Q317442 - BUG: ORM or ER Source Models May Become Corrupt After Building Projects

The information in this article applies to:

- Microsoft Visio for Enterprise Architects -- part of Microsoft Visual Studio .NET (2002), Enterprise Architect Edition

SYMPTOMS

An ORM source model or an ER source model becomes corrupt after building the Project.

CAUSE

Failure to save source model changes before building (or rebuilding) can cause the source model to become unreadable or to be converted to a simple drawing with no underlying model.

RESOLUTION

After making any change to an ORM source model (or an ER source model), be sure to save the model before you build (or rebuild) a project from it.

Scenario 1: Open a source model before adding it to a project

Suppose you have open an ORM source model (or an ER source model) that you wish to add to a database model project. This may be a new source model you are working on, or a previous source model you have reopened. If you make any change to the source model (even simply moving a shape), you must SAVE the most recent state of the model BEFORE adding the model to the project. Once you have added the source document to the project, it is good practice to save the project before building it. After the project is built, if you make any new changes to the source model, SAVE those changes BEFORE rebuilding the project. If you have already rebuilt a project without previously saving the changes to the source model, make sure you SAVE the source model BEFORE exiting from Visio.

Scenario 2: Open a project before adding a source model to it

Suppose you first create a database model project, and then add an existing ORM source model (or an ER source model) to it. If you make any change to the source model (even simply moving a shape), you must SAVE the most recent state of the model BEFORE building the project. After the project is built, if you make any new changes to the source model, SAVE those changes BEFORE rebuilding the project. If you have already rebuilt a project without previously saving the changes to the source model, make sure you SAVE the source model BEFORE exiting from Visio.

STATUS

Microsoft has confirmed this to be a bug in the Microsoft products listed at the beginning of this article.

REFERENCES

Additional query words: EA

Q317443 - BUG: ORM Source Model May Become Corrupt after Migration from a ER or Database Model

The information in this article applies to:

- Microsoft Visio for Enterprise Architects -- part of Microsoft Visual Studio .NET (2002), Enterprise Architect Edition

SYMPTOMS

An ORM source model becomes corrupt after migrating any change from a database model or an ER model.

CAUSE

The source model was not saved prior to being closed.

STATUS

Microsoft has confirmed this to be a bug in the Microsoft products listed at the beginning of this article.

MORE INFORMATION

Suppose that you have used the Database Model Diagram solution to build a project that includes an ORM source model (or an ER source model). Now suppose that you make some change to the logical database model (e.g. rename a column), and save this change to the project. When saving, you are prompted whether you wish to migrate the changes back to the source model. If you answered "Yes" to the migration, you must save the source model again before closing it. One way to do this is to open the source model in the project window (e.g. by double-clicking it) and then choose File > Save from the menu.

Failure to save a source model after changes have been migrated to it can cause the source model to be converted to a simple drawing with no underlying model.

REFERENCES

Additional query words: EA

Qnnnnnn - BUG: In an ORM Source Model, Detaching a Role Connector Fails to propagate to Duplicate Predicates

The information in this article applies to:

- Microsoft Visio for Enterprise Architects -- part of Microsoft Visual Studio .NET (2002), Enterprise Architect Edition

SYMPTOMS

In an ORM source model, detaching a role connector fails to propagate visually to duplicate predicates.

CAUSE

Suppose an association (e.g. A has B) is duplicated in the graphical representation of an ORM source model, and a role within its predicate is then detached from the role's object type (e.g. to reattach it to another object type). Although this change is made internally in the model, this change is not displayed on graphical duplicates of the predicate.

RESOLUTION

After completing a change to a role attachment, manually make this same change to all other graphical instances of the role attachment. If the role connection was moved to a different object type that is not displayed on a page containing a duplicate of the predicate, drag another instance of this object type onto that page from the business rules editor to enable the new connection to be displayed.

Scenario 1: Move a role connection to a new object type displayed on all relevant pages

Suppose that pages 1 and 2 of an ORM source model both display the following fact types:

F1: Employee has office in Building

F2:	Room is in Building

Now suppose that you want to change F1 to:

F3:	Employee has office in Room

To do this, you make the following change to page 1: detach the second role of F1 from Building and then attach it to Room.

If you look at page 2, it still shows the duplicate of F1 unchanged, even though this change has been made in the model (e.g. F1 on page 2 will verbalize as F3). To avoid confusion, you should ensure the displays are consistent. To do this, manually repeat on page 2 the change made to page 1, i.e. detach the second role of F1 from Building and then attach it to Room.

Note that this problem also exists in the unlikely case where the duplicate predicate appears on the same page as the original.

Scenario 2: Move a role connection to a new object type that is not displayed on all relevant pages

Suppose that page 1 of an ORM source model both display the following fact types:

F1:	Employee has office in Building

F2:	Room is in Building

Suppose that page 2 displays a duplicate of F1, but does not display the object type Room.

Now suppose you want to change F1 to:

F3:	Employee has office in Room

To do this, you make the following change to page 1: detach the second role of F1 from Building and then attach it to Room.

If you look at page 2, it still shows the duplicate of F1 unchanged, even though this change has been made in the model (e.g. F1 on page 2 will verbalize as F3). To avoid confusion, you should ensure the displays are consistent. To do this, first drag an instance of the Room object type onto page 2 from the Business Rules window, then manually repeat on page 2 the change made to page 1, i.e. detach the second role of F1 from Building and then attach it to Room.

STATUS

Microsoft has confirmed this to be a bug in the Microsoft products listed at the beginning of this article.

REFERENCES

Additional query words: EA

Qnnnnnn - BUG: In an ORM Source Model, Moving a Subtype Connector may Delete Subtype Definitions and Duplicate Subtype Connectors

The information in this article applies to:

- Microsoft Visio for Enterprise Architects -- part of Microsoft Visual Studio .NET (2002), Enterprise Architect Edition

SYMPTOMS

In an ORM source model, moving a subtype connector may delete subtype definitions and duplicate subtype connectors.

CAUSE

Subtype connectors attach to connection points on the supertype and subtype. These connection points are hidden by default, but may be displayed by selecting View > Connection Points from the main menu. If the subtype connector is moved, at least one end may become detached from any connection point, at least temporarily (e.g. to switch the connector to a more central connection point for better alignment). This effectively causes the former subtype to no longer be treated as a subtype in the underlying model, so its definition and any duplicate displays of its subtype connector are removed.

RESOLUTION

If you need to reposition a connector to a subtype for which you have supplied a subtype definition, first copy the definition either to the clipboard or to a text box, so that it can be pasted back into subtype definition pane after completing the repositioning. If you have duplicated the subtype and supertype (e.g. on another page), you can redisplay their connection by dragging the Subtype Relationship arrow from the ORM source Model stencil and connecting it to the object types. Alternatively, you can right-click the subtype or supertype, and choose Show Relationships from its context menu – however this will also display any associations played by the object type.

Scenario 1: Move a subtype connection where the subtype has a subtype definition

Suppose that page 1 of an ORM source model displays the fact type Person has Gender, as well as the object type Woman, and a subtype connection from Woman to Person. Suppose also that a definition exists in the Subtype definition field of the Database Properties sheet for Woman (e.g. each Woman is a Person who is of Gender 'F').

Now suppose that you wish to move the subtype connector either indirectly (by moving either the subtype or supertype shape) or directly (by disconnecting it at one end). *Before* doing so, first copy the subtype definition either to the clipboard or to a text box, so that it can be pasted back into subtype definition pane after completing the repositioning. You can copy text to the clipboard by selecting it and choosing Ctrl-C. Moving the subtype connector deletes the subtype definition. *After* repositioning the subtype connector, you can paste the subtype definition text back to the Subtype definition field by inserting the cursor in the desired position and choosing Ctrl-V.

Scenario 2: Move a subtype connection where the subtype is duplicated elsewhere

Suppose that page 1 of an ORM source model displays the fact type Person has Gender, as well as the object type Woman, and a subtype connection from Woman to Person.

Suppose that page 2 displays a duplicate of Person and Woman, but does not display the subtype connection between them. To display this connection on page 2, drag the Subtype Relationship arrow from the ORM source Model stencil and connect it from Woman to Person.

Now suppose that you move the subtype connector on page 1 either indirectly (by moving either the subtype or supertype shape) or directly (by disconnecting it at one end).

Doing this removes the duplicate subtype connection on page 2 (it is still in the underlying model, just not in the display for the duplicate). To restore display of this connection on page 2, drag the Subtype Relationship arrow from the ORM source Model stencil and connect it from Woman to Person on page 2.

Note that this problem also exists in the unlikely case where the duplicate subtype/supertype appears on the same page as the original.

STATUS

Microsoft has confirmed this to be a bug in the Microsoft products listed at the beginning of this article.

REFERENCES

Additional query words: EA

Qnnnnnn - BUG: Migrating Logical Model Changes to an ORM Source Model Takes Too Long

The information in this article applies to:

- Microsoft Visio for Enterprise Architects -- part of Microsoft Visual Studio .NET (2002), Enterprise Architect Edition

SYMPTOMS

Migrating changes from a logical database model to an ORM source model may take a very long time.

CAUSE

A logical database model project may be built from one or more ORM source models. On saving changes made directly to the logical database model, you are prompted whether you wish to migrate these changes back to the source model(s). If you answer Yes to this prompt, and the logical database model is large, migration sometimes takes a very long time (e.g. over an hour) to complete, even if you made only minor changes. This is caused by a bug that forces the migration process to do unnecessary extra work.

RESOLUTION

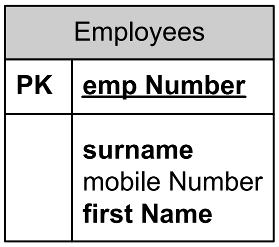
Don't say Yes to the Migrate prompt unless you made a structural change to the logical model, or you want to reuse the ORM source model(s) in another project. Non-structural changes are: renaming tables; renaming columns; moving columns that are not part of the primary key to a different non-key position in the table; reordering columns within a primary key. These non-structural changes are automatically stored in the mapping details for the project, so do not need to be explicitly migrated. The only time you might consider migrating non-structural changes is if you want to reuse the source model(s) in different projects, and want these naming and column ordering choices enforced there. However, you can usually configure an ORM source model to generate the table and column names you want, avoiding the need to ever rename them after mapping.

Examples of structural changes include: adding a column or table; deleting a column or table; adding, deleting or changing a foreign key relationship; changing the required status of a column etc. For structural changes, it is always better to make the equivalent changes to the ORM source model, and then map those changes to the logical model by rebuilding the project. If you do this, the changes are safer since they are often better understood at the ORM level, and it avoids the need to ever migrate changes back to an ORM model.

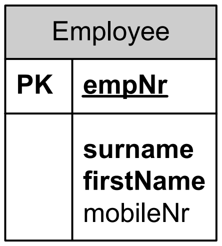
If you have made a structural change to a logical database model, and want to update the source model(s) accordingly, then you can often get much faster performance by saying No to the migration prompt, and then performing the source model update by choosing the menu option: Database > Project > Update Source Models.

Scenario 1: Making non-structural changes to a logical database model

Suppose that a logical database model built from an ORM source model contains the following table scheme:



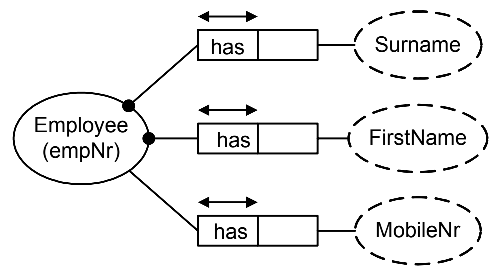
Now suppose you rename the table and some columns, and move the last column up one position to produce the following table scheme:



On saving these changes, answer No to the migration prompt, unless you want to reuse the source model in another project. Since these changes were non-structural, they will be preserved in the mapping the next time you rebuild this project.

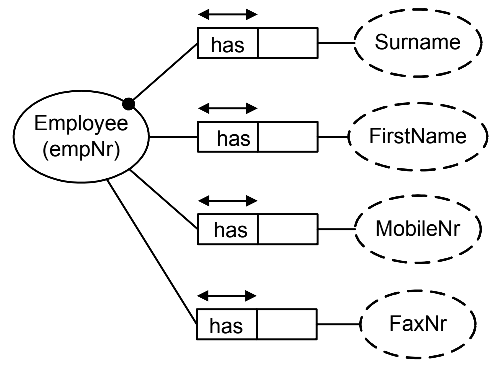
Scenario 2: Making structural changes to a logical database model

Suppose the ORM model below is used to build the logical database model shown at its right.



Employee	
PK	**empNr**
	surname
	firstName
	mobileNr

Now suppose you want to make a structural change to the logical model, such as making firstname optional and adding an optional faxNr column, to produce the table scheme shown below. Instead of making these changes to the logical model, and migrating them to the ORM source model, change the ORM model appropriately as shown, and then propagate those changes to the logical model by rebuilding it, to get the same result.



Employee	
PK	**empNr**
	surname
	firstName
	mobileNr
	faxNr

STATUS

Microsoft has confirmed this to be a bug in the Microsoft products listed at the beginning of this article.

REFERENCES

Additional query words: EA