



Product Documentation

ER/Studio® Data Architect

New Features Guide

Version 9.5/XE3

Published December 20, 2012

© 2012 Embarcadero Technologies, Inc. Embarcadero, the Embarcadero Technologies logos, and all other Embarcadero Technologies product or service names are trademarks or registered trademarks of Embarcadero Technologies, Inc. All other trademarks are property of their respective owners.

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, CodeGear™ RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.

December 20, 2012

Contents

- New Features Summary 4
 - General Updates 4
 - DBMS Updates 4
 - New Platform Versions 4
 - New Platforms 5
 - New Operating Systems 5
 - New DBMS Features 5
 - Repository Enhancements 5
- Details of New Features 6
 - Partial Layout 6
 - Custom Datatypes 8
 - Universal Naming Utility Provides Access to Object Editors 10
 - Universal Mappings 11
 - Creating Universal Mappings through the Compare/Merge Utility 12
 - Creating a Universal Mapping using the Universal Mapping Editor 15
 - Viewing Universal Mappings 17
 - GENERATED BY DEFAULT for DB2 UDB 9.x 18
 - Range-partitioned Universal Tablespaces for DB2 z/OS 9.x and 10.x 19
 - ORGANIZE ON for Netezza 6.0 22
 - COLLECT STATISTICS for Teradata 12.0 and 13.x 24
 - CHECKSUM for Teradata 12.0 and 13.x 26
 - Group Support in Repository Security Center 27

NEW FEATURES SUMMARY

The new features supported by ER/Studio Data Architect 9.5 fall into the following categories.

- [General Updates](#)
- [DBMS Updates](#)
- [Repository Enhancements](#)

GENERAL UPDATES

Graphics Improvements: You can now select multiple objects (entities, views, and shapes) and apply a layout command. All relationships, view relationships or lines that belong to the selected objects will be affected by the layout. You can use existing layout commands from the menu and toolbar. If you want to layout the entire submodel, no objects should be selected. For more information, see [Partial Layout](#).

Custom Datatypes: Using the Datatype Mapping Editor, you can now create your own custom datatypes. For more information, see [Custom Datatypes](#).

Universal Naming Utility: You can now access Object Editors from within this utility. For more information, see [Universal Naming Utility Provides Access to Object Editors](#)

Universal Mappings: Mappings you create for Tables, Attributes, Views and Shapes are now automatically matched when comparing and merging models. In addition, mapping you force in the Compare and Merge Utility can be saved in the target model for use in future compare/merge operations. You can see and edit the mappings you create on the Where Used tab of the object. For more information, see [Universal Mappings](#).

DBMS UPDATES

NEW PLATFORM VERSIONS

IBM DB2 z/OS 10.x

Microsoft SQL Server 2012

Netezza 6.0

Teradata 13.x

PostgreSQL 9.x

NEW PLATFORMS

GreenPlum 4.x

NEW OPERATING SYSTEMS

Windows 8

NEW DBMS FEATURES

DB2 LUW 9.x: TIMESTAMP datatypes now support generated by default option. For more information, see [GENERATED BY DEFAULT for DB2 UDB 9.x](#).

DB2 z/OS 9.x and 10.x: Range-partitioned Universal Tablespaces. For more information, see [Range-partitioned Universal Tablespaces for DB2 z/OS 9.x and 10.x](#).

Netezza 6.0: Organizing keys now supported. For more information, see [ORGANIZE ON for Netezza 6.0](#).

Oracle 10g and 11g: Capacity planning of tables can now be based on the average row size.

Teradata 12 and 13: ST_GEOMETRY datatype is now supported.

COLLECT STATISTICS and CHECKSUM clauses are now supported. For more information, see [COLLECT STATISTICS for Teradata 12.0 and 13.x](#) and [CHECKSUM for Teradata 12.0 and 13.x](#)

PostgreSQL 8.0 and 9.x: Support has been added for uuid, money, xml, and serial datatypes; uuid is mapped to the UNQUEID logical type; money is mapped to MONEY logical type, xml is mapped to XML logical type; and serial is mapped to SERIAL/INTEGER logical type.

REPOSITORY ENHANCEMENTS

Windows 8: Now supported for ER/Data Architect Repository.

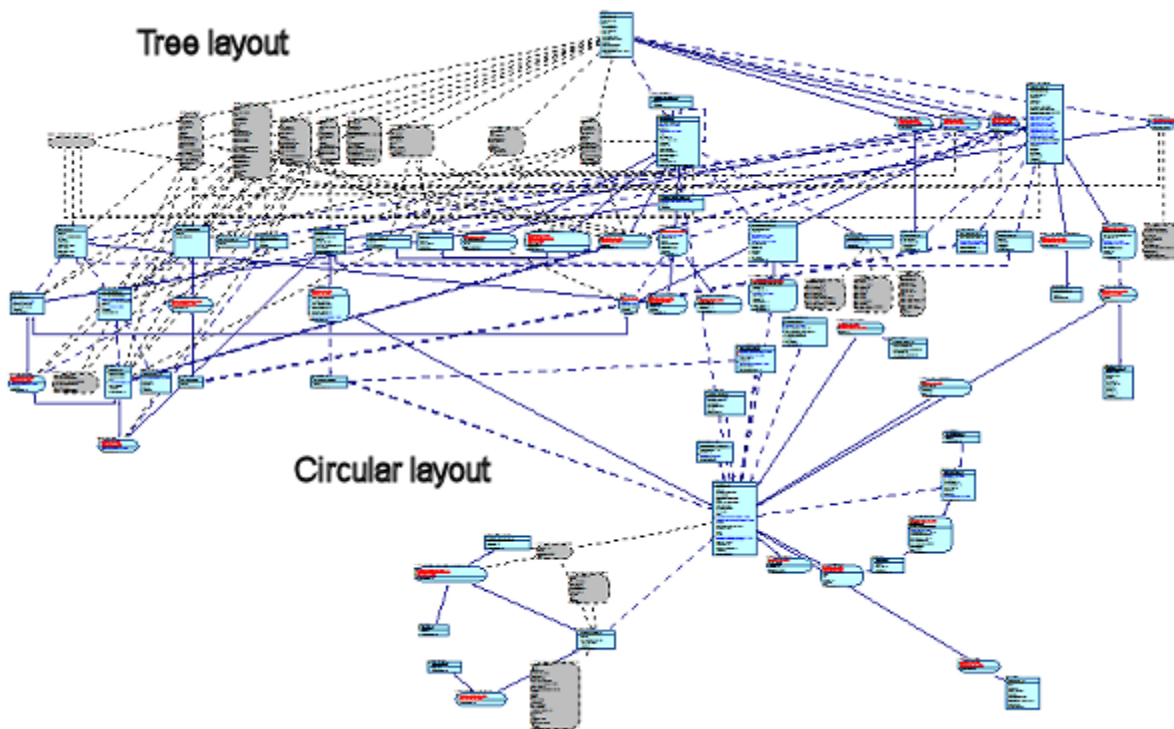
Security Center: Security groups are now supported. For more information, see [Group Support in Repository Security Center](#)

DETAILS OF NEW FEATURES

PARTIAL LAYOUT

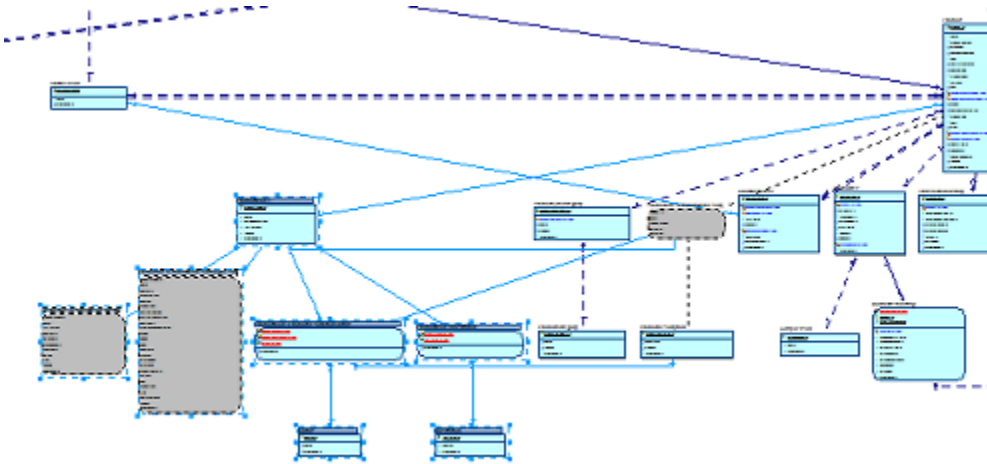
You can now select multiple objects (entities, views, and shapes) and apply a layout format to that selection only. All relationships, view relationships or lines that belong to the selected objects will be also affected by the layout. This allows you to choose a different layout style to draw attention to specific parts of the model, such as objects recently added or a special relationship that exists between certain entities. If you don't have any objects selected, the layout style chosen will apply to the entire model.

In this example, both Tree and Circular layouts are used.



To change the layout of a portion of model:

- 1 Drag across the relevant objects to make a broad selection.



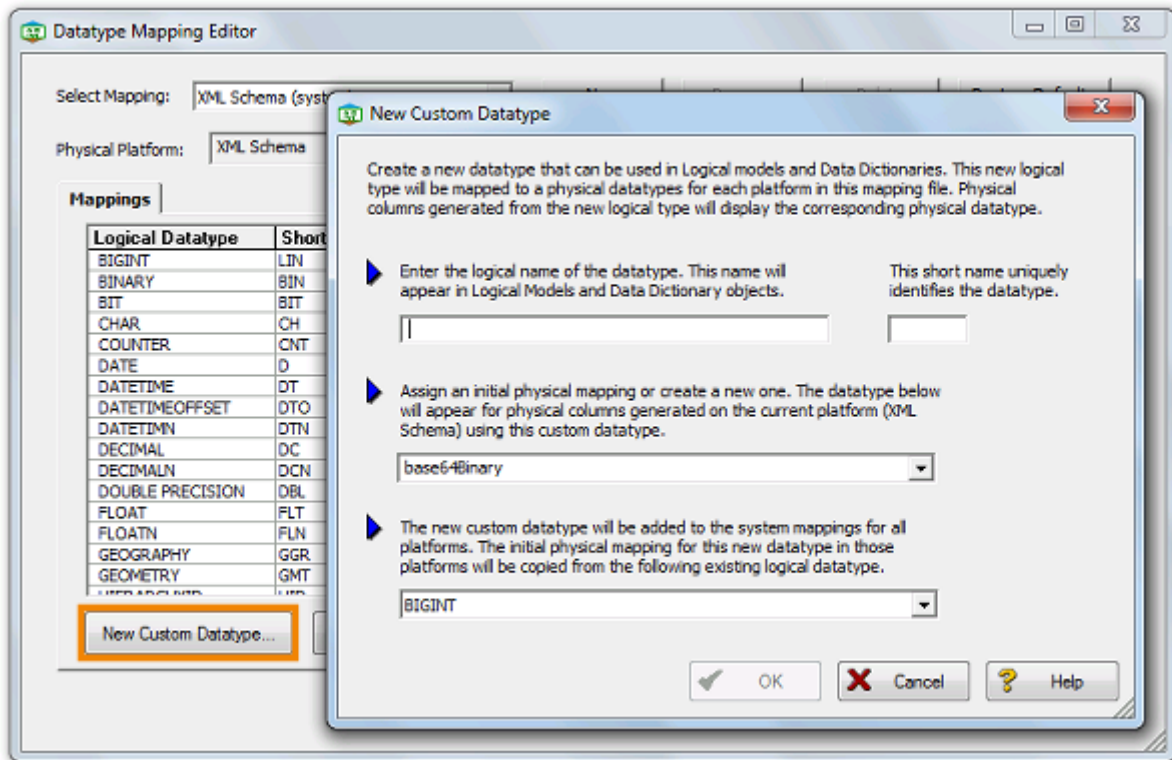
- 2 Control-click to add or remove entities, views, shapes, relationships, view relationships, and lines from the selection as required.
- 3 Choose the layout style to apply to the selection from either the **Layout** menu or by clicking the appropriate layout icon on the toolbar, and click **OK** on the warning dialog that appears..

Only the objects you selected are laid out using the different style.

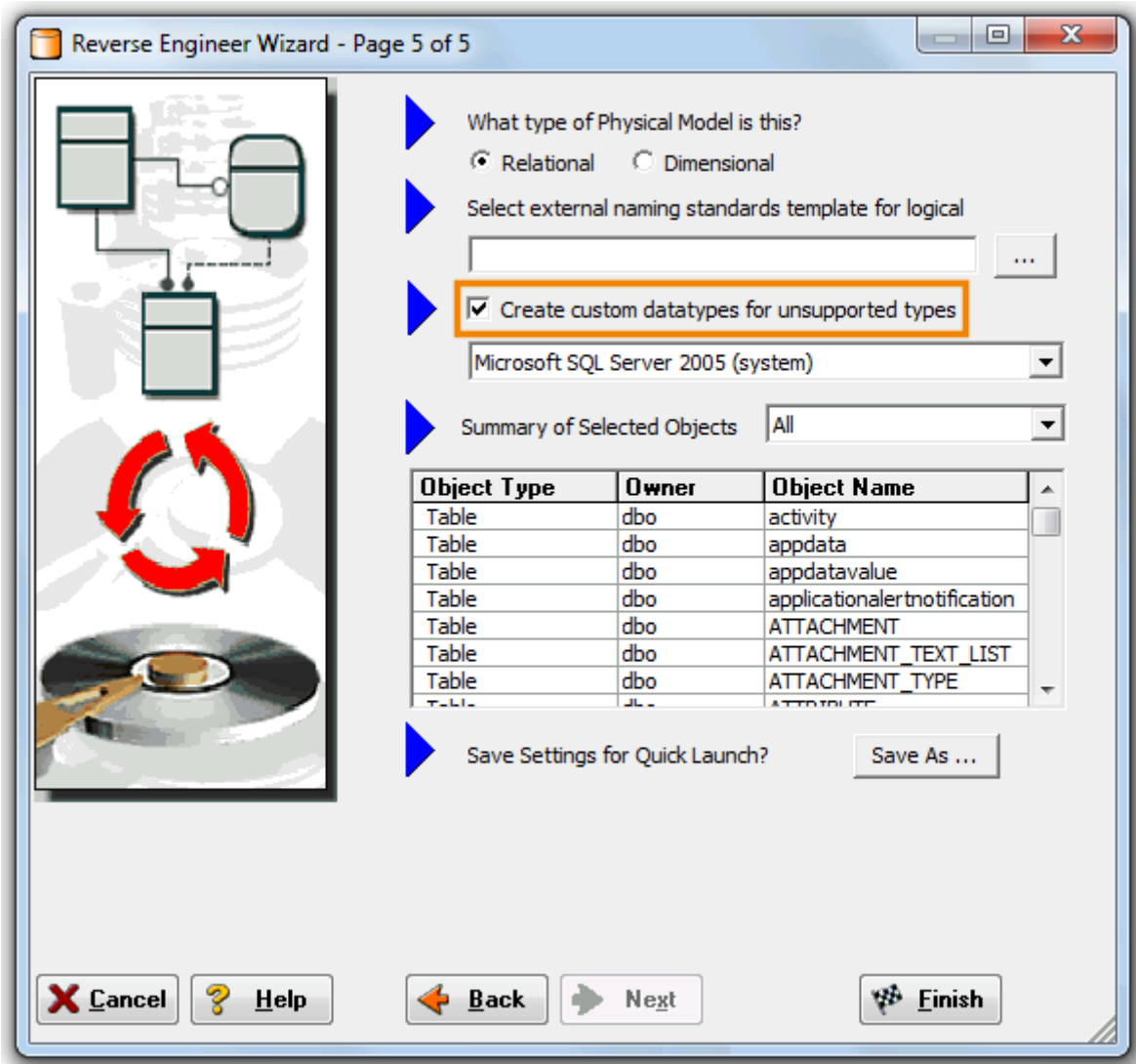
CUSTOM DATATYPES

Some platforms have datatypes that are not supported by ER/Studio Data Architect. You can define custom datatypes in the Datatype Mappings editor to use so unsupported datatypes will not be assigned a default datatype, such as CHAR (10) when importing, reverse engineering or merging databases that contain those unsupported datatypes.

The Datatype Mapping Editor has been updated to accommodate this new functionality.



A new option, **Create custom datatypes for unsupported types** has been added to the Reverse Engineering Wizard, which if selected, any datatypes not recognized will be added as new logical and physical types to the selected mappings file.



If the selected mappings file contains new datatypes because you added them or they were added by a previous reverse-engineering process, then those datatype mappings will be used.

NOTE: Width and scale information of unsupported and unmapped datatypes is not retrieved in the reverse-engineering process. If you need this information, create a custom datatype manually through the Datatype Mapping Editor before reverse-engineering.

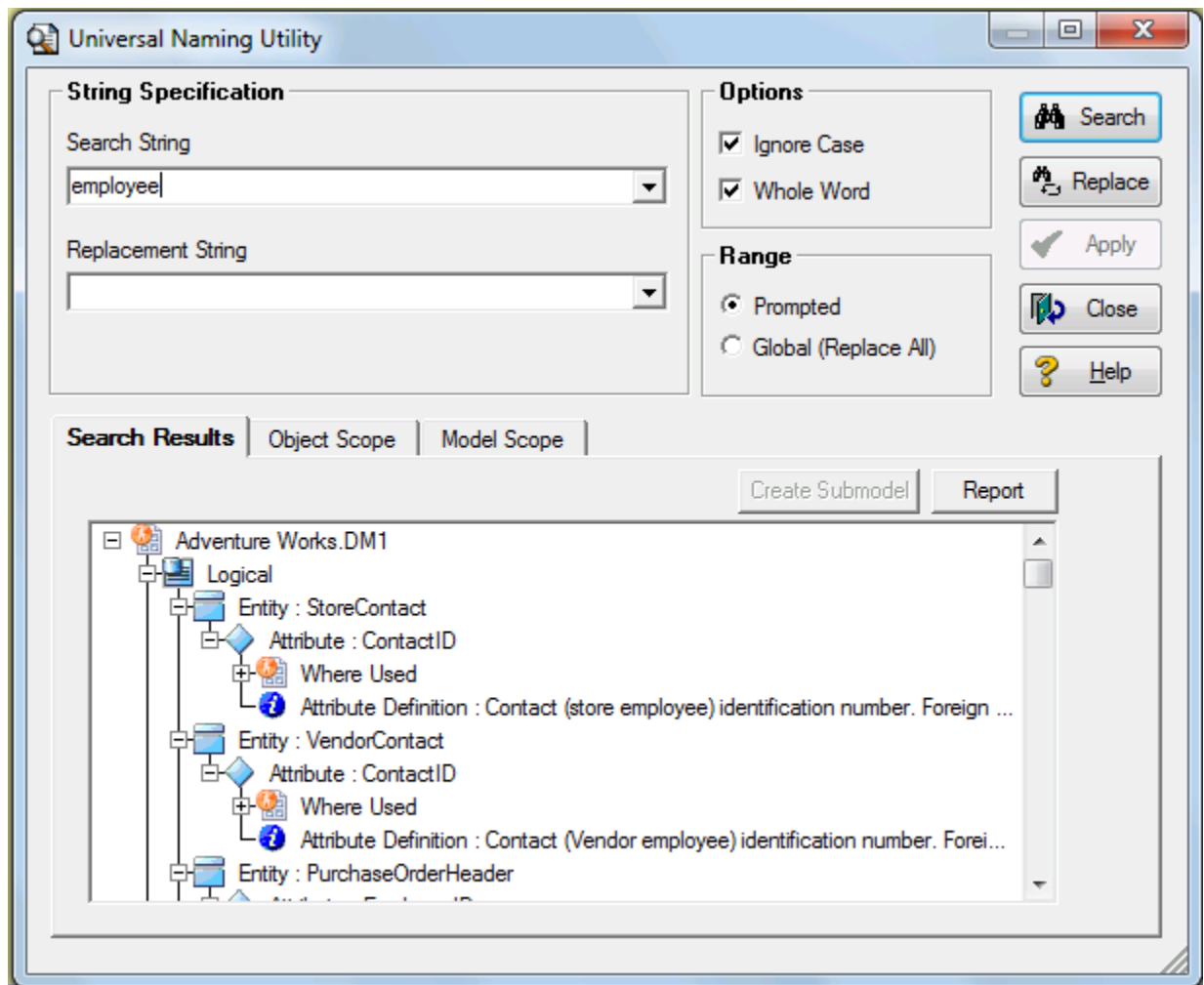
Usage Example

The PERIOD datatype for Teradata 13.x is not supported natively by ER/Studio. However, you can reverse engineer columns from a Teradata database that uses the PERIOD datatype by checking the option to create a custom datatype.

UNIVERSAL NAMING UTILITY PROVIDES ACCESS TO OBJECT EDITORS

On the **Search Results** tab of the Universal Naming Utility (UNU,) you can double-click on an object and open the editor for that object. In this process, you are asked to save any changes you made in the UNU; the UNU closes; the object is located and selected in Data Model Data Dictionary, or Data Lineage tree; and the object editor opens with the object selected.

- 1 From the **Tools** menu, select **Universal Naming Utility**.
- 2 Enter a **Search String** and then click **Search**.
- 3 On the **Search Results** tab, double-click any object and the editor for that object appears.



For example, if you double-click an **Entity**, the **Entity Editor** appears. If you double-click a **Procedure**, the **Procedure SQL Editor** appears.

UNIVERSAL MAPPINGS

Universal Mappings have been implemented to support enterprise-level modeling in large organizations with many systems and overlapping data and processes. They allow you to map objects between models in the same dm1 file, between models in separate dm1 files, and between an ER/Studio Data Architect diagram and a ER/Studio Business Architect workspace in the Repository. These mappings are retained and can be used when comparing and merging. They are automatically applied in the **Compare and Merge Utility** if they exist.

What is a mapping?

A mapping is a link between two objects. There are two types of mappings that serve different purposes.

- **Mappings within a design layer:** This horizontal type of mapping shows reuse within a design layer, such as logical to logical, physical to physical and conceptual to conceptual. The scope of this mapping is limited to mapping objects of the same type, such as entity to entity, table to table, and column to column.
- **Mappings across different design layers:** This vertical type of mapping shows traceability or where used, upstream and downstream across different design layers, such as: conceptual to logical; logical to physical; and logical and physical.

Why are cross-repository mappings important?

- **Impact Analysis:** Understanding where common objects are located. Traceability between conceptual and logical models.
- **Scalability:** A mapping system across dm1s and workspaces reduces the need for large models that have everything in them. You can now perform Where Used (ER/Studio DA) or Impact Analysis (ER/Studio BA) that can include information on multiple dm1 or workspace files. Organizing objects into different dm1s or workspaces and then mapping related objects scales much better.
- **Performance:** Smaller dm1 or workspaces improves performance and manageability.
- **Enterprise Modeling:** Facilitates an enterprise view of your data. Typically this starts with identifying subject areas, then conceptual entities and then logical/physical models from the conceptual. Once the hierarchy is in place, you can see upstream from logical/physical objects and downstream from conceptual and process models.

The following topics help you understand how Universal Mappings are implemented in ER/Studio:

- [Creating Universal Mappings through the Compare/Merge Utility](#)
- [Creating a Universal Mapping using the Universal Mapping Editor](#)
- [Viewing Universal Mappings](#)

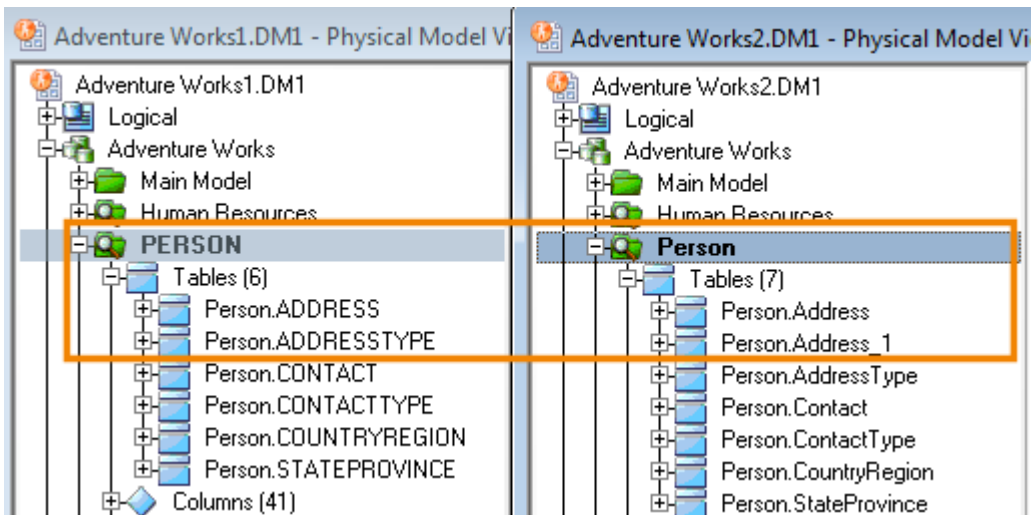
CREATING UNIVERSAL MAPPINGS THROUGH THE COMPARE/MERGE UTILITY

The following illustrates the process of creating and using a **Universal Mapping**:

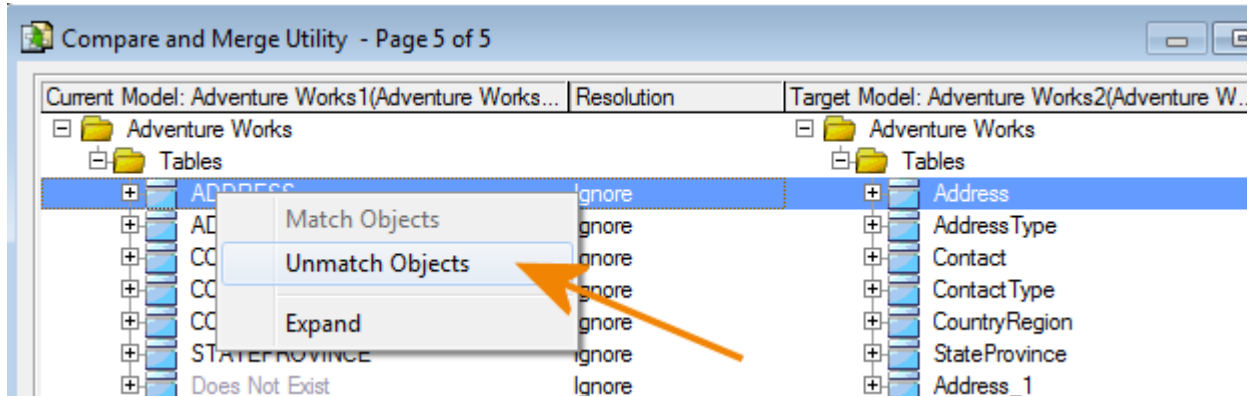
NOTE: **User-Defined Mappings** can be used in a similar way when comparing between two models in the same dm1 file. Existing user-defined mappings will now be considered when matching in the compare/merge, and you will have the option to save matches as **User-Defined Mappings** for intra-diagram comparisons.

NOTE: The precedence for matches in the Compare/Merge Utility is as follows, from high to low precedence: force matches, universal mappings, name matches, attribute sequence number and datatype matches. In other words, if the datatype matches between two attributes but the universal mapping matches one of those attributes to a different attribute, then the second match will be automatically applied. You can use **Force Match** in the **Compare and Merge Utility** to implement a different match.

In the following image, we see two dm1 files that differ slightly.

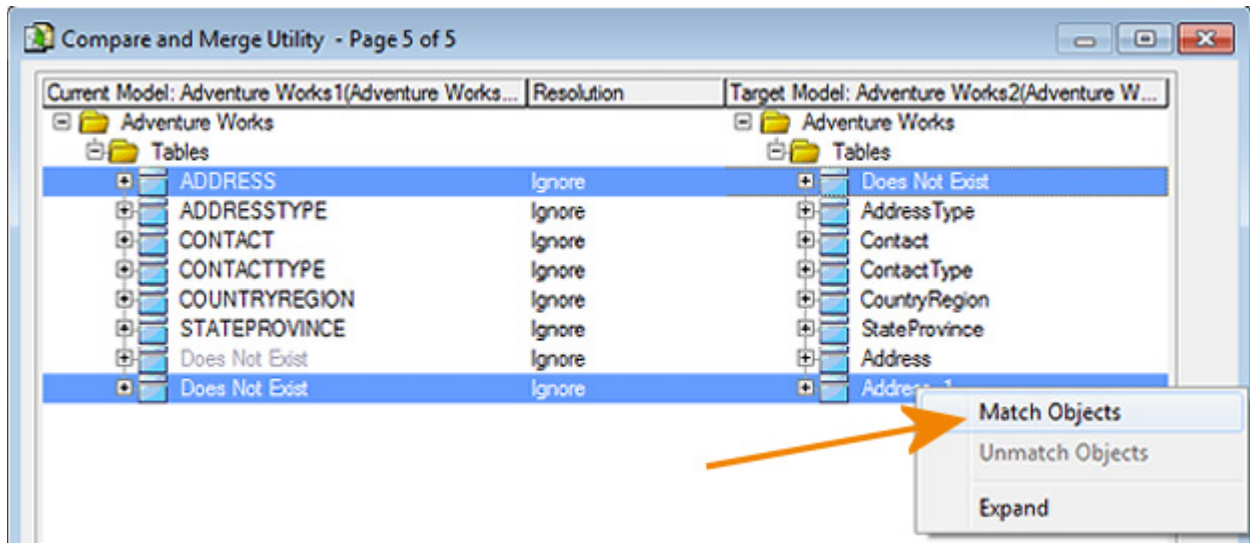


The **Compare and Merge Utility** automatically matches Person.ADDRESS to Person.Address, but if we want to match Person.ADDRESS to Person.Address1, then we can unmatched these attributes and create a different match.

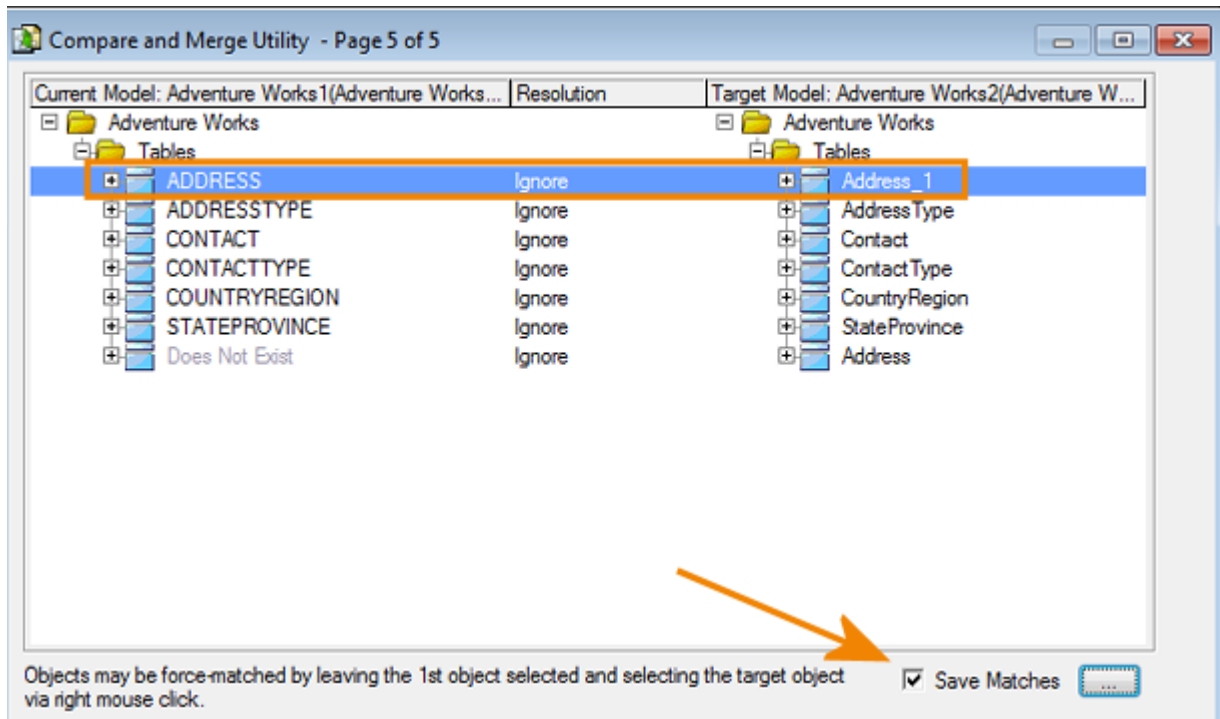


To illustrate the power of Universal Mapping, we can match two objects and then have the matches saved.

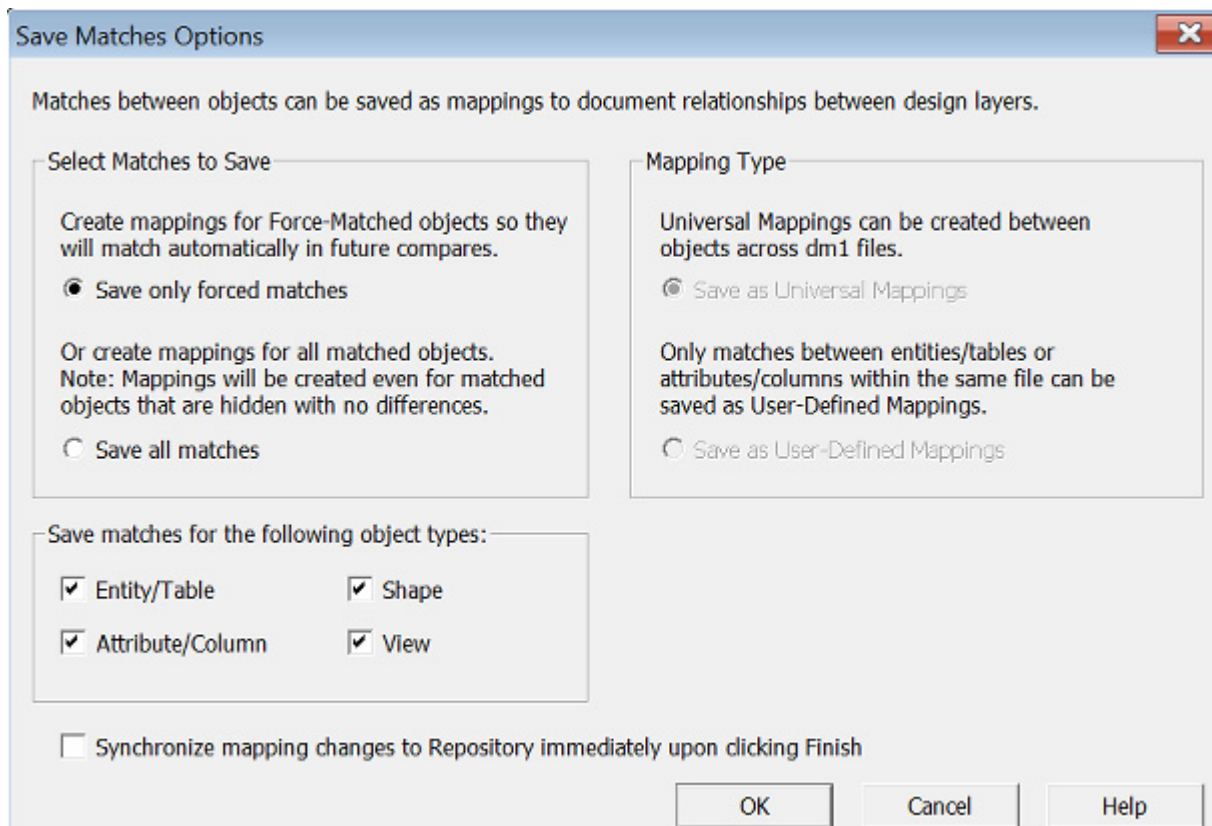
As indicated in the **Compare and Merge Utility**, "Objects may be force-matched by leaving the 1st object select and selecting the target object via right mouse click."



Now that the match is made, we can choose to save the match to use the next time we compare these two models. The matches are saved in both the current and target models.



You can click the ellipses to see the **Save Matches Options** and make your choices.



A checkbox for each object type supported, Entity/Table, Attribute/Column, View, and Shapes is listed and all are selected by default. Only objects of the types selected are eligible to have matches saved as mappings.

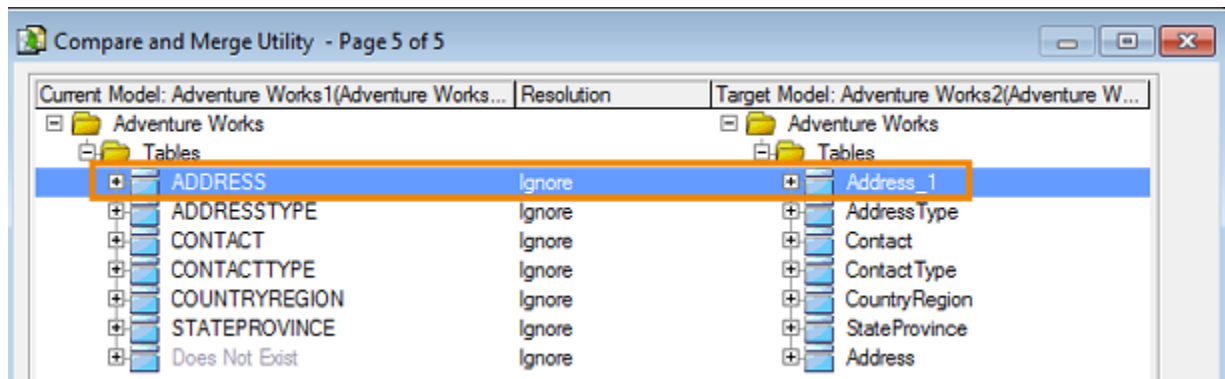
When comparing models in the same diagram, you can choose between **Save as User-Defined Mappings** or **Save as Universal Mappings**.

When comparing models in different diagrams, **Save as Universal Mappings** is selected. This is the only option available.

If you are comparing two diagrams in the Repository, the option to **Synchronize mapping changes to Repository immediately upon clicking Finish** becomes available.

Click **OK** on the **Save Matches Options** and then click **Finish** to complete the compare/merge.

When you run the **Compare and Merge Utility** again, you will see that the matches you forced and saved are used by default.



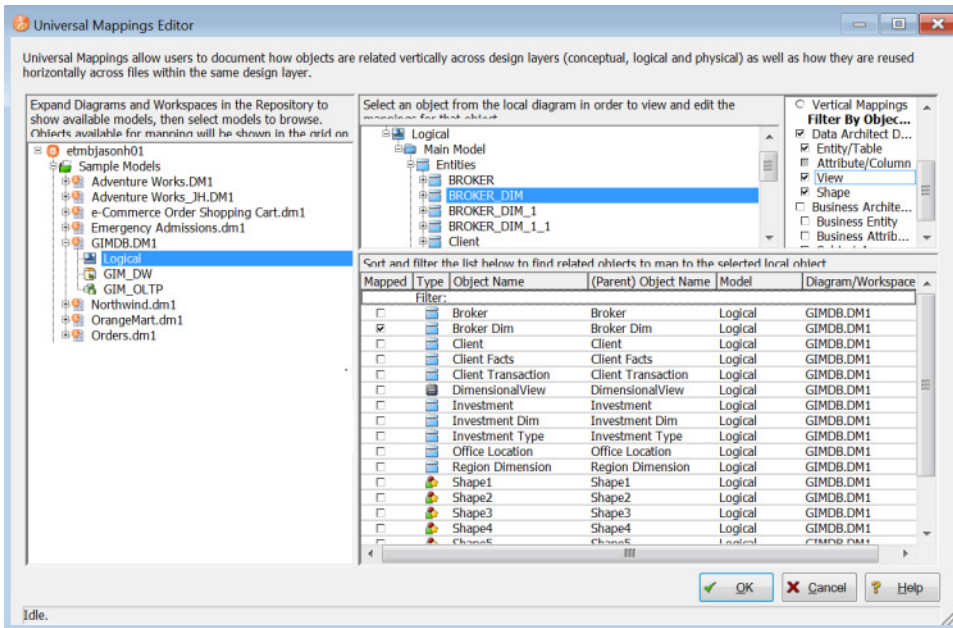
CREATING A UNIVERSAL MAPPING USING THE UNIVERSAL MAPPING EDITOR

The Universal Mappings Editor provides the following functionality:

- **Local Objects** You can edit the Universal Mappings of a source object without exiting the Universal Mapping Editor.
- **Browse/Filter/Search:** You can view objects in other diagrams and workspaces that can be mapped to the currently selected object. Because the number of objects can potentially be very large, you can narrow down the list by using the filters available and sorting the related objects.

- **Existing Mappings:** You can see which objects are already mapped, as well as the description and any other data associated with the mapping.

- 1 In the diagram window, right-click the object you want to map or change the mapping for and select **Universal Mappings**.



- 2 In the Universal Mapping Editor, in the center of the window, as indicated onscreen, **Select an object from the local diagram in order to view and edit the mappings for that object.**

- 3 On the far left of the screen select a model in the tree. Once you have selected the model, the data for it is retrieved from the Repository.

The list of mappable objects that displays on the bottom right of the editor is generated from your selection in the Repository tree. If you select a diagram node, then all objects in its models will be included in the mappable object list. If you select a Project, then all objects in diagrams in that project will be included.

- 4 On the far right of the editor, select the filters to use in order to see the objects of those type you can map to.

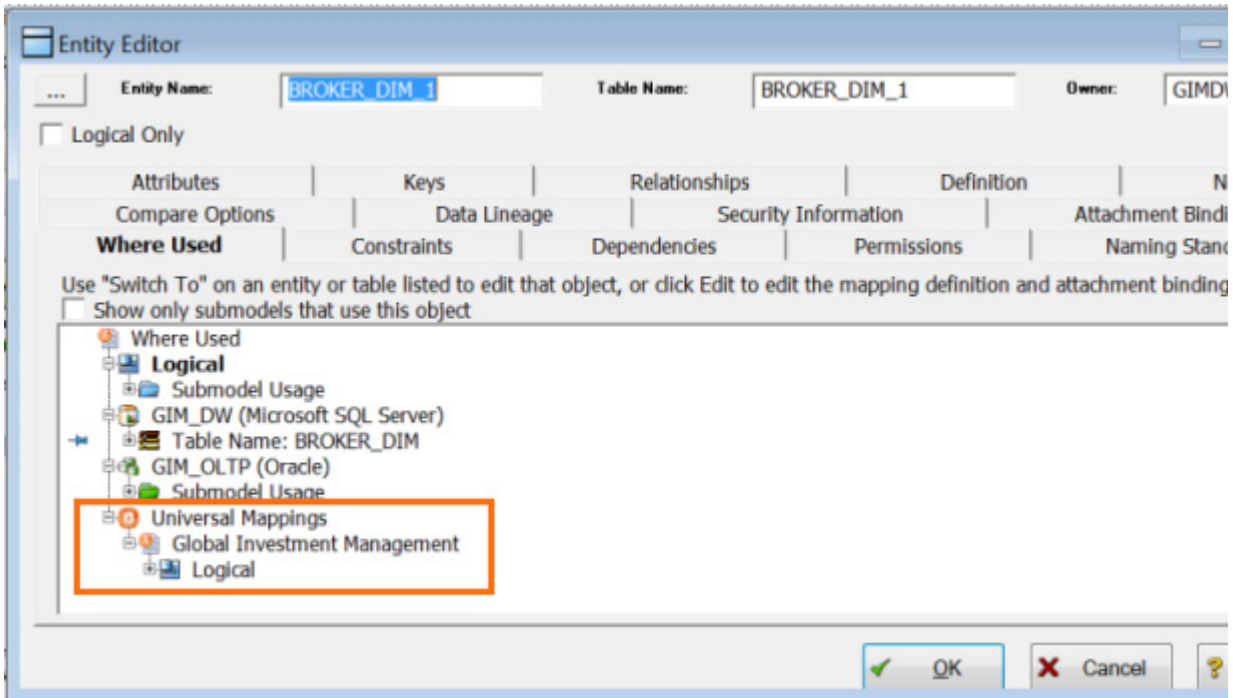
NOTE: You can map ER/Studio DA entities/tables, attributes/columns, views, shapes and ER/Studio Business Architect entities, attributes, and subject areas to each other, including mapping ER/Studio DA objects to ER/Studio BA objects.

- 5 At the bottom of the editor, click a column name to reorder the list of related objects to find the object to map to the local object.
- 6 Deselect or select the target object's **Mapped** checkbox to remove or add a mapping.
- 7 Click **OK** to complete the change.

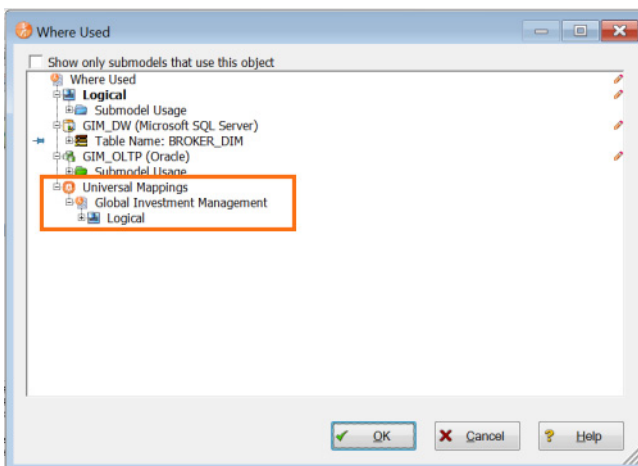
- 8 Save your diagram and in the case of Repository diagrams, check in the affected object to save the mapping change to the Repository.

VIEWING UNIVERSAL MAPPINGS

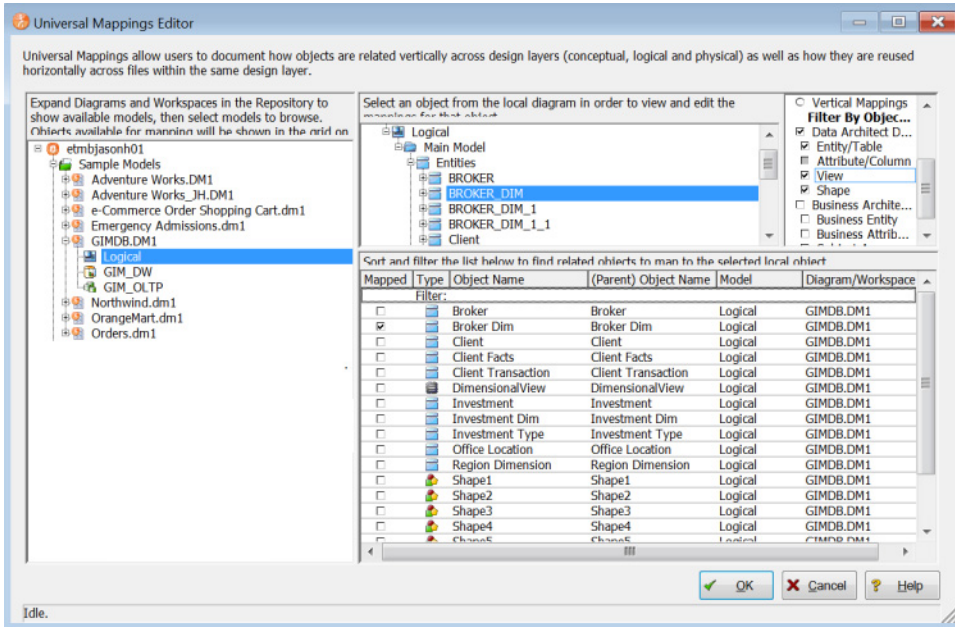
Universal Mappings are displayed in the object's **Where Used** tab.



You can also see the mappings of an object by right-clicking the object and selecting **Where Used**.

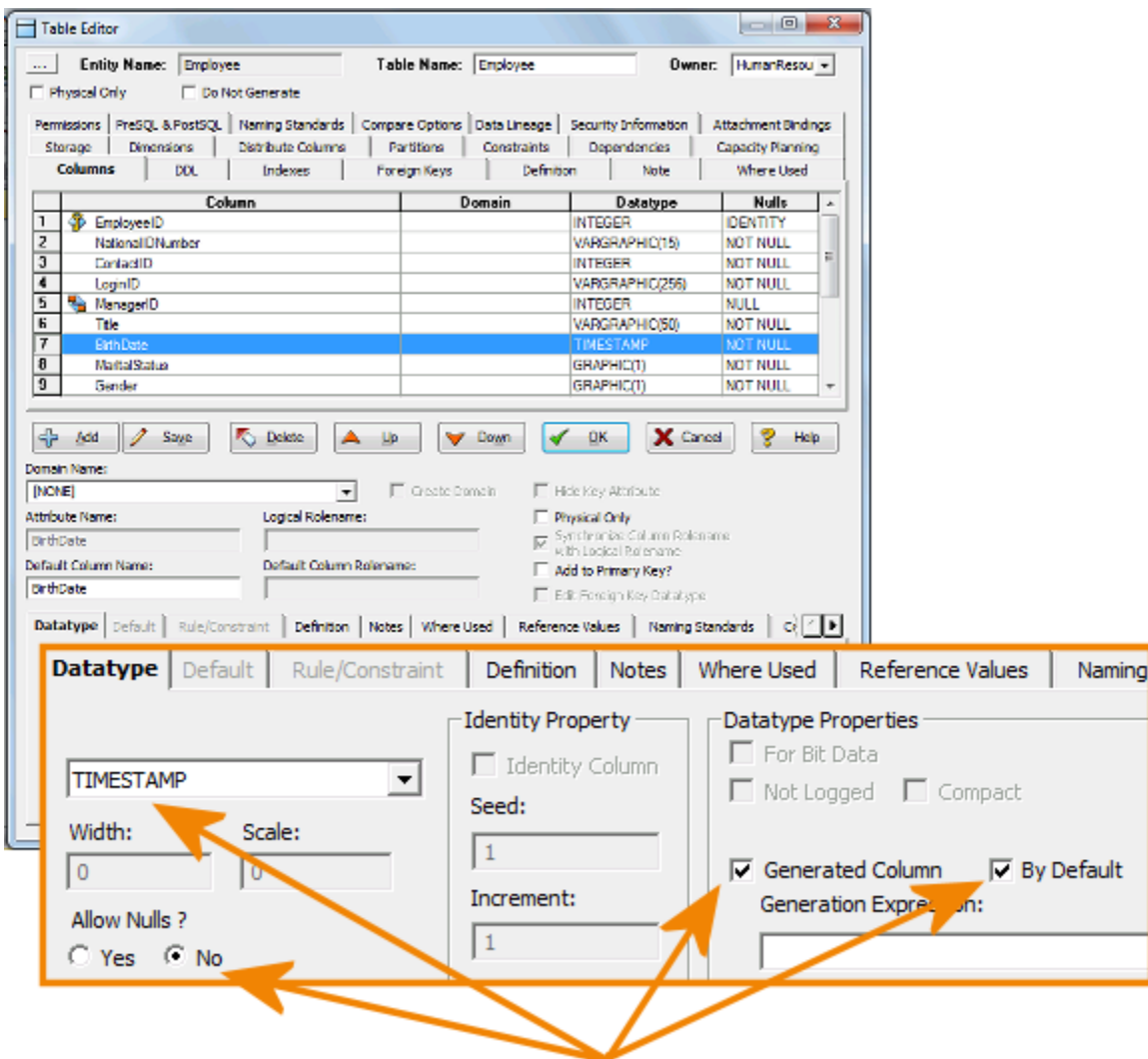


And you can see the mappings of an object in the **Universal Mappings Editor**.



GENERATED BY DEFAULT FOR DB2 UDB 9.X

A generate **By Default** option is now available for a **TIMESTAMP** datatype if the column does not allow nulls (**Allow Nulls?** is set to **No**), and you have selected **Generated Column**.



The following is an example of the SQL produced for a column when these options are selected
 HireDate TIMESTAMP NOT NULL GENERATED BY DEFAULT FOR EACH ROW ON UPDATE AS ROW
 CHANGE TIMESTAMP.

RANGE-PARTITIONED UNIVERSAL TABLESPACES FOR DB2 z/OS 9.X AND 10.X

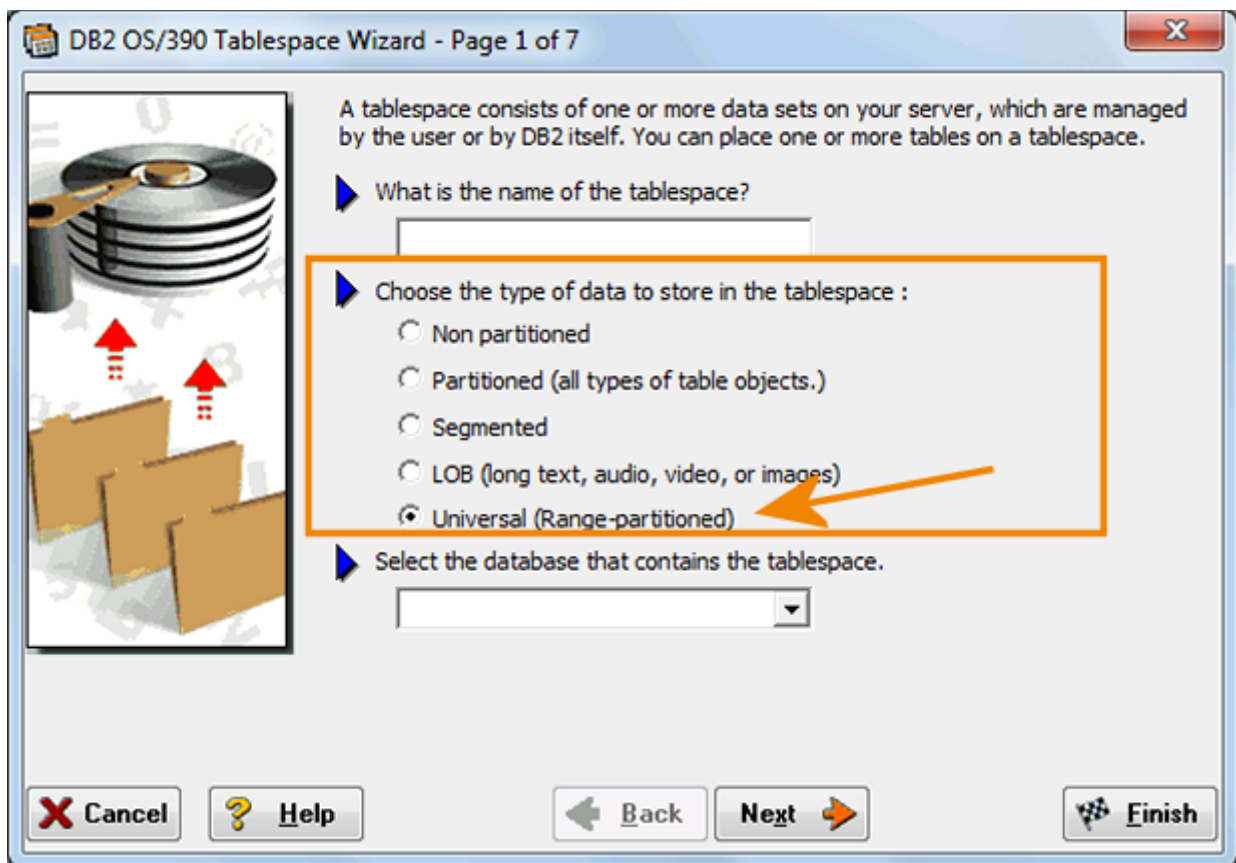
The Tablespace Editor now for IBM DB2 z/OS (OS/390) 9.x and 10.x lets you specify a new type of tablespace, **Universal (Range-partitioned)**.

A range-partitioned tablespace is segmented and based on partitioning ranges. It contains a single table and you can create any type of index on tables in the range-partitioned tablespace.

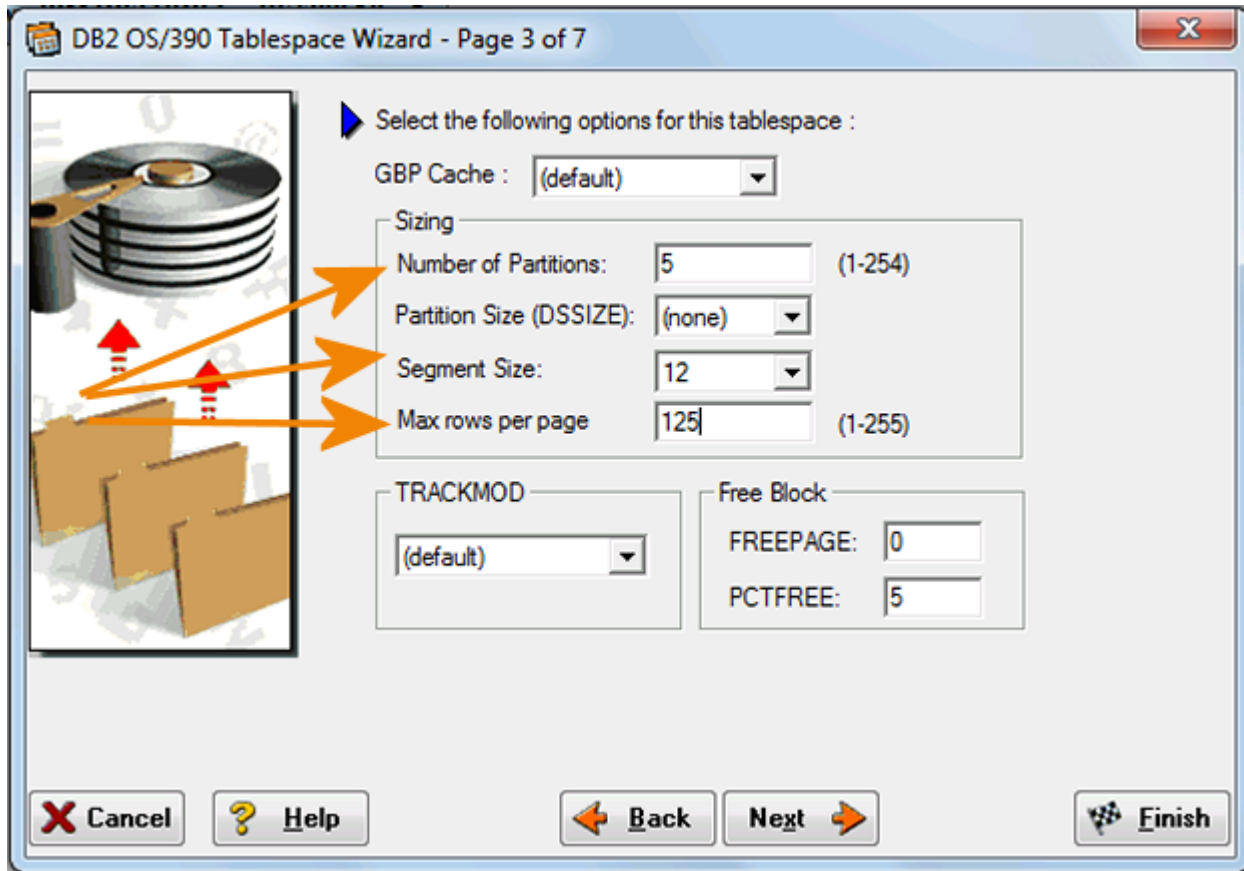
The following describes the minimal steps necessary to create a universal range-partitioned tablespace.

- 1 In the Data Model Explorer tree, for a DB/2 OS/390 9.x or 10.x physical model, right-click the Tablespace object and select **New Tablespace**.
- 2 Enter a name for the tablespace and then select **Universal (Range-partitioned)**.

TIP: You can also specify the type of data to store in the tablespace on the **Name & Type** tab of the **Tablespace Editor**.



- 3 Select other options you want on page 1 and page 2.
- 4 On page 3 of the Tablespace Wizard, specify the **Number of Partitions**.
- 5 Select a value in the **Segment Size** list.



6 Specify the **Max rows per page** or accept the default of 255.

7 Click **Finish**.

The following is an example of the DDL produced when creating a universal range-partitioned tablespace.

```
CREATE TABLESPACE URPT
```

```
  NUMPARTS 5
```

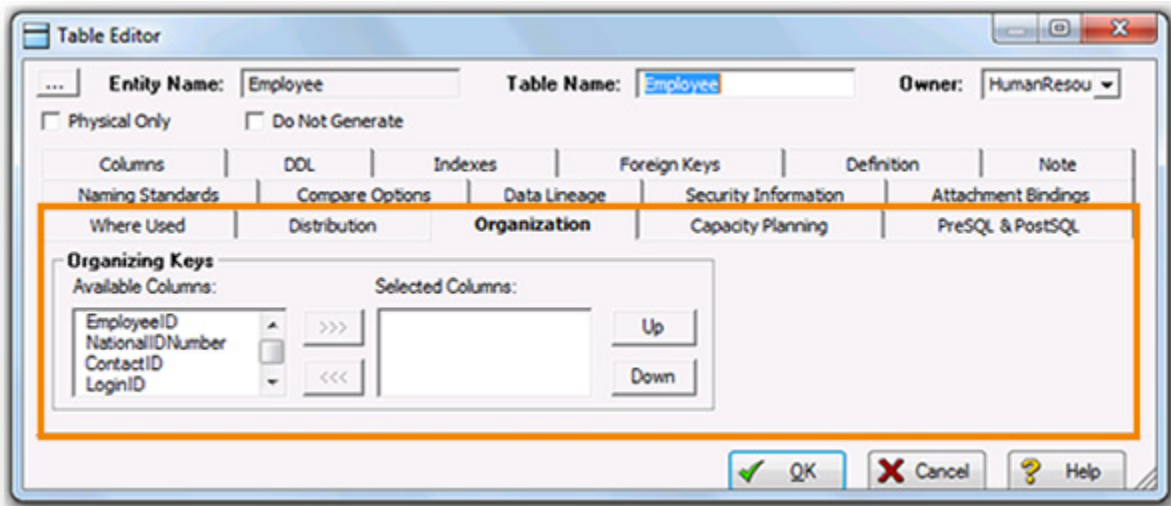
```
  SEGSIZE 12
```

```
  MAXROWS 125
```

```
:
```

ORGANIZE ON FOR NETEZZA 6.0

Organizing keys are now supported for the Netezza platform. The Table editor has been expanded by adding a new Organization tab for the Netezza 6.0 platform.



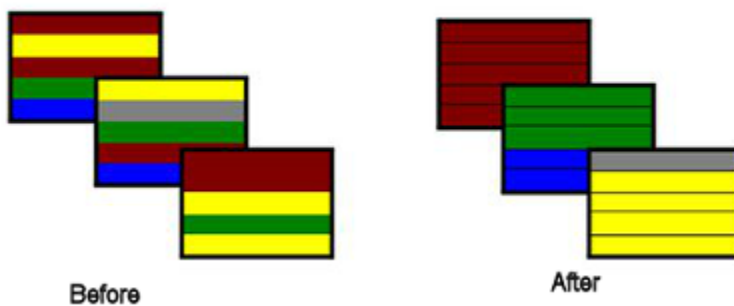
On the **Organization** tab of the **Table Editor** for the Netezza 6.0 platform, you can select up to four columns of a table on which to base the organization of the table. Tables with columns specified in the ORGANIZE ON clause are called Clustered Base Tables (CBT). Organizing the table records in this way causes Netezza to save the selected table records in the same or nearby extents. This can enhance query performance and disk I/O.

The following is an example of the SQL produced by specifying organizing keys:

```
ORGANIZE ON (EmployeeID, ManagerID)
```

CBTs are most often used for large fact or event tables with many millions or billions of rows. Scanning records of such large tables requires full disk scans to gather the relevant records. Organizing tables to correspond with your queries can take advantage of zone maps to improve performance.

Using ORGANIZE ON Clause

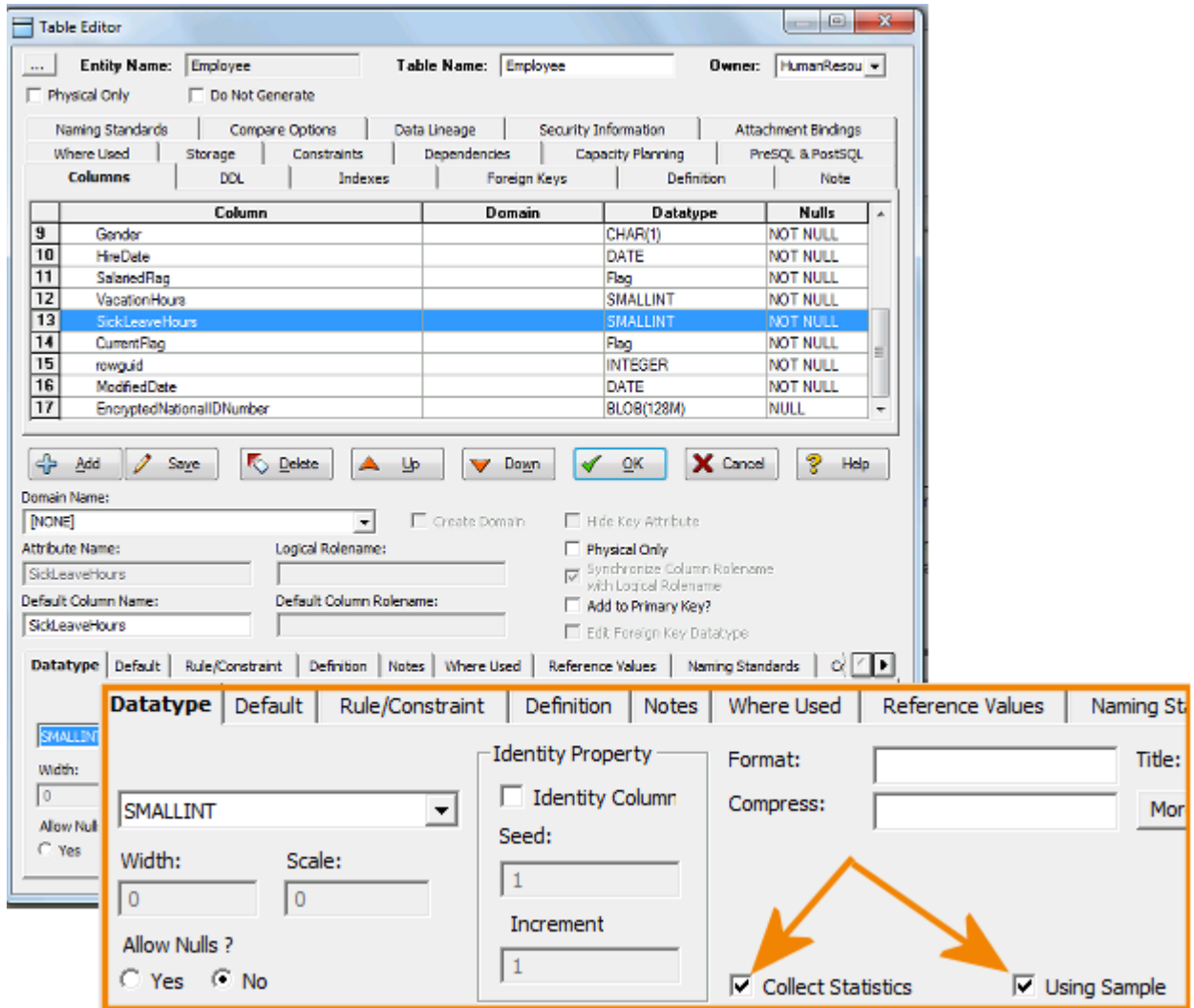


- **Available Columns:** Displays all the columns available to add to the organization key. Select the column you want to add to the organizing keys and move it to the Selected Columns box. Use the left and right arrows to move columns to and from the Selected Columns box. You can select a maximum of four columns to use in the organizing key.
- **Selected Columns:** Displays the columns that make up the organizing key.
- **Up/Down** buttons: Let you reorder the columns in the organizing key. The column order can affect the access speed. The most frequently accessed columns should be at the top of the Selected Columns list.

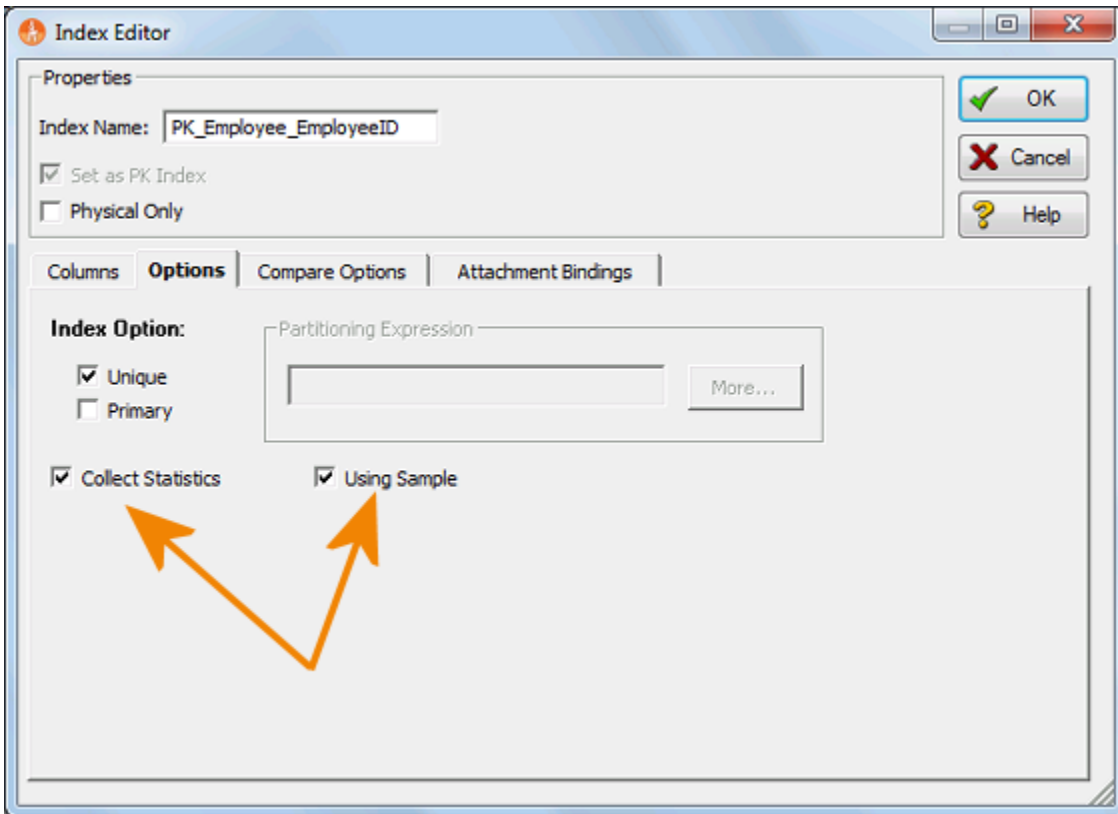
COLLECT STATISTICS FOR TERADATA 12.0 AND 13.X

The Table and Index editors have been updated to accommodate **Collect Statistics** options for the Teradata platform.

The following shows the new **Collect Statistics** and **Using Sample** options on the **Datatype** tab of the **Table Editor**.



The following shows the new **Collect Statistics** and **Using Sample** options on the **Options** tab of the **Index Editor**.



The following describes the purpose of these options and their effect on the SQL generated for the physical model:

Collect Statistics: (`COLLECT STATISTICS`) Collects demographic data for one or more columns of a table, computes a statistical profile of the collected data, and stores the profile in the data dictionary. Collecting full statistics involves scanning the base table, and sorting to determine the number of occurrences for each unique value. The Teradata Optimizer uses this profiling data when it generates its table access and join plans. Full statistics is generally recommended for relevant columns and indexes on tables with less than 100 rows per Access Module Processor (AMP).

This is an example of the SQL generated when **Collect Statistics** has been selected in the **Table Editor** for a column in the table.

```
COLLECT STATISTICS ON table_1 COLUMN column_1;
```

This is an example of the SQL generated when **Collect Statistics** has been selected in the **Index Editor** for an index in the table.

```
COLLECT STATISTICS ON table_1 INDEX unique_1;
```

Using Sample Collecting sampled full-table statistics is generally faster than collecting full table statistics, but may not produce the best table access and join plans. When you specify

this option, Teradata will determine the appropriate percentage to the data to scan. The best choices for USING SAMPLE are columns or indexes that are unique or 95% unique.

This is an example of the SQL generated when **Collect Statistics** and **Using Sample** have been selected in the **Table Editor** for a column in the table.

```
COLLECT STATISTICS USING SAMPLE ON table_1 COLUMN column_1;
```

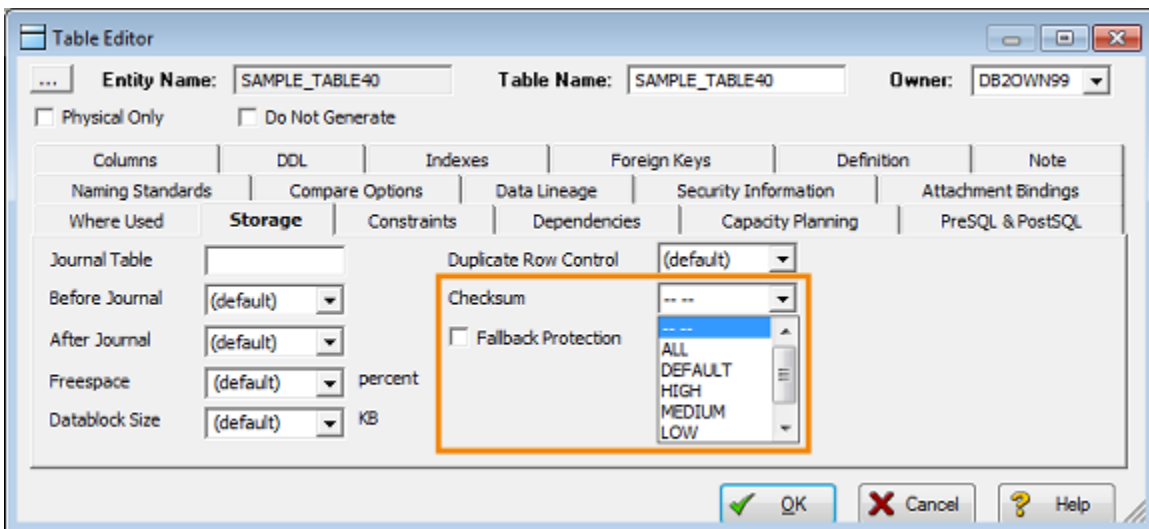
This is an example of the SQL generated when **Collect Statistics** and **Using Sample** have been selected in the **Index Editor** for an index in the table.

```
COLLECT STATISTICS USING SAMPLE ON table_1 INDEX unique_1;
```

CHECKSUM FOR TERADATA 12.0 AND 13.X

The **Table Editor** has been changed to accommodate the new **Checksum** options for the Teradata platform.

The following shows the new **Checksum** and **Medium** options on the **Storage** tab of the **Table Editor**.



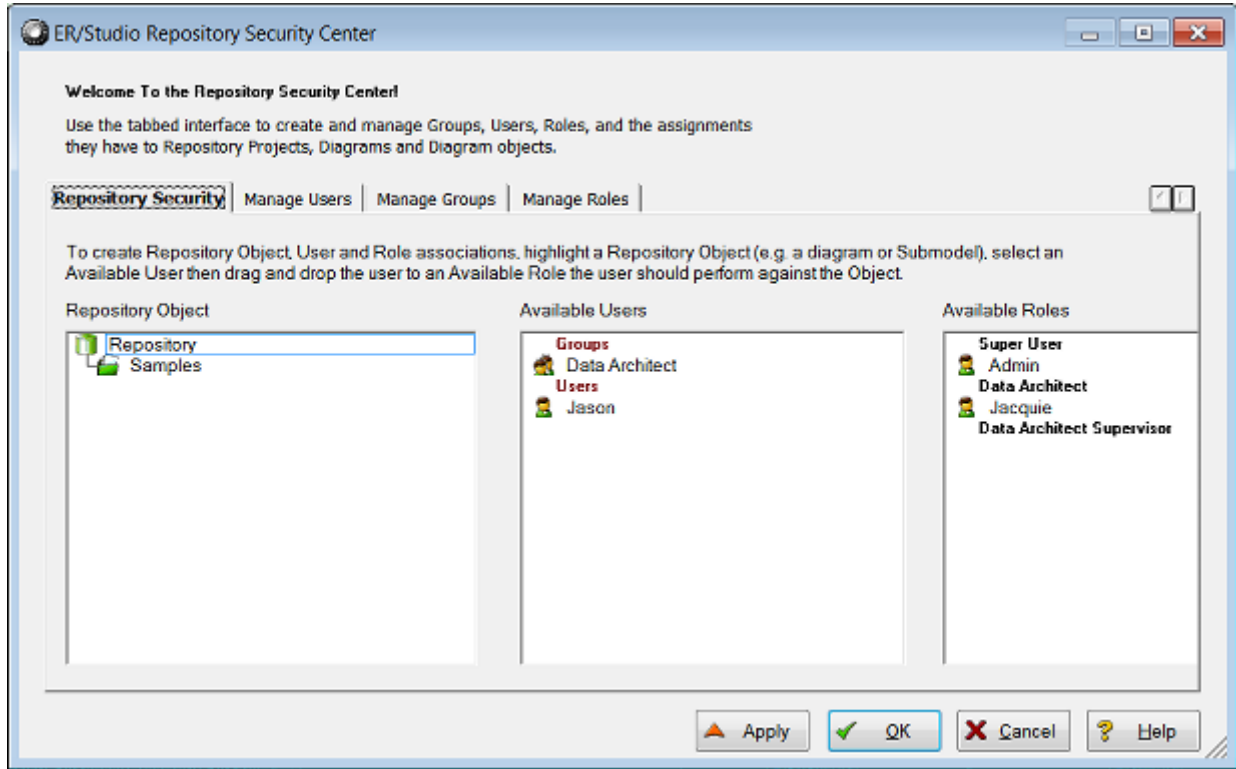
Checksum (CHECKSUM) : Detects and logs disk I/O errors. The choices available determine the percentage of data used to compute the checksum for the table. This is an example of the SQL generated with **Checksum > Medium** has been selected on the **Storage** tab of the **Table Editor**.

Checksum = Medium

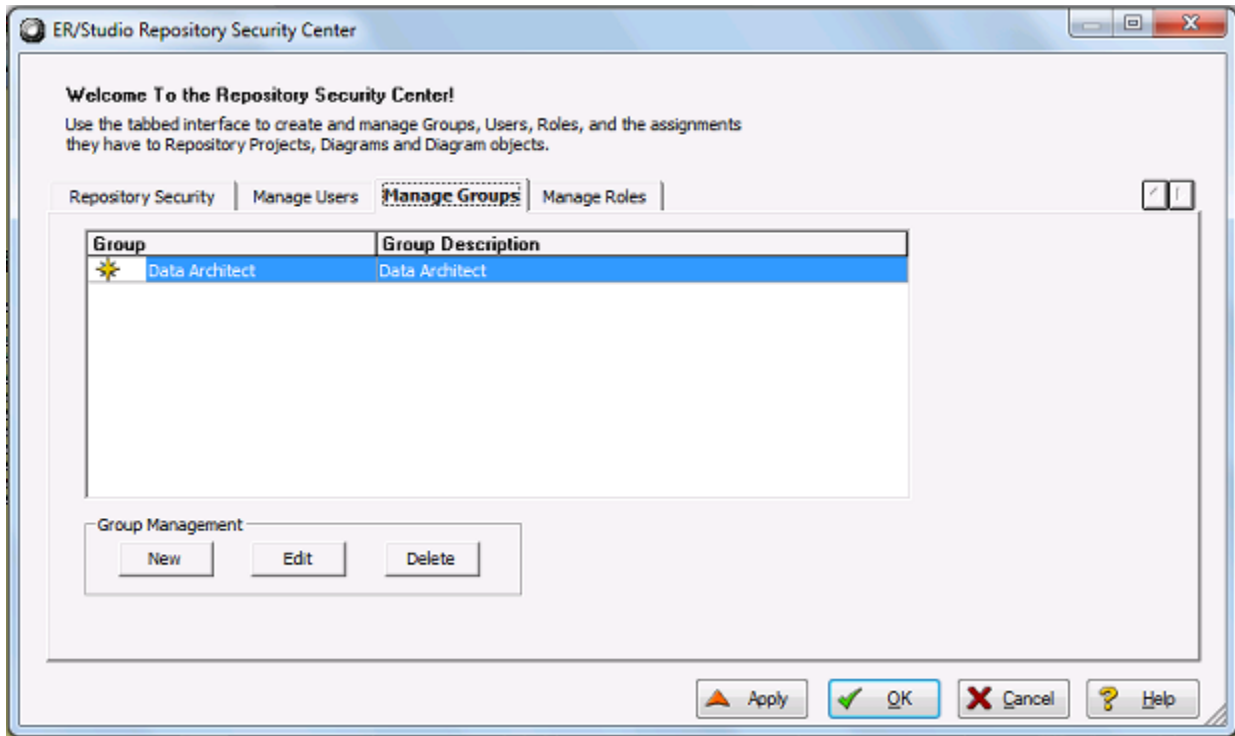
GROUP SUPPORT IN REPOSITORY SECURITY CENTER

To make it easier to handle permissions, group support has been added to the ER/Studio Repository Security Center. Repository administrators can assign roles to the group at repository, project and diagram levels, then simply add or remove users from the group to apply the settings to the users quickly and easily. The Repository administrator can either manually create a group and assign users to it or use an LDAP/Active Directory group and add users to the Repository to a group based on LDAP/Active Directory settings.

On the **Repository Security** tab, the list of **Available Users** displays **Groups** first and then individual **Users**.

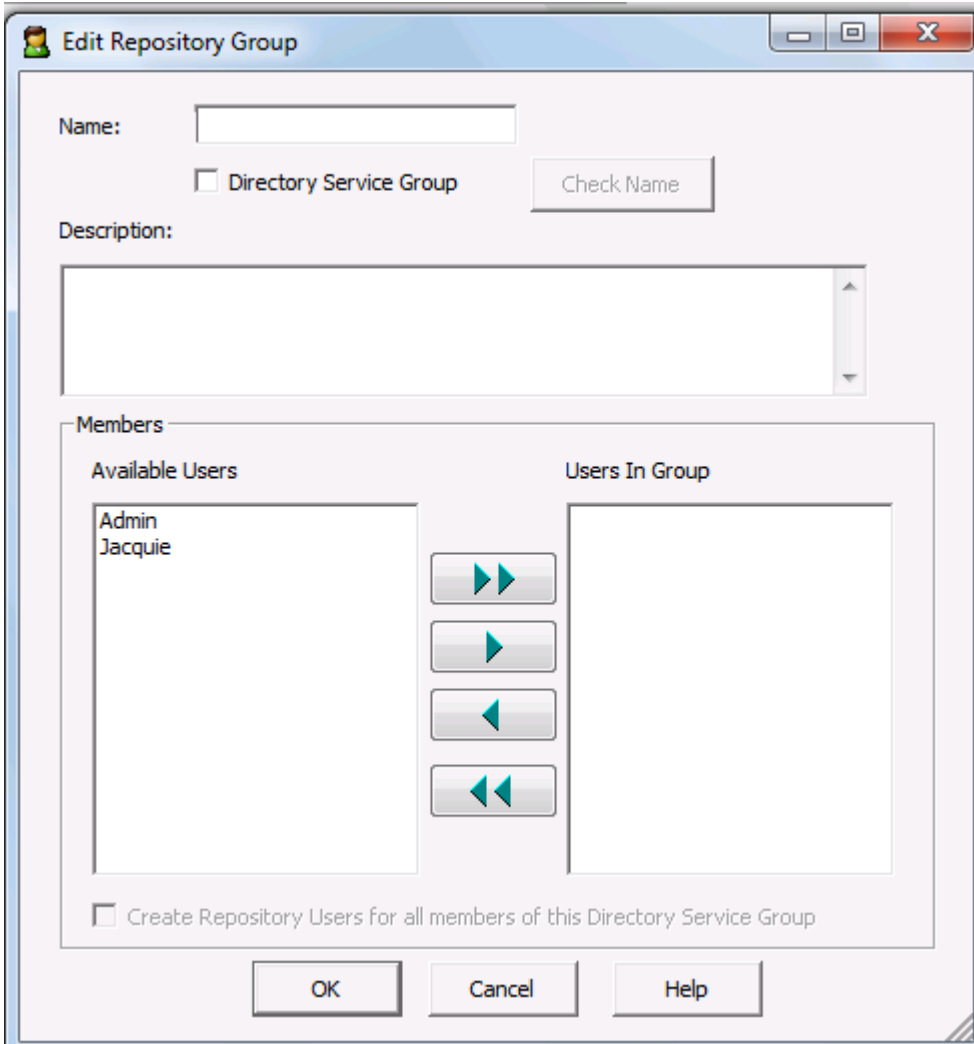


A new **Manage Groups** tab has been added to the **ER/Studio Repository Security Center** where you can add, edit, and delete groups.



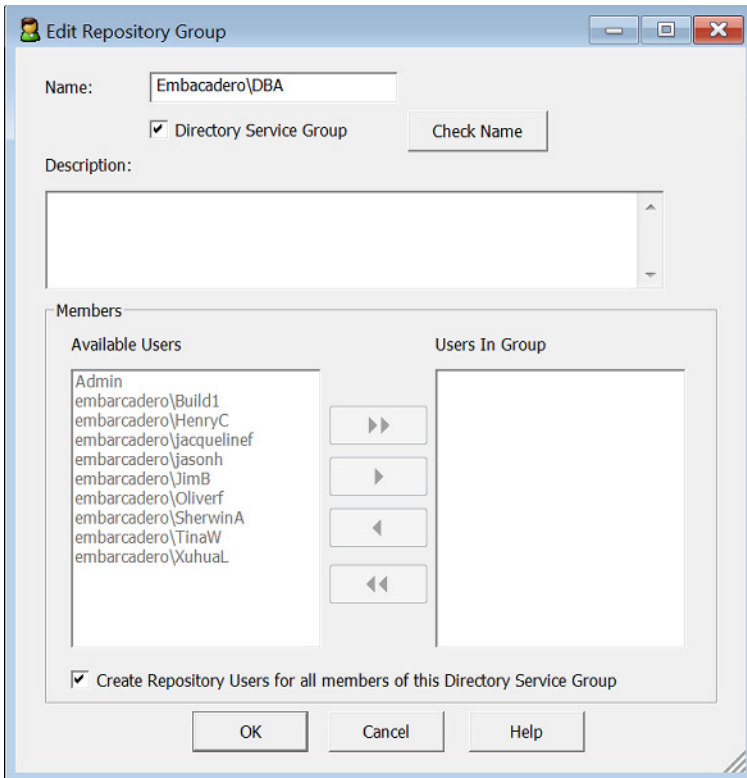
Create, Update, and Delete Group

- 1 Choose **Repository > Security > Security Center**.
- 2 Click the **Manage Groups** tab and select **New**.



- 3 Provide a name for the group and optionally, a description of the group.

- 4 If you want to create a group based on an LDAP/Active Directory group, click **Directory Service Group** and another option becomes available at the bottom of the dialog, **Create Repository Users for all members of this Directory Service Group**. If you select this option, then repository users in this group are automatically added to the list of Repository users. If you do not select this option, then you will need to create the users manually and as you add LDAP/Active Directory Service users, they will be added to the Repository groups that correspond to the LDAP/Active Directory Service groups that they are members of.



If you want to manually populate the group, do not select **Directory Service Group**, from the list of **Available Users**, select the group members by Ctrl-clicking each user, or click the double right-facing arrow to select all the users. You can also remove group members by clicking the left-facing arrow.

Notes

- **Delete a group:** On the **Manage Groups** tab, you can delete a group by selecting the group and clicking **Delete**. Deleting a group does not delete the group members but the group members will no longer have the permissions afforded to them by the group membership.
- **Edit a group:** On the **Manage Groups** tab, you can edit the group name and list of group members by selecting the group, clicking **Edit**, and making your changes on the **Edit Repository Group** dialog.

Assigning a Role to a Group

Assigning a role to a group grants the group all the permissions selected in the role.

- 1 Choose **Repository > Security > Security Center**.
- 2 Select a Repository item you want to assign permissions to.
- 3 *If the group is not currently assigned a role*, on the **Repository Security** tab, click the group name in the **Available Users** column and drag it onto a role name in the **Available Roles** column.

If the group was already assigned a role, on the **Repository Security** tab, click the group name in the **Available Roles** column and drag it onto another role name in the **Available Roles** column.

- 4 Click **Apply** and then click **OK** to exit the **Security Center**.
- 5 Notify users in the group that they must log out and then log in again to receive the security updates.

NOTE: If a user has been assigned the No Access role for an object, but that user is a member of a group with permissions for the object, then the user will have all permissions assigned to the group. All permissions available to a user through direct assignment to roles are added to the permissions available to any groups the user is a member of. Permissions are additive, so the user gets all the permissions assigned to the user whether they were assigned through a group or directly to the user.