

# Creating an Excel connection

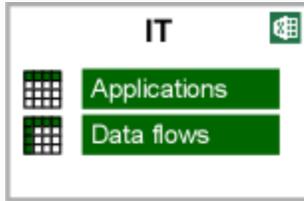
[Expand/Collapse All](#)

To illustrate the procedure to create a basic Excel connection, the following example is used.

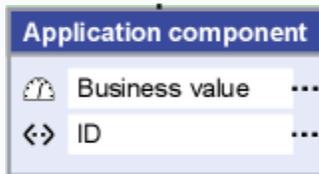
## **Example**

You want to exchange model information between Enterprise Studio and Excel about application components, their business value, and their data flows (relationships with data objects).

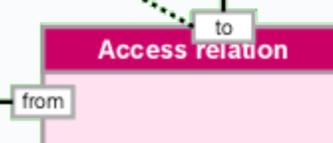
To realize the information exchange, the following connection with two mapping tables will be created:



Applications		
A	B	C
Application component	Business value	ID



Data flows	
Application component	Access relation
	Data object



The following is done to create the connection:

- Add an Excel connection as basis for the mappings.
- Define two mappings, one with a regular table for the application components and their business value, and one with a relation cross-reference table for the relationship between the application components and data objects.
- Define the Excel worksheets and their columns, rows and cells by configuring the mapping tables.

- Define the model data on the Enterprise Studio side by adding metaobjects and their properties, and metarelations.
- Map the model data from the tool side to the Excel worksheet by creating relations between the model elements and the mapping tables.

There is no prescribed order in which the table and the meta-elements of a mapping should be configured to create the mapping. The preferred order may depend on what you want to use the connection for.

If you use it for import, you may prefer to start with the table, since you have an Excel file as starting point. If you want to export, you may prefer to start with the model data from the tool side by adding the meta-elements.

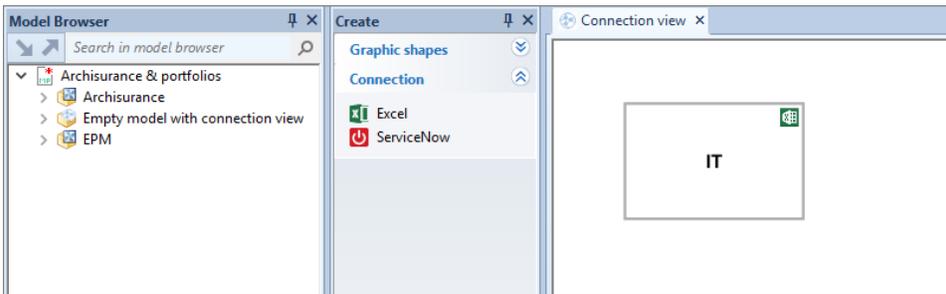
**On this page:**

- [Adding a connection](#)
- [Defining mappings](#)
- [Configuring tables](#)
- [Adding metaobjects and metarelations](#)
- [Creating relations between the mapping elements](#)
- [Specifying mapping options for import and export](#)

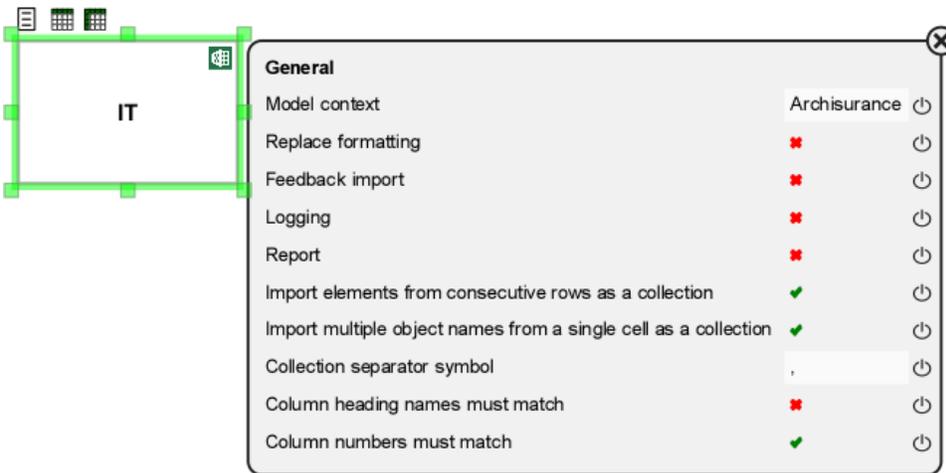
## Adding a connection

To add a new Excel connection to your Connection model, follow these steps:

1. In the **Create** pane, click the **Excel** connection , click in the open connection view, and type the name of the new connection.



2. Select the connection object, and then click  to set the connection properties. Options with a check mark  are activated, options with a cross  are deactivated. Click the check mark or cross to change the status.



Select the model (element) in the model package hierarchy that should act as root for looking up and creating objects in the context of this connection.

An existing Excel file may contain additional formatting information, such as logos or company names. In the worksheet, this formatting information must be located in a row above or column to the left of the exported table. Example:

	A	B	C	D	E	F
1		<b>BiZZdesign</b>				
2		<b>Relation</b>	<b>Data object</b>			
3		<b>Application component</b>	Customer File Data	Damage Claim Data	Insurance Policy Data	Insurance Request Data
4		Bank System				
5		Call center application				
6		Claim Data Management		X		
7		CRM System				
8		Customer Data Access	X			
9		Financial Application				

If the formatting information can be replaced upon export, activate this option. The worksheet will then only contain the exported data, the formatting info will be removed.

If the formatting information must be kept, deactivate the option, and make sure that the starting row and column of the table are correctly set at the [table settings](#), so that the exported table data will start in the column and row next to the formatting information, and not on top of it.

Activate this option if feedback about the import must be provided. During import the Excel file is opened, and live feedback is provided by means of coloring cells and comments to the cells. Example:

	A	B	C	D	E
1	<b>Relation</b>	<b>Data object</b>			
2	<b>Application component</b>	Customer File Data	Damage Claim Data	Insurance Policy Data	Insurance Request Data
3	Bank System				
4	Call center application				
5	Claim Data Management				
6	CRM System				
7	Customer Data Access	X			
8	Financial Application				
9	Home & Away Policy Administration				

Could not import object (ID invalid or incomplete)

It is not useful to activate this option when using [ADOdb](#) for import and export.

If activated, a log file with feedback about the export and import is created. The log file will be placed in the same folder as the Excel file.

If activated, a table with information about created, modified and matched objects and relations will be generated and shown after performing an [export](#) or [import](#).

In case some property represents a collection of elements, these elements are assumed to be specified on consecutive rows by default. If the property value is assumed to be specified in a single row, deactivate this option.

Multiple objects can be specified by their name in a single cell to import a property that represents a collection of object references, or to import multiple relations, one for each object (acting as end point of the relation). Deactivate this option if you do not want this.

By default, a comma is used to separate object names if imported as a collection. If you want to use another symbol, type it in this box.

Deactivated by default. Activate this option if you want the names of the column headings in the Excel worksheet to be matched with the names of the mapping table columns specified in the connection. If the names do not match, the data will not be imported.

Activate this option if the number of columns is not known, or if you are going to import or export [timed values](#), except when importing timed values using ADOdb.

Activated by default. The number of a modeled column must match the column number in the Excel worksheet; otherwise column numbers are ignored and only the relative ordering of the modeled columns is considered. In that case, deactivate this property and activate matching on column heading names.

This option must be deactivated if you are going to import or export [timed values](#) (because the column number is not known up front), except when importing timed values using ADOdb.

3. When you are ready, close the settings window

## Defining mappings

After the connection has been created, one or more mappings must be defined. One mapping represents one Excel worksheet.

### Example

For the example connection, two mappings are defined. One with a regular table, and one with a relation cross-reference table.

1. In the connection view, select the Excel connection object, and then click:



to add a mapping with a regular table

 to add a mapping with a relation cross-reference table

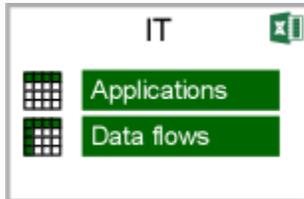
2. Name the mapping. It will also be the name of the table in the mapping.

If you are using the connection for export, this name will be the name of the worksheet in Excel. Take note that Excel has a few [limitations](#) with respect to naming worksheets.

If you are going to use the connection to import data, make sure the name of the mapping/table matches the name of the Excel worksheet you want to import from!

### Example

For the example connection, the result is a connection with two mappings:



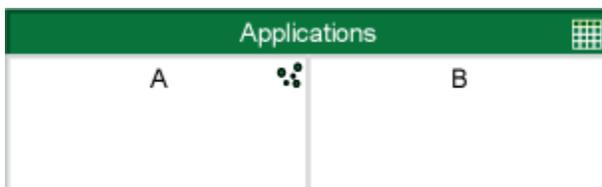
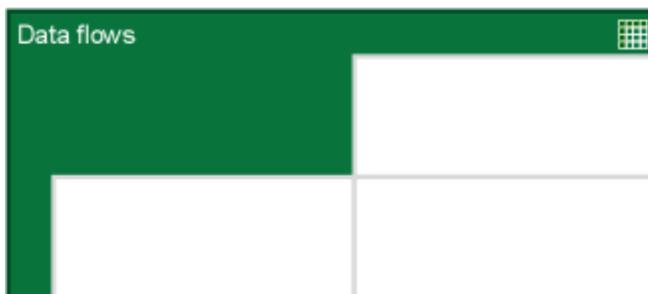
## Configuring tables

Next step is configuring the tables to correctly represent the Excel worksheet you want import data from or export data to. Do this for each mapping you have added.

### Example

For the example connection, both tables are configured, the regular table and the relation cross-reference table.

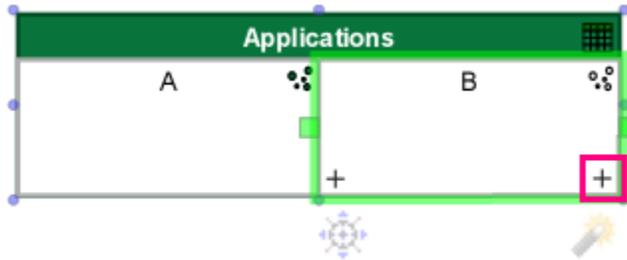
1. Open a table by double-clicking the mapping. A new diagram opens containing an empty mapping table. Depending on the chosen mapping, it is a regular table or a relation cross-reference table.

A screenshot of a table configuration window titled "Applications". The window has a green header bar with the title and a grid icon. Below the header, there is a table with two columns. The first column is labeled "A" and the second column is labeled "B". In the center of the table, there is a small icon consisting of four dots arranged in a square.A screenshot of a table configuration window titled "Data flows". The window has a green header bar with the title and a grid icon. Below the header, there is a table with two columns. The first column is labeled "A" and the second column is labeled "B". The table is currently empty.

2. Regular table only: Determine the number of needed columns for the Excel worksheet, add the additional columns and name them.

If you are going to use the connection for import, you can quickly add the columns by [importing](#) them from the Excel file instead of adding and naming them manually. See [Importing table column headings from Excel](#).

- a. To add columns manually, select a table column, and then click the plus sign on the left or right side to add a column to the left or the right.



The total number of columns must equal the number of relations you want to create between the model data and the mapping table. In the end each table column must have a mapping relation attached.

- b. Name the table columns. The names represent the column headings in the Excel worksheet. To name the columns, you have the following possibilities:

*Name automatically:* In that case do nothing yet; when you further on in this procedure [create relations](#) between the meta-elements and table columns that have not been named yet, the column automatically gets the name of the linked meta-element.

*Name manually:* Select the column, press F2 and type the name.

3. Regular table only: By default the first table column is set as the population column. The population column is the column that defines the population for an export, the objects in this column will be taken as a starting point for the export. If you want to set another column as the population column, select that column, and then click  in the upper right corner.
4. Optional: If the first column and row of the table in the Excel worksheet starts (for import) or must start (for export) in another than the first column or row, then specify the start column and row number for the table. Do this for each table, if needed.

Select the table title, click , and set the following properties.

**Table properties**
✕

First column	1	⏻
Row with column headings	1	⏻
Column heading names must match	✔	⏻
Column numbers must match	✖	⏻
Import column headings		

Set the number of the column in the Excel file from which the export or import must start. Default value is 1, representing Excel column A.

Set the number of the row in the Excel file from which the export or import must start. The first row contains the column headers, i.e., the names of the model elements and properties these columns contain. Default value is 1.

Default setting is based on what is set at connection level. Change it if needed. If different, the setting at table level overrides the one on connection level.

Activate this option if you want the name of the column heading in the Excel worksheet to be matched with the name of the mapping table column. If the names do not match, the data will not be imported.

Activate this option if the number of columns is not known, or if you are going to import or export [timed values](#), except when importing timed values [using ADOdb](#).

Default setting is based on what is set at connection level. Change it if needed. If different, the setting at table level overrides the one on connection level.

If activated, the number of the table column must match the column number in the Excel worksheet; otherwise column numbers are ignored and only the relative ordering of the modeled columns is considered. In that case, deactivate this property and activate matching on column heading names.

This option must be deactivated if you are going to import or export [timed values](#) (because the column number is not known up front), except when importing timed values using ADOdb.

5. When you are ready, close the properties window.

### Example

For the example connection, the result of configuring the tables is a regular table with three columns, and a relation cross-reference table:

Applications		
A	B	C
Application component	Business value	ID

Data flows	

## Adding metaobjects and metarelations

Next step is adding metaobjects and metarelations to define the model data from Enterprise Studio that must be mapped to the Excel worksheet. In case of a regular table, add one metaobject or metarelation. In case of a relation cross-reference table, add two metaobjects and a metarelation.

After adding the meta-elements, properties may need to be added to the meta-elements if the mapping tables contain defined properties.

### Example

In case of the example connection, one metaobject including properties is added to the regular mapping table, and two metaobjects and a metarelation are added to the relation cross-reference mapping table.

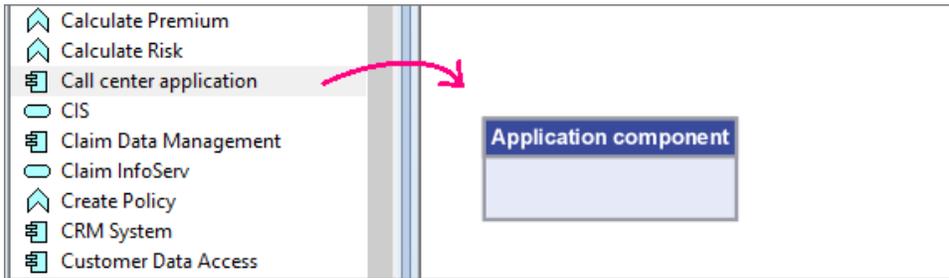
Additionally, properties are added to the metaobject for the regular mapping table.

## Adding a meta-element

To add a metaobject or metarelation to a mapping, follow these steps:

1. In the model browser, select a model object or relation of the type you want to add to the mapping, and drag it onto the diagram. It can be any object or relation.

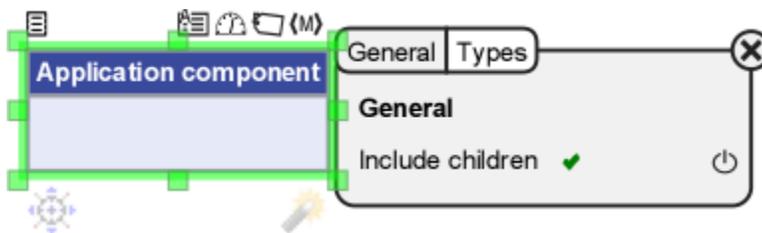
A metaobject or metarelation is placed on the diagram. It has the name of the element type of the selected element. In the figure below a metaobject representing application component is added by dragging an application component.



Alternatively you can add a meta-element by [using the Create pane or the quick-create pop-up window](#).

2. Only for metaobjects, optional: By default, any children of an object of the selected object type will automatically be included in an import or export. If you do not want any child elements to be included, do as follows:

- a. Click the object type, and then click the control.
- b. On the **General** tab next to **Include children**, click the check mark to deactivate the option. The check mark will turn into a red cross.



## Adding meta-element properties

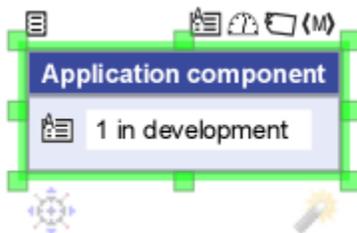
If one or more columns in the table represent object or relation properties, these properties also need to be added to the metaobject or metarelation in order to be mapped to the table columns. These properties can be an attribute, metric, tag, and metadata.

To add a property to a meta-element, do as follows:

- Select the meta-element, and then click the control of the type of property you want to add. If needed, add multiple properties.

### attribute

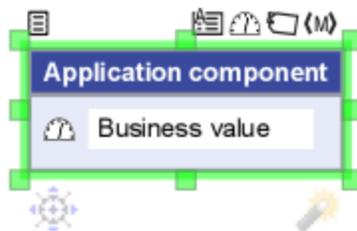
Add an attribute. In the selection window that appears, select an attribute, and click **OK**. Example:



### metric

Add a metric. Metrics are used to describe numerical properties of objects. In contrast to attributes metrics can be added to a model, whereas attributes have to be introduced in the tool configuration.

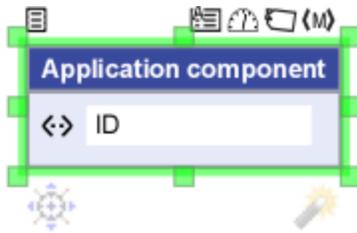
In the selection window that appears, select a metric from the model package, and click **OK**. Example:



### metadata

Add metadata. Two types of metadata are available: ID and type name of the element. Metadata can be exported and can (only) be used for matching during import. This is because type name and ID cannot be changed for existing objects. For more information about using metadata as property, see [Using metadata as meta-element property](#).

In the metadata selection window, select the desired metadata type, and click **OK**. Example:

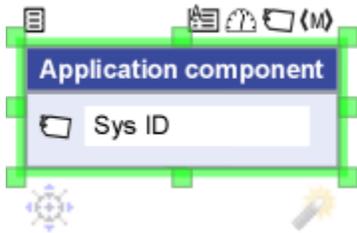


### tag

Add a tagged value. A tagged value represents some data that is linked (tagged) to an object. Tagged values are meant for advanced usage such as complex viewpoints (scripts) or complex import scenarios, but can also be used for properties that are not of interest to the end-user, but which are used for matching existing elements from earlier imports or exports.

Tagged values should typically not be used to model properties of an object, metrics and attributes are used for this. More information about using tagged values can be found in the [BiZZdesign Scripting Reference](#).

After clicking the control, type the name of the tag, and click **OK**. Example:



### Example

For the example connection, adding metaobjects and metarelations and adding properties to them has resulted in the following:

Regular table

Applications		
A	B	C

Application component	
Business value	
ID	

Relation cross-reference table

Data flows	

Data object

Application component

Access relation	
from	to

## Creating relations between the mapping elements

Now, relations must be created between the model data and table in the diagram for the actual mapping of the data. The *mapping* relation is used for this. Each column in a regular mapping table must have a mapping relation attached that connects the column with a metaobject, metarelation, or its properties (attribute, metric, tag, or metadata).

The *link* relation is used for connecting metarelations and metaobjects. Which relations you need to create, depends on your mapping. In the end, each column or cell in the mapping table must have a relation attached.

### Example

For the example connection, mapping relations between the regular mapping table and the metaobject and its properties (attribute and metadata) are created, and mapping relations between the relation cross-reference table and the metaobjects and metarelation.

Link relations are created between the metaobjects and the metarelation in the mapping with the relation cross-reference table.

## Creating a mapping relation

In a mapping with a regular table, a meta-element is linked to the table column representing the element type, any meta-element properties are linked to the columns representing the respective properties. In a mapping with a relation cross-reference table, the metaobjects (or their property, if set) are linked to the end cells of the table representing the column and the row. The metarelation is linked to the central cell representing the cross-reference.

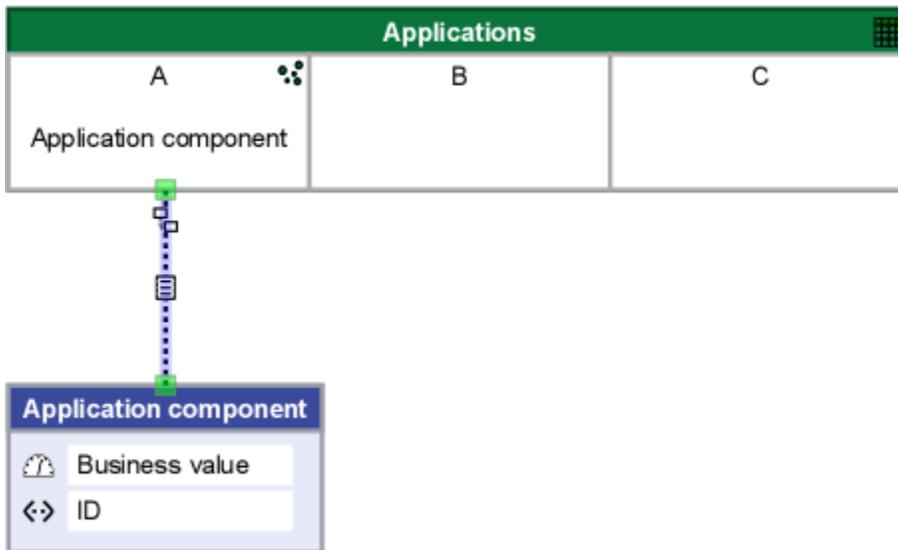
If you have not named the table columns yet, they will automatically be named the moment you map them to a meta-element or property. They will get the name of the meta-element or the property they have been mapped to.

In case you draw a mapping relation to a column that already has a mapping relation, a new column is added to the left or right of the column, depending on where you attach the relation to the column (left or right from the center).

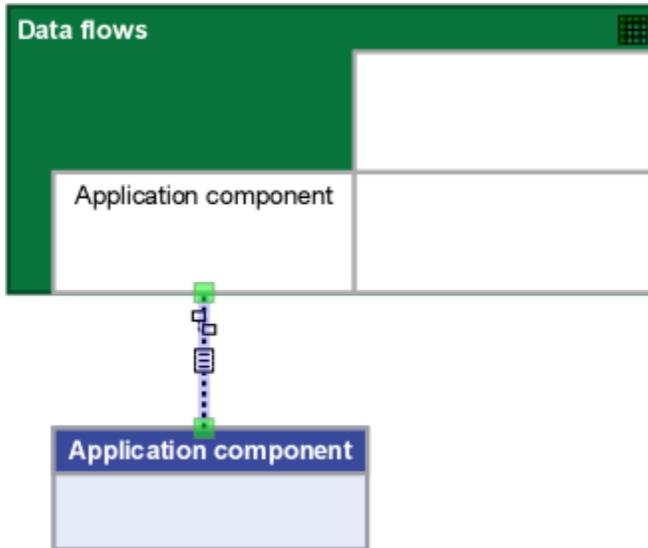
To create a mapping relation, do as follows:

- Select the meta-element or a property in the meta-element, and draw a mapping relation  to the table column or cell representing the element type or property. Do this for every element and property present in the mapping until each column or cell in the mapping table has a relation attached.

Regular table



Relation cross-reference table

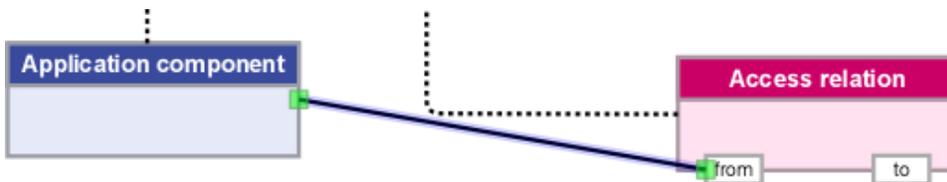


Relation lines are automatically laid out optimally when connecting the components. That is default behavior for modeling a mapping. In case a relation does not have the optimal layout, use the  control on a selected relation to [lay out the mapping relations](#).

## Creating a link relation

To create a link relation, do as follows:

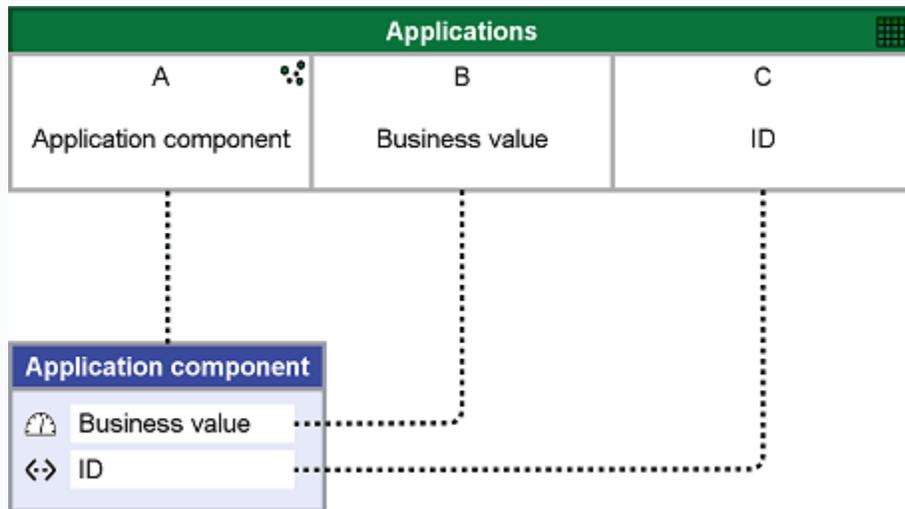
- Select the metaobject that must be linked to the "from" connector (endpoint) of the metarelation, and draw a link relation  between the metaobject and the metarelation connector. Do the same for the other metaobject and the "to" connector of the metarelation.



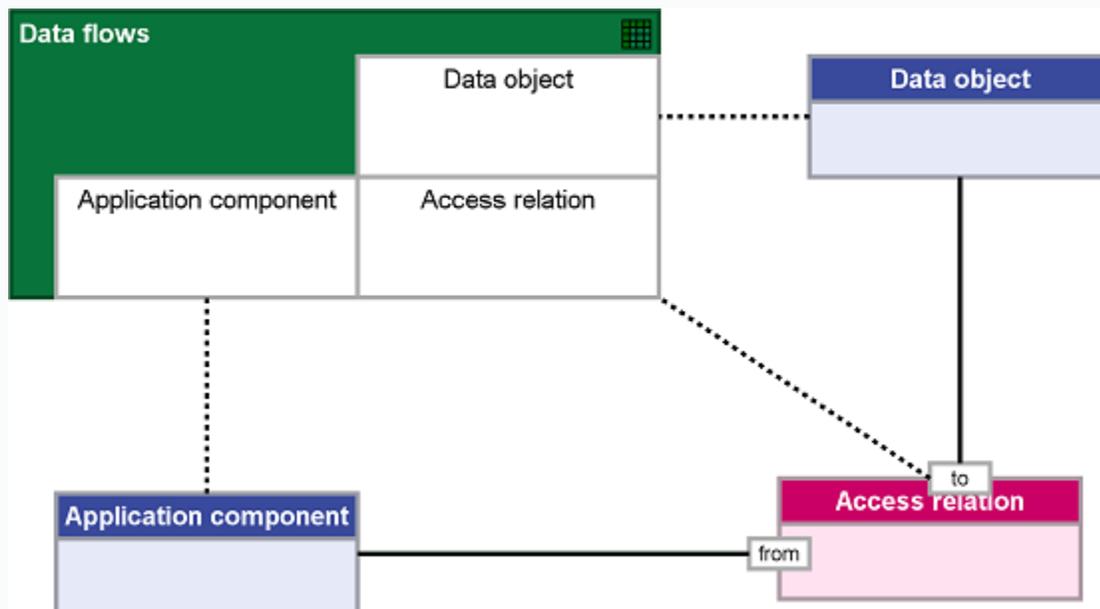
### Example

For the example connection, the result of creating the relations looks as follows:

Regular table



Relation cross-reference table



Your mapping may not look as organized as the example above; the relations and connectors are probably positioned differently. In order to make your mapping look more organized, you can reposition one or more relations in the mapping by dragging them, and [relocate the "from" and "to" connectors](#) of the metarelation.

## Specifying mapping options for import and export

Once mapping relations have been created, their settings for import and/or export must be specified. For each mapping relation in your connection mapping(s), specify the mapping options that are relevant to your use of the connection. Depending on what you are going to use the connection for, set the options for import or export, or for both.

The availability of an option in a mapping relation and its default setting may depend on the mapping table in which the relation is created (a regular or relation cross-reference table), and the component the mapping relation is attached to (metaobject, metarelation, or property).



Import and export options of a mapping relation for a metaobject in a regular table

To specify the properties for a mapping relation, follow these steps:

1. Select a mapping relation, and then click  on the relation line. The settings window pops up.
2. Click the **Import** tab, or the **Export** tab to specify the settings.

**Import:**

If activated, the presence of a value on the Excel side is mandatory. Upon import, if a value is missing, a message will appear indicating this.

If deactivated, missing values are allowed.  
Only available on mapping relations in a regular mapping table.

If activated, the object or relation type or property attached to the mapping relation is used to identify the related element on import. If attached to an object or relation type, identification is done on the element's name. If attached to a property, it is the property's value. This implies that the value of this type or property cannot be changed on import.

If deactivated, there is no matching, so the value can change on import. If replacement has been deactivated (**Replace** option), and the **Add** option has been activated, and there is no matching, import will create duplicate elements in your model.  
If activated, a new model object or relation is created and added to the model if no match with an existing one is found on import. In case of a property value, the value is added to the model. Deactivate this option if you do not want this to happen. Example when activated:

	A	B	C	D	E
1	<b>Relation</b>	<b>Data object</b>			
2	<b>Application component</b>	Customer File Data	Damage Claim Data	Insurance Policy Data	Insurance Request Data
3	Bank System				
4	Call center application				
5	Claim Data Management		X		
6	CRM System				
7	Customer Data Access	X			
8	Financial Application				
9	Home & Away Policy Administration				
10	Policy Data Management			X	
11	Risk Assessment				
12	Web portal				

Created object

If activated, the name of the object or relation in the model may be replaced by the imported name. In case of a meta-element property the property value may be replaced by the imported value. Deactivate the option if replacement is not allowed.  
In case of a mapping relation between a meta-element and a mapping table: the object or relation will be removed from the model if it is not present in the import. If deactivated, the object or relation is preserved in the model.

In case of a mapping relation between a property and a mapping table: can only be activated if **Mandatory** has been deactivated. If **Remove** is activated, a non-mandatory property value will be reset to its default value if the value is not present in the import. In case of a metric, the metric value will be removed (metrics do not have a default value). When importing a set (of multiple values), the values that are not present in the import will be removed from the set in the model.

By default, the model context specified at the [general connection settings](#) is searched to find a match for imported objects and relations. If you want to use a different model context for this specific mapping, click in the box and select the model (element) in the model package that should be searched. This context overrides the general connection setting.

By default, the model context specified at the [general connection settings](#) is used to create objects that could not be matched on import (**Add** option). If you want to use a different model context for this specific mapping, click in the box and select the model (element) in the model package that should be used. This context overrides the general connection setting.

### Import and export

If you have a mapping script available for processing imported or exported values, you can specify it here. Click in the box and select the script. Upon import/export, the mapping script is invoked to process the imported/exported values. This script should be defined as a viewpoint.

### Export

Objects, relations have a default name when added to the model. Properties also have a default value. If the option is activated, objects and relations without a user-defined name, or properties without a user-defined value will be exported. If deactivated, default names and values will not be exported.

By default, objects from the model context specified at the [general connection settings](#) are exported. If you want to use a different model context for this mapping, click in the box and select the model (element) that should be used for exporting objects. This context overrides the general connection setting.

Only available for export if the mapping relation is between a timed attribute/metric and a mapping table, and if **Time-value mapping** in the table column's properties is set to "Define value per date". If activated, timed values for all associated dates will be exported. If deactivated, define the dates for which the timed values should be exported in **Time domain**.

Only available if export option **Export time-value mappings for all current dates** is deactivated. Click in the box next to **Time domain** to set the dates for which the timed values should be exported, or copy an existing time domain from another mapping (if available) by clicking  and selecting one.

3. When you are ready, close the settings window.

Your Excel connection is now completed.

## Related articles

- [Using timed data in the Excel import and export](#)
- [Import and export of Excel worksheets using an ADOdb connection](#)
- [Data integration with Excel](#)
- [Structure of an Excel connection](#)
- [Exporting model data to an Excel workbook](#)