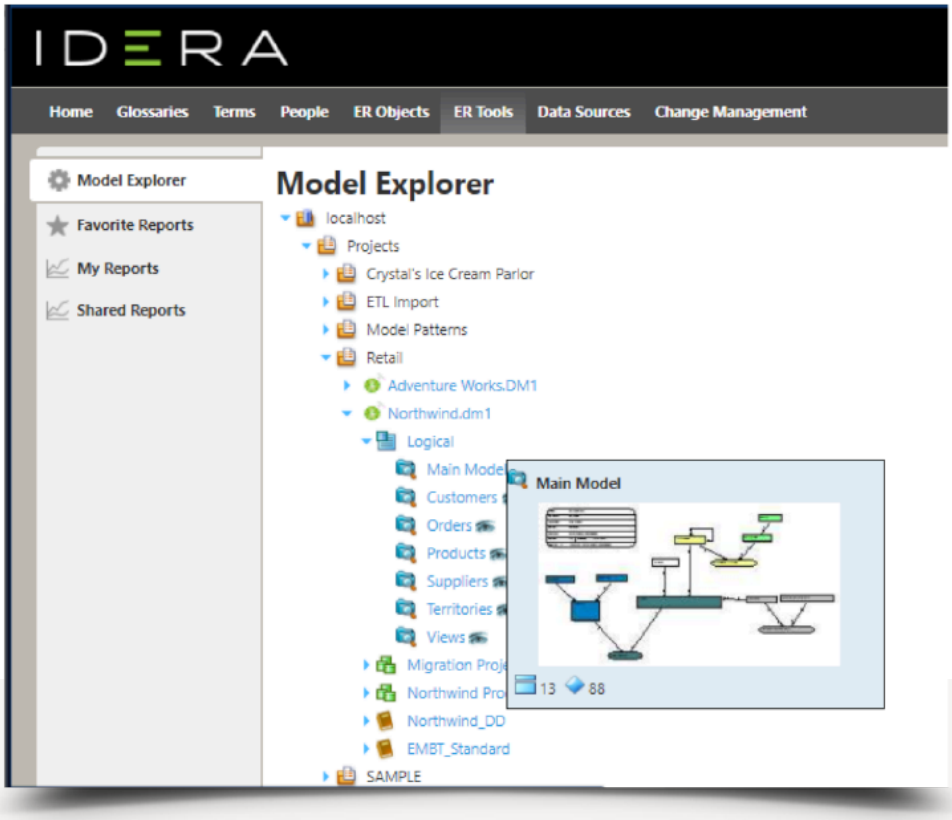


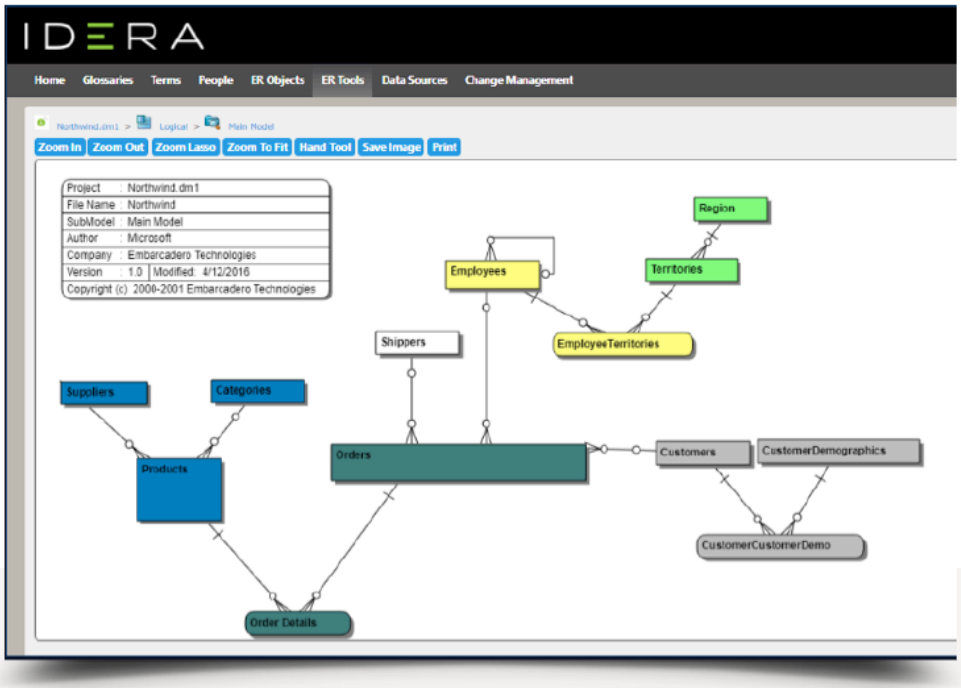
The term has now been linked to the ER object and can also be related to other terms, objects, and glossaries to show additional relationships. It's as simple as selecting the item to relate and they will be connected. Team members can also "Follow" entities so that any subsequent changes will show up in their home page stream.



View and Navigate Interactive Data Models

The models are capable of being viewed at the most granular level and are also actively linked to created terms within Team Server. Navigate to ER Tools > Model Explorer and open a model. Hovering over the model name will show a preview of the model contents, and clicking the "eye" icon next to the model name will open a full-size image of the model in a separate browser window.


IDERA
Home
Glossaries
Terms
People
ER Objects
ER Tools
Data Sources
Change Management
Model Explorer
Favorite Reports
My Reports
Shared Reports
Model Explorer
localhost
Projects
Crystal's Ice Cream Parlor
ETL Import
Model Patterns
Retail
Adventure Works.DM1
Northwind.dm1
Logical
Main Model
Customers
Orders
Products
Suppliers
Territories
Views
Migration Proj
Northwind Pro
Northwind_DD
EMBT_Standard
SAMPLE
Main Model
13
88

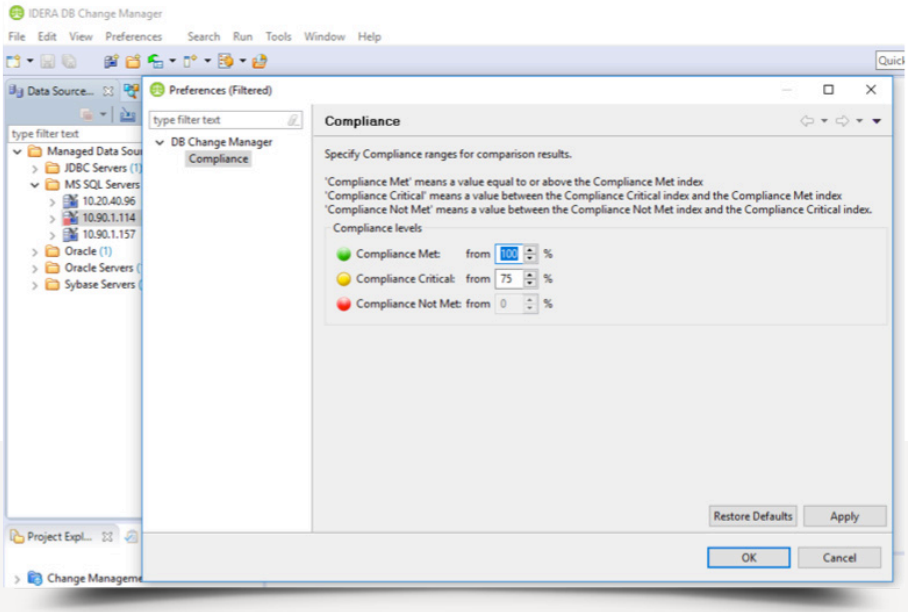

IDERA
Home
Glossaries
Terms
People
ER Objects
ER Tools
Data Sources
Change Management
Zoom In
Zoom Out
Zoom Lasso
Zoom To Fit
Hand Tool
Save Image
Print
Project : Northwind.dm1
File Name : Northwind
SubModel : Main Model
Author : Microsoft
Company : Embarcadero Technologies
Version : 1.0 Modified: 4/12/2016
Copyright (c) 2006-2001 Embarcadero Technologies
Region
Employees
Territories
EmployeeTerritories
Shippers
Suppliers
Categories
Products
Orders
Customers
CustomerDemographics
CustomerCustomerDemo
Order Details

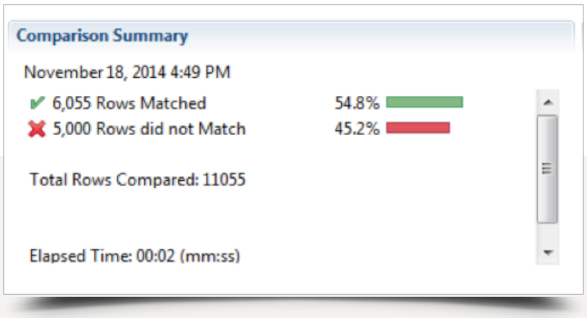
MANAGING CHANGE IN THIRD-PARTY DATABASES

Using DB Change Manager

When you're faced with supporting even a few databases from third-party vendors, one of the challenges lies in knowing when the schema or specific data changes, if that database is used as a source for part of your data warehouse. It is possible to use ER/Studio to identify the schema changes using the compare and merge capability but a more robust alternative is offered by the DB PowerStudio product, DB Change Manager.

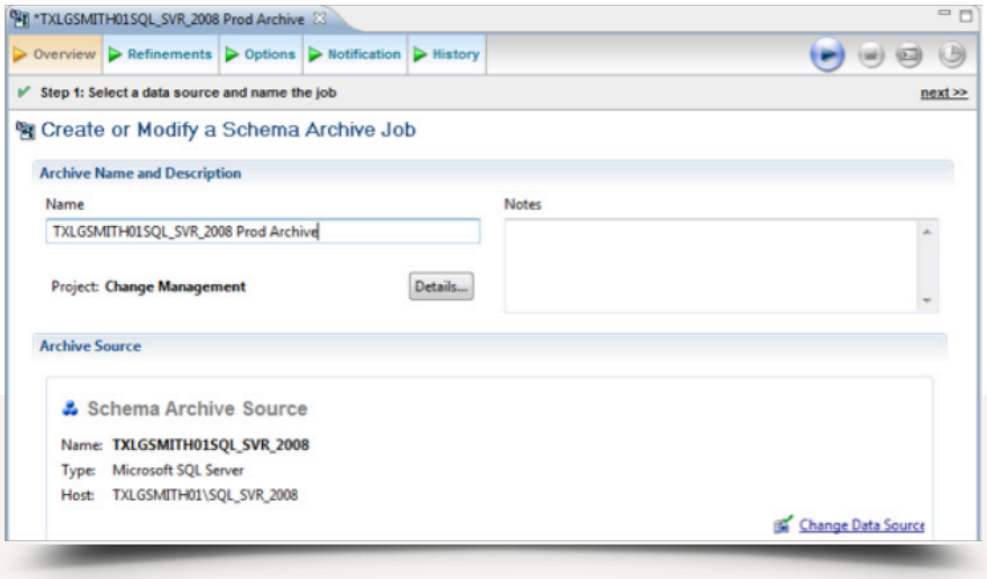
The first step is to connect to your data source. DB Change Manager allows you to set the level of compliance that you want through the Preferences > Compliance dialog for use in the comparison summary as shown below.



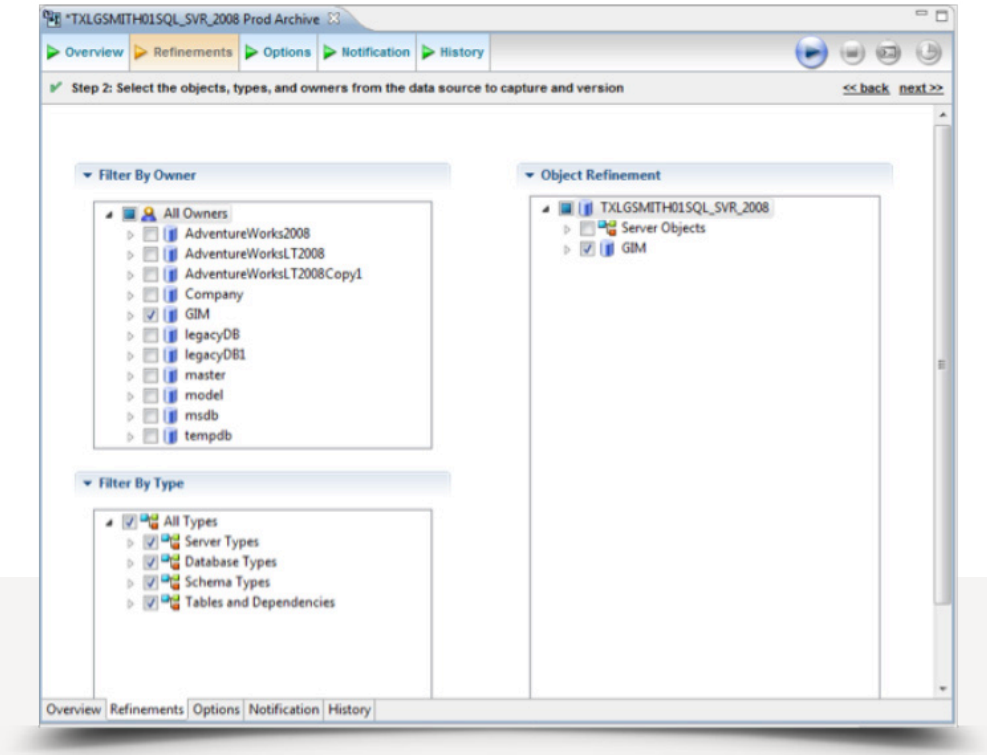


A comparison can be performed for the Schema, Data and the Configuration of a data source. The source can be a live data source or an archive of the desired area for comparison. Creating a baseline schema archive is a good step toward identifying changes in a schema.

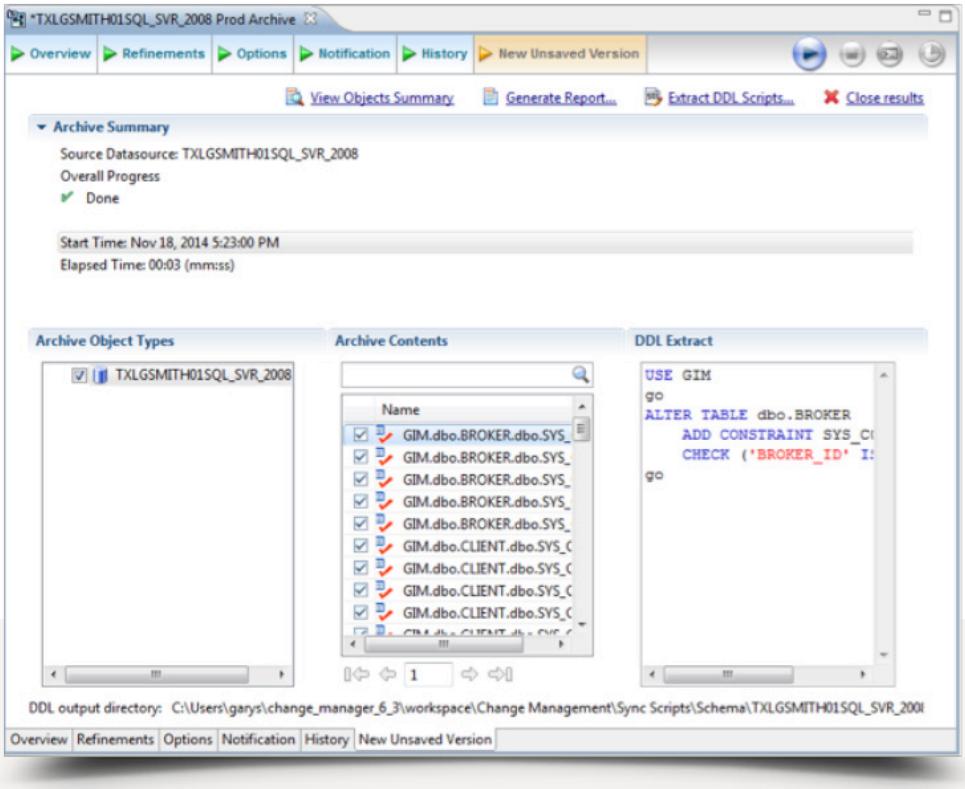
To create your schema archive select File > New > Schema Archive Job and select the data source you wish to archive.



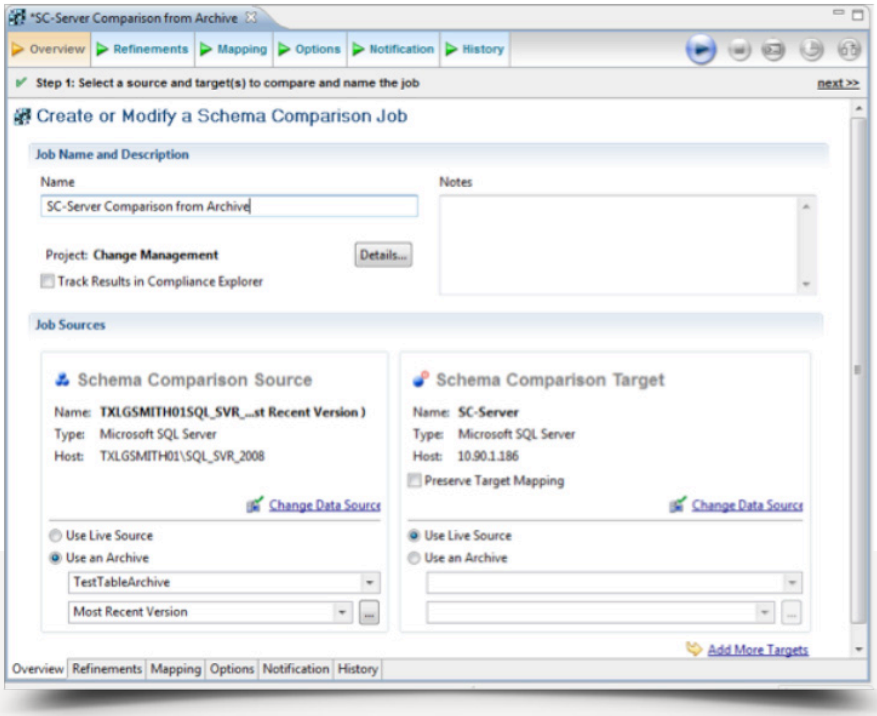
Refine your archive job to include the objects and types you wish to include in the archive. In this example we are including only the GIM database.



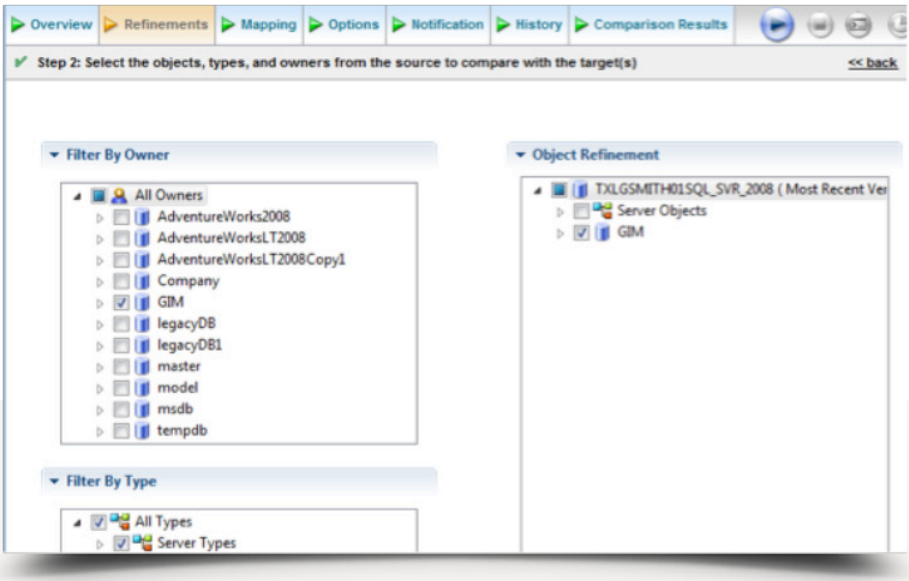
Once the archive is completed, you can examine the contents if you wish, but you'll want to save the archive for a future schema comparison job. In this case we've actually archived a source on one server and will compare it to a source on another server. This is also useful in development environments to identify what the differences are between Production, Development, and QA databases.



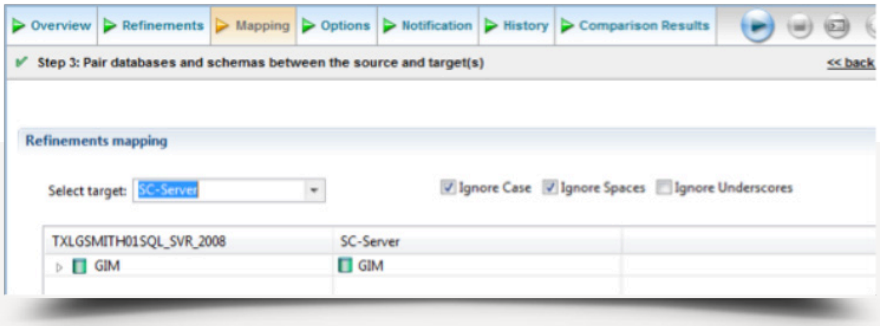
At the start of a schema comparison job you'll select an archive for use in the comparison and the live data source to compare against. In this case we are going to compare GIM databases from the archive taken on TXLGSMITH01SQL_SVR_2008 against the live GIM database on the SC-Server. Our premise is that the schemas should be the same.



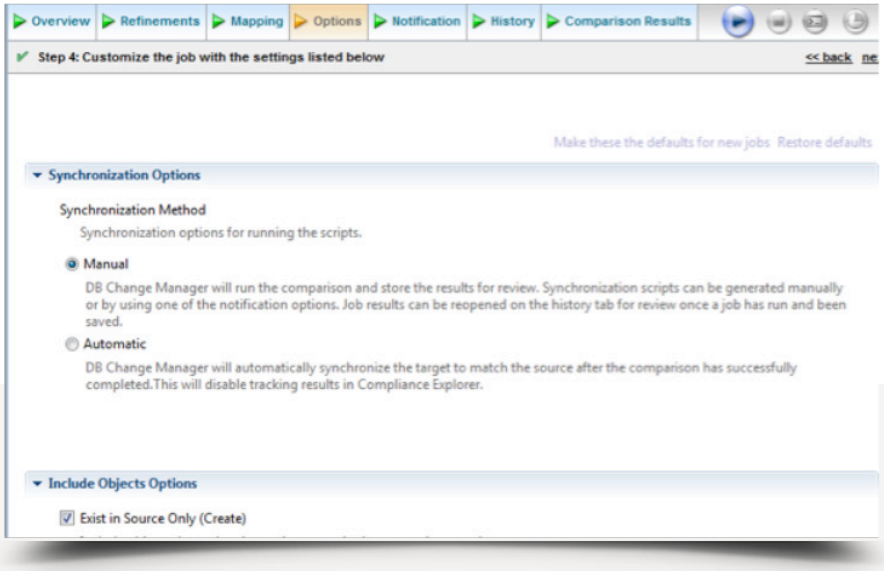
You should select the same objects for the comparison as you did for your archive.



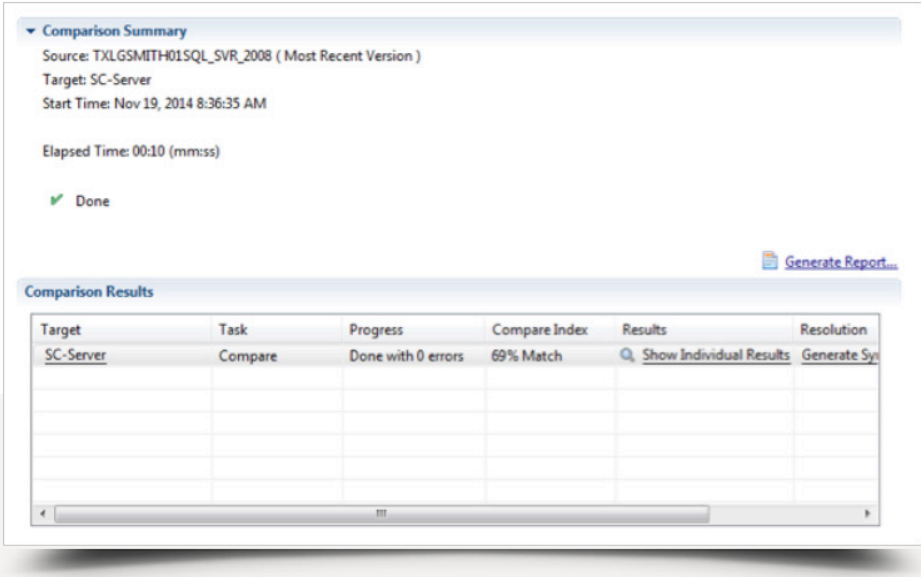
You'll see a mapping pairing the databases and schemas for the desired comparison.



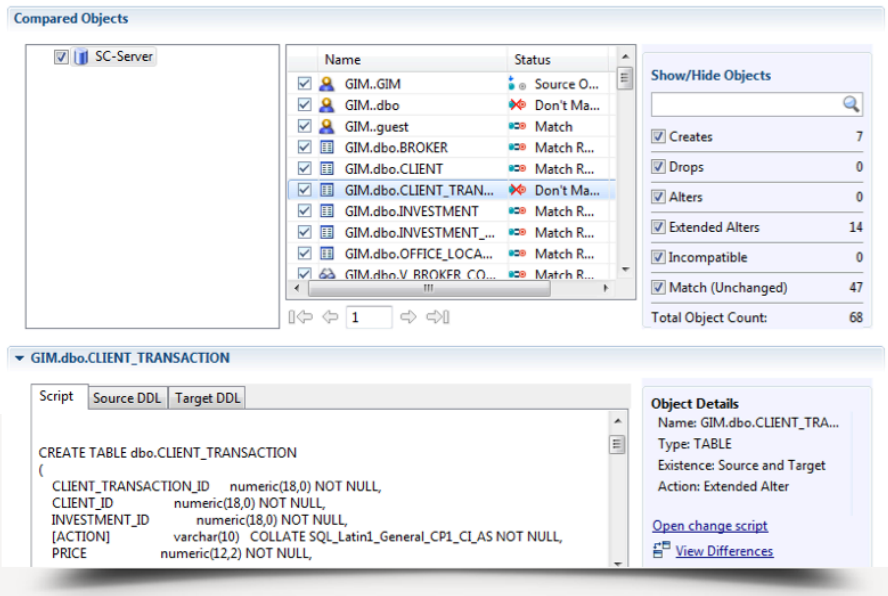
There are a number of options including the ability to automatically synchronize the database with the archive. You can also extract the synchronization DDL if you wish.



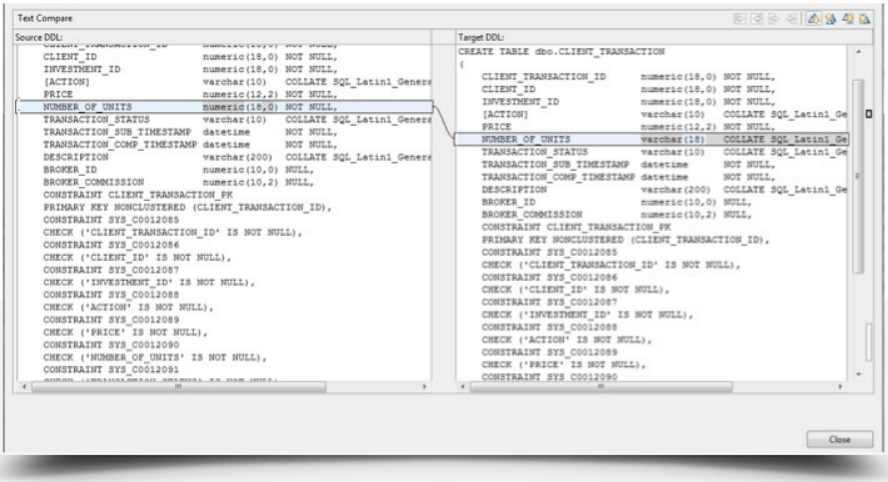
Our comparison summary shows a 69% match. You can click on Show Individual Results to see what objects do not match and view the DDL for synchronization.



You can click View Differences and see exactly what's different in the schema in the Individual Results view by selecting items that don't match in the list.

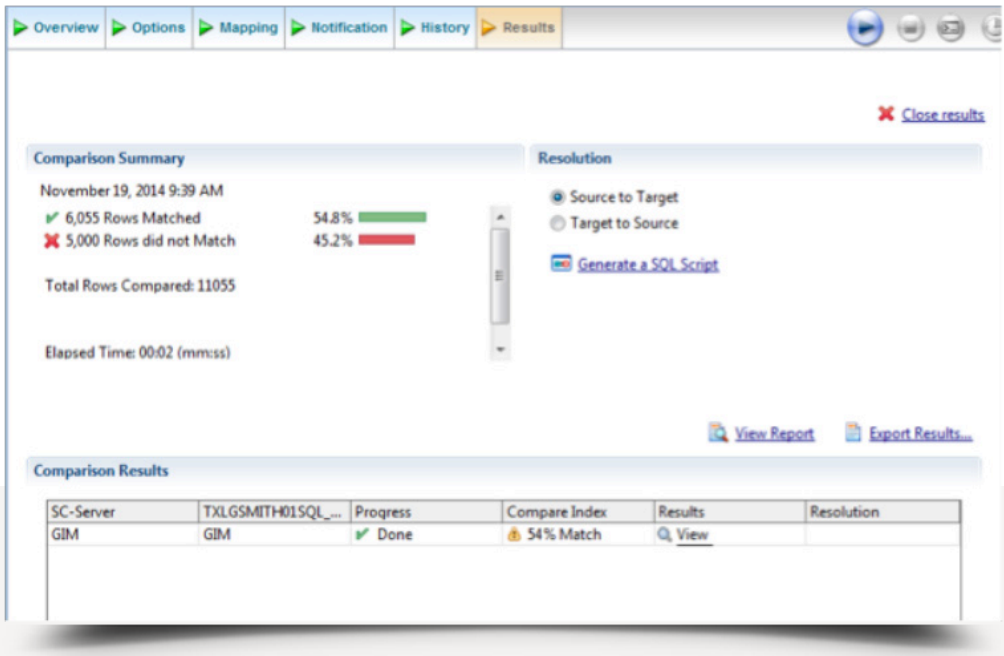


Here we can see that the NUMBER_OF_UNITS column is a numeric field in the archive but has been changed to varchar in the target GIM database on the SC-Server.

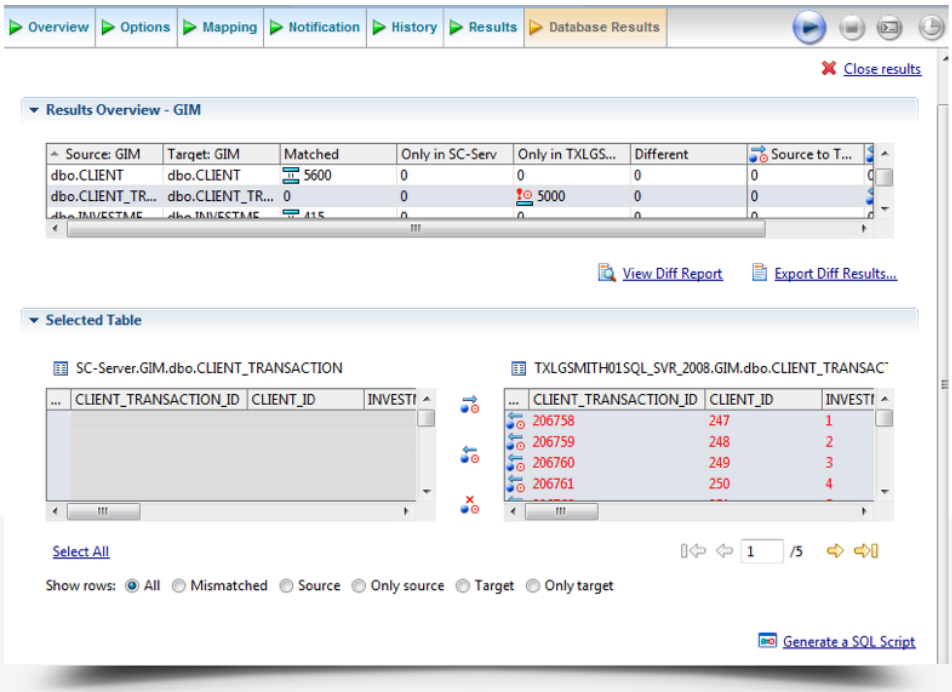


Data Comparison

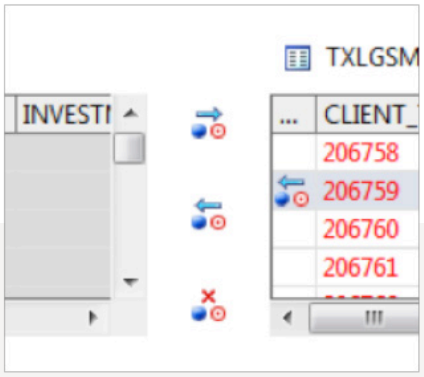
A data comparison job has a similar process but you'll compare two live data sources rather than taking an archive of the database for obvious reasons. In this example the result is a 54% match.



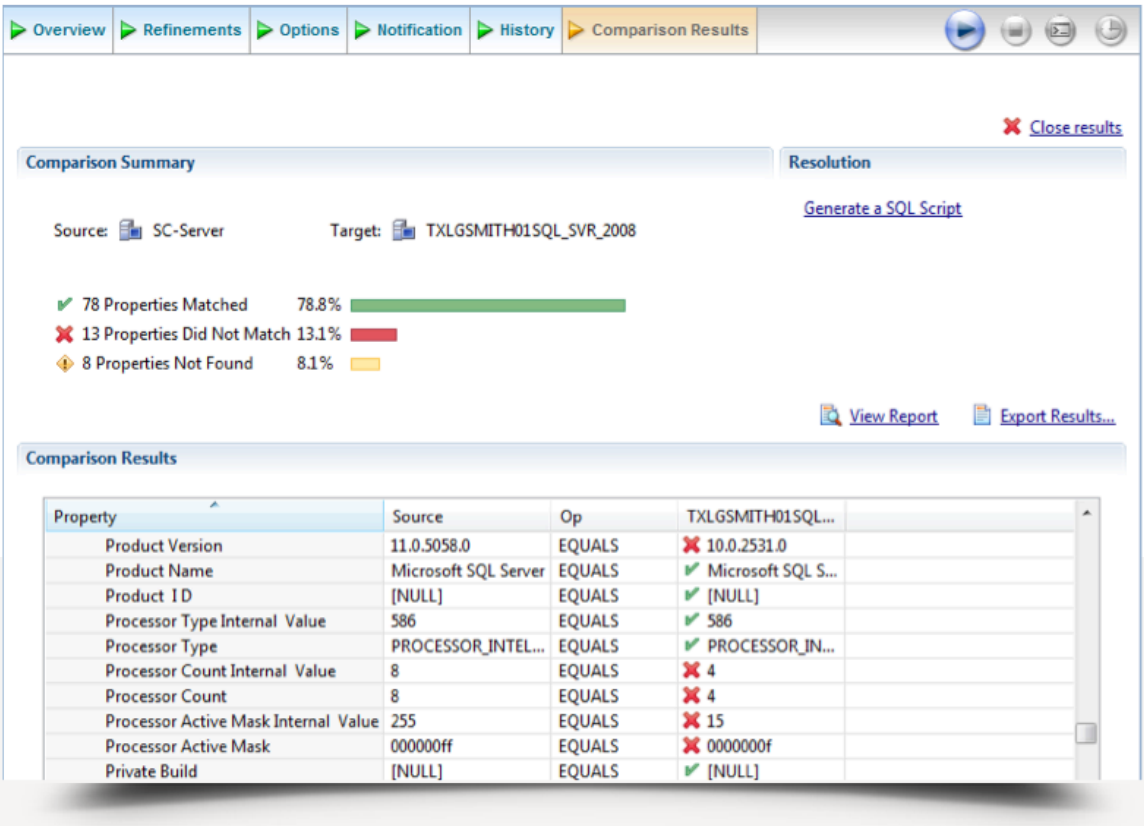
Selecting View in the Results column will provide details on differences and the ability to synchronize the data between the two compared sources.



You can move all rows or selected rows of mismatched data using the icons shown below and generate the SQL script. Data can be moved in either direction by selecting source to target or target to source and clicking Generate a SQL Script.



A configuration comparison job can also be run against an archive or between two live data sources. This may be of limited use in the workflow we're discussing but it's worth mentioning since the tool can be used to identify differences in configurations where the expectation is that they are the same, or if you simply wish to document actual configurations. As with other comparisons, full reports on the comparison are available.



DB CHANGE MANAGER ADVANCED CAPABILITIES

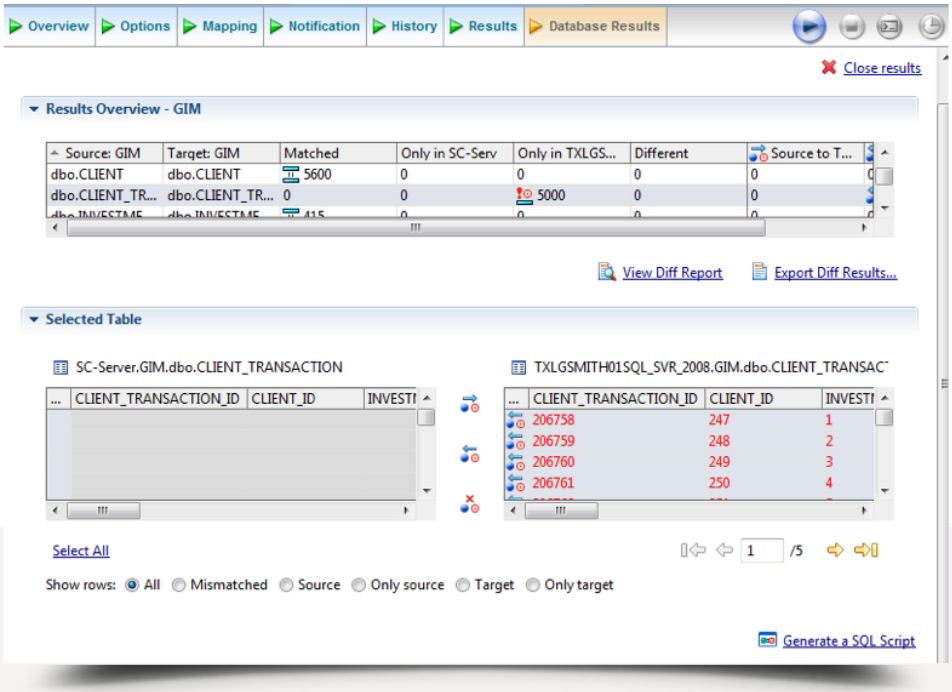
Job Scheduling

DB Change Manager can also be used to schedule any of these jobs or run them from the command line once the job has been saved.

Data Masking

Data masking is a way of securing sensitive data during the development or testing phases of a database development project. It is often performed as a security or compliance measure that protects important information. By masking valid production data, you can provide a copy of the data that is “scrambled” but still represents your production environment.

DB Change Manager lets you specify masking rules for moving data between a source and a target in a data comparison job. You can set rules for individual columns, tables, and entire databases. When you run a data comparison with the Automatically Synchronize option on, the data on the target is replaced with data from the source and any items configured with a masking rule will be masked. You can then use the masked data in your development and testing environments.



CONCLUSION

Data professionals need useful tools to effectively manage the complex landscape of third-party applications and databases. Good data quality can be achieved from properly building out and maintaining your organization's data warehouse. IDERA ER/Studio Enterprise Team Edition and DB PowerStudio help you develop a workflow to build data warehouse models and manage the third-party databases, so that you can fully define the environment and track any changes to the schema or data. Gain better control and visibility of your data warehouse and third-party databases with IDERA.

IDERA understands that IT doesn't run on the network – it runs on the data and databases that power your business. That's why we design our products with the database as the nucleus of your IT universe.

Our database lifecycle management solutions allow database and IT professionals to design, monitor and manage data systems with complete confidence, whether in the cloud or on-premises.

We offer a diverse portfolio of free tools and educational resources to help you do more with less while giving you the knowledge to deliver even more than you did yesterday.

Whatever your need, IDERA has a solution.