Axapta Integration

User Manual

Project :	e-Con 3.5
Author:	To-Increase
Company	To-Increase B.V.
Date	March 16, 2006

Document Information

Document number	
Version	1
Status	Final
Title	Axapta Integration
Subject	User Manual
Author	To-Increase
Department	Development
Manager	Marijn van Poelje
Project	e-Con 3.5
Last saved	1-1-1601 1:00

Revision history

Version	Date	Status	Changes
1	01-03-2006	Final	

Table of Contents

Chapter 1 Getting Started

In this chapter, you will find an introduction to e-Con. Read this section to get a quick overview of the product and the manual.

1.1 Introduction

We all know that today's business processes are dynamic. The market is changing constantly, as are the needs of customers. If your company's business processes are flexibly organized, you can adjust them at a moment's notice and stay ahead of the competition. e-Con makes this possible. e-Con is a powerful tool which allows you to record and manage all dynamic information relevant to your organization.

Companies are increasingly using the Internet to offer their customers personalized service. Order fulfillment, e-Business, e-Commerce and e-Collaboration business models require integration of new Internet technologies with your current business processes. All of this is possible if your business software is web-based. e-Con functions as a web-based Business Intelligence Tool. It navigates the user through a number of questions, while automatically carrying out other processes unseen. And provides you with a customer-specific product that can be manufactured.

In this manual, e-Con for Axapta is explained in detail. The manual consists of a few main parts. In Chapter 1, you will find an overview of our product. Read this section to gain a good understanding of the different possibilities of e-Con. New features are also listed, and the help procedures are described. Chapter 2 describes installation. Read Chapter 3 to understand the architecture of e-Con. For a brief overview of how a model is built in e-Con, read Chapter 4. This chapter provides a step-by-step explanation of how to set up your first easy model. Chapters 5 through 8 completely explain Axapta integration. All the tables and fields are described in detail in Chapter 5. Chapters 6 and 7 provide a lot of "how to" information: how to set up, enter and use the different e-Con functionalities, step by step. Consult these chapters while modeling. Chapter 8 tells you how to set up a product model. Chapter 9 describes the e-Con Studio, how to set up rules, which functions and attributes are available, how to use them, etc. Chapter 10 provides you with information about troubleshooting. If errors occur while modeling or running the model, just read this chapter. In Chapter 11, you will find a Glossary. The main terms used are listed and explained there.

1.2 Installation

See the Installation Manual on the Product CD.

Chapter 2 The e-Con Product

This chapter describes the e-Con product and its architecture. First, a general overview is provided, followed by a more detailed description of the architecture.

2.1 General Overview

e-Con inserts records at the Back End, in this case the Axapta SQL database. In Axapta, the Generic Model is built (the records and fields that are to be part of the data structure and how that structure is to be configured). Modeler intelligence is added to the model, in the form of rules such as IF – THEN – or ELSE. Actual configuration takes place at the Front End. The choices made by the end user influence the value of the configured records and fields.

The concept of e-Con is based on the following:

- Component-based
- A generic solution, both functionally and technologically
- State-of-the-art technology (e.g. use of XML, Internet technology)

This means that e-Con is not a solution built exclusively for Axapta. e-Con will also work with other ERP packages, databases and even stand-alone systems. Integration of Internet technology as the basis of e-Con's functionality demonstrates that e-Con is a true state-of-the-art business solution.

The objective of e-Con is to make all required configuration functionality available in any situation. Within the scope of technological possibilities, total flexibility is offered. "All that should be configured, can be configured" is a motto designed to clarify that e-Con can be used for configuring products, service calls, time sheets, prospect data, and much more.

2.2 Configuration Processes

As described above, you can configure everything with e-Con. Some possibilities are listed below.

- Customers, prospects, suppliers, items
- Customer-specific items, BOMs (Bills of Materials) and Routings
- Customer-specific services
- Mortgage quotes and mortgage contracts
- Insurance policies
- Questionnaires for customer surveys in CRM systems
- Projects and project budgets
- Warehouse environments and optimal inbound/outbound operations
- Sample orders for quality control
- General ledger accounts
- Service/maintenance contracts

- Service orders
- Sales contracts for products and services
- Service/maintenance schedules
- Purchase orders for products and services
- Alternative routings in BOMs or recipes
- Recipes for input-driven quality
- Any entity or business process that involves business rules

Now we will describe some common processes in more detail.

2.2.1 Product Configuration

In the product configuration process, product data is generated by using questions and answers based on a generic configuration tool model. The result of a product configuration generally includes:

- Product data, such as descriptions, texts, etc.
- Product Bill of Materials, or "BOM"
- Product Routing

A product configuration can be used in a number of areas:

- Engineering
- Production
- Sales
- Etc.

The end result can be a complete product specification (= object) or a product specification that needs further engineering. In both cases, all possible end results must be first known and designed in the starting model.

Example:

A common car can have different engines and, depending on the choice of engine, different types of transmissions. Depending on the type of transmission - manual or automatic - a different production robot is needed. The costs will also depend on the components used and production methods employed. Specially-designed cars can be based on a common car, but then with some parts engineered to meet customer specifications. Like trucks, for example. The unknown part in the configuration is called a "black box". It requires further engineering before it is clear.

The Bike demo model is a typical example of product configuration.

2.2.2 Sales Configuration

A sales configuration focuses more on sales processes than on product data. However, product data can be a part of the configuration, in addition to the sales order or quote. The result of a sales configuration generally includes:

- A sales order or sales quote
- Specific client information

- Sales prices
- Product data

A sales configuration can be used in many situations:

- Sales, on-line or off-line
- Telemarketing and telesales
- Etc.

The end result is a sales document (order or quote) with all the necessary commercial information needed for the sales processes. It may also contain customer requirements for additional engineering or product configuration.

Example:

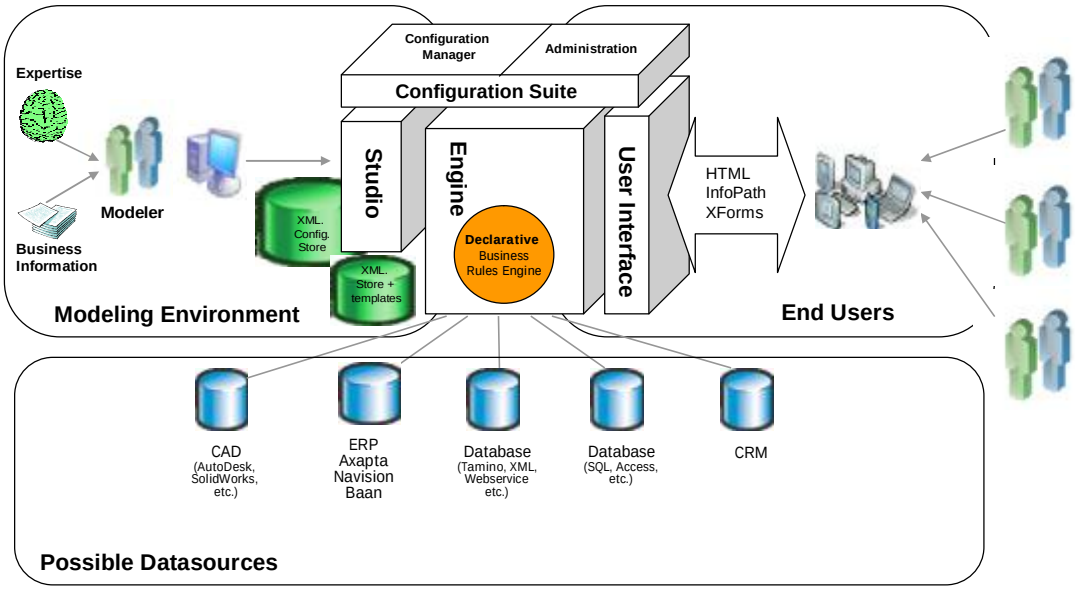
Sales prices and discounts depend on specific customer details, like their order history and contracts. The commercial price is not based solely on the products sold but also on the services rendered.

2.2.3 Customer Configuration

As indicated above, a configuration tool like e-Con can be used to configure many other things besides products and orders. In some cases, for example, the positioning of a prospective customer or current customer is based on a complex set of issues, such as the customer's relationship to other customers (in the case of multinationals). These interrelationships can present a very complicated picture. But e-Con can support the process of dealing with this interrelated information. In many situations, the e-Con end user will be entirely unaware of the complexity of it all, because they only have to deal with e-Con's simple, clear questions.

2.3 Architecture

Architecture of e-Con



The figure above shows the architecture of the total e-Con concept. The design of e-Con is so flexible because of its use of XML documents with which it can be integrated, regardless of ERP system or database Back End used. The Back End needs to have a dedicated XML generator in order to generate model data in the XML language. The Connector and Generator supply the XML generator. Rules are added to the XML data model in the Studio.

Configuration takes place in the Front End and the result is saved as an XML document and downloaded to the Back End by means of the Generator and Connector.

Besides the basic functional building blocks, there are also optional components. One example is the Report Generator, which can be used to print documents such as product specifications of configurations.

2.3.1 Important Conceptual Features of e-Con

The concept of a configuration tool is based on the existence of a generic model and the possibility to extract a realistically configured object from that model. The model itself is a collection of all the possible combinations of features and options, and in addition to data, it also contains rules. The rules enable the user to extract the correct data in order for the model to become a realistically configured object. Answers to questions provided during the configuration process trigger the execution of rules.

The configuration tool is generic, which means that it is not a dedicated product configuration tool or sales configuration tool. All kinds of models are possible - for instance, all the data of the Axapta database can be used to build a model and can therefore serve as an object for configuration.

Rules can be "imperative," which means that the sequence of the questions determines the end result. This is the true of most configuration tools. In e-Con, the rules are "declarative," which means that the sequence of the questions has no influence on the end result of the configuration. This is a distinct advantage of e-Con, which makes the maintenance of data structures and rules much easier.

Example:

Starting position:

Engine = 2.0 l.
Number of gears = 4
AC = No

Rules:

If Gears = 5 then AC = Yes
If Engine = 2.0 then Gears = 5

Imperative result:

Engine = 2.0 l.
Number of gears = 5
AC = No

Declarative result:

Engine = 2.0 l.
Number of gears = 5
AC = Yes

When a value is changed, only the applicable rules are re-triggered. In this example, the first rule will be executed again because the number of gears has been changed by the second rule.

e-Con is web-enabled. Web technology is one of the architectural components of e-Con. The configuration takes place in a web browser (Microsoft Internet Explorer 5.5). The models are stored electronically as XML documents, which serve as direct input for the web browser.

Using the latest (Microsoft) web technology, it is possible to deliver a flexible, customizable end user web interface (Front End). The presentation of e-Con to the end user can be customized, for example, the sequence of questions asked, and the dynamic visibility and non-visibility of questions.

Another feature is the simplicity of working off-line. This is mainly based on the fact that e-Con is web-enabled. The models used for e-Con are electronic documents (XML) with all the model information (data and rules) in them. On a local off-line computer, these electronic documents can be used in a web browser, and the configuration can be done off-line. Of course, it is necessary to upload the configured data to the network once the computer is on-line again.

In most cases, it is necessary to have some knowledge of computer language in order to design and maintain the model rules. A dedicated Studio has been designed for e-Con. One of the main purposes of this Studio is to add rules to the model. The Studio is a user-friendly tool with "drag and drop" functionality. It has an adaptive shell around the possible rules and functions so an end user can maintain the model rules.

Chapter 3 How to Build Your First Model

The purpose of this chapter is to give you a brief overview of how a Generic Model is built in e-Con. It outlines the main steps of creating a model. However, you will have to study this manual more closely to get more detailed information about all the aspects and possibilities of the Studio.

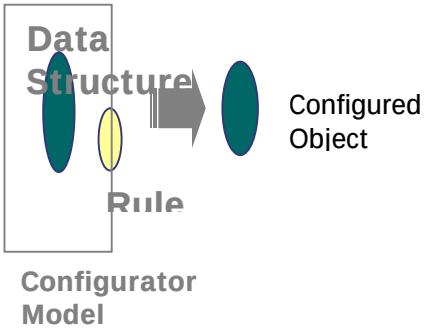
In order to gain more experience with modeling, we recommend consulting our training manual. This manual presents a lot of exercises, from simple models to the more complex. An expanded version of the training manual is also available. It provides a complete step-by-step explanation of all the exercises.

The first section of this chapter provides an overview of modeling with e-Con. The various aspects of modeling are discussed. The second section explains how to build the data structure, step by step. The result will be a finished data structure for a simple bike. The third section will describe how to create business rules in the e-Con Studio. This will result in the addition of some simple rules to the bike. Finally, how to integrate this model into the business process is described.

3.1 Overview of e-Con Modeling

An e-Con configuration model consists of two parts: the data structure and the business rules. See the diagram below.

The gray arrow in this figure represents interaction with the end user. The end user



selects the options and answers the questions of the configuration model, which results in a configured object.

The data structure is built in Axapta and the business rules are built in the e-Con Studio. The data structure in Axapta consists of a few elements listed hierarchically:

- e-Con Project (optional)
- Generic Model
- Version
- Entity
- Entity Member

The highest mandatory level is the Generic Model. It consists of a name and a description. An e-Con project is actually the top level, but it's not mandatory to use a project. Projects are useful and actually highly recommended when there is more then one model involved, the so called sub models. The second level is the Version. Just like the BOM and Routings in Axapta, a generic model can have versions. The next level is the Entity. Using Entities, you determine into which table e-Con is to insert records. For example, if you want to create an e-Con model for the configuration of customers, an Entity for CustTable is required because Axapta stores customers in the CustTable. The lowest level is that of the Entity Member. Using members, you can specify an Entity in more detail. It is not enough to just tell e-Con into which table records must be inserted, because a record has many fields that probably have to be configured in a certain way. That kind of information is set up in the members of an Entity.

There are three types of Entities members:

- Field Property
- Additional Property
- Relation

An Entity member of the "Field Property" type is used when you want to enter information in a field or change a field in the table concerned. For example, if you want to enter the name of the customer in your configuration model, you have to add a field property that is linked to the Name field (assigned to names) in Entity CustTable (assigned to customers).

An Entity member of the "Additional Property" type is used when you want to request or store information that isn't related to any Axapta table field. For example, in the Bike demo model, we ask the user for the model and type. That information isn't available as a field in Axapta. So those are additional properties, or actually, they are a type of variable.

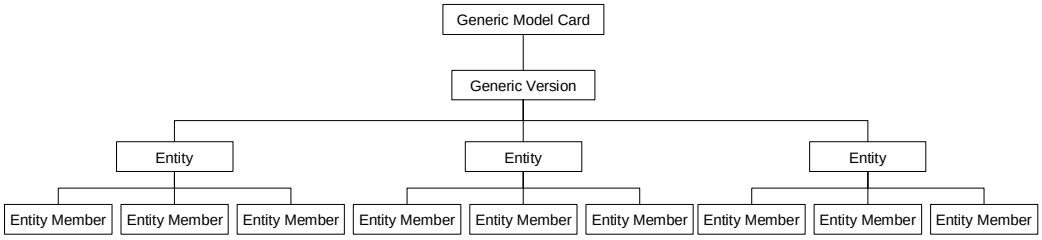
The last type, "Relation," is used to set up the relationship between different Entities. Here, we will use the bike model as an example, too. The bike model consists of a lot of Entities, including an Entity for inserting an "Item", a "BOM version", a "BOM Header" and several "BOM Lines". All those Entities have a certain relation because there is a relation in Axapta between those records, as well. The "BOM version" corresponds with an "Item", and is linked to the item via the "Item no." in the "BOM version" record.

The same goes for the "BOM Header" and "BOM Lines", which of course all have a relation to the "BOM version".

The relations that exist in the Axapta Database have to be translated for the Entities. To do this, we use the "Relation" member type.

In the example of the bike, there is a member relation in Entity InventTable (Item) for setting up the relationship between the item and the relevant BOM version. In Entity BOMVersion (BOM version), there are a few relations as well to set up the relationships between the BOM version and the BOM Header (entity BOMTable) and the BOM Lines (entities BOM).

See the figure below for the data structure hierarchy.



Once the data structure is complete, it will be translated into XML by a special function called "Generate XML". An XML file is then created with the data structure in it.

All the Entities and the members are available in the XML file and are represented by labels. The XML file does not have Entity numbers (table numbers) or member numbers (field numbers). Instead, everything is based on labels. Consequently, when you are building the data structure, you have to add a label to every Entity and member. Always use a descriptive label, because this makes it much easier to create business rules later on. For example, we will add the label "NAME" to the field property of Entity CustTable (Customer) that we discussed before.

So labels are always added to your data structure for these reasons:

- Labels are used in the XML file to represent the Entities and members
- The label description of the label is used as a description at the Front End
- The version label has another special purpose – the label ID will be used as the filename for the XML file created from your Generic Model.

Within the e-Con Studio, business rules can be added to this data structure. The data structure represented by the labels is also available in the Studio. Drag and drop functionality makes it very easy to use members from the data structure in your rules.

Furthermore there are a lot of attributes and functions available to meet special requirements, like price calculation, member (in)visibility, member defaults, consulting the Back End (Axapta Database), etc.

After completing the rules, you have to exit the Studio and save the rules. Now the model is ready to use. The last step is to integrate the model into your business processes, if desired. This makes it possible to start up the model from a certain form in Axapta, like a Sales Order.

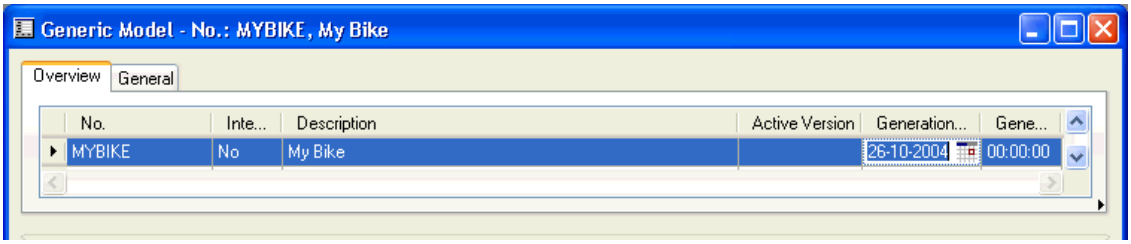
3.2 How to Create Your Generic Data Structure

This section provides step-by-step instructions on how to build a generic data structure. The most elemental steps are described. In order to get a more complete idea of all the possibilities, you may need to study the manual and perhaps the training manual in detail. This section will serve as a good starting point to introduce you to modeling.

- In this section, we will build a simple data structure for a bike. This bike consists of an item itself, a BOM version and lines. The BOM consists of two lines, one line for the frame and one for the wheels.
- First import the demo bike model, because it will add the necessary items to the database. See chapter 6.5 or Installation manual.

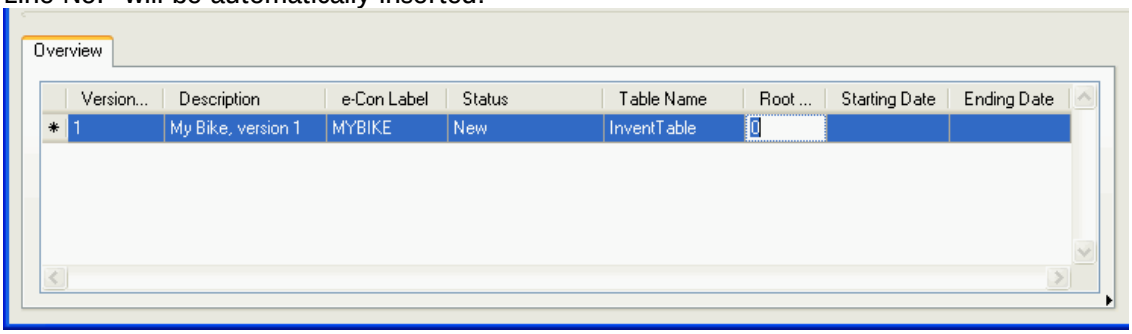
Let's get started.

1. The first step is to create a model.
   - Select "Models" from the e-Con main menu. The Generic Models are shown.
   - Insert a new line with Ctrl-N.
   - Enter a No. and description.

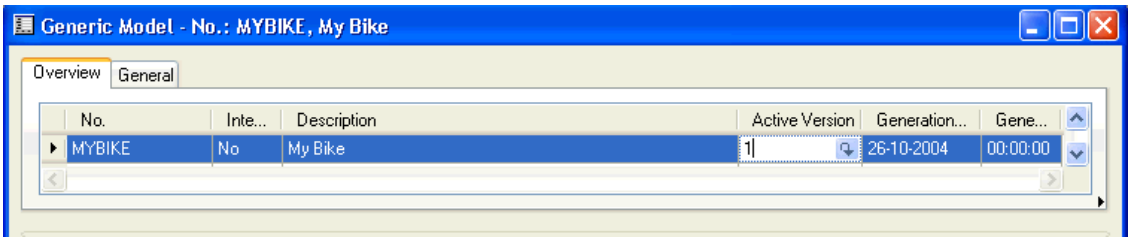


2. Create a version.
   - Go to the lines underneath the models and insert a new line with Ctrl-N.
   - Enter a version number (e.g. 1) and a description. Add a label. This label will be the name of the XML file once the data structure is generated. If the label doesn't already exist, create a new one. Click the lookup button in the "Label" field. The Labels form will appear. Insert a new record and enter the label ID and description. If the label also exists as Axapta label the field “Axapta label” will be check-marked automatically. Select the label and "Past selected label". Notice that "New" is displayed in the "Status" field of this version. Then enter the Root Entity. This tells e-Con which Entity (table record) to insert first of the data structure. This is in fact the top level of the data structure, in this case Entity InventTable. “Root

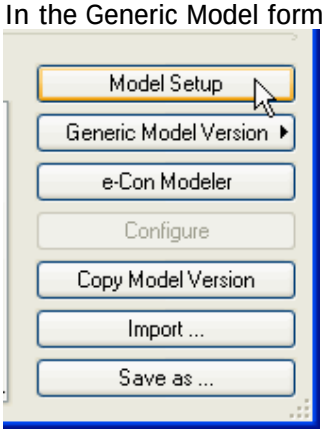
Line No.” will be automatically inserted!



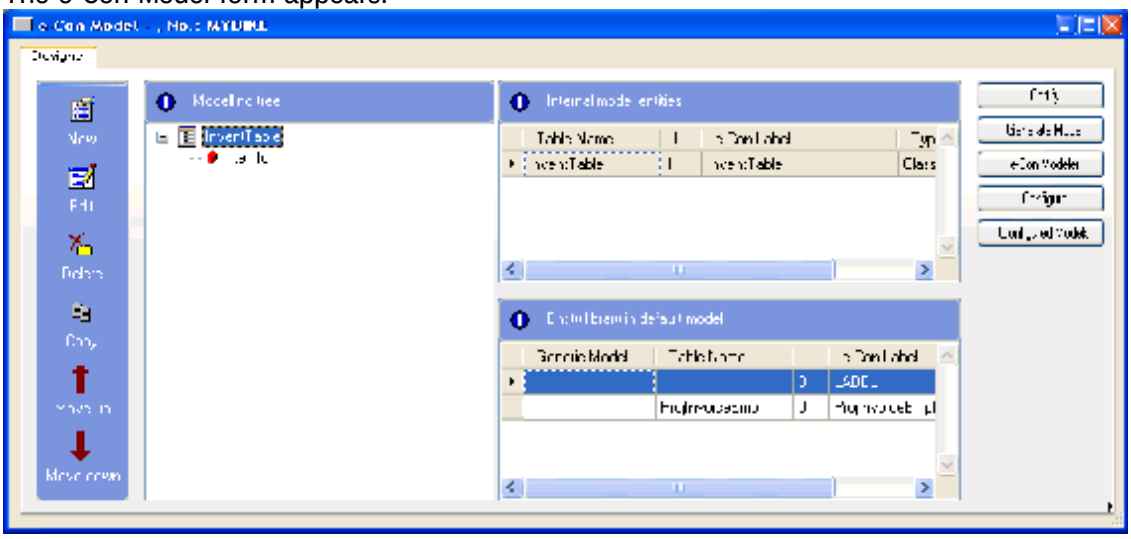
- Go to the Generic Model line and select the version as the active version for the Model.



3. Set up the model.
   - In the Generic Model form, click Model Setup.

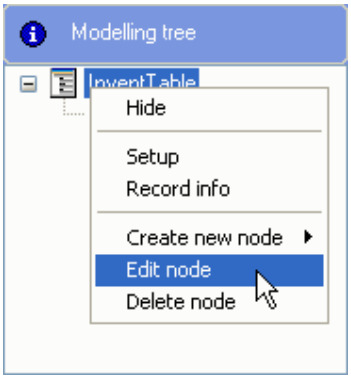


- The e-Con Model form appears.

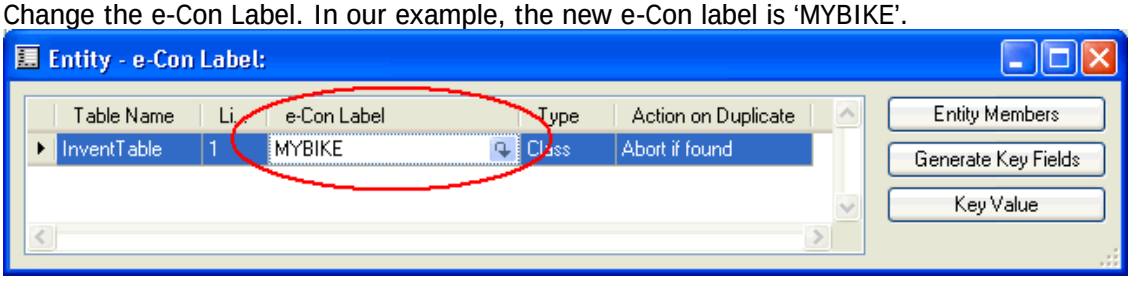


Every record in Axapta has one or more fields that are strictly necessary. They are called key fields. So, because we want to insert records into the Axapta database, we have to take that into account. Those key fields are field properties that we need anyway. So the Modeler does not have to find out what the key fields of every table are. The root entity's key fields are automatically entered as field properties. In our example, the Itemid property is the InventTable's key field.

- Rename the tree. To do so, in the Modelling tree, right-click 'InventTable' and click 'Edit node'.

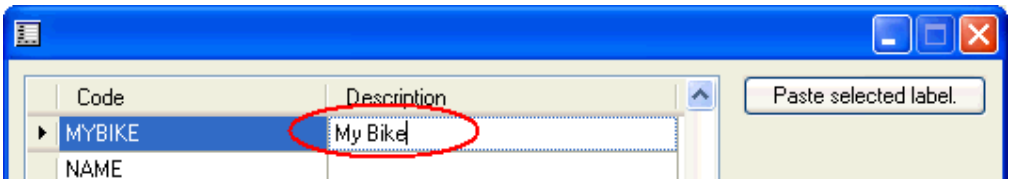


- Change the e-Con Label. In our example, the new e-Con label is 'MYBIKE'.

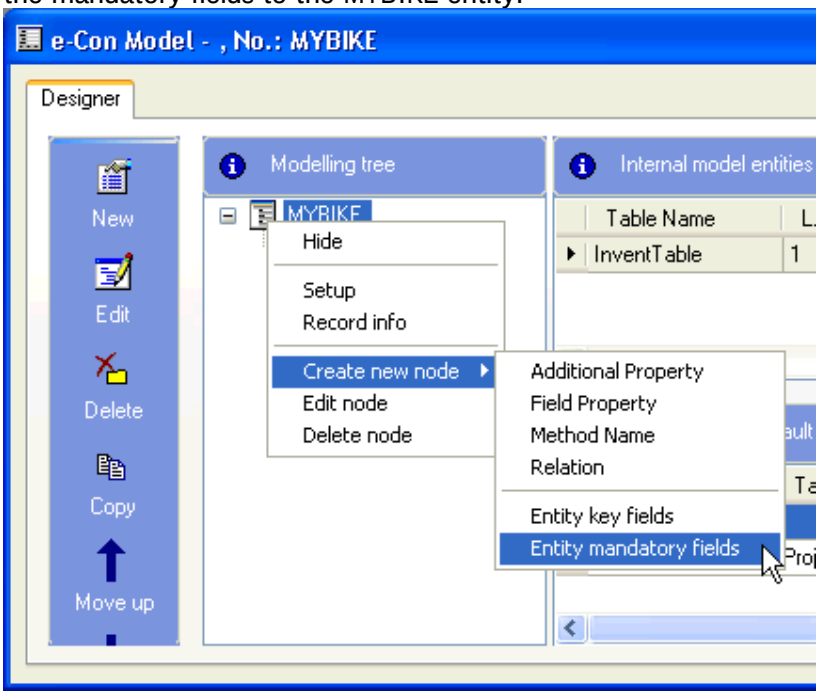


Tips & Tricks:

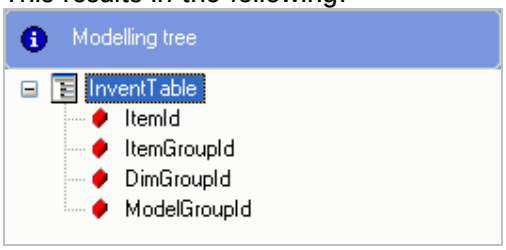
If you change the e-Con Label, make sure that the new e-Con label's Description is also specified! The descriptions are used in the final model. To specify a description, browse in the e-Con Label field of the Entity form and add a description to the e-Con label's Code.



4. Add the entity members:
   - To beale to create a valid record in the entity's table in Axapta, you must add the mandatory field properties to the entity in modelling tree. To do so, right-click the entity and click 'Create new node', 'Entity mandatory fields'. In our example, add the mandatory fields to the MYBIKE entity.



This results in the following:



- Check the key fields and mandatory fields property settings. To do so, right-click the property and click 'Edit node'. The Entity Member form appears.

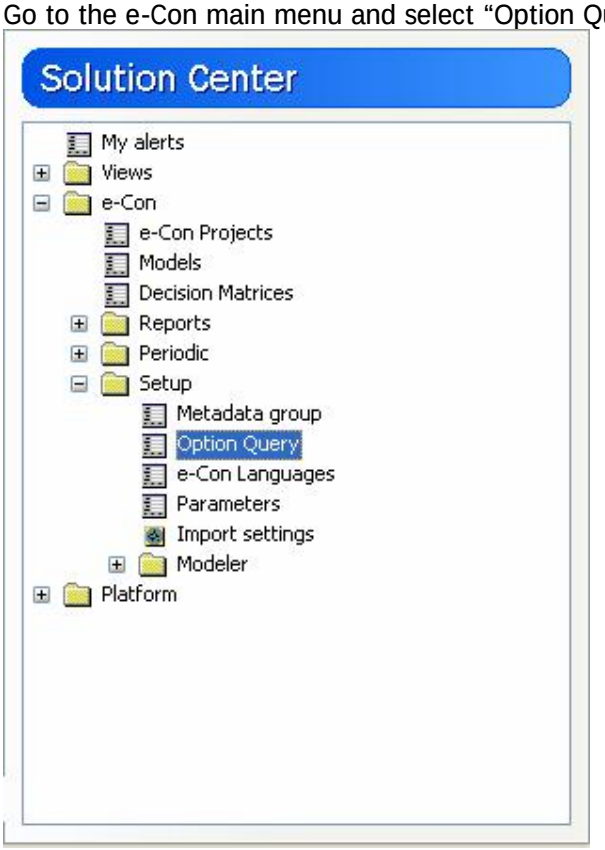
- You can enter the Default Code and the No. Series on the Field tab. Check the properties on the following settings:

Entity member	Editable	Constant	Default Code	No. Series
ItemId	Yes			Bike
ItemGrouId		Yes	eCon Bike	
DimGroupId		Yes	Std-Dim	
ModelGroupId		Yes	DEF	

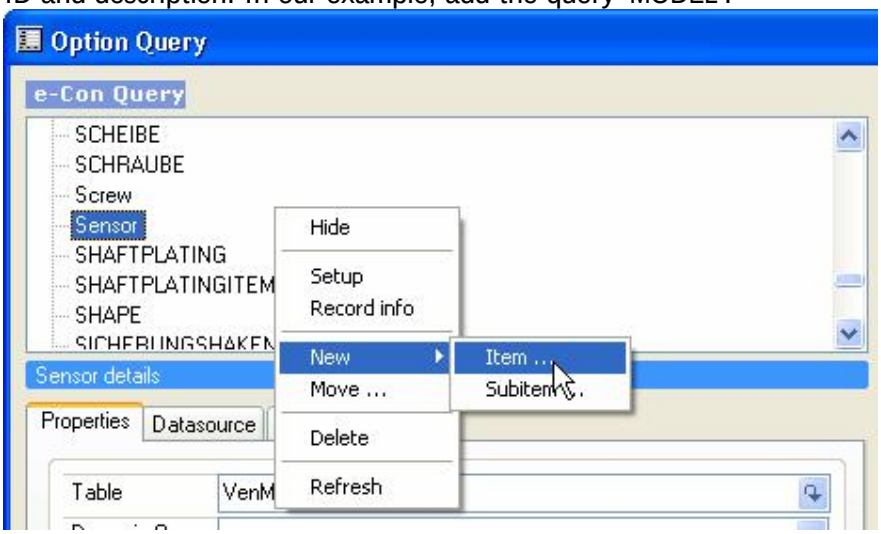
- You can add extra properties to the entity. You can add:
  - Field properties, which are used to enter data in or to retrieve data from table fields in Axapta. To add a field property right-click the entity and click 'Create new node', 'Field Property'. Pick the field for which you want to add a property.
    If required, you can check and edit the field property's settings. To do so, double-click the field property. Only check the settings on the Overview tab and the Field tab..
  - Additional properties, which are used to enter additional data, which will not be entered in or retrieved from an Axapta table. To add an additional property, right-click the entity and click 'Create new node', 'Additional Property'. Enter the aditional property's settings, as required.
  - Relations, which are used to relate the entity to another Axapta table. To add a relation, right-click the entity and click 'Create new node', 'Relation'. Pick the relation that you want to add.
    Add the relation's entity. To do so, right-click on the relation and click 'Create new node', 'Entity'. As a result, the entity as specifeid in the relation is added and the relation's settings are completed. Also the relatione entity's key fields are added a field properties.
    If required, you can check and edit the relation's settings. To do so, double-click the relation. Only check and edit the settings on the Overview and the Relation tab.
    You can also add the mandatory fields and extra properties to this entity.

  In our example, to be able to:
  - Enter the bike's description, add a field property. Pick field 'ItemName'. Select the Editable and Visible check boxes.
  - Enter the item type, add a field property. Pick field 'ItemType'. Select the Constant check box. The field property's Default Code must be 'BOM'.
  - Enter the bike's model, add an additional property 'Model' with a field length of 30. Also select the Editable, Mandatory, and Visible check boxes.
  - Relate to the BOM version in your model, add a relation. Pick relation 'BOMVersion'. Select the Editable, Visible, and Constant check boxes.

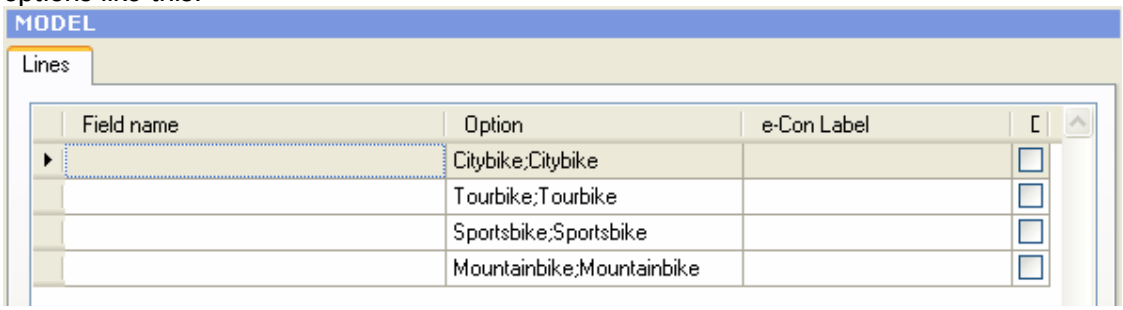
5. Add option queries to entity members, if required.
   To offer the end user the possibility to select options from an option box, you can add option queries to properties. To add option queries:
   - Go to the e-Con main menu and select “Option Query” from “Setup”.



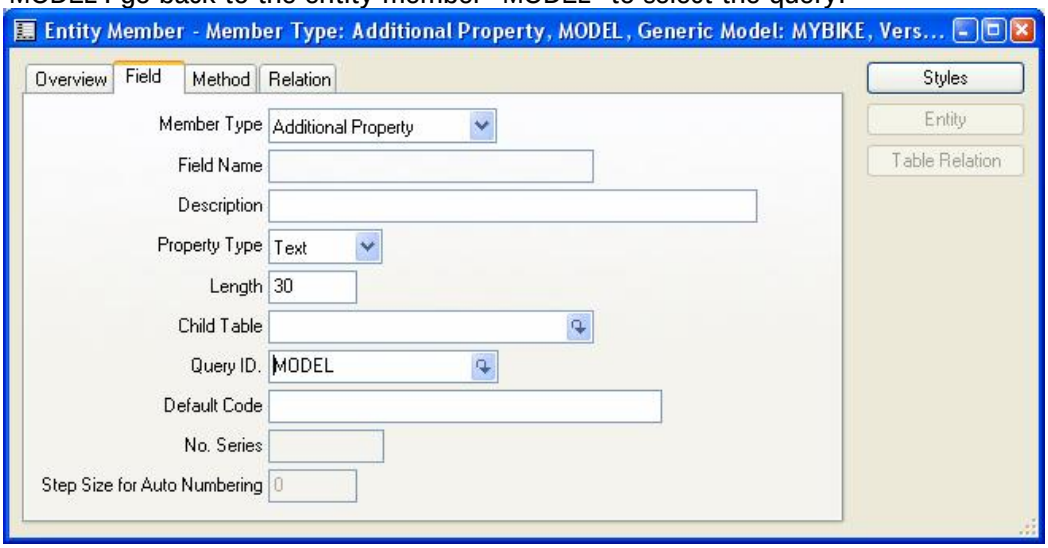
   - Add a new query by selecting ‘New’ – ‘Item’ from the context menu and give it an ID and description. In our example, add the query ‘MODEL’.



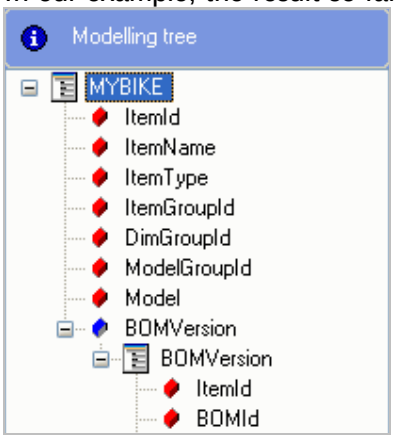
- Go to the Option Query Lines area and add the options. In our example add options like this:



- Exit this form and link the option query to the property you want. To do so, double-click the property and enter the option query in the Query ID field on the Fields tab. In our example, link the option query to the additional property 'MODEL'. go back to the entity member "MODEL" to select the query.



In our example, the result so far is as follows:



Complete the model in our example as follows:

First complete the BOMVersion entity and entity members:

6. Check the key fields of the BOMVersion entity on the following settings:

Entity member	Editable	No. Series
ItemId	Yes	
BOMId	Yes	Bike

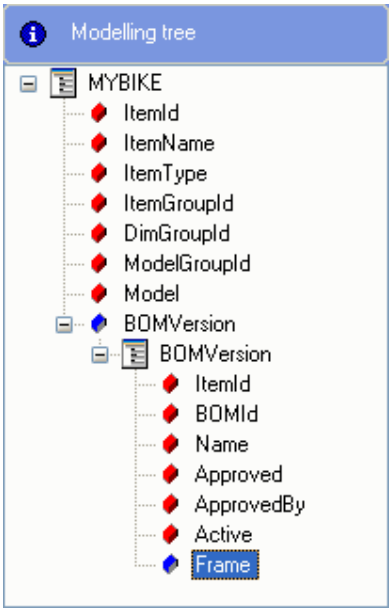
7. Add the following field properties to the BOMVersion entity:

Entity member	Editable	Constant	Default Code
Name	Yes		
Approved		Yes	Yes
ApprovedBy		Yes	Econ
Active		Yes	Yes
FromDate	Yes		

8. Add the following relations to the BOMVersion entity:

Pick relation:	e-Con Label	Editable	Visible	Constant
BOM	Frame	Yes	Yes	Yes
BOM	Wheel	Yes	Yes	Yes

The result so far is:



As the 'Frame' and 'Wheel' relations are almost the same, you can, for example, first build the 'Frame' relation and then copy it. After copying, you only need to rename the relation copy to 'Wheel' and its 'Frame' relation (the relation to the Item table) to 'Wheel'. So, first only add the 'Frame' relation, its entity and its entity members. To do so:

9. Add the mandatory fields to the 'Frame' relation entity.
10. Check the 'Frame' relation entity's key fields and mandatory fields on the following settings:

Entity member	Editable	Constant	Step size autonumbering	Default Code
BOMId	Yes			
LineNum	Yes		1	

RecId	Yes			
ItemId	Yes			
UnitId		Yes		Pcs

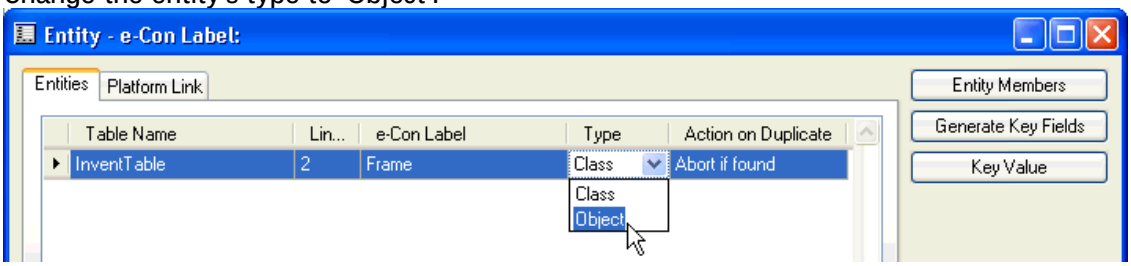
11. Add the following field properties to the ‘Frame’ relation entity:

Entity member	Editable	Constant	Default Code
BOMType		Yes	Item
BOMConsump		Yes	Variable
BOMQty	Yes		1
InventDimId		Yes	Axapta

12. Add the following relation to the ‘Frame’ entity:

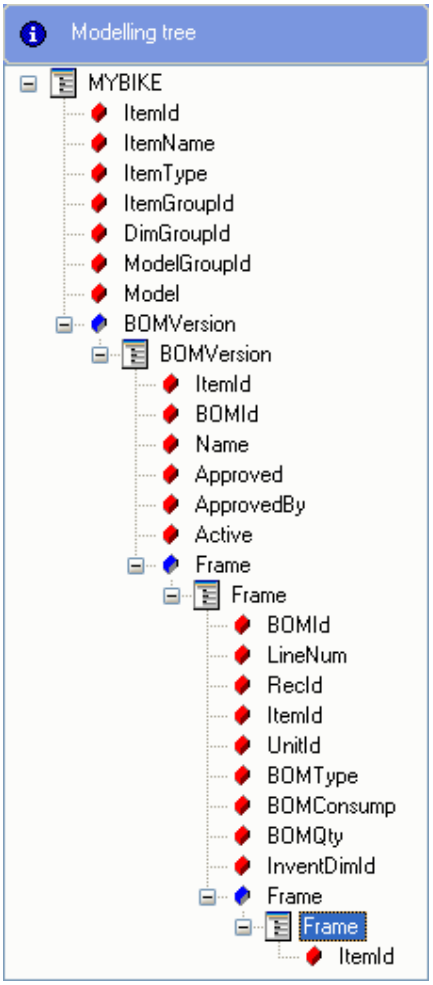
Pick relation:	e-Con Label	Editable	Visible
ItemId	Frame	Yes	Yes

13. Add the ‘Frame’ relation’s entity.
14. Change the entity’s type to ‘Object’.



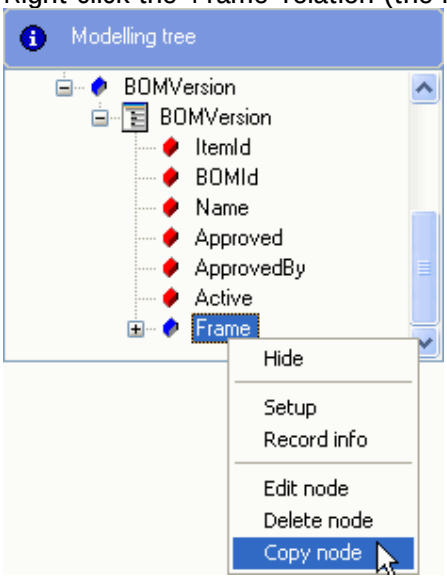
An entity can be of the type "Class" or "Object." An Entity of the "Class" type is a generic entity, which means that the linked table will be filled by e-Con. An Entity of the "Object" type is not a generic entity, which means that the linked table is not filled by e-Con. Entities of type “Object” are only used to retrieve information.

The result so far is as follows:

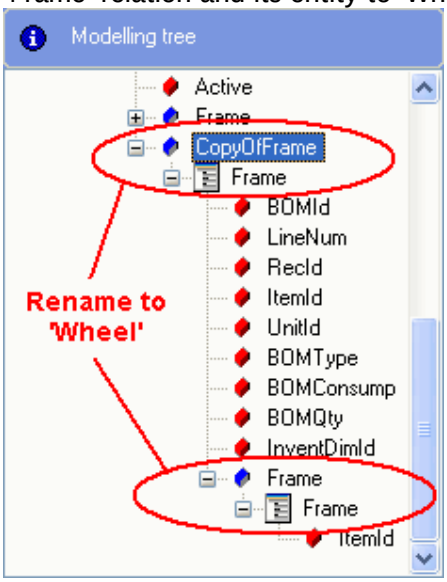


Now copy the ‘Frame’ relation and rename the relation’s copy to ‘Wheel’. Also rename the ‘Frame’ relation (the relation to the InventTable table) to ‘Wheel’. To do so:

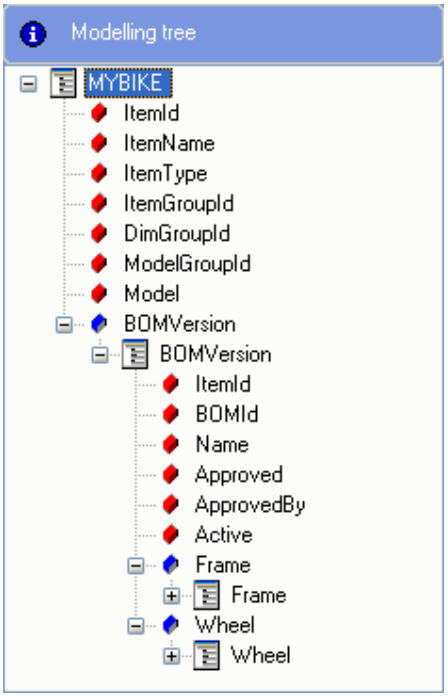
15. Right-click the ‘Frame’ relation (the relation to the BOM table) and click ‘Copy node’.



16. Rename the ‘CopyOfFrame’ relation and its entity to ‘Wheel’. Also rename the ‘Frame’ relation and its entity to ‘Wheel’.



The result so far is as follows:

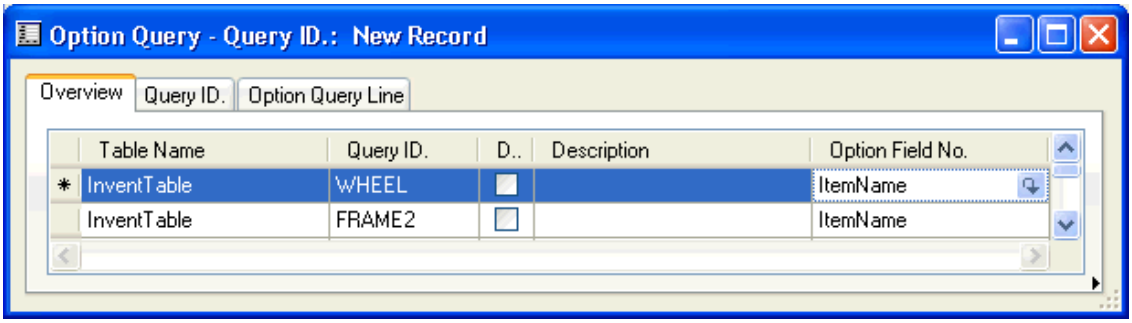


Add the folowing option:

- 'FRAME2' to the 'Frame' relation (to the InventTable).
- 'WHEEL' to the 'Wheel' relation (to the InventTable).

To add the option queries to the relations:

17. Add the option queries in the Option Query form. The Option Field No. must be 'ItemName'.



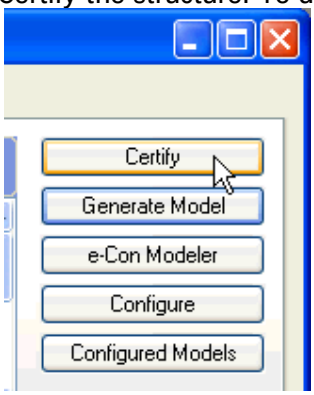
18. Add the following option query lines:

Option Query	Field Name	Option
FRAME2	ItemName	Frame*
WHEEL	ItemId	1200

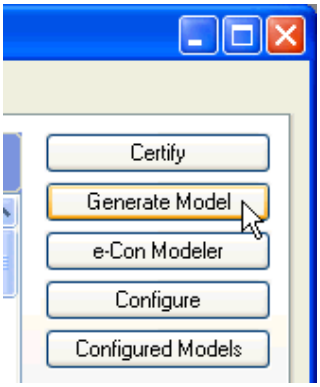
19. Actually add the option queries to the relations.

Once the data structure is finished, you can certify the structure and generate the model.

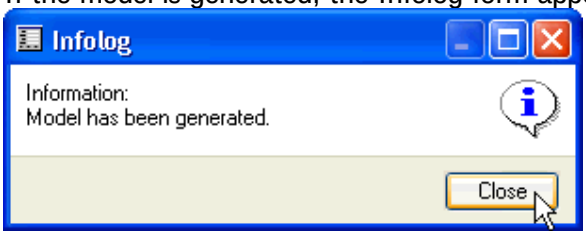
20. Certify the structure. To do so, in the e-Con Model form, click Certify.



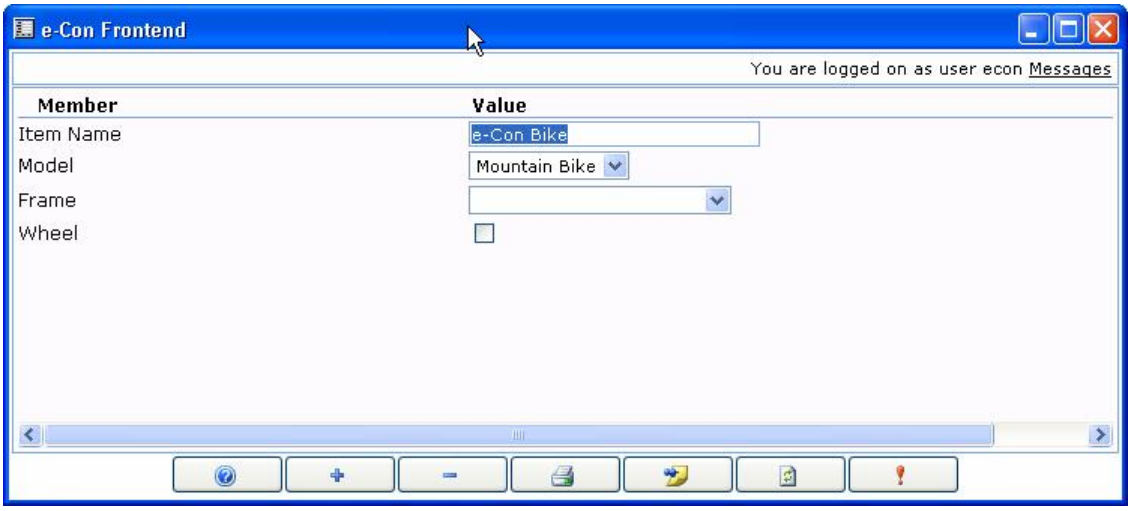
21. Generate the model. To do so, in the e-Con Model form, click Generate Model.



22. If the model is generated, the Infolog form appears. If everything is OK, click Close.

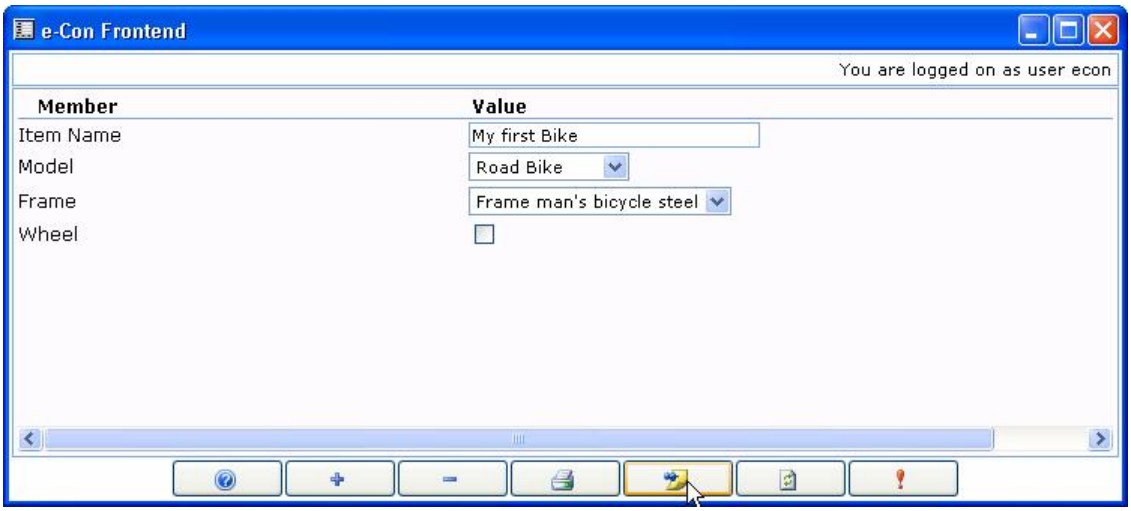


23. View the result. To do so, in the e-Con Model form, click Configure. The result will look as follows:

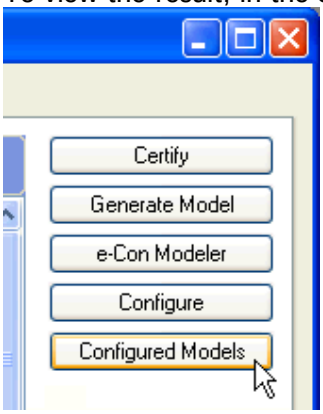


Finally, configure your first bike, process the configuration and view the result in Axapta. To do so:

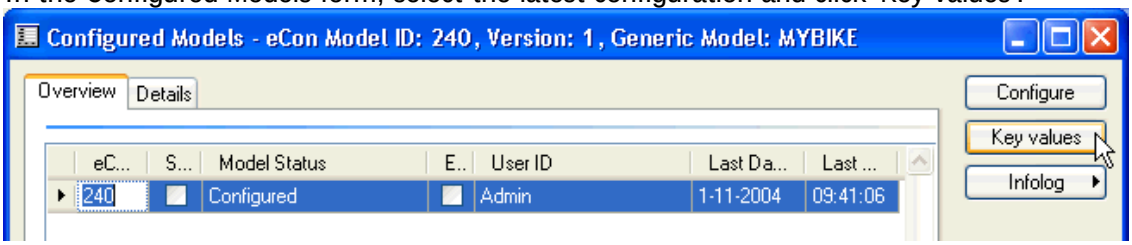
24. In the e-Con Model form, click Configure and enter the configuration in the e-Con Frontend.



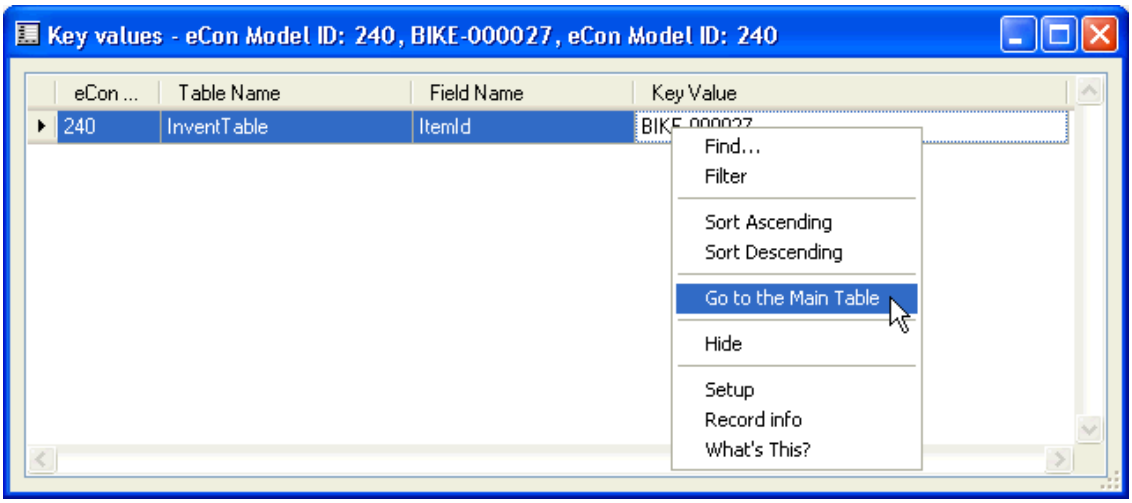
25. Click Process to actually enter the configuration in Axapta.
26. To view the result, in the e-Con Model form, click Configured Models.



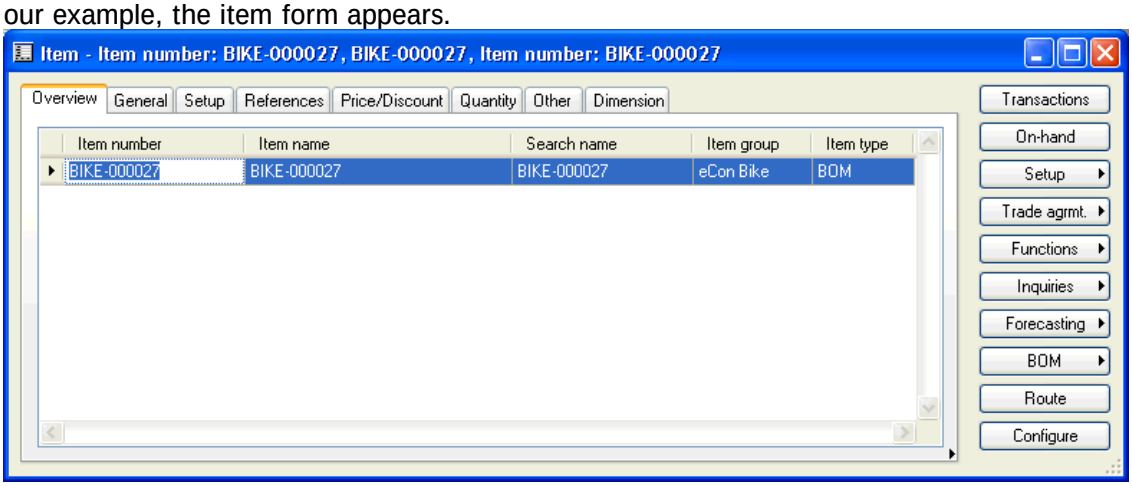
27. In the Configured Models form, select the latest configuration and click 'Key values'.



28. To look for the item that is created in Axapta, on the Key Value form, right-click the key value record and click 'Go to the Main Table'.



As a result, the main table's form in Axapta appears, displaying the created item. In our example, the item form appears.

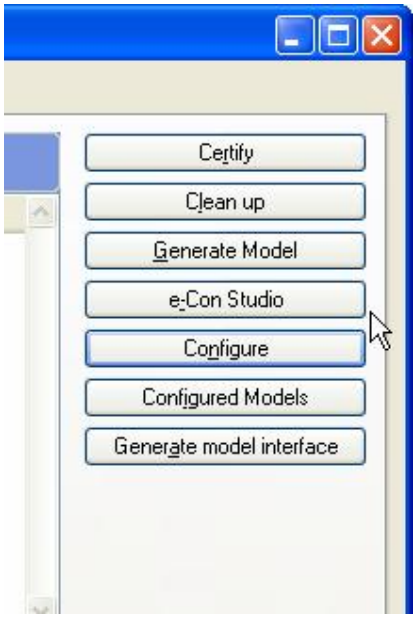


3.3 Add Your Business Rules

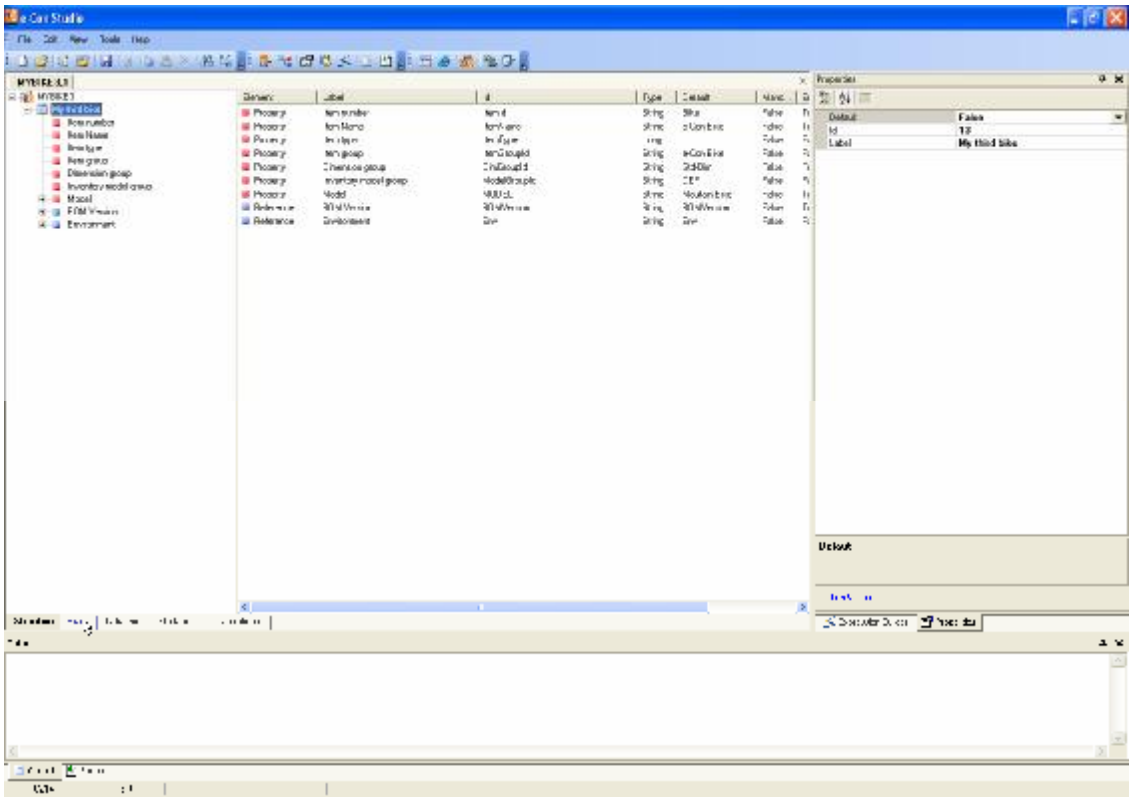
This section describes how some business rules can be added to a model. The starting position is the Bike model that is described in the previous section. We will add the following business rules step by step:

- The name of the item and the BOM Version of the bike that should be entered with the chosen model.
- A default value for the model.
- Some dependencies between the chosen model and the default value for the frame.
- Some dependencies between the chosen model and the visibility of the wheel.
- Some dependencies between the chosen model and the options possible for the frame.

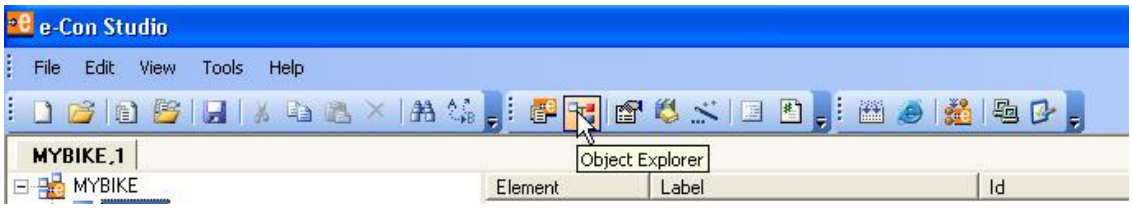
1. The business rules are added in the e-Con Studio. To do this, start the “e-Con Studio” with the button.



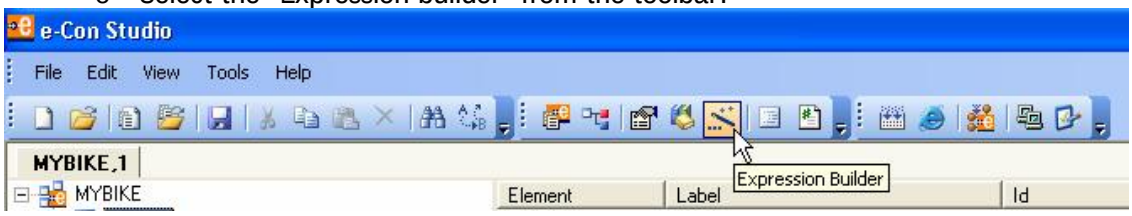
2. Add a new rule:
   - Select the rules view from the model by selecting the "Rules" Tab.



- Activate the “Object Explorer” and “Expression Builder”. The “Object Explorer” is needed to drag and drop model properties in to a rule. The “Expression Builder” is a great tool that’s helping us in the creation of rules.
    - Select the “Object Explorer” from the toolbar.

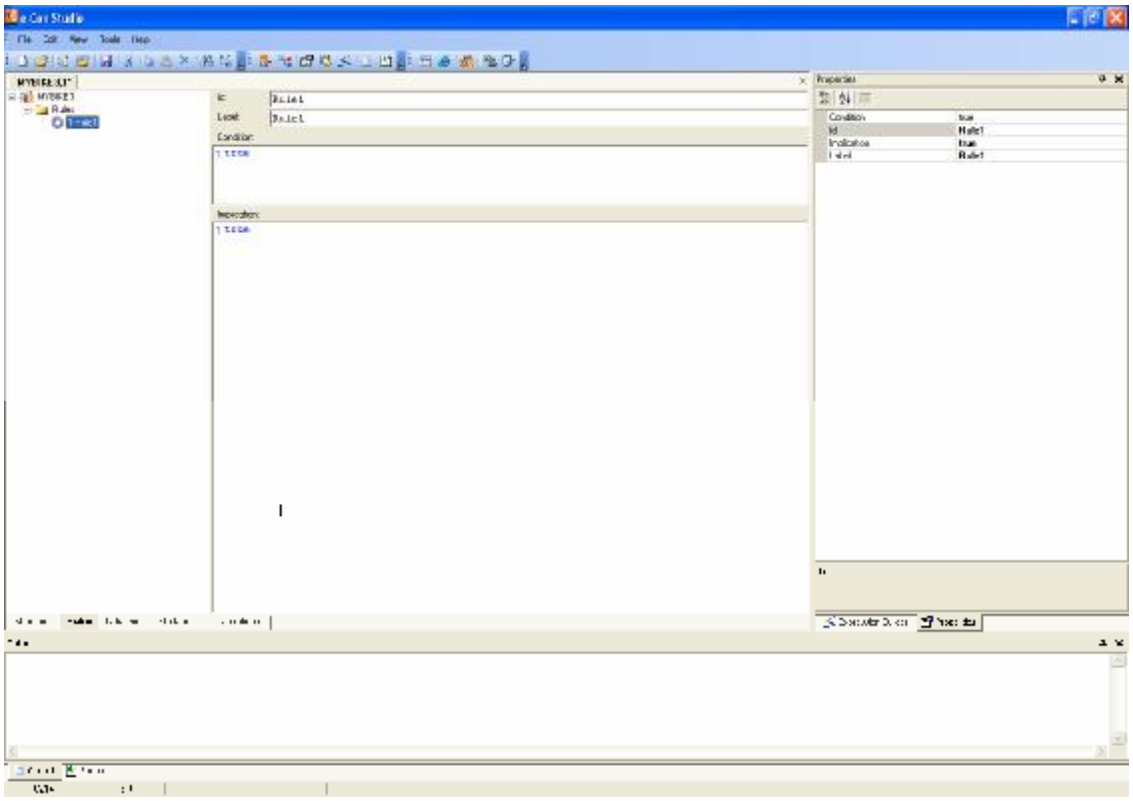


    - Select the “Expression builder” from the toolbar.



- Select the ‘”Rules Map” from the tree, right mouse click and select "New Rule" from the menu.

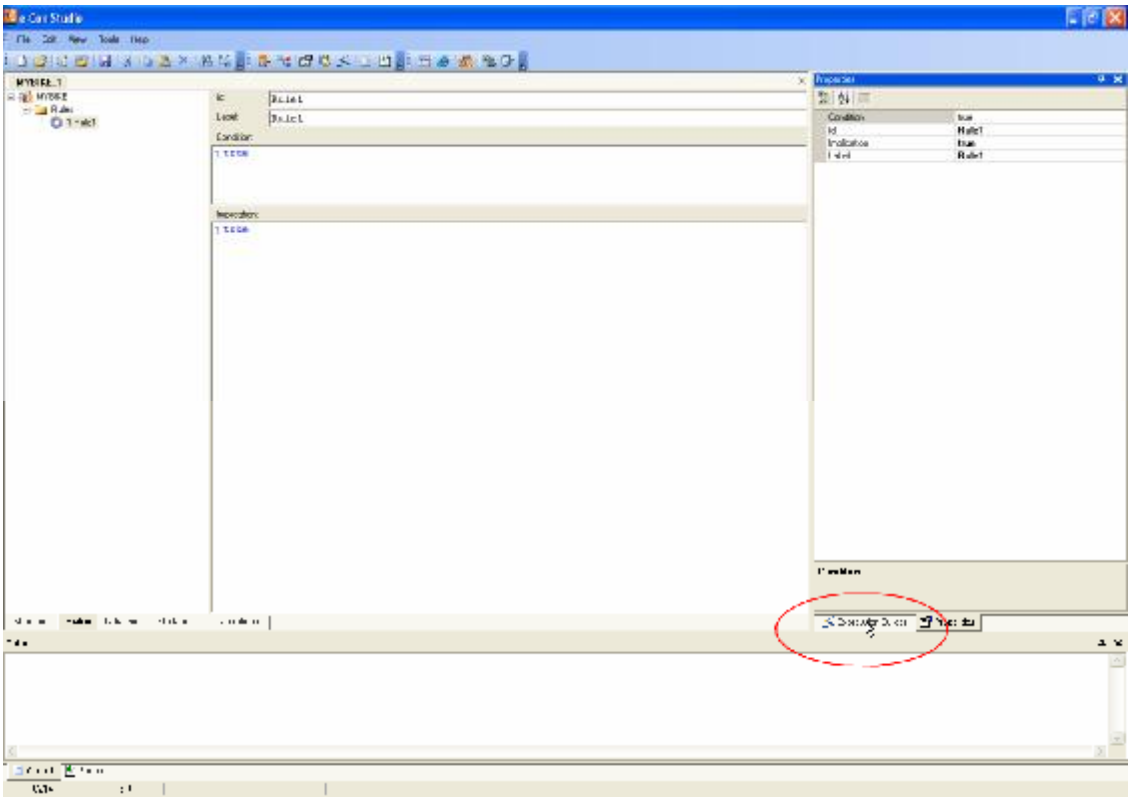
- As result a rule is added in the folder and shown in the edit pane of the studio:



We can enter the rule in this pane. Besides the rule “id” and “label” there is a "Condition" section and an "Implication" section. The "Condition" section is for when the rule is to occur, and the "Implication" section is for what is to occur.

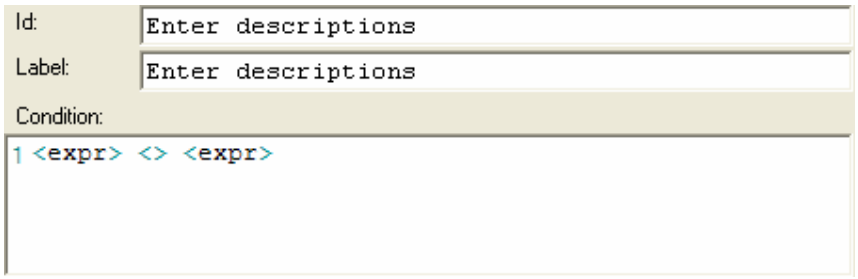
The first rule we want to create will be explained step by step. In this rule, we will enter the name of the item and the BOM version of the chosen model:

- Enter a descriptive id and label like: "Enter descriptions"
- Move the cursor to the "Condition" section
- Select the “Expression Builder”.

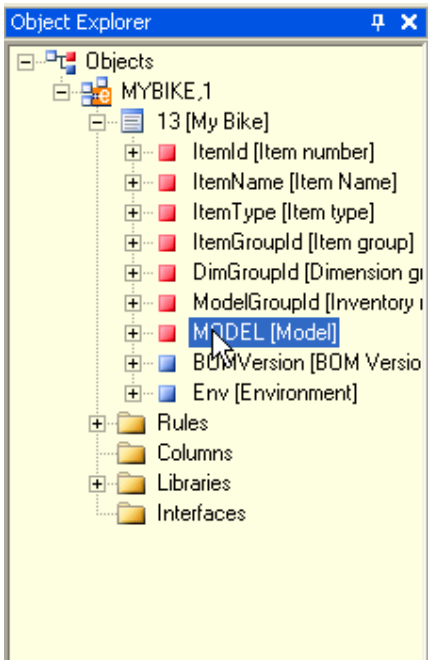


- Select the "Comparison" category, followed by the “Left <> Right” function and click copy.

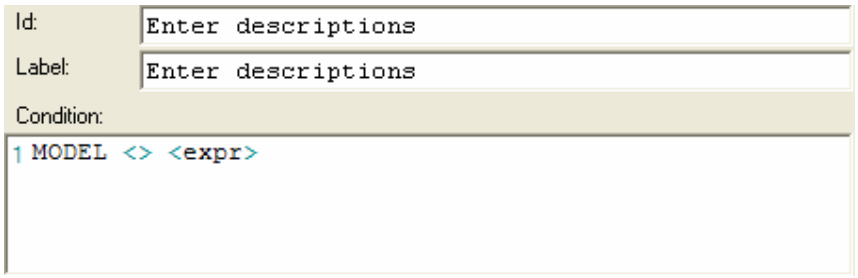
o The following will appear in the "Condition" section.



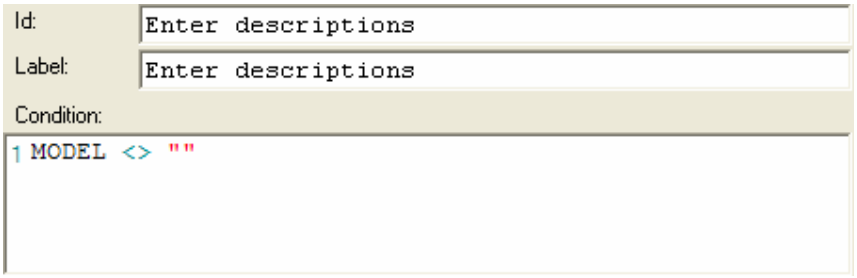
- Browse to the “Object Explorer”, expand the model structure and select the property ”model”.



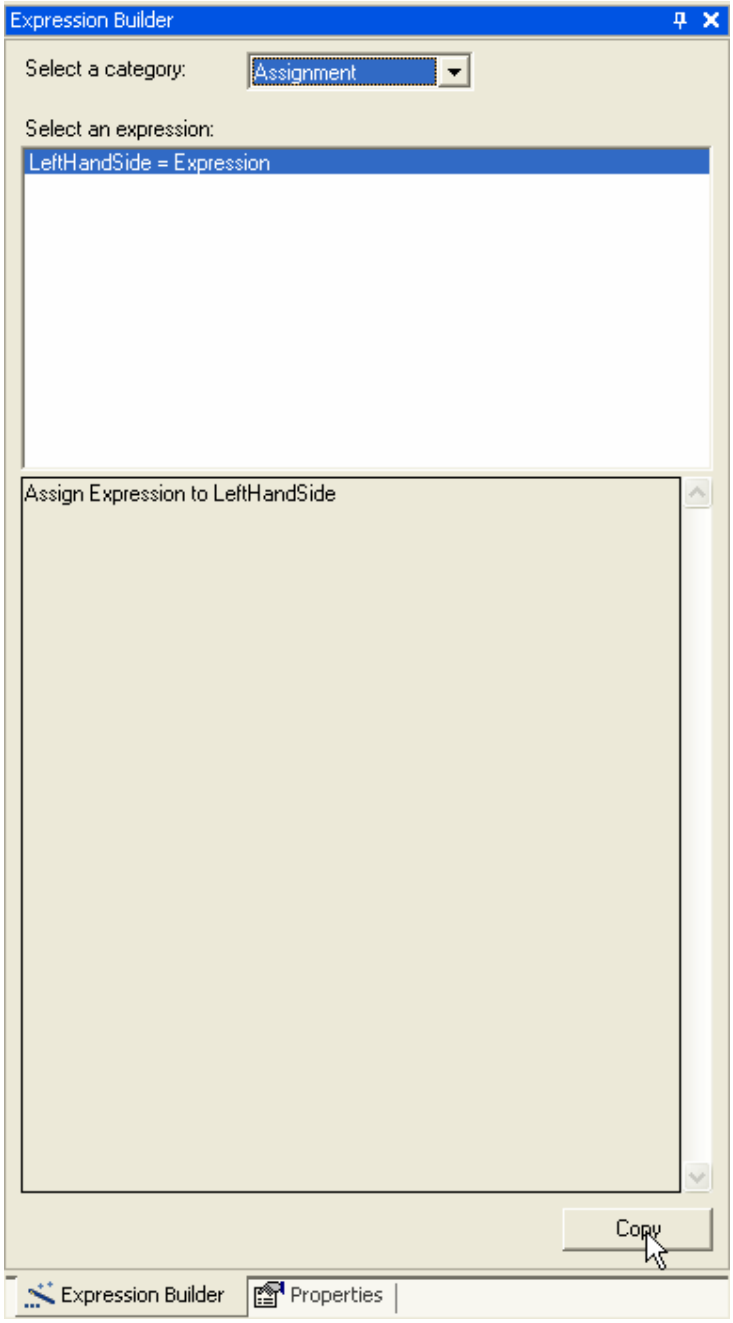
- Use the mouse to drag this member to the left "<expr>" expression code in the "Condition" section and drop it there. You will see that this member will replace the expression.



- Replace the right "<expr>" expression code with "" (2 double quotation marks). Now the condition tells us: This rule should be executed when a bike model is selected (when "MODEL" is not equal to ""; that is, when MODEL is not empty).

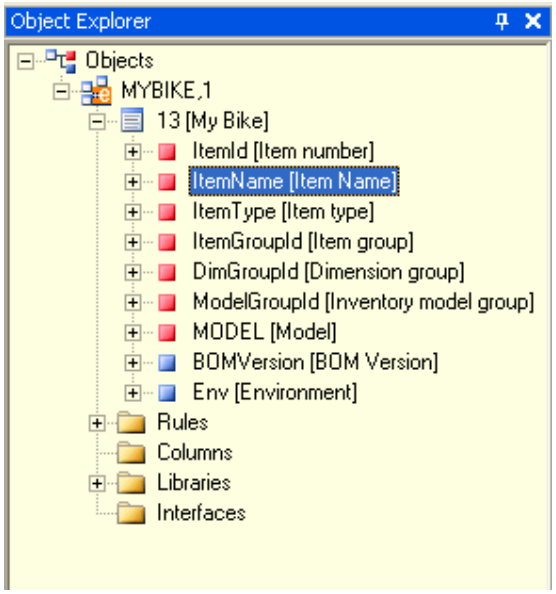


- Place the cursor in the "Implication" section of the rule, select the “LeftHandSide=Expression” function from the “Expression Builder” and click “Copy”. This function is available in the category “Assignment”.



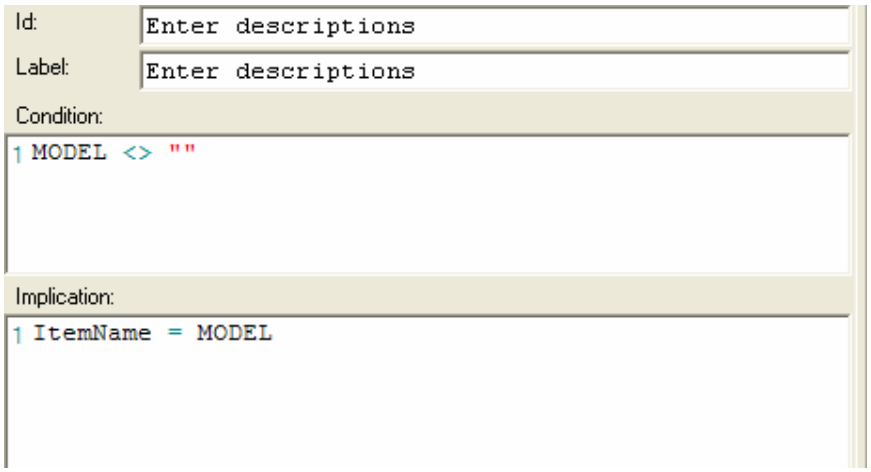
o This will result in:

- Select "ItemName" from the data structure.



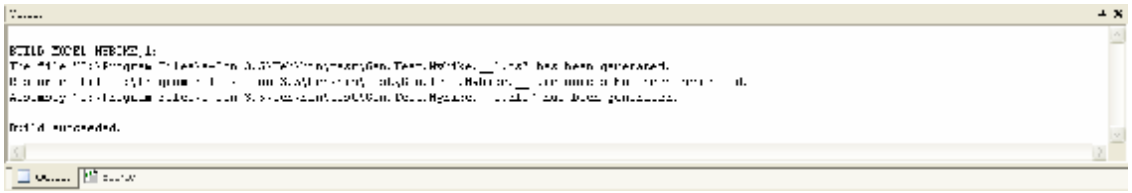
- Drag and drop this member into the left "<expr>" expression code in the "Implication" section.

- Select "Model" from the data structure, and drag and drop it into the right "<expr>" expression code.

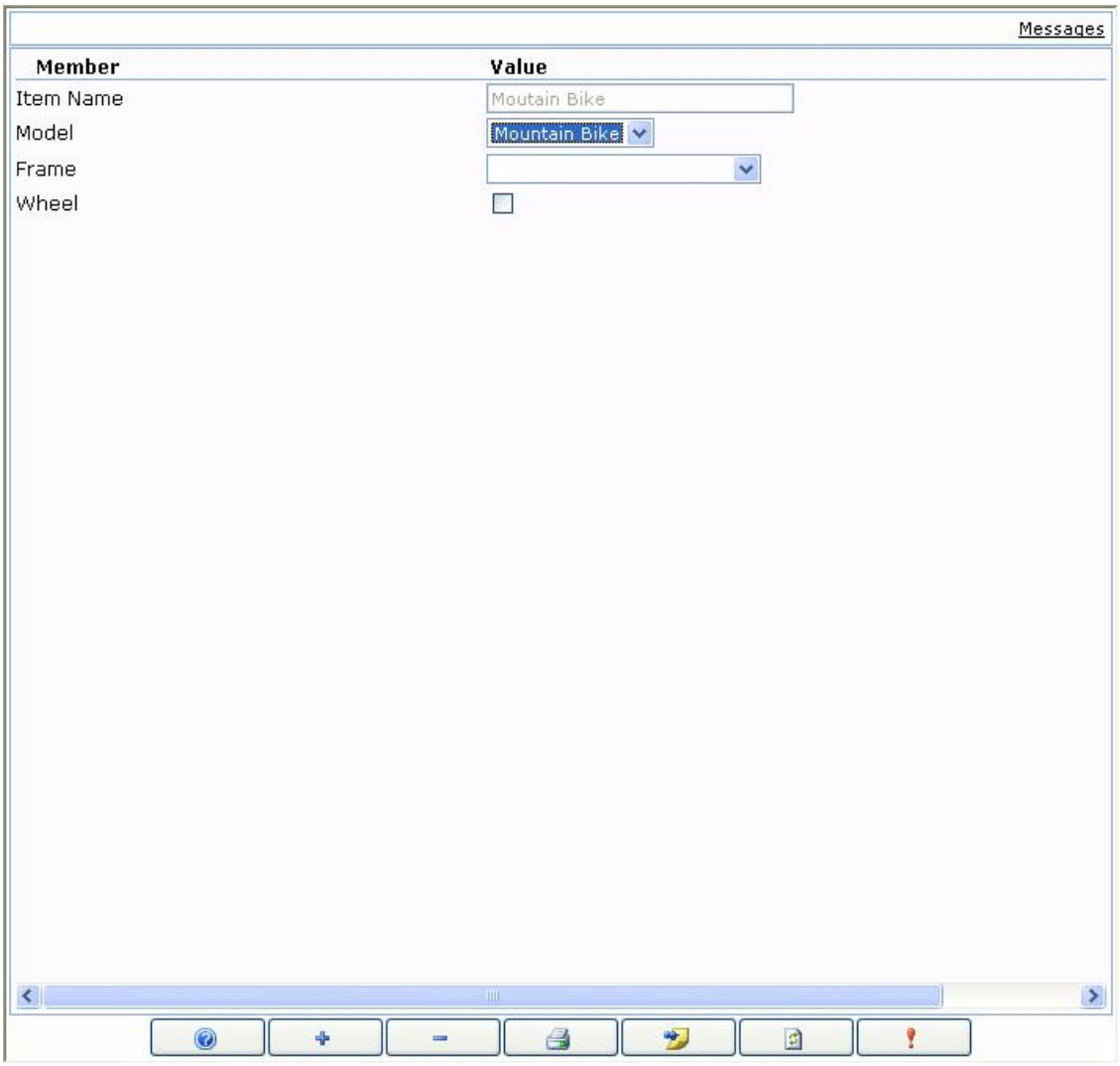


- Before we continue the rule by entering the name of the BOM Version, let's test this rule.
- Click "Browse" in toolbar.

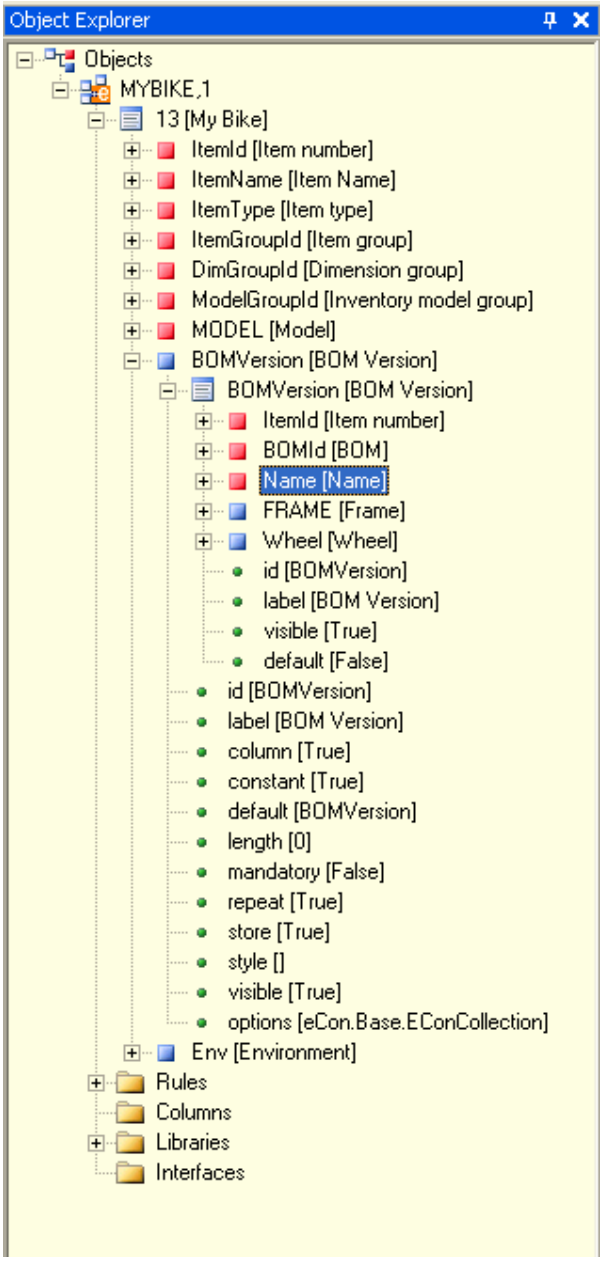
- Now it's possible for you to test your model, but it isn't possible to save the configuration in Axapta. This "Browse" functionality is for testing purposes only. Notice that the model is compiled before the browse is performed. The results from the compile or build is logged in the output window at the bottom of the studio.



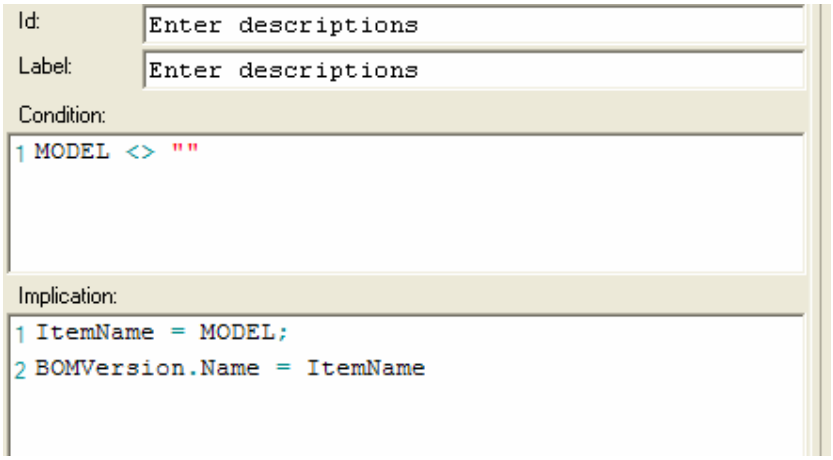
- Select a model of bike. You will see that the description is entered.



- Now we will expand the rule. To do this, switch back to the model and select the rule.
- Select the "Implication" section with the mouse, and add a semicolon after the rule that already exists.
- Press <Enter> and copy the “LeftHandSide=Expression” function from the “Expression Builder”. Siply entering “=” is possible as well.
- Replace the left "<expr>" expression code with the "BOMVERSION.NAME" member. Click the "BOMVersion" member to expand the menu and retrieve this member. Use the drag and drop functionality to place this member in the rule.

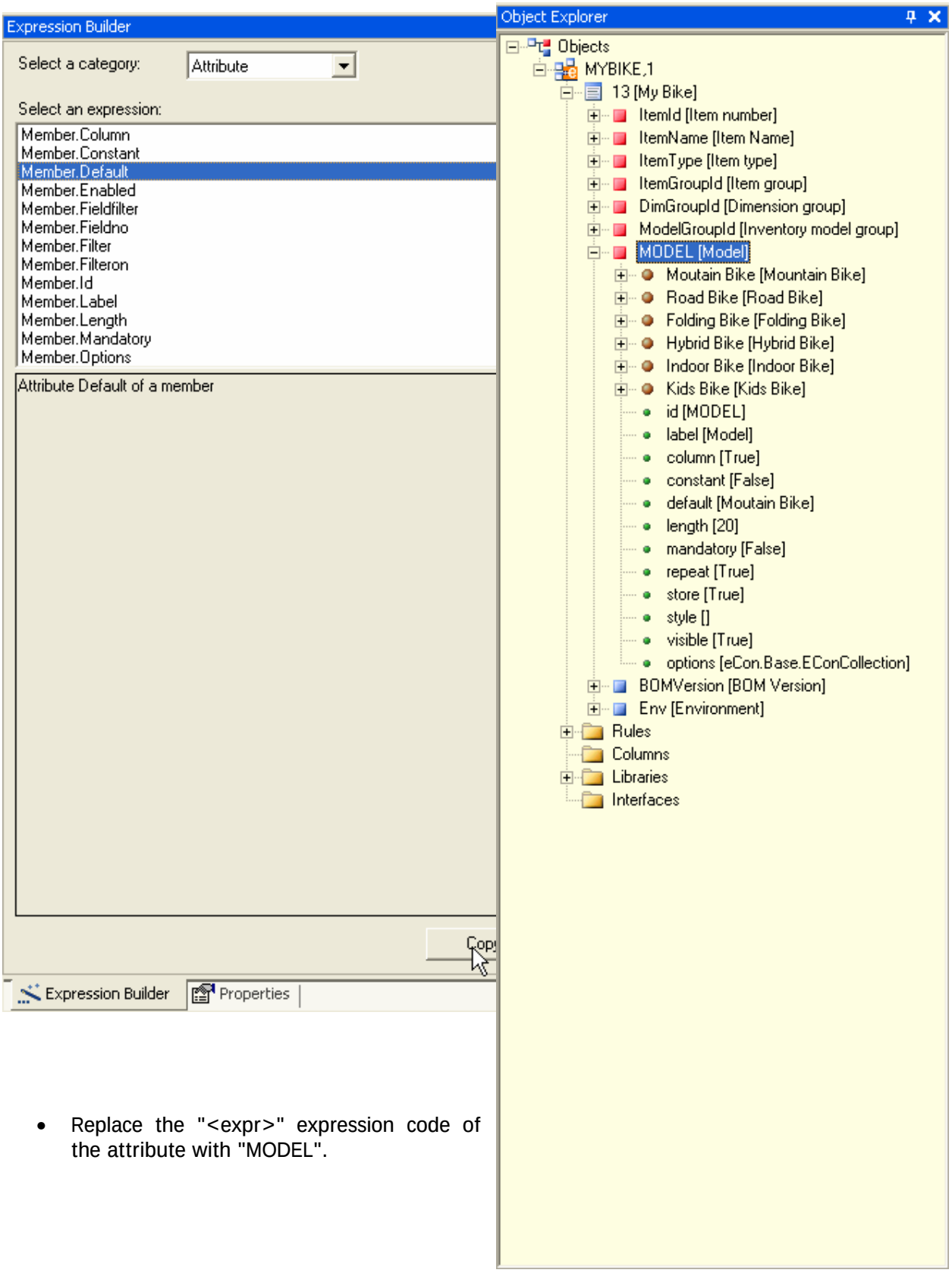


- Replace the right "<expr>" expression code with the “ItemName" member. Now the name of the BOM Version is set up to be the same as the name of the item.



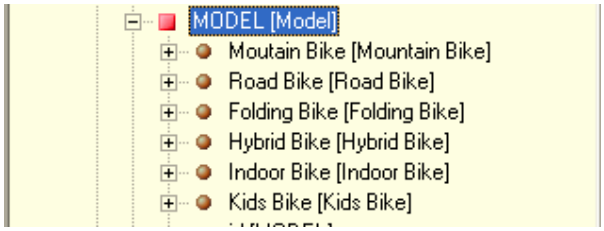
Tips & Tricks:
When you enter more than one implication, you have to separate them with a semicolon.

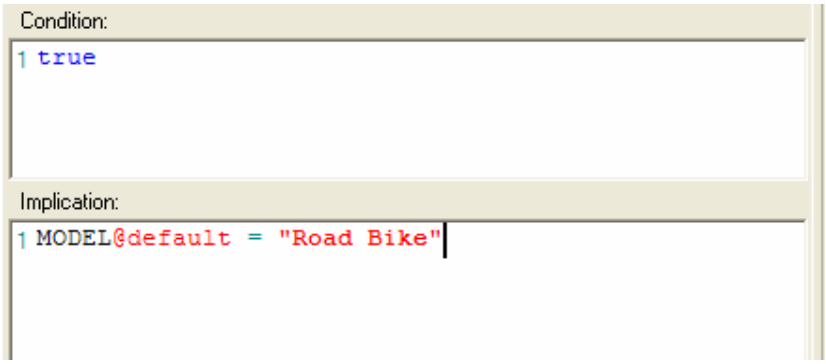
3. Now we will add more rules. In the following rule, we will set up a default value for the "Model" member.
   - Select the "Rules" folder.
   - Right mouse click and select “New Rule”. Enter an id and label for the rule, such as "Default for Model".
   - De default condition is "true" which is fine for this rule. This rule should always be executed so the condition is "true".
   - Copy the “LeftHandSide=Expression” function from the “Expression Builder”. Siply entering “=” is possible as well.
   - No we will use the “@default” attribute to assign a default value for the model field. The @default attribute can be copied from the “Expression Builder”. It’s located in the category “attributes”. Another possibility is to drag and drop the model property including the “@default” attribute from the “Object Explorer”.



- Replace the "<expr>" expression code of the attribute with "MODEL".

- Replace the right "<expr>" expression code of the rule with the "RoadBike" option .The options can also be retrieved from the data structure. Just click "MODEL" to expand the list.



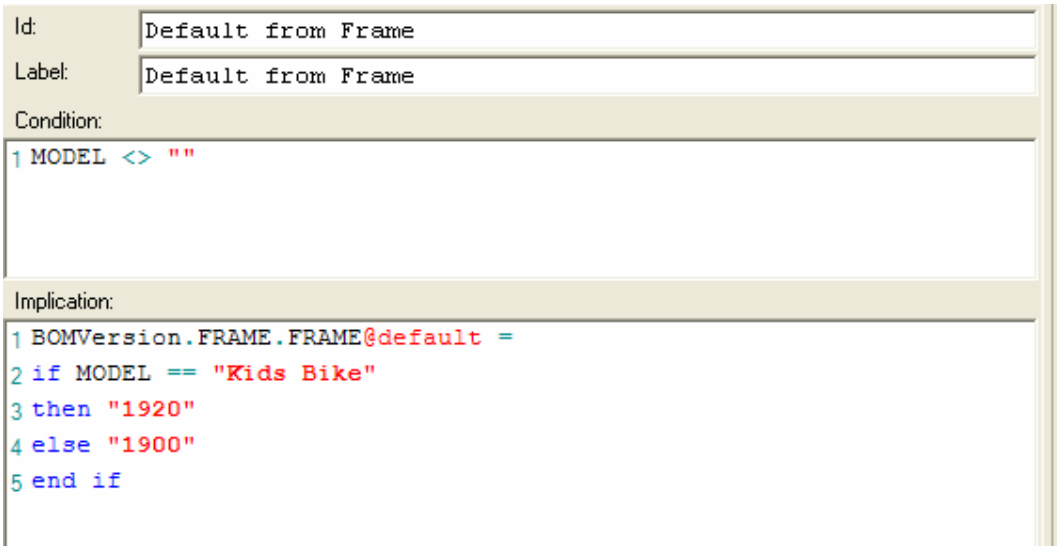


- Check the results by using the "Browse" function to view your model.

4. The rule we will add next will add dependencies between the model and the default for the frame. If the model is "Kidsbike" then the default for the frame should be "1920 Frame man's bicycle ALU" for all other models the default frame should be "1900 Frame man's bicycle Steel".
   - Add a new rule and enter the following in the "Id" and "Label" fields: "Default for Frame".
   - For the condition, enter: MODEL <> ""
   - For the implication, enter:

```
1 BOMVersion.FRAME.FRAME@default =
2 if MODEL == "Kids Bike"
3 then "1920"
4 else "1900"
5 end if
```

   - With this "if-then-else" structure, we accomplish that the default code for the frame is "1920" when the MODEL is equal to "Kidsbike", and in all other cases, the default is “1900”. The “if then else” can be copied from the “Expression Builder” and is located in the “Conditional” categorie.

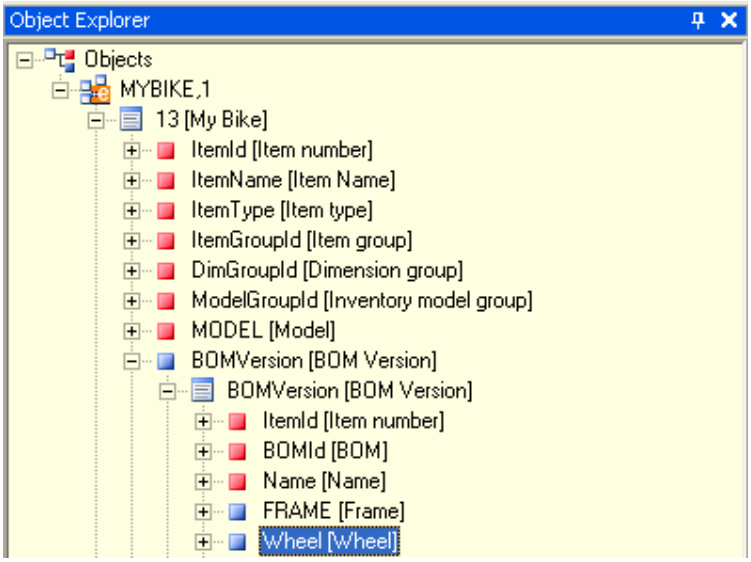


- Check the results by using the "Browse" function to view your model.

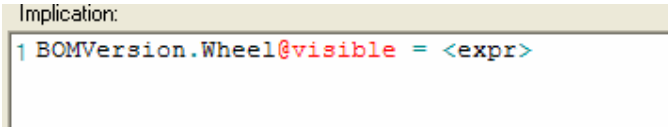
Tips & Tricks:

An " if-then-else" structure always consists of three elements: If .. then.. else... Such a structure also always ends with "end if". As much “end if” as “if” exists.

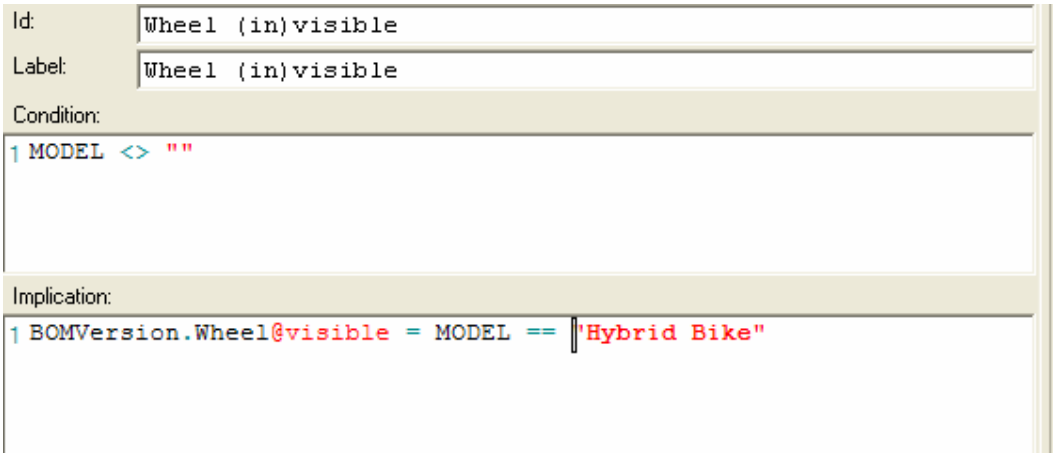
5. In the next rule, we will configure a rule so that when the model is "Hybridbike", the wheel can’t be selected. Making the wheel member invisible will do this.
   - Add a new rule and enter the following in the "Id" and "Label" fields: "Wheel (in)visible".
   - For the condition, enter: MODEL <> ""
   - Select the "Implication" field. Enter the "=" sign, and select the "@visible" attribute in the left "<expr>" expression code. Select the member that represents the relation of the BOM Version to the BOM Line of the wheel.



- Replace the "<expr>" expression code of the attribute with this member.
- For the rest of the implication, ente the @visible attribute:



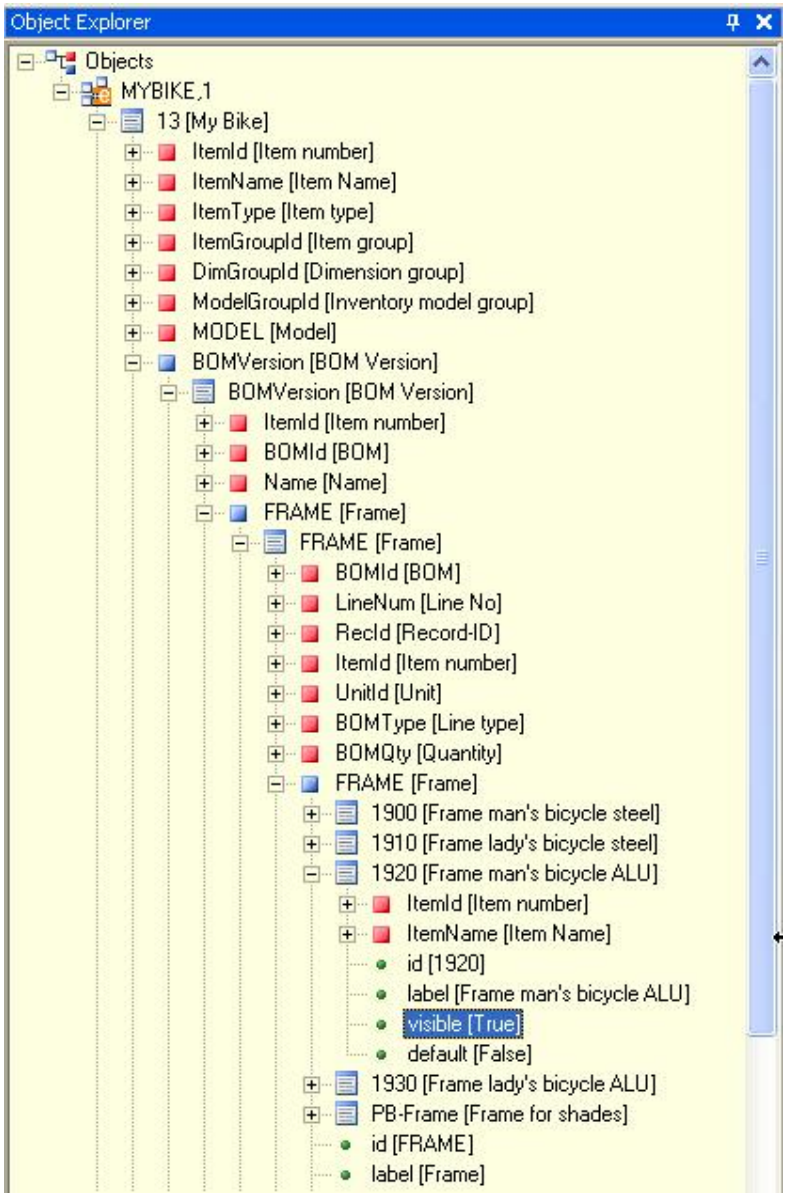
- When "True" is assigned to the "@visible" attribute, it will cause the relevant member to be visible. "False" will cause it to be invisible. The expression (MODEL <> "Hybridbike") will also result in a true or false answer, so this will work.



- Check the results by using the "Browse" function to view your model.

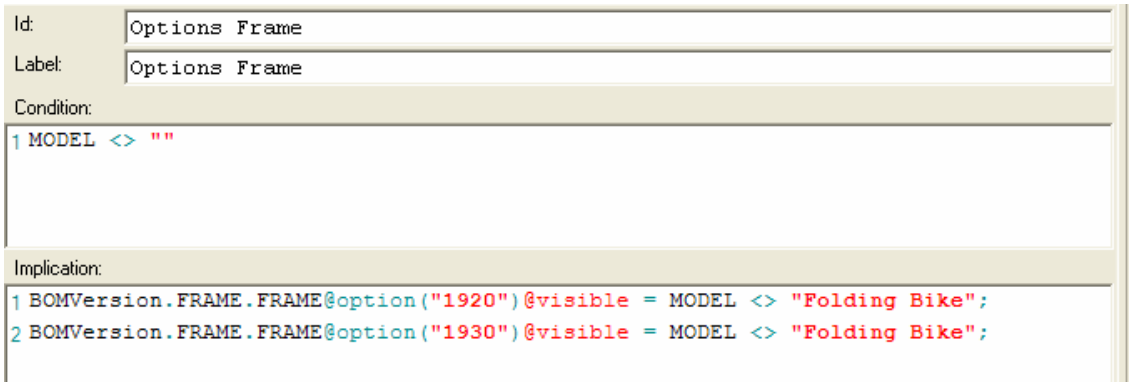
6. The last rule we will add is: If the Model is equal to "Foldingbike", it will be impossible to choose ALU frames.

- Add a new rule and enter the following in the "Id" and "Label" fields: "Options Frame".
- For the condition, enter: MODEL <> ""
- Select the "implication" field. Enter the "=" sign and select the "option.visible" attribute from the “Expression Builder” at the left site of the asssignment. This attribute is located in the “Option Attribute” category. This attribute makes it possible to make an option of a certain member (in)visible. Another possibility is to directly drag the property option including the “@visible” attribute from the “Object Exporer”.



- Enter in the right "<expr>" expression code: MODEL <> "Foldingbike"

- Add the same rule for the other ALU frame. The final result should look like this:



- Check the results by using the "Browse" function to view your model.

7. The business rules are finished, so exit the Studio and save your rules.

Chapter 4 Integrate Your Model into Your Business Process

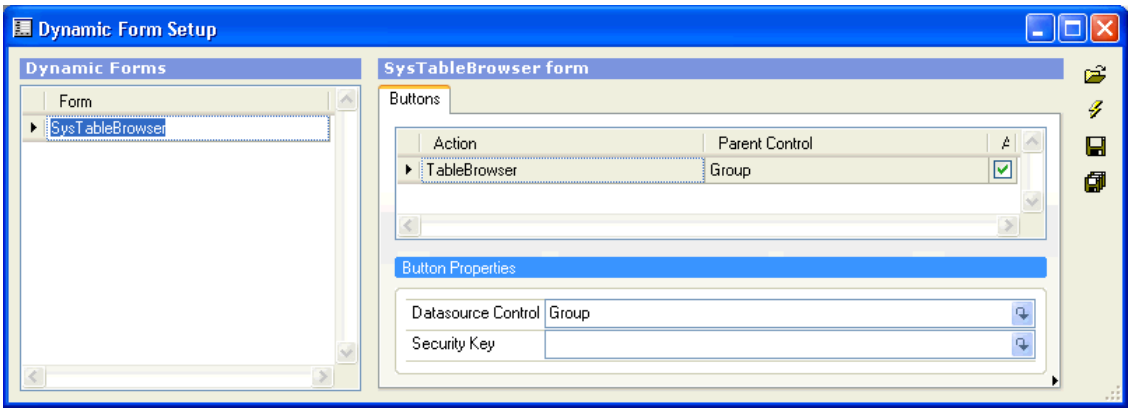
It would be very nice if the model could be better integrated into daily business. Making it possible to enter configurations starting from Axapta forms, provides a better integration. To do so, you can add buttons to Axapta forms to start the e-Con Frontend in which you can enter configurations. You can use the dynamic buttons functionality to add buttons to Axapta forms.

You are advised to only use dynamic buttons on basic data forms, like the InventTable and the CustTable. So, do not use this functionality on, for example, the SalesTable form.

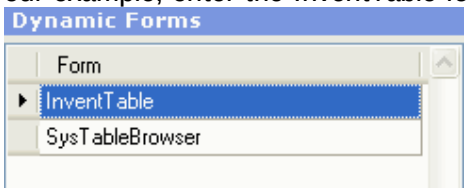
In our example, a 'Configure' and a Configure (new) button is added to the InventTable form.

To add a button to a form:

1. In the Solution Center, select Platform, Dynamic Forms, Dynamic Forms

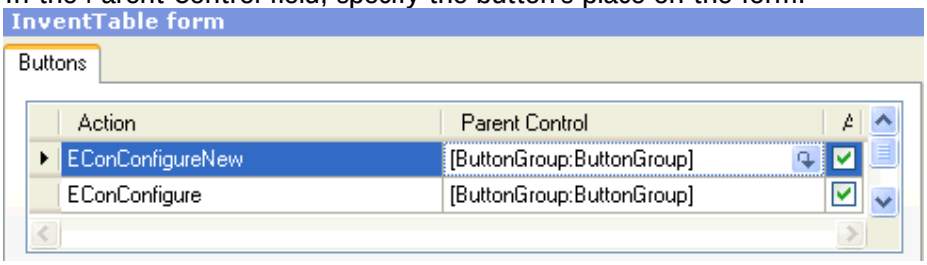


2. In the Dynamic Forms pane, enter the form to which you want to add the button. In our example, enter the InventTable form.

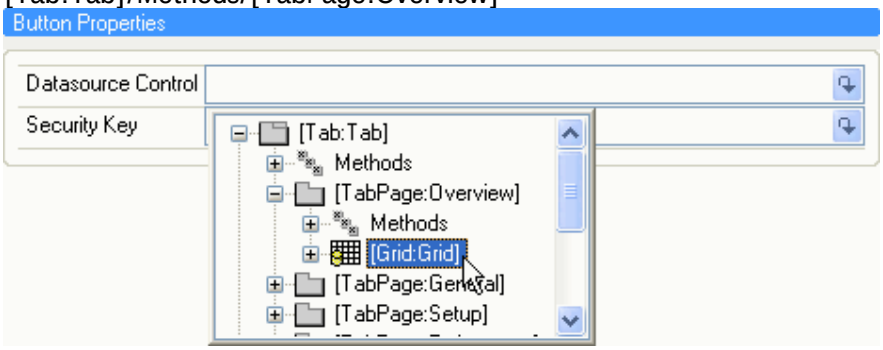


- In the upper-right pane, create a new action.
- In the Action field, enter the action that you want to add to the form. The following actions are possible:
  - EConConfigure; this action only opens existing configurations. So, if no configuration exists for an item, the e-Con Frontend is not started.
  - EConConfigureNew; this action starts the e-Con Fronten in which you can create a new configuration.

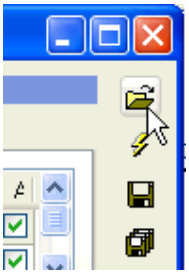
- In the Parent Control field, specify the button's place on the form.



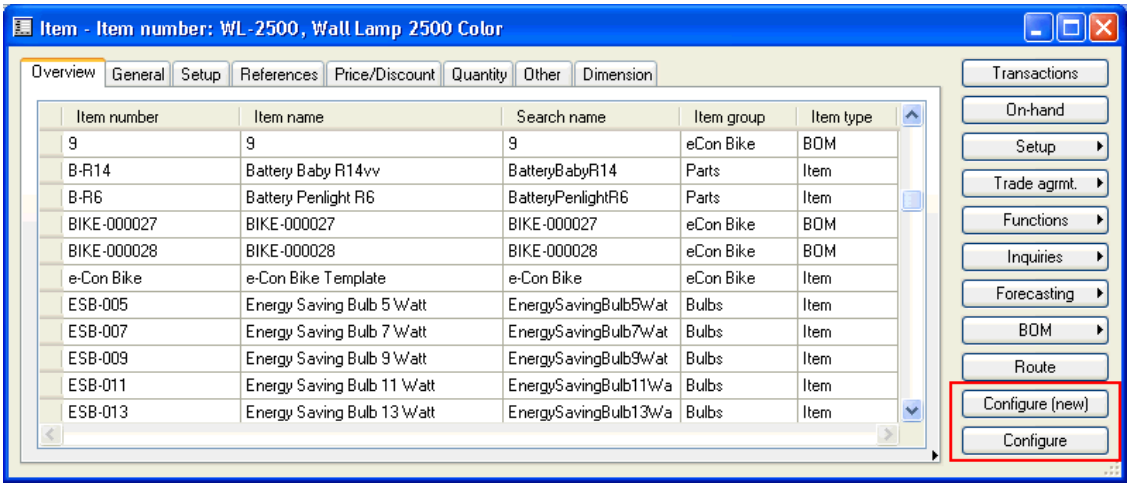
- In the Datasource Control field, enter the grid. To do so, browse on this field and select the grid. Usually, you can find the grid under [Tab:Tab]/Methods/[TabPage:Overview]



If the buttons are added to the form, you can test the buttons. To test the newly added buttons, on the Dynamic Forms form, click the 'Open form' button at the upper-right side of the form.

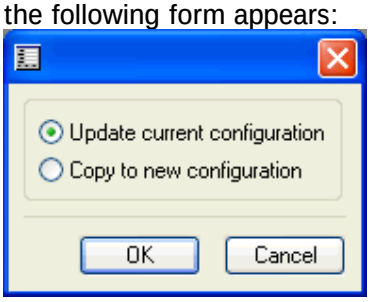


The form, to which the buttons are added, is opened. In our example, this is the InventTable (item) form.



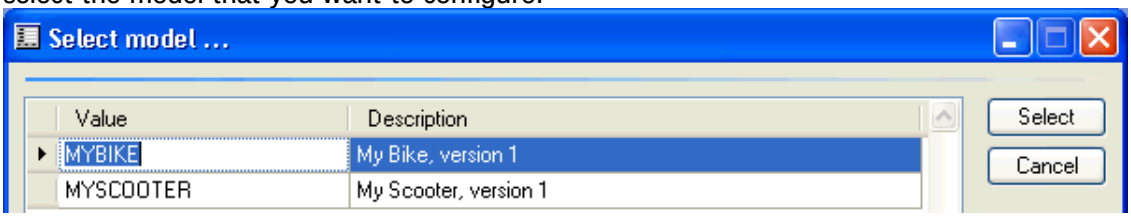
If you want to:

- Configure or copy an existing configuration, click the Configure button. If you do so, the following form appears:



Choose whether you want to update the current configurations or copy it to a new configuration.

- Enter a new configuration, click the Configure (new) button. If only one model has an active certified version with as root entity the concerned table, this model will directly started. In our example this root entity is the InventTable. If, however, several certified models with the same root entity exist, a list is shown from which you can select the model that you want to configure.



After the configuration is processed, the form from which the configuration is opened, appears, displaying the just configured record.

Chapter 5 e-Con Axapta Overview

5.1 Purpose of Axapta e-Con Module

The purpose of the e-Con Axapta connector is:

- To make the data structure of the model
- To translate the configured XML object into objects that can be imported into the Axapta database.

In fact, the e-Con Axapta Connector is the Back End of e-Con.

5.2 Tables

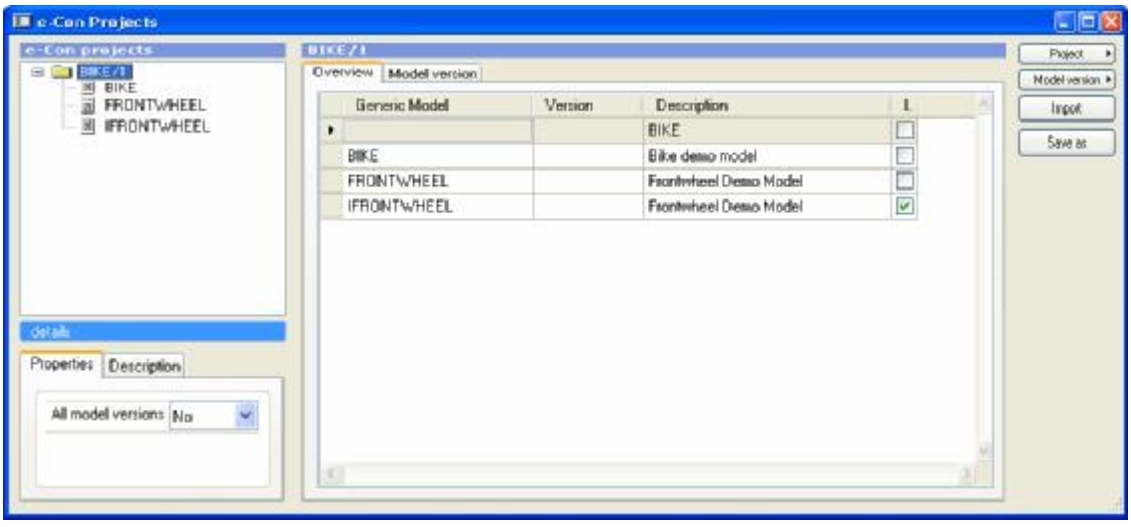
This section describes all the tables and forms that are used in the e-Con Axapta Connector.

5.2.1 e-Con Project

e-Con Projects can be used to manage your models. Models and sub models can be tight together in a project. Convenient tools are available in the projects like:

- Generate XML files from all models within the project
- Open de e-Con studio for a project to maintain the models within the project

Version is available as well.



5.2.1.1 Fields of the Overview Tab

No.

Code of the Generic Model. You can select a model to be added to the project.

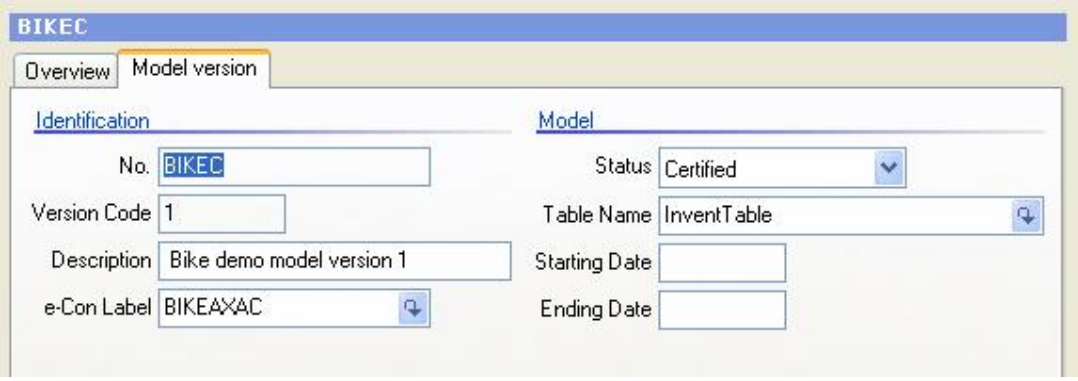
Version

The version of the Generic Model added to the project.

Description
Description of the Generic Model added to the project.

Interface Model
Indicator if the model added to the project is an interface model. Checked when the particular model is an interface model.

5.2.1.2 Fields of the Model Version Tab



No.
Code of the Generic Model selected within the project.

Version
The version of the Generic Model selected within the project.

Description
Description of the Generic Model selected within the project.

e-Con label
Label of the version of the Generic Model selected within the project.

Status
Status of the Generic Model selected within the project.

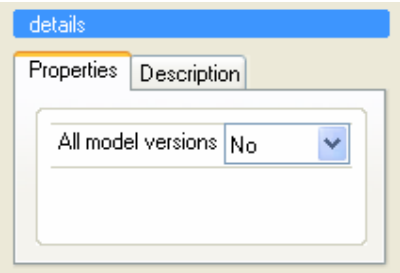
Table Name
Table Name of the main entity of the Generic Model selected within the project.

Starting Date
Starting date of the version of Generic Model selected within the project.

Ending Date

Ending date of the version of of the Generic Model selected within the project.

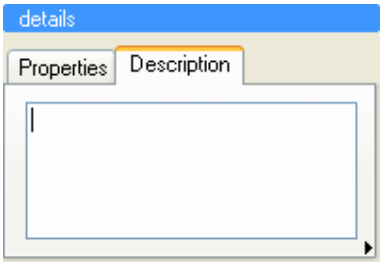
5.2.1.3 Fields of the Properties Tab (details pane)



All model versions

Two options yes and no. If no is selected automatically the active version of the selected model is added to the project. If yes is selected the model version to be added can be manual selected.

5.2.1.4 Fields of the Description Tab (details pane)



Description

A description can be added here for the model added to the project.

5.2.1.5 Project Button

Get Model Version

A list of models is shown. A model can selected here and added to the project.

Generate XML

Before the Generic Model can be used in the Front End and in the e-Con Studio the XML document and .Net assembly dll has to be created. The XML document consists of the data structure and rules that are created in the e-Con Studio. The data structure is generated. This can be done with the "Generate XML" function. When this function is selected, both the XML document and .Net assembly dll are created for all models within the project.

e-Con Studio

Start the e-Con Studio for of this Project. In the studio, amongs other, rules can be added to the models within the project. Rules can be used to add intelligence to the data structure. For example, if the type of the bike is a mountain bike, no mudguards can be chosen. After rules are added to the e-Con Studio, the rules are stored in the XML document of the Generic Model.

5.2.1.6 Model Version Button

Models

Select and opens the model view for the selected model.

Designer

Opens the e-Con model designer for the selected model. In this form the data structure of the model can easily be maintained.

Generate XML

Before the Generic Model can be used in the Front End and in the e-Con Studio the XML document and .Net assembly dll has to be created. The XML document consists of the data structure and rules that are created in the e-Con Studio. The data structure is generated. This can be done with the "Generate XML" function. When this function is selected, both the XML document and .Net assembly dll are created for all models within the project.

e-Con Studio

Start the e-Con Studio for of this Project. In the studio, amongs other, rules can be added to the models within the project. Rules can be used to add intelligence to the data structure. For example, if the type of the bike is a mountain bike, no mudguards can be chosen. After rules are added to the e-Con Studio, the rules are stored in the XML document of the Generic Model.

Included models

Shows included sub models for the selected model in a tree view. Makes only sense when sub models are used in the selected model.

5.2.1.7 Other buttons

Import

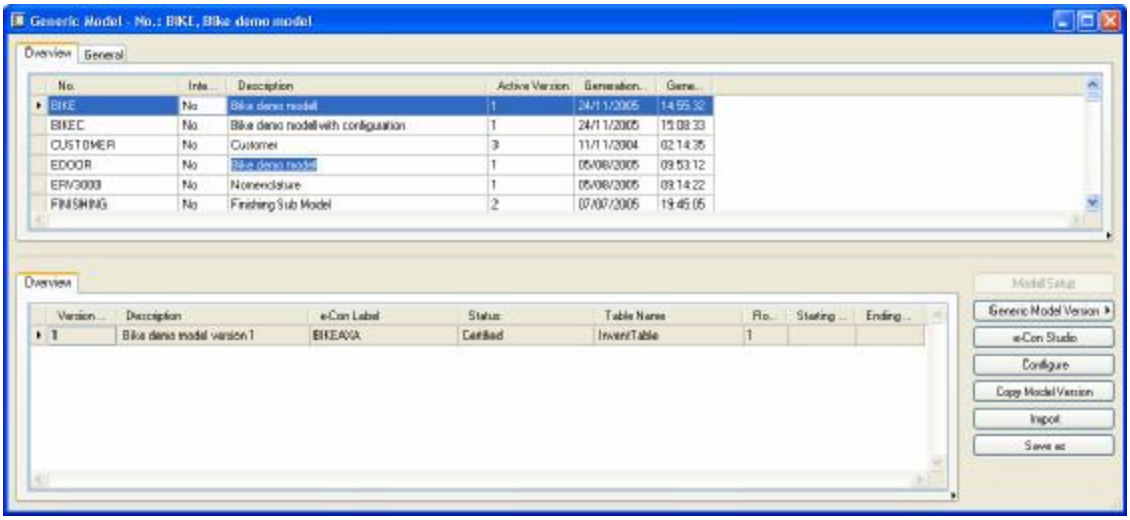
To import an project

Save as

To export an project

5.2.2 Generic Model

The highest level of the structure is the Generic Model. On the Generic Model form, the data for the generic model and version is entered, and an active version is selected. With Models in the e-Con menu the Generic Model form is started.



5.2.2.1 Fields of the Overview Tab

No.

Code of the Generic Model. You can enter a maximum of 20 characters, alphanumeric.

Interface Model

To mark a model as an interface model. Interface models are needed when sub models are used. A sub model needs an interface to be able to interface with another model.

Description

Description of the Generic Model. You can enter a maximum of 50 characters, alphanumeric.

Active Version

The active version of the Generic Model. A generic model can have different versions. For example, a bike comes in three colors. If you want to add more colors but keep the original model unchanged, then copy the model to a new version. Certify this version. Once this new version is selected as the active one in the Generic Model Card and the XML is generated, it will be used for configuration.

Generation Date

Date the last XML document was generated.

Generation Time

Time the last XML document was generated.

5.2.2.2 Fields of the General Tab

Include to Object

When "Include to Object" is check-marked, the configured object of this generic model will be stored as a separate XML file object. This child object can be reconfigured independently of its parent object. For example, if you use the BIKE demo model to configure a Bike e-Con will generate one XML object file for both generic models (BIKE and FRONTWHEEL), unless the "Include to Object" field is check-marked in the FRONTWHEEL model. In that case, two separate XML object files are created.

Online Recovery

To activate the “Online Recovery” for this particular model. With “Online Recovery” the user has the possibility to recover erroneous configuration. If the processing of a configuration fails the next time the user starts the e-Con model, the user is asked if he want to reopen the erroneous configuration. After reopening the configuration a reprocess can take place.

Online Synchronizing

The "online synchronizing" field has to do with the reconfiguration of an already configured object. When you reconfigure an object it will be synchronized with the active version of the base model. There are three ways of synchronizing:

- Manual: When the flag 'Synchronize' of the configuration is check marked the configuration will be synchronized.
- On Confirmation: You will be asked if you want to synchronize, no matter if the flag ‘synchronize’ is check marked.
- Automatically: The object is always synchronized, no matter if the flag ‘synchronize’ is check marked.

For example:

On 23/03/01, a customer orders a series of 100 customer-specific. With the help of e-Con, a customer-specific Item with its customer-specific Bill of Material and Routing will be generated. On the 12th of April the same year, the customer again orders a series of 100 bicycles which are identical to the series ordered a few weeks before, except for the color. This time the customer wants a special color. So a search for the item which had been generated a few weeks ago is done, and based on this item, a new item will be configured. If the "Online Synchronizing" field is check-marked, the already configured object will be merged with the basis model (the configuration tool model "BIKENM.xml") before the user can reconfigure the object.

When the model is merged, all the data from the object that was already configured is imported to the newest version of "BIKENM.xml" and saved. Now all changes made to the base model are available in the reconfigured object.

Infolog detail

A setting to determine the level of detail to be shown in the Axpata infolog after processing this model into Axapta.

There are four possibilities:

- All: All the info is shown in the infolog;
- Errors and warnings: Only errors and warnings are shown in the infolog;
- Errors only: Only errors are shown in the infolog;
- None: Nothing is shown in the infolog.

Limit

The maximum number of messages to be shown in the infolog.

5.2.3 Overview on version

Version Code

Code of the version. You can enter a maximum of 20 characters, alphanumeric.

Description

Description of the version. You can enter a maximum of 50 characters, alphanumeric.

e-Con Label

This label is used as name of this Generic Model in the XML document. Choose a short description without spaces.

Status

Status of the active version. There are four different statuses:

New: Whenever a new version is inserted, the default status is "New". A new version can be inserted by selecting the "Copy Model" option or by inserting a new line in the Version form. When the status is "New", the data structure can be changed. When a version has this status, it cannot be used for configuration. Change the status to "Certified" when you need it for configuration. Once the status of a version has changed from "New" to "Under Development", "Certified" or "Closed", the Status can't be changed to "New".

Under Development: This status is used to modify the data structure. When you want to change the data structure of a version, this status has to be chosen. Change the status to "Certified" when you want to use the version for maintenance of rules or for use in the Front End.

Certified: Only when the status is "Certified" can it be used for the Front End or for maintenance of rules. This means when the Key Value of the Root Entity is selected in the Sales Order, the Front End is started, but only if the status is "Certified". When the status is changed to "Certified", a number of checks are performed to ensure the data structure is correct:

If the Member Type is "Relation", has the link (if needed) been made?

Do the relations defined refer to existing Entities?

If the Member Type is "Field Property" is a "Table" field selected?

Etc.

Closed: This status is selected when a version has to be blocked for use. Use this status for models without any data or if certain options are no longer available.

Table Name

The main Entity to which this Generic Model is linked. The rest of the Entities are linked/refer to this Entity. For example, when a new generic item is created, with Bom version, Routing and Bills of Materials (BOM), the Root Entity will be the InventTable, Item. Once configuration is complete, a new item is inserted in the Item table.

Root Line No.

The line number of the main Entity to which this Generic Model is linked. The rest of the Entities are linked/refer to this Entity. It is possible to have a more then one entity for a table. The Line No. will make every entity unique.

Starting Date

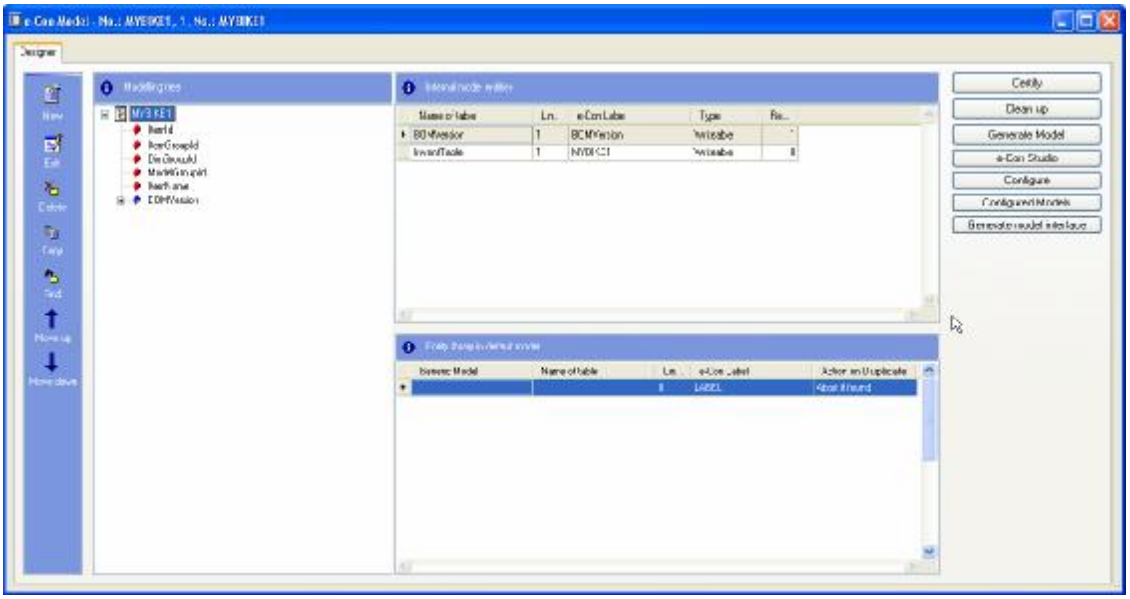
The starting date of the period in which the version is valid.

Ending Date

The ending date of the period in which the version is valid. A version can only be used for configuration while it is valid.

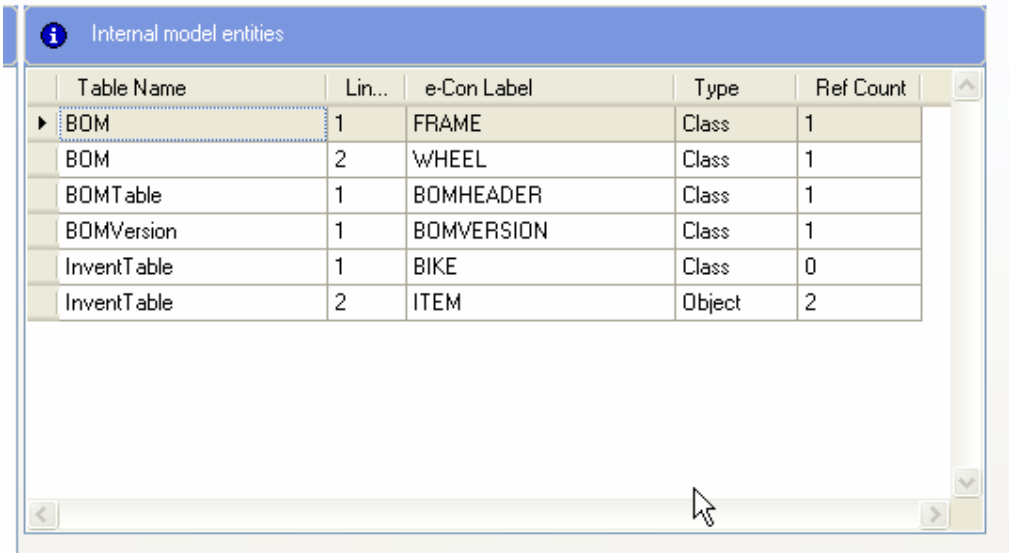
5.2.4 Model Setup Button on Generic Model form (only available when the version is “under development”)

Opens the e-Con Model form.



In this form the data structure of the model can easily be maintained. It is different from the method used in chapter 4.2.

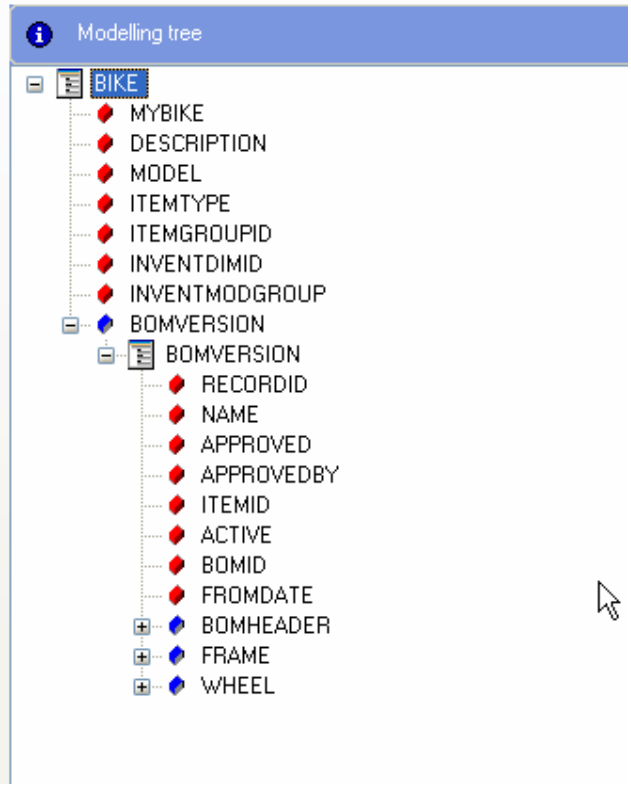
Adding end maintaining entities



In this area the entities of the model are shown.

With Ctrl-N new records (Entities) can be added to the data structure, see 5.2.13.

When selecting an Entity in this part of the form with the “New” button new entity members can be added to the entity. With the “Edit” button the entity can be maintained, see 5.2.13.



Parts of the data structure can be expanded to select the right Entity or Entity member.

Selecting an Entity (e.g. BIKE) with “New” Entity members can be added and with “Edit” the Entity can be changed, see 5.2.15.

Selecting an Entity member (e.g. MODEL) with “Edit” the Entity member can be changed, see 5.2.15.

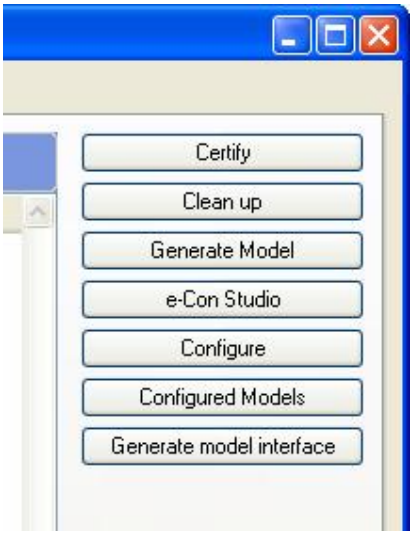
Selecting an Entity or Entity member with “Copy” the Entity or Entity member is copied. The new id of the Entity or Entity member will be “Copy of …”.

With the “Find” can be searched for an Entity or Entity Member within the data structure.

Selecting an Entity or Entity member with Delete it will be deleted from the data structure.

Selecting an Entity or Entity member with Ctrl ↑ or Ctrl ↓ the sequence in the data structure and therefore in the Front End can be changed.

5.2.5 e-Con Model form Buttons



Certify
The data structure is certified. The status of the version becomes certified.

Clean Up

The model is checked for unused entities. These entities are available in the model entities but not used in the tree. After the check e-Con is asking for a confirmation to delete the unused entities from the model.

Generate XML

Before the Generic Model can be used in the Front End and in the e-Con Studio the XML document and .Net assembly dll has to be created. The XML document consists of the data structure and rules that are created in the e-Con Studio. The data structure is generated. This can be done with the "Generate XML" function. When this function is selected, both the XML document and .Net assembly dll are created.

e-Con Studio

Start the e-Con Studio for the active version of this Generic Model. Rules can be used to add intelligence to the data structure. For example, if the type of the bike is a mountain bike, no mudguards can be chosen. After rules are added to the e-Con Studio, the rules are stored in the XML document of the Generic Model.

Configure

Start the Front End for the active version of this Generic Model. Use this function to view the layout of the Model in the Front End, or to test the model. Once the Front End is saved or "Cancel" clicked, Axapta returns to the Generic Model Card. However, if the Front End is started from the sales order line, the system returns to the sales order line.

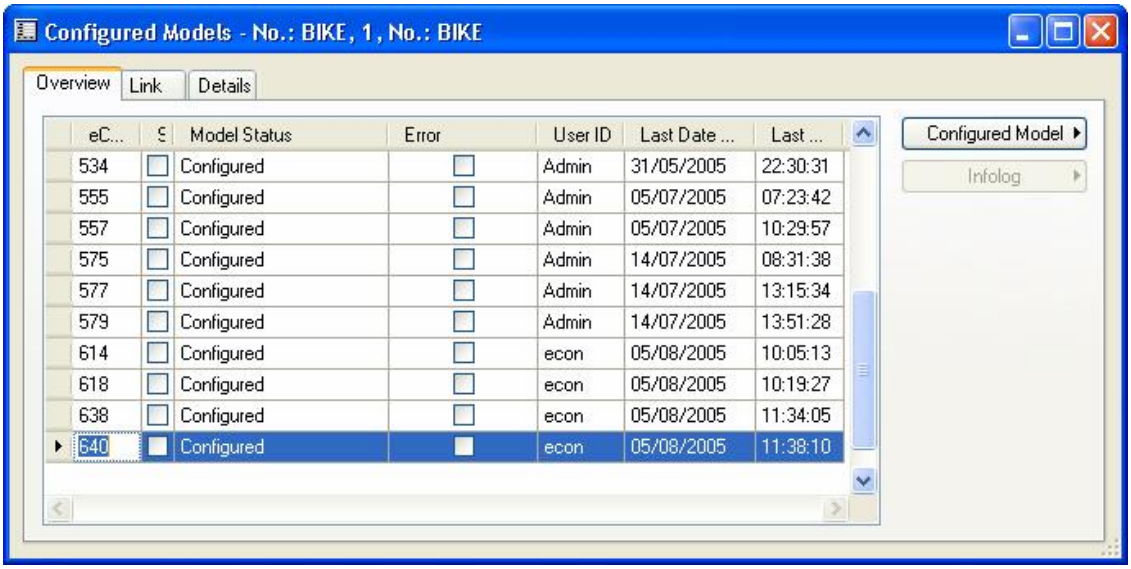
Configured Models

This form displays the objects that have been configured from the Front End. Once the "Save" button is clicked in the Front End, this list will be updated.

Generate Model Interface

This function will generate an interface model. Interfaces are used for sub models and needs to be created in order to interface with other models. All members with the “To interface” field checked will be added to the interface model when the “Generate Model Interface” function is executed. The user will be asked for the interface model id, version and label.

5.2.6 Configured Models



eC...	S	Model Status	Error	User ID	Last Date ...	Last ...
534	☐	Configured	☐	Admin	31/05/2005	22:30:31
555	☐	Configured	☐	Admin	05/07/2005	07:23:42
557	☐	Configured	☐	Admin	05/07/2005	10:29:57
575	☐	Configured	☐	Admin	14/07/2005	08:31:38
577	☐	Configured	☐	Admin	14/07/2005	13:15:34
579	☐	Configured	☐	Admin	14/07/2005	13:51:28
614	☐	Configured	☐	econ	05/08/2005	10:05:13
618	☐	Configured	☐	econ	05/08/2005	10:19:27
638	☐	Configured	☐	econ	05/08/2005	11:34:05
640	☐	Configured	☐	econ	05/08/2005	11:38:10

5.2.6.1 Fields of the Overview Tab

e-Con Model ID

e-Con will automatically create this value. It is a serial number.

Synchronize

If set to Yes before Configure the model will be synchronized during reconfiguration.

Model Status

The status of the model: new, configured, etc.

Error

This field contains an error code if there’s a problem with a configuration.

User ID

User ID of the user who configured the object.

Last Date Modified

The last processing date of the configured object.

Last Time Modified

The last processing time of the configured object.

5.2.6.2 Fields of the Link Tab

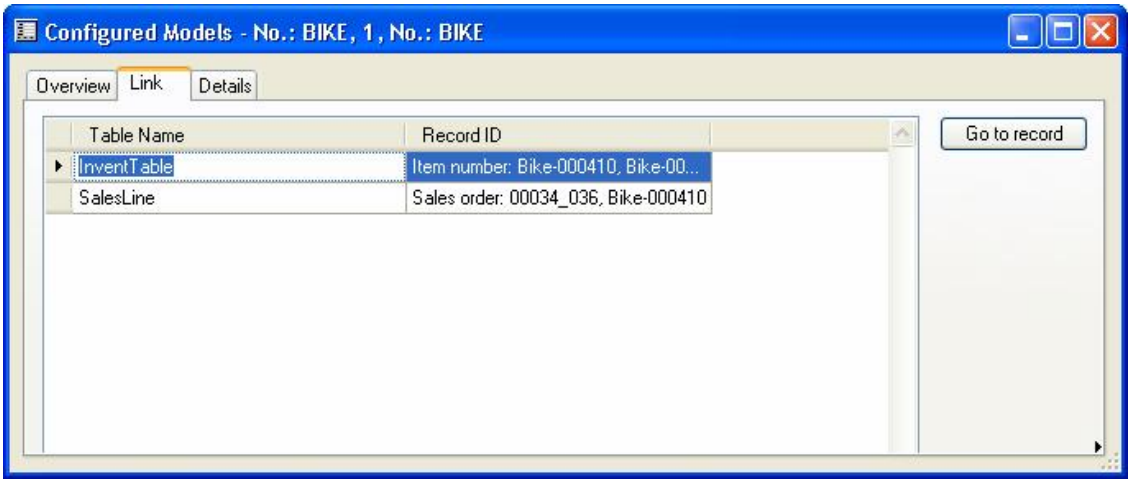


Table Name

The table linked to the configured model. This table is populated by this XML configuration

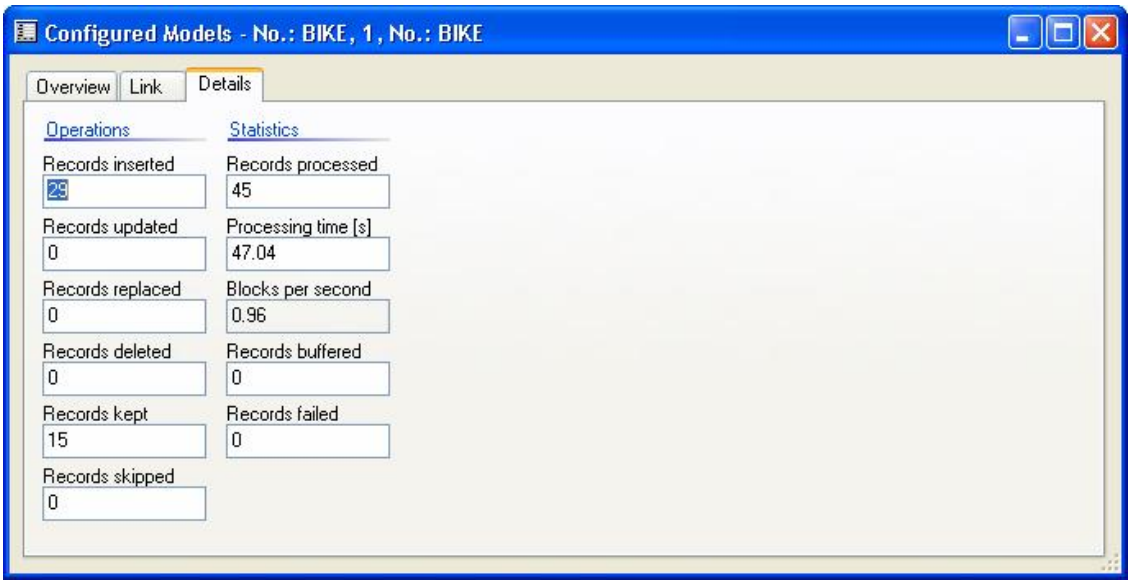
Record Id

The link to the actual record of the table

Button "Go to record'.

Clicking this button will jump to the table and record selected

5.2.6.3 Fields of the Details Tab



Information about number of records inserted, updated, delete etc. are displayed here. Performance indicators like processing time is here available as well.

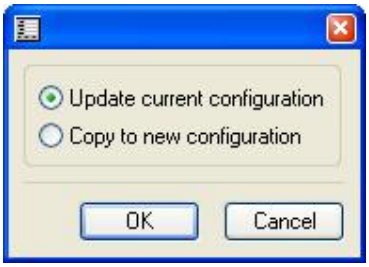
5.2.7 Configured Model Buttons

Process

An erroneous configuration can be reprocessed. Only enabled when the error flag is set for the particular configuration.

Configure

This activates the "Configure" option for the selected model.



Key values

Shows the Key Value(s) of the configured object.

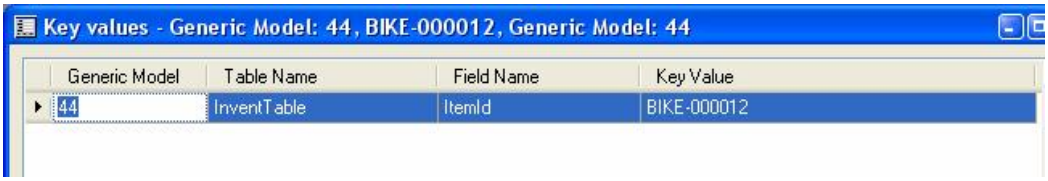
Included Object

Displays the included objects of this configured object. Inlcuded objects does only exist when sub models are used in the configuration model and when the flag 'include to objects' is set for the particular sub model. If so, a separate configured object (xml file) is created for the sub model part of the configuration.

Report

To print an e-Con report (both html or MS Word) from the selected configurations. Shows a list of reports available for the model. After selecting the report the document is printed.

5.2.8 Key Values



Generic Model

E-Con will automatically create this value. It is a serial number.

Table Name

The table name where the configured object is stored.

Field Name

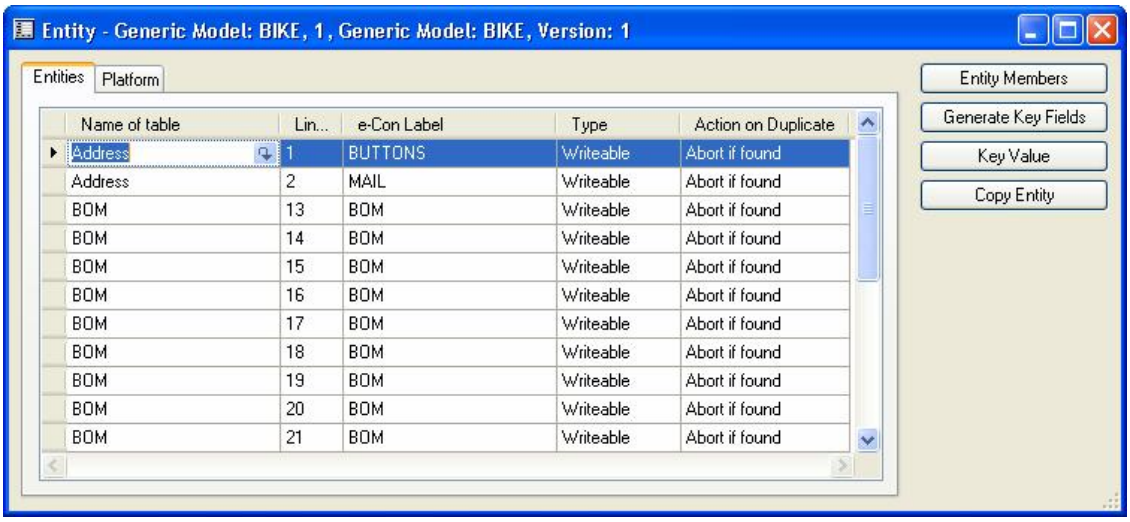
Field name of the key field that is used.

Key Value

Key Value of the configured object.

5.2.9 Entities button on Generic Model form (only available when the version is "under development")

Opens the Entity form



Once configuration is complete, records are inserted into the Axapta database. The definition of a record is determined by the Entity definitions.

In the Back End, the Generic Model is the name for the e-Con model. The model itself consists of a structure with properties, options, relations and objects. The structure of the model is built with Entities.

The key of an Entity consists of a Table Name and a Line No.

Tips & Tricks:

When the default model is entered in the e-Con setup, the Label and Members (Entity lines) of an Entity are copied from the default model. When an Entity of a table is inserted for the second time in the same model (for example a Bill of Material line), the Entity is no longer copied from the "Default model", but from the same Entity above.

Table Name

Name of the Axapta Table. You can use the lookup button to select the right table.

Line No.

Line No. is present because a table can be used on multiple levels in the model. This number makes the Entity unique. Only numbers can be used. Use no. 1 for the first Entity of a certain table, 2 for the second, etc.

e-Con Label

Label for the Entity that is used in the XML document. The code of the label will be used in the rules of the e-Con Studio, the description of the label will be used in the Front End.

Type

Writeable (generic part of model (generic item) which has to be written into the database Read only (lowest level of model, which is linked to a table record) which only has to read data from the database.

Action on Duplicate

The action to be taken if the generation process of the Axapta Back End finds a duplicate Key Value when inserting the record into Axapta.

A duplicate Key Value can only occur if the Key Value(s) are specified in the e-Con Front End and no "No. Series" are used in the Entity members.

The following actions are available:

Stop Process: This is the default value. If there is a duplicate value, the process is stopped and Axapta generates an error message.

Use Database Record: The generation process will use the record found in the database; any data entered in the e-Con Front End is ignored for this Entity.

Modify Database Record: The record is used and the data entered in the Front End is used to modify the record.

If the Key Value of the Root Entity is entered in the e-Con Front End or generated by the rules, and a "No. Series" is used, then both values (Front End value and "No. Series" value) are combined into one single value.

5.2.10 Entity form Buttons

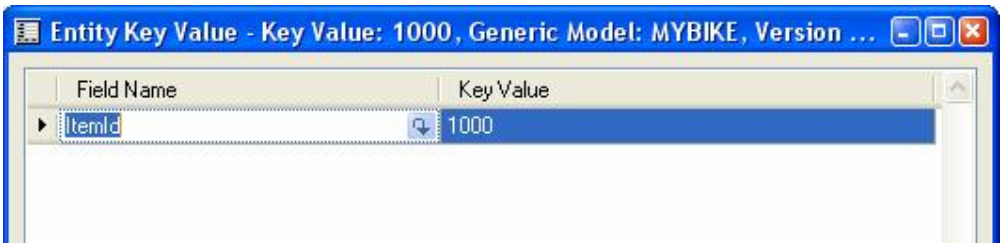
Entity Members

Opens the Entity Member form. The members corresponding with the Entity can be entered in this form. See the section on Entity Member form.

Generate Key Fields

The "Generate Key Fields" function must be used once an Entity has been entered. It will generate the key fields as the first members of the entity, which are necessary.

Key Value



The key field names and key values, if filled, are shown.

About Key Values:

- A Key Value can be added to each Entity and is a pre-existing record of the table that is represented by the Entity. This Key Value will act as a kind of template. All the fields that are not entered by e-Con, fields that are not represented by a field property in the Entity, will be entered with the value of the relevant fields from the Key Value.
- The Key Value that is added to the Root Entity has another special function, too. When this Key Value is entered in the Sales Order line, the appropriate e-Con Model will be called up automatically when the “Configure” button is activated.

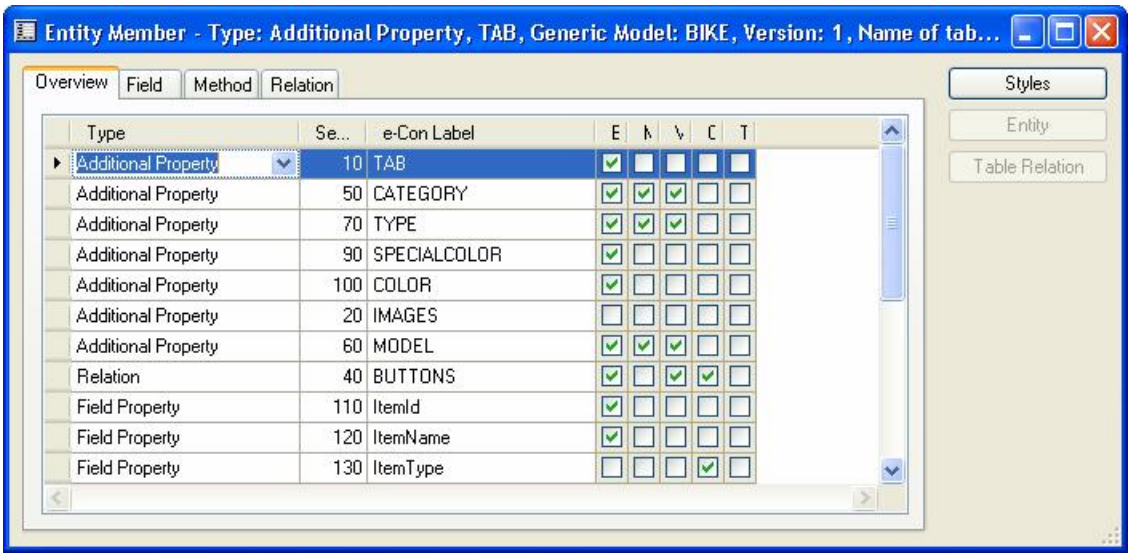
See also chapter 4.4.

Copy entity

When using this function the selected Entity will be copied with all the members and key values. Only the e-Con Label will not be filled, because it has to be unique.

5.2.11 Entity Member form

Once the Generic Model and the Entities are filled in, the details of each Entity have to be entered. When used in reference to a data model, the "members" of an Entity are the same as fields in a table.



5.2.12 Fields on Overview tab

Member Type

There are four options:

Field Property: link to a table field

Additional Property: field that does not exist in the Entity table, but is used in the Front End, and/or in the e-Con Studio.

Relation: relation to another Entity

Display method link to a table display method foeld

Sequence

Determines the shown sequence in the Front end during configuration. The step size in Setup Parameters in used to determine the steps.

e-Con Label

Label of the Entity Member. Using this label, you can see which Entity member you are working in the e-Con Studio. The code of the label is used for reference in the rules, the description is used in the Front End. It makes the Entity Member unique in the XML data structure file.

Editable

Checkbox for whether this member is editable in the Front End.

Mandatory

Checkbox for whether this member is mandatory in the Front End.

Visible

Checkbox for whether this member is visible in the Front End.

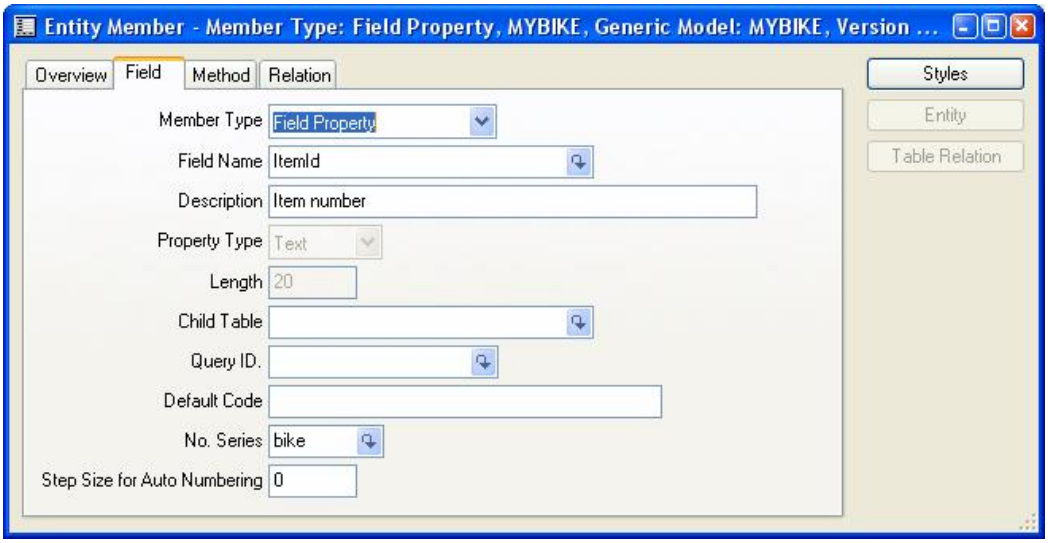
Constant

Checkbox for whether this member is constant in the Front End. If the member type is "Relation", and there only is one option, the level is skipped in the Front End.

To Interace

Checkbox for whether this member must be part of the interface. Interfaces are used for sub models and needs to be created in order to interface with other models. All members with the “To interface” field checked will be added to the interface model when the “Generate Model Interface” function is executed.

5.2.13 Fields on Field tab



Member Type

There are four options:

Field Property: link to a table field

Additional Property: field that does not exist in the Entity table, but is used in the Front End, and/or in the e-Con Studio.

Relation: relation to another Entity

Method name link to a table display method field

Field Name

This is only entered if the member type is "Field Property". Here, the field of the table is selected.

Description

Description of the member. In the case of a field property, the description entered is the description of the field chosen as the Field No. If the member type is "Additional Property" or "Relation", the description can be entered for information purposes. It will not be used in the XML document.

Property Type

The Property Type has to be entered if the member type is "Additional Property". The following options are available:

Text:	For text fields, like color
Integer:	For numeric fields, like length
Decimal:	For prices, like Special price
Boolean:	For true/false checkboxes
Date:	For data fields
Time:	For time fields

Length

Length of Property Type (has to be defined only if the member type is "Additional Property"). Choose normal values like 12 for decimal and 30 for text.

Child Table

If the member type is "Field" or "Additional Property":

When the Child Model is left blank and the child table is filled, a direct relation to a table is created. Using a query, records can be selected from the chosen table.

Query ID.

ID of the Query used. The usage of the query depends on the selected member type:

Field property:	Selection of the records chosen in the Child table.
Add. Property:	A query can be made in table no. 0. This means a set of options can be filled in.

Default Code

The default option code can be specified in the "Default Code" field for field properties and additional properties.

No. Series

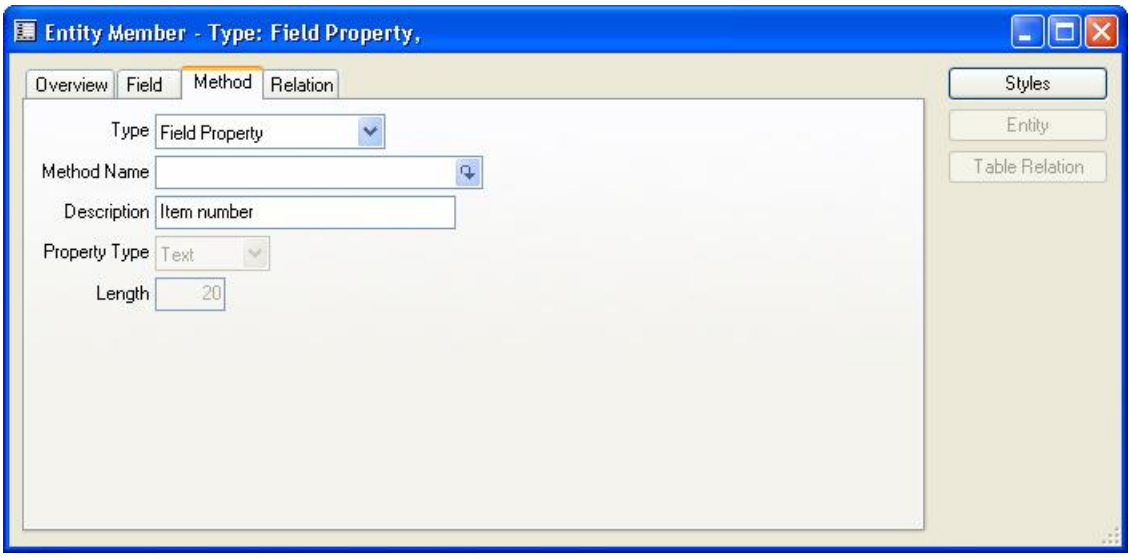
When an object is inserted in an Axapta table after configuration, the code is retrieved from this No. Series. When no number series is entered, the code of the record that is inserted into Axapta, such as "Item", has to be generated by the rules. When both the number is generated from the rules, and a number series is entered, the two fields are

combined. For example, if "H0001" is the number created by the number series and "MENSBIKE" is created by the rules, the item number will be MENSBIKEH0001.

Step Size for Auto-numbering

When Auto-numbering as part of the primary key is needed the step size can be filled in here. The step size will be added when the next record is added to the database.

5.2.14 Fields on Method tab



Member Type

There are four options:

Field Property: link to a table field

Additional Property: field that does not exist in the Entity table, but is used in the Front End, and/or in the e-Con Studio.

Relation: relation to another Entity

Method name link to a table display method field

Method name

This is only entered if the member type is "Display Method". Here, the display method of the table is selected.

Description

Description of the member. In the case of a display method.

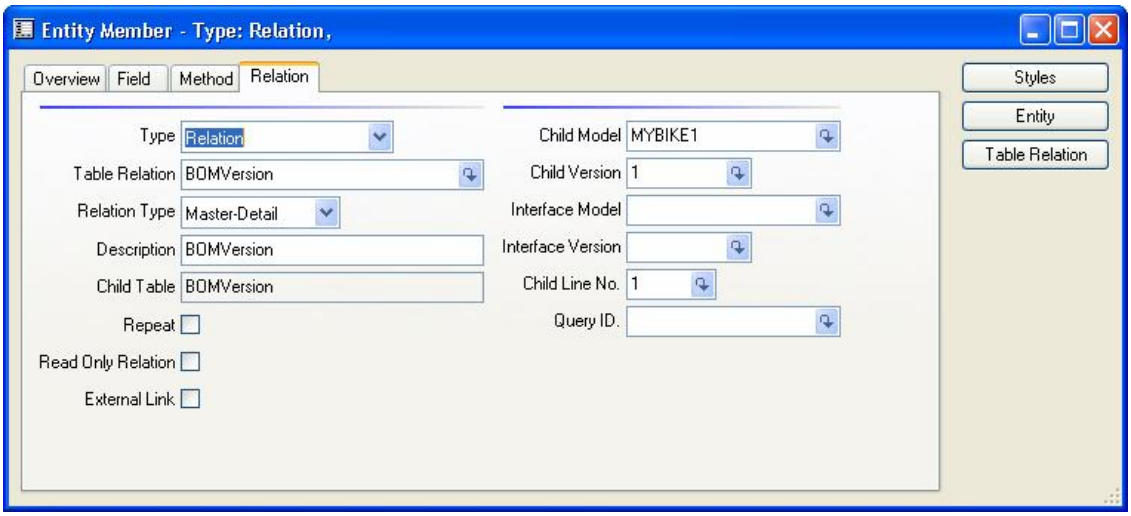
Property Type

Type of the display method, not editable.

Length

Length of Property Type (has to be defined only if the member type is "Additional Property"). Choose normal values like 12 for decimal and 30 for text.

5.2.15 Fields on Relation tab



Member Type

There are four options:

Field Property: link to a table field

Additional Property: field that does not exist in the Entity table, but is used in the Front End, and/or in the e-Con Studio.

Relation: relation to another Entity

Method link to a table display method table field

Table Relation

This is the table relation from the platform used in this relation. Table relation information is required during the processing of the XML document.

Relation Type

The Relation Type is derived from the table relation. This field is needed to insert the records in Axapta in the right order. There are two options:

Reference:	Reference between two Entities. Example: "item" and "BOM version" - both records can exist without each other.
Master/Detail:	1 to N relation between Entities (1 to N), such as "BOM version" and “BOM line"; first the BOM version exists, and then the BOM line is linked to the BOM version. The primary key of the detail is an extension of the key of the master.

Description

Description of the member. In the case of a field property, the description entered is the description of the field chosen as the Field No. If the member type is "Additional Property" or "Relation", the description can be entered for information purposes. It will not be used in the XML document.

Child Table & Child Line No.

If the member type is "Relation":

These two fields combined refer to an Entity. If the Child Model differs from the current model, the Root Entity will be selected.

If the member type is "Field" or "Additional Property":

When the Child Gen. Model is left blank and the child table is filled, a direct relation to a table is created.

Repeat

Checkbox for whether this member should be repeated in the Front End. It can only be marked if the member type is "Relation" and the relation type is "Master/Detail". If the new reference is mandatory and there is only one visible option available, then the reference will be expanded automatically, once the "New" button is clicked.

Read Only Relation

The related entity will only be used to deliver data to the model. No record based on the related entity will be inserted in the database after configuration.

External Link

The relation is not referring to a child model and child table in Axapta. The relation is used to refer to an external model. (stand alone model). Only an empty reference is created during the generation of the XML document. Details, like the interface and sub model, must be added in the e-Con studio. With the external link ‘stand alone’ e-Con models can be integrated in ‘Axapta’ e-Con models.

Child Model

Can only be filled in if member type is "Relation". Here you can use the lookup button to select the Generic Model to which the relation refers. When another model than the

current one is selected, this will result in inclusion of that model. Now the Root Entity is automatically chosen as the Child Table and Child Entity.

Child Version

Active version of the selected Generic Model.

Interface Model

The interface model to be used for the relation. Interfaces are only required when the relation refers to a sub model (child models differs from the current model).

Interface Version

The interface model version to be used.

Query ID

ID of the Query used. The usage of the query depends on the selected member type.

Field property: Selection of the records chosen in the Child table.

Additional Property: A query can be made in table no. 0. This means a set of options can be filled in.

Relation to an Object: A filter can be created for an Axapta table

Relation to a Class: Using a filter, a reference to another Entity is created.

Tips & Tricks: The primary key fields of an Entity have to be defined as "Field Property" in the Entity lines (Entity members). These field properties are created automatically when the "Generate Key Fields" function is run in the Entities form.

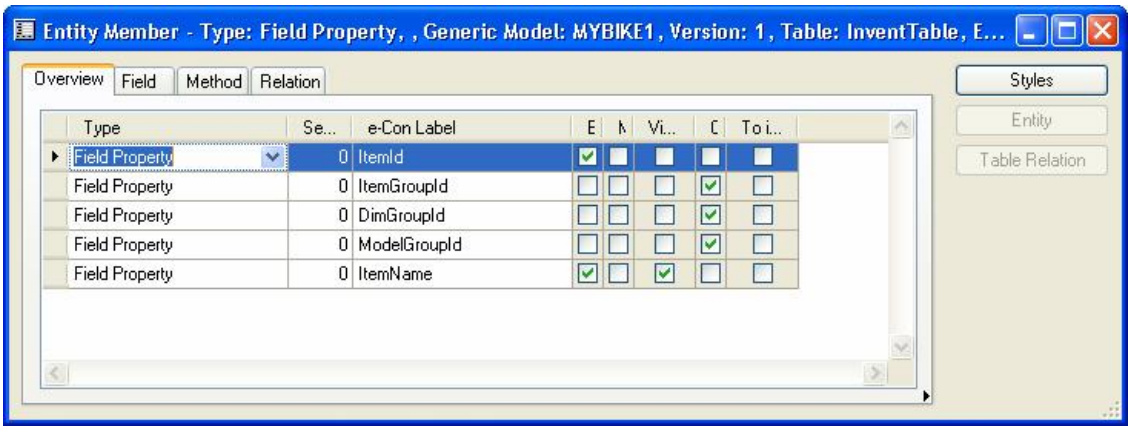
There are quite a few fields to be defined for each member on the Entity Card. The chart below shows which fields are mandatory and which are optional, according to member type.

- ˜ Mandatory
- º Optional
- • Blank, or automatically filled in by the system

Member type	Field Property	Additional Property	Relation
Label	˜	˜	˜
Sequence	º	º	º
Field No.	˜	•	•
Description	º	º	º
Property type	•	˜	•
Length	•	˜	•
Relation type	•	•	˜

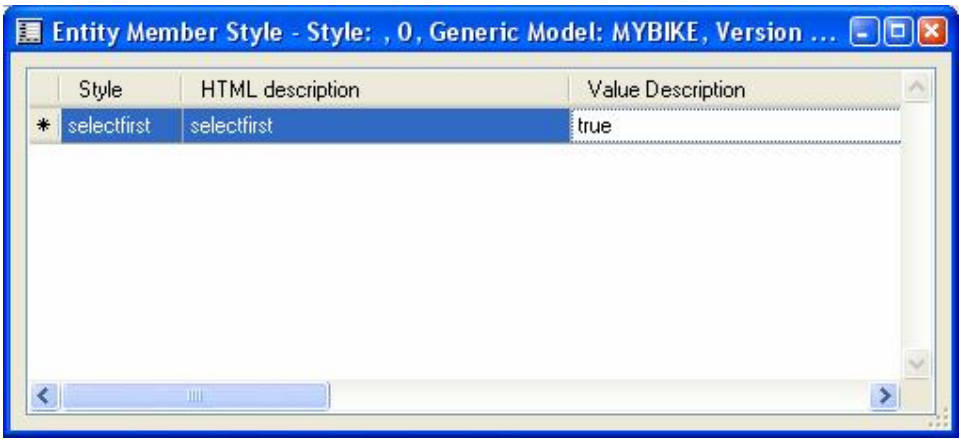
Repeat	•	•	o
Editable	o	o	o
Mandatory	o	o	o
Visible	o	o	o
Constant	o	o	o
Child model	•	•	~
Child version	•	•	~
Child table	o	o	~
Child ent. no	•	•	~
Query ID	o	o	o
Default code	o	o	•
No. Series	o	•	•
Step size	o	•	•

5.2.16 Entity Member Buttons



5.2.16.1 Styles

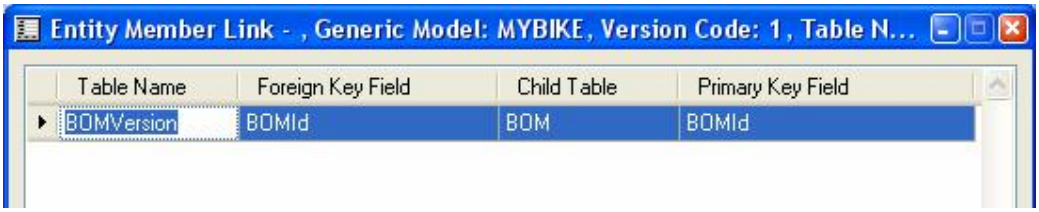
Here, the style of an Entity member can be filled in. This style influences the behavior of the entity member in the Front End.



5.2.16.2 Entity

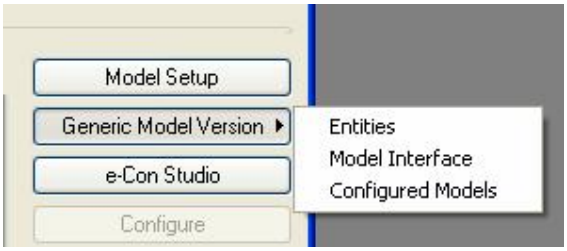
When the selected Entity Member is of Member Type Relation the related Entity form can be opened.

5.2.16.3 Table Relation



Normally it is not necessary to tell the data structure what fields must be equal to have a valid relation between to table records. When the model version is certified this is checked. If it is needed to fill the related fields manually is will be told during certifying of the model version. Then it is necessary to fill this form. Add a line with CTRL-N and select the appropriate fields.

5.2.17 Generic Model Version Button on Generic Model Form



Entities

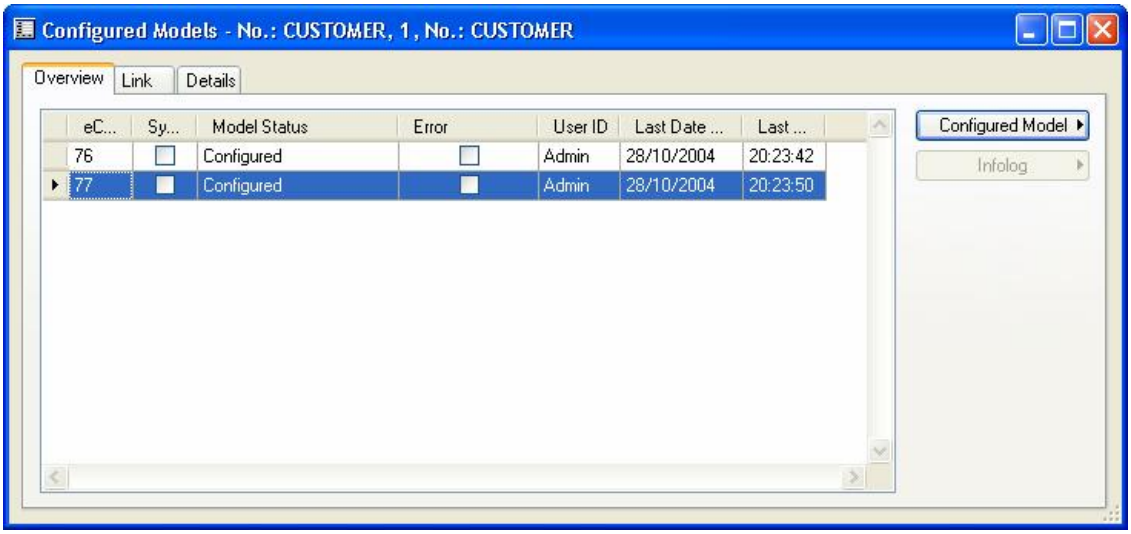
Displays the entities from the selected model. See section 5.2.23

Model Interface

Shows the interface for the current model. Only if the model is referenced from another model, the model is a sub model, an interface is required and must be selected here.

Configured Models

This form displays the objects that have been configured from the Front End. Once the "Save" button is clicked in the Front End, this list will be updated.



e-Con Model ID

e-Con will automatically create this value. It is a serial number.

Synchronize

If set to Yes before Configure the model will be synchronized during reconfiguration.

Model Status

The status of the model: new, configured, etc.

Error

This field contains an error code if there's a problem with a configuration.

User ID

User ID of the user who configured the object.

Last Date Modified

The last processing date of the configured object.

Last Time Modified

The last processing time of the configured object.

5.2.18 e-Con Studio Button on Generic Model form

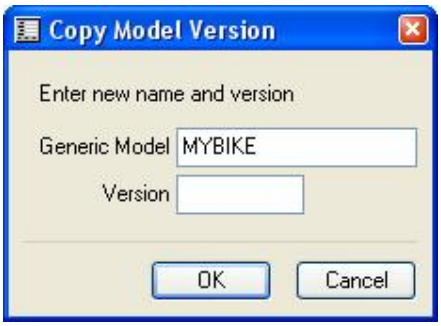
Start the e-Con Studio for the active version of this Generic Model. Rules can be used to add intelligence to the data structure. For example, if the type of the bike is a mountain bike, no mudguards can be chosen. After rules are added to the e-Con Studio, the rules are stored in the XML document of the Generic Model.

5.2.19 Configure Button (only available when the version is “certified”)

Start the Front End for the active version of this Generic Model. Use this function to view the layout of the Model in the Front End, or to test the model. Once the Front End is saved or "Cancel" clicked, Axapta returns to the Generic Model Card. However, if the Front End is started from the sales order line, the system returns to the sales order line.

5.2.20 Copy Model Version button

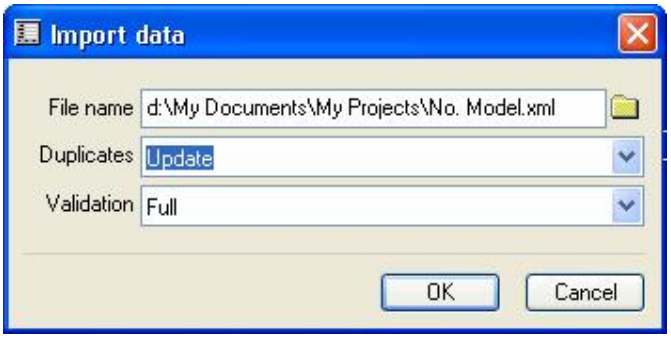
It starts the function to make a copy of the selected model version.



Fill the Generic Model and Version and OK. New versions for the same model can be made, but also new version of other models or a whole new model can be made.

5.2.21 Import Model Button

With this function an export e-Con model file can be imported in the Axapta database.



File Name

The file name of the e-Con export model file can be selected here.

Duplicates

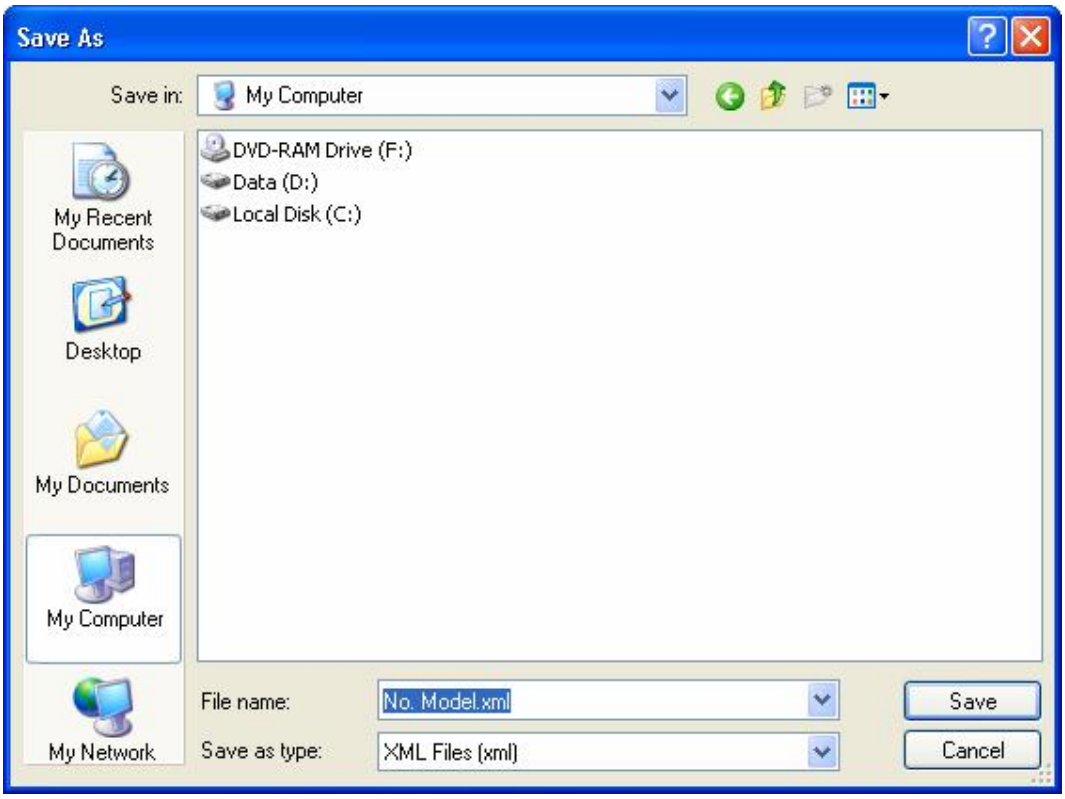
Set up how to handle with duplicates found during import. Typical e-Con labels and e-Con queries could already exist.

Validation

Set up how the Axapta record validation should be performed at the import of the e-Con model.

5.2.22 Save as Button

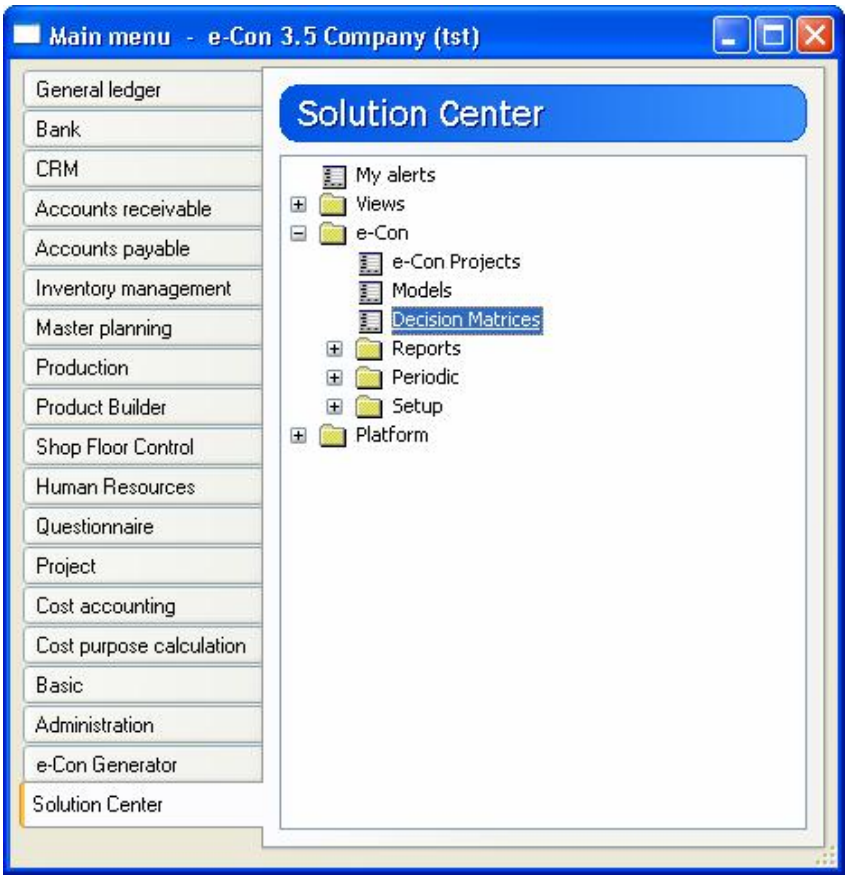
With this function an export file for the selected model version can be made to be imported in another Axapta database.



5.2.23 Decision Matrices

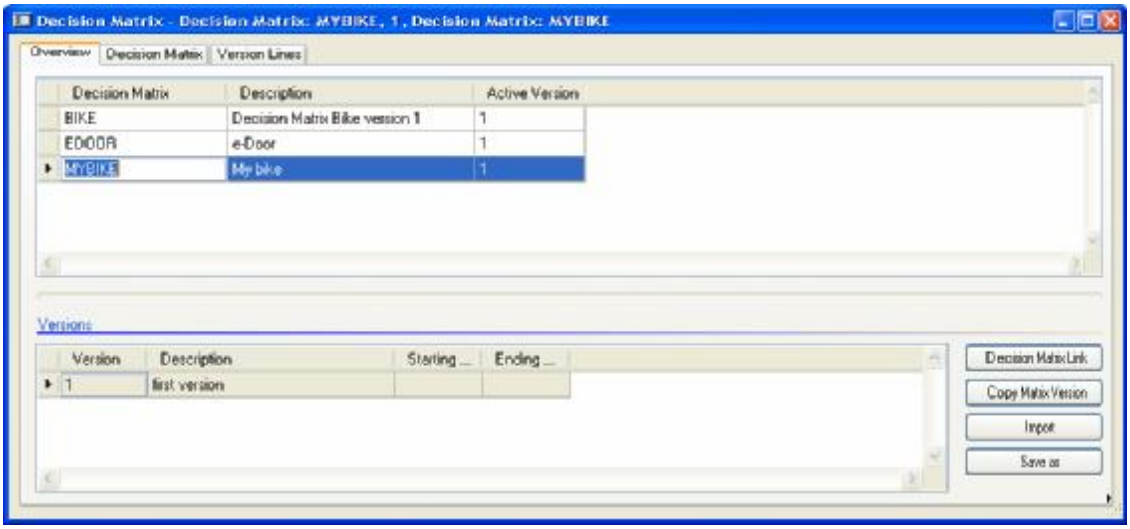
A Decision Matrix is used to set up interdependencies between member options.

You can find Decision Matrices in the e-Con menu, by going to the "Decision Matrices".



It is possible to determine more then one decision matrix and every decision matrix can have one or more versions.

5.2.23.1 Overview Tab



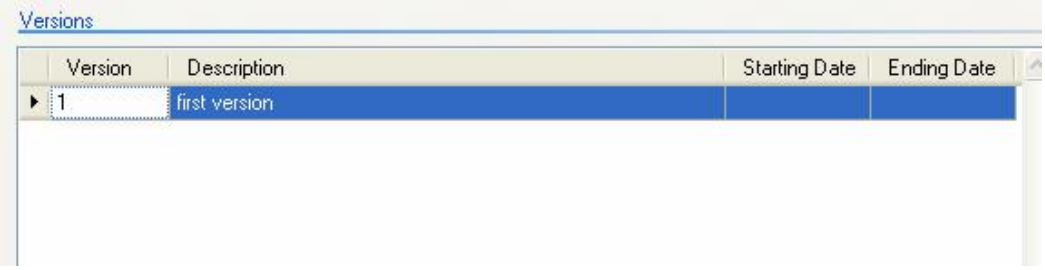
Decision Matrix

Unique alpha-numeric code for the decision matrix.

Description
Description of the decision matrix.

Active version
One of the determined versions is the active version of the decision matrix.

5.2.23.2 Version lines



Add versions to the decision matrix.

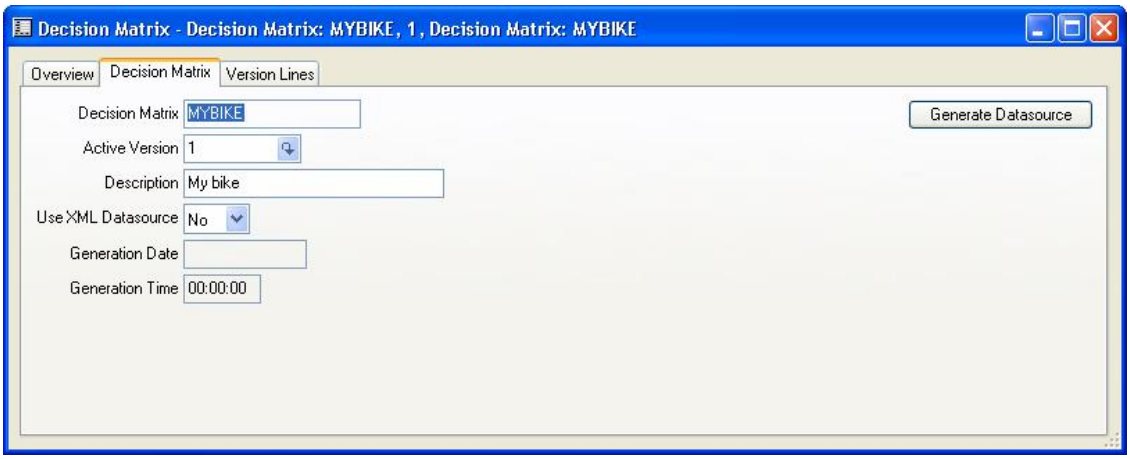
Version
Code of the version. You can enter a maximum of 20 characters, alphanumeric.

Description
Description of the version. You can enter a maximum of 50 characters, alphanumeric.

Starting Date
The starting date of the period in which the version is valid.

Ending Date
The ending date of the period in which the version is valid. A version can only be used for the decision matrix while it is valid.

5.2.23.3 Decision Matrix Tab



Decision Matrix

Id of the decision matrix

Active Version

The version that is active.

Description

Description of the decision matrix

Use XML Datasource

Field for whether this decision matrix is available in XML as well.

Generation Data

Date of generation of the XML file of the decision matrix. (Only filled up when the decision matrix is available in XML format)

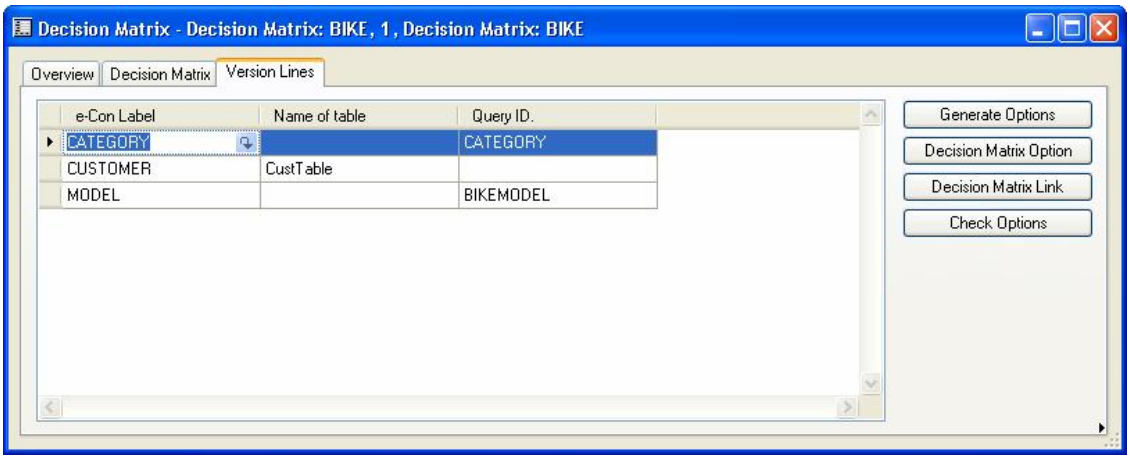
Generation Time

Date of generation from XML file from the decision matrix. (Only filled up when the decision matrix is available in XML format)

Button Generate Datasource

This function is used to export the selected decision matrix to a XML file. The XML file generated is stored in the folder Datasources in the eCon data folder.

5.2.23.4 Version Lines Tab



In the version lines the queries used in the decision matrix are determined, per version.

e-Con Label

Label of the version line (option set).

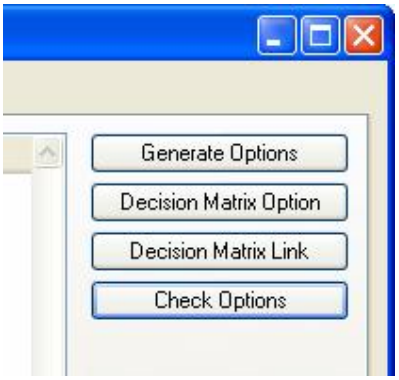
Name of table

Table linked to the query that will deliver the option set. Empty if the query is not based on an Axapta table.

Query ID

Select the query that will deliver the options.

5.2.23.5 Functions Button

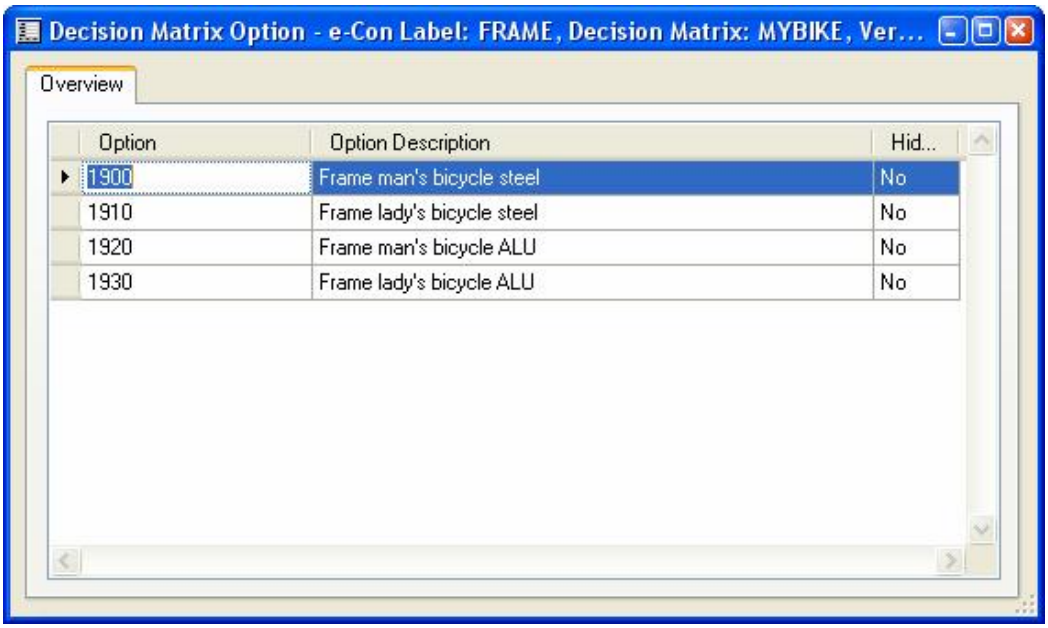


Generate Options

Generates the options for the version line, based on the Query linked to the version line.

Decision Matrix Option

Options that are generated are shown.

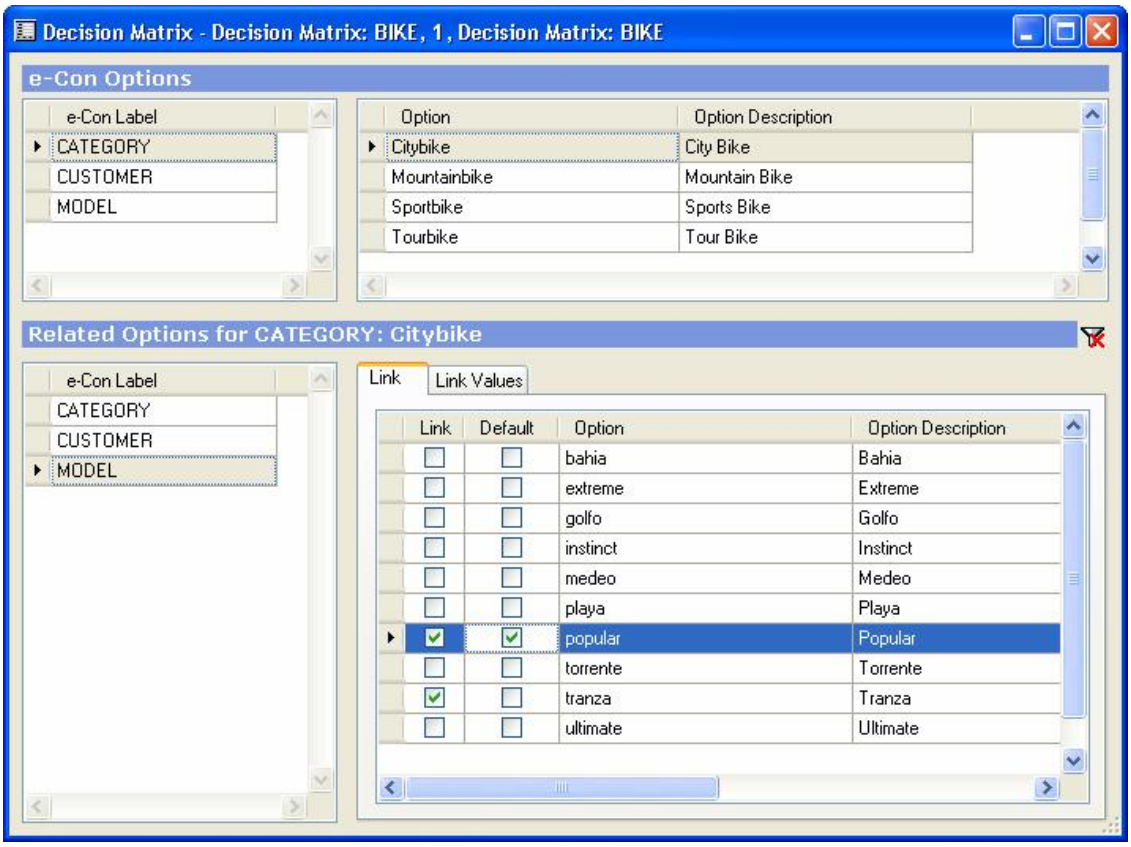


Options can be added manually.

With “Hide in Matrix” options can be left out the decision matrix.

Decision Matrix Link

The form were the relation between the options can be entered is opened.



5.2.23.6 Link Tab

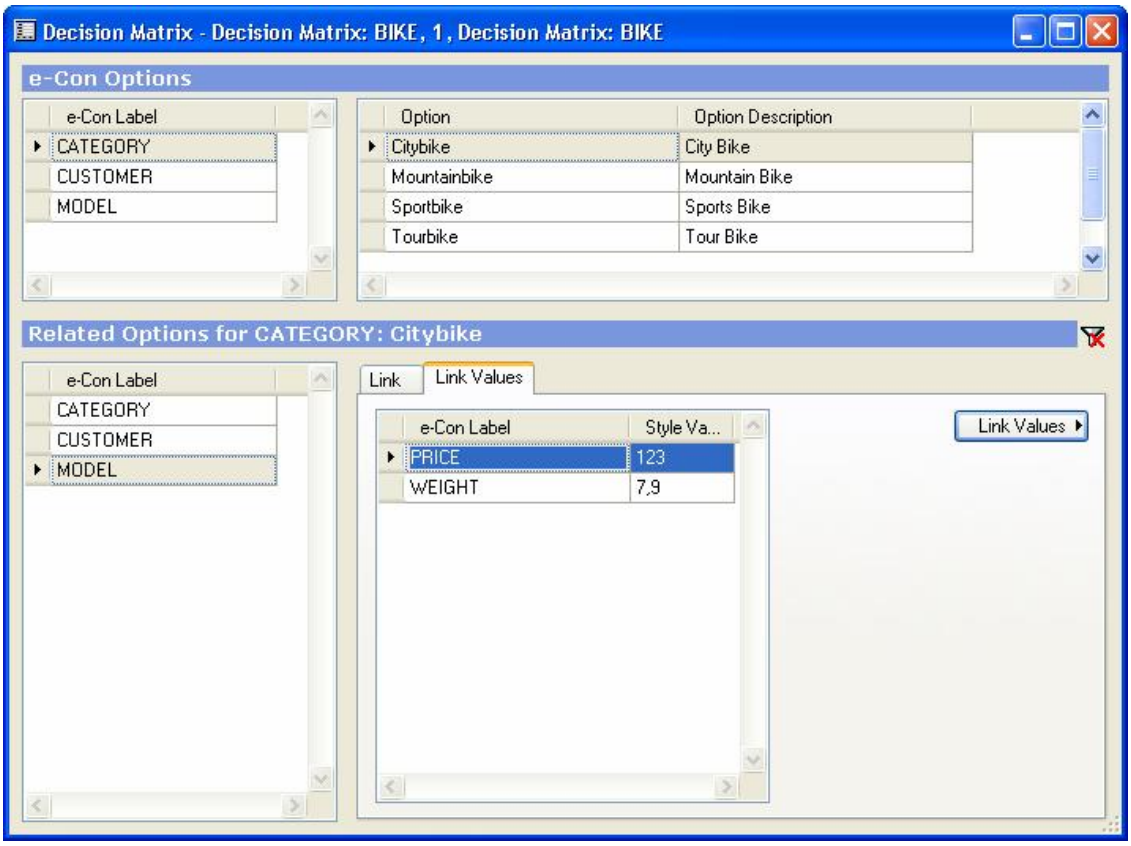
Link

The valid option for the upper part selection can be set here. (In this case popular and tranze are valid options for the Category Citybike)

Default

The default option for the upper part selection can be set here. (In this case popular is the default option for the Category Citybike)

5.2.23.7 Link Values



e-Con label

The label, identifier, of the linked value. This value is linked to the option checked at the link tab.

Style Value

Linked value itself.

5.2.23.8 Link Values button - Defaults

Default label and value can be set up here. These labels and values will be defaulted in for every checked option of the decision matrix.

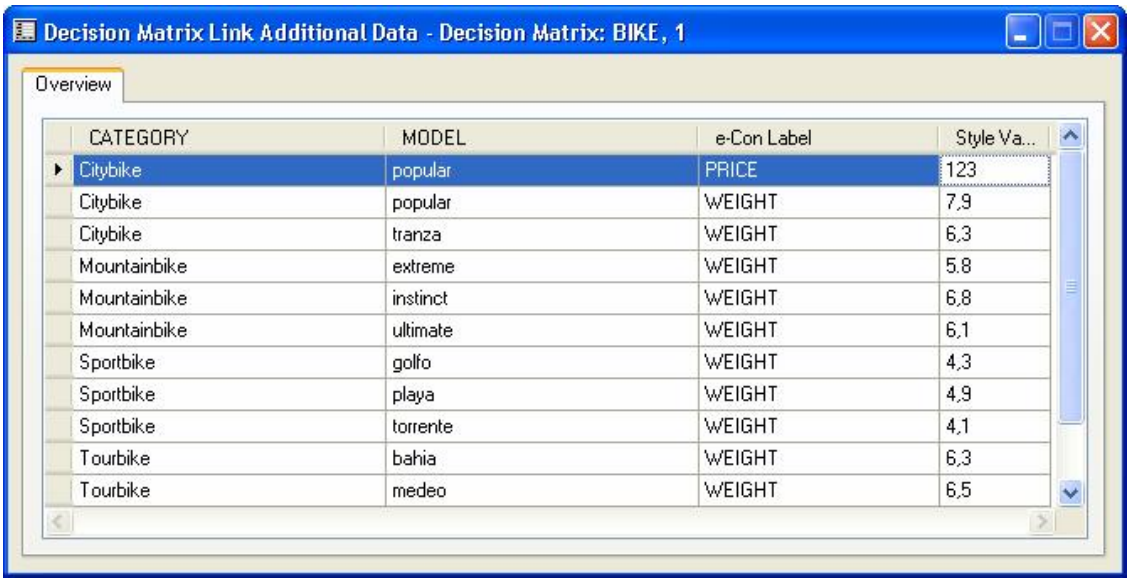
e-Con Label:

The default label, identifier, of the linked value

Style Value:

Default value itself.

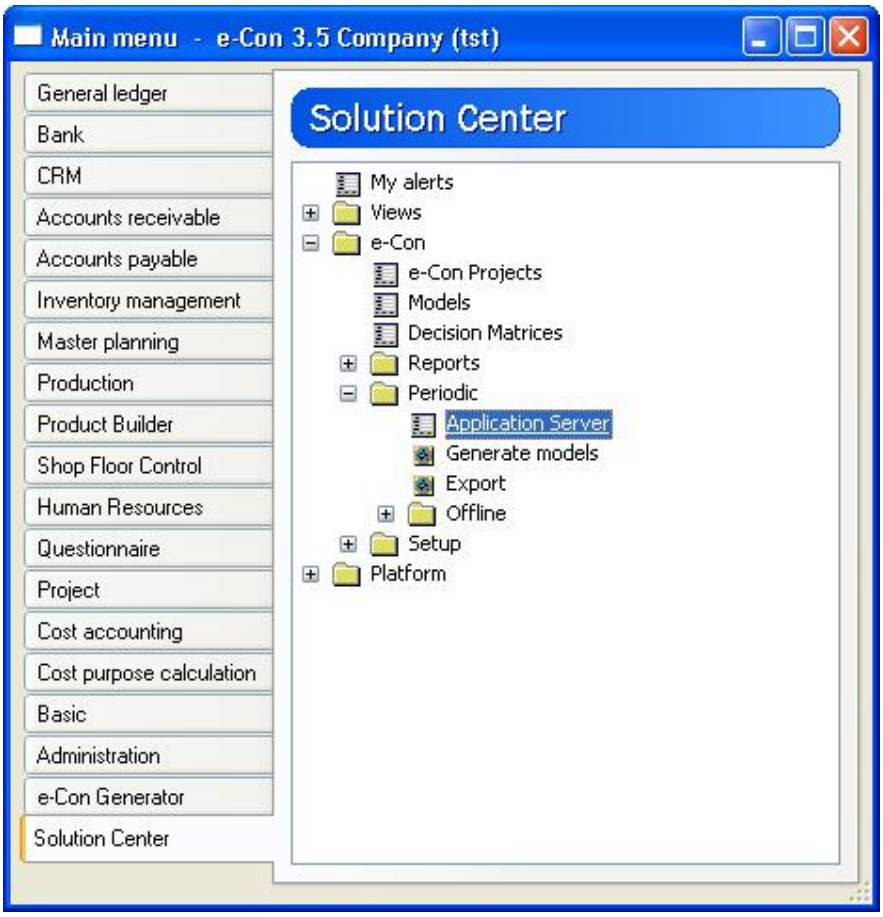
5.2.23.9 Link Values button - Overview



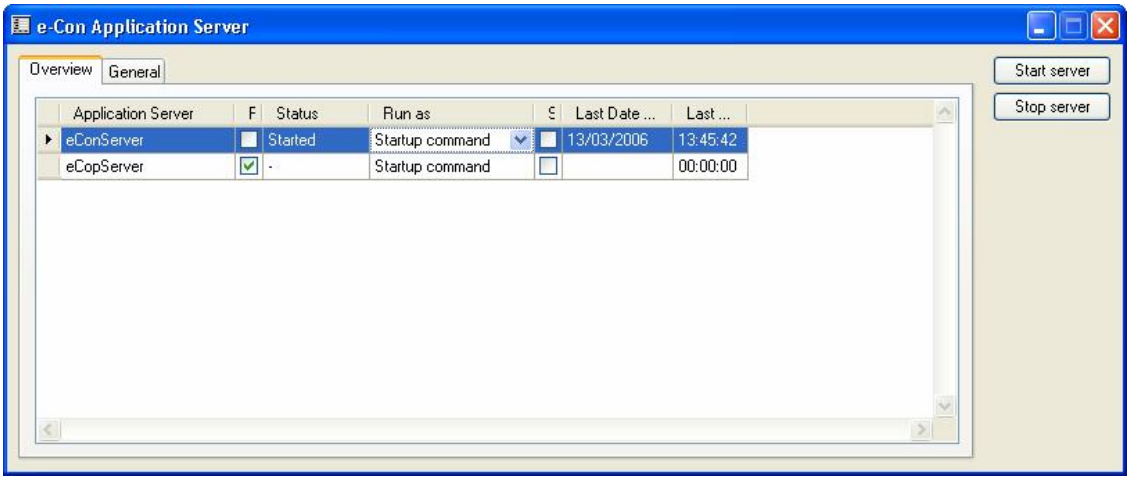
An overview of all linked values added in the decision matrix.

5.2.24 Application Server

Application server can be found in the Periodic menu. The application server is the server that handles the request of e-Con for the real time data access to Axapta.



5.2.24.1 Overview tab



Application Server

Name of the application server. Default name is eCon Server and eCop Server. Any name is allowed here.

Processing

This field indicates if the Application Server is used for Document Processing. The Application Server can either be used to consult the Axapta database during the configuration process or to process the XML document after saving. The last option is only used when the Post functionality is used. (Post functionality allows to save and process a configured document without exiting the e-Con UI)

Status

Shows the status of the Appication Server, the correct status is only displayed if you use the buttons (Start Server and Stop Server) to start and stop the Application Server. There are 4 statuses:

- Stopping The Application Server is stopping. After status started the status changes to stopping after clicking the button Stop Server. Only when the Application Server is stopped correctly the status changed to stopped
- Stopped The Application Server is not running (or not correct)
- Starting The Application Server is starting. Clicking the Start Server button will first change the status to starting. After the correct start up of the server the status becomes started.
- Started The Application Server is running in the right way

Run as

This field has two options batch and client.

- Client: The client is used as an Application Server. Clicking the start button runs the current client as the e-Con application server
- Start up command: An Axapta batch process running as a service runs as the e-Con application server

Stop

Indicator when the Application Server must stop. This field is checked after activation of the Stop Server function.

Last Date modified

The last date of stopping or starting the application server

Last Time modified

The last time of stopping or starting the application server

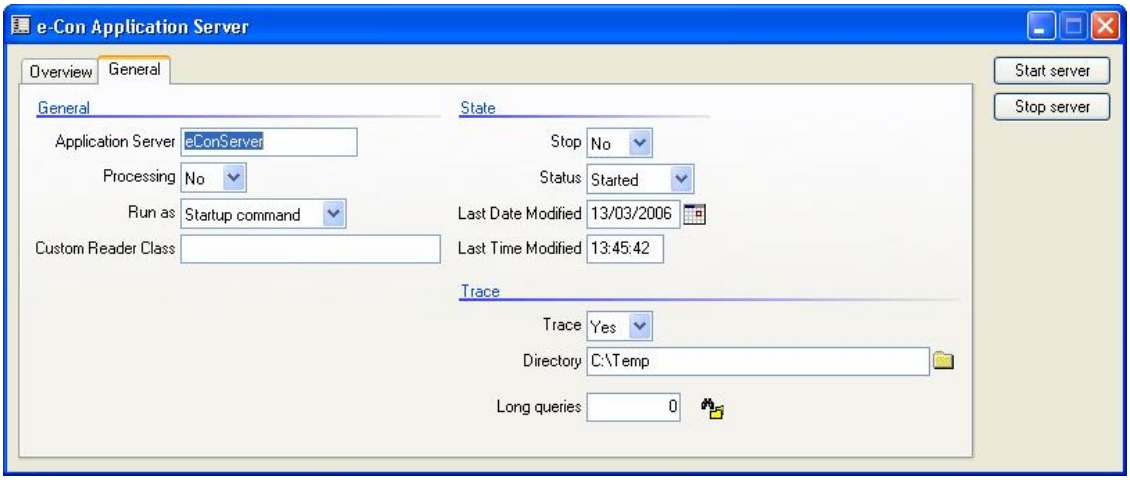
Start Server – button

To start the selected application server

Stop Server – button

To stop the selected application server

5.2.24.2 General tab



Application Server

Name of the application server. Default name is eCon Server and eCop Server. Any name is allowed here.

Processing

This field indicates if the Application Server is used for Document Processing. The Application Server can either be used to consult the Axapta database during the configuration process or to process the XML document after saving. The last option is only used when the Post functionality is used. (Post functionality allows to save and process a configured document without exiting the e-Con UI)

Run as

This field has two options batch and client.

- Client: The client is used as an Application Server. Clicking the start button runs the current client as the e-Con application server
- Start up command: An Axapta batch process running as a service runs as the e-Con application server

Custom Reader Class

When custom classes are executed with the e-Con Data Function 'CallFunction()' the class containing these custom class definitions must be specified here. Read more about this in the 'SDK Axapta integration'

Stop

Indicator when the Application Server must stop. This field is checked after activation of the Stop Server function.

Status

Shows the status of the Appication Server, the correct status is only displayed if you use the buttons (Start Server and Stop Server) to start and stop the Application Server

Last Date modified

The last date of stopping or starting the application server

Last Time modified

The last time of stopping or starting the application server

Trace

With this setting the trace functionality can be activated. Activating trace will log all communication between e-Con and Axapta. All the XML messages send via the message queues are logged and stored in the folder as indicated in the field 'Directory'.

Directory

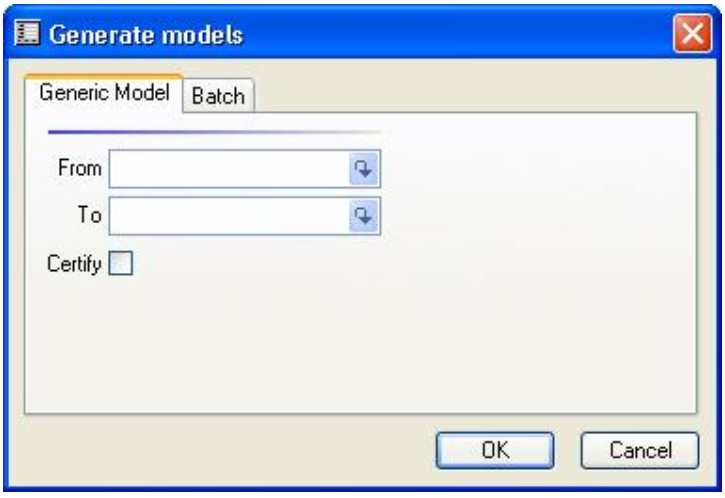
Location where the trace function stores the trace files.

Long Queries

A special function for performance analyses. All queries with a duration longer then specified in this field are gathered.

5.2.25 Generate models

Batch generate of models. Useful to update the XML files of the models with for example latest item and price information.



From:

Model from where XML generation must start. (Start of range)

To:

End of range of models to be generated.

Certify:
When check marked models in range are certified before generation.

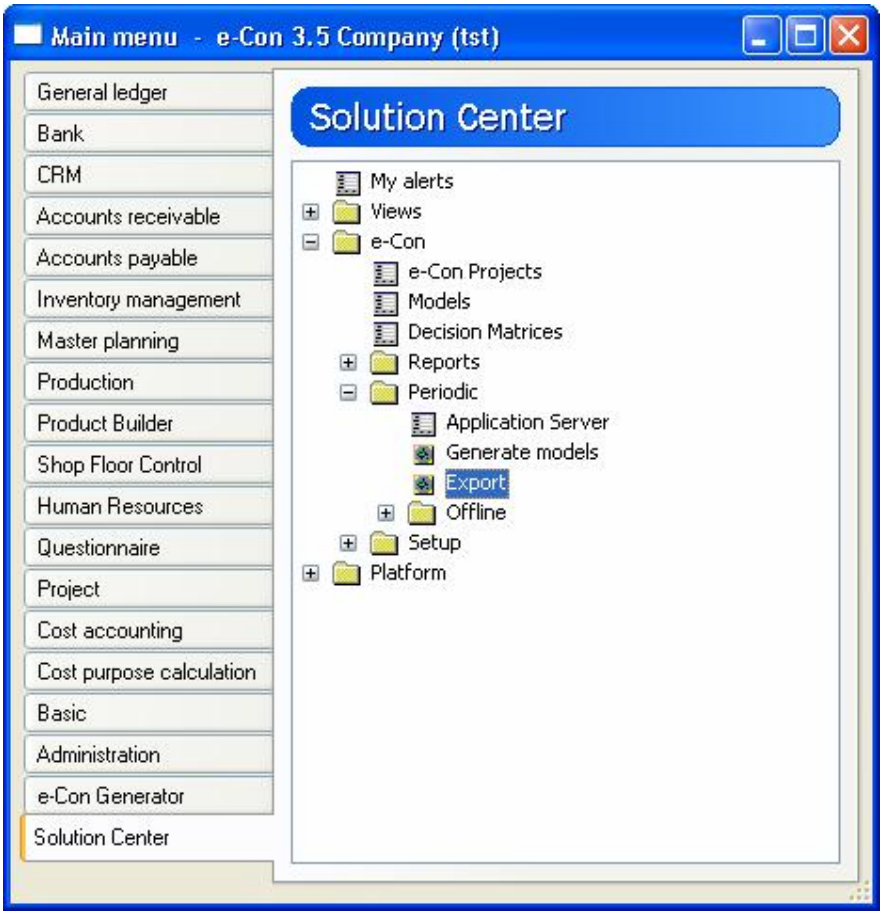
5.2.26 Export

Export of setup and data.

Export setup will export the following data:

- Queries
- Styles
- XML labels
- Context Transitions
- Metadata groups
- e-Con parameters

Export data will export all the models.
These export functions are useful to export the whole e-Con environment.

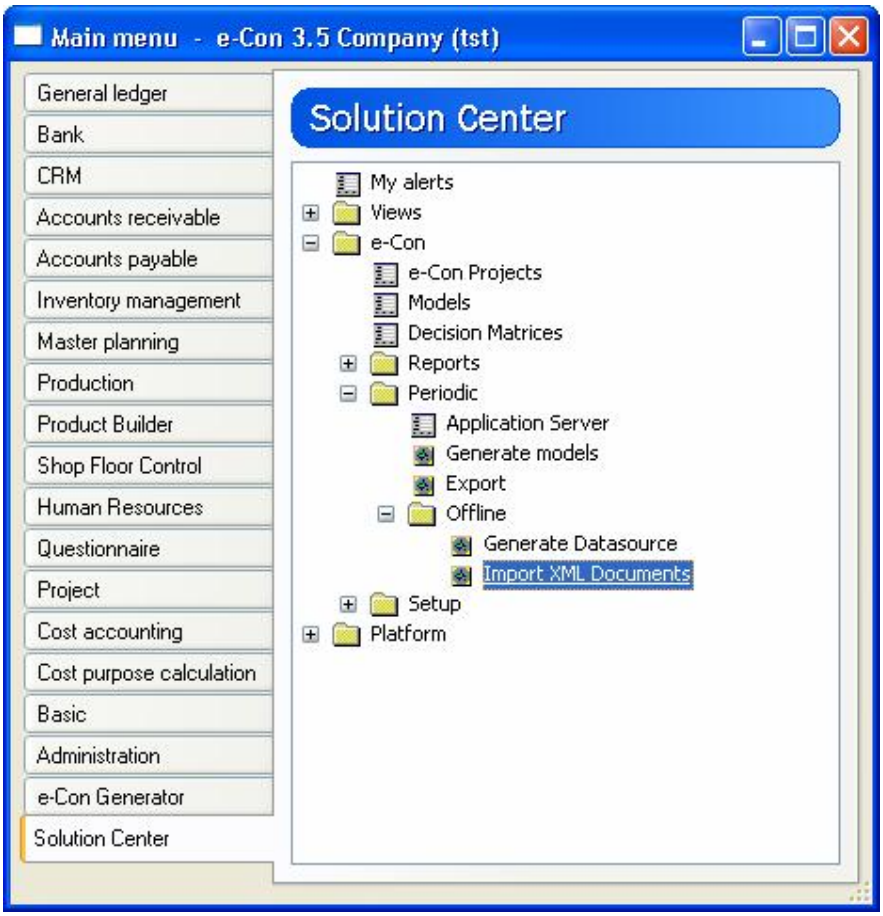


Export setup:
Check this field when the set up (queries, styles etc.) must be exported.

Export data:
Check this field when the data (e-Con models) must be exported.

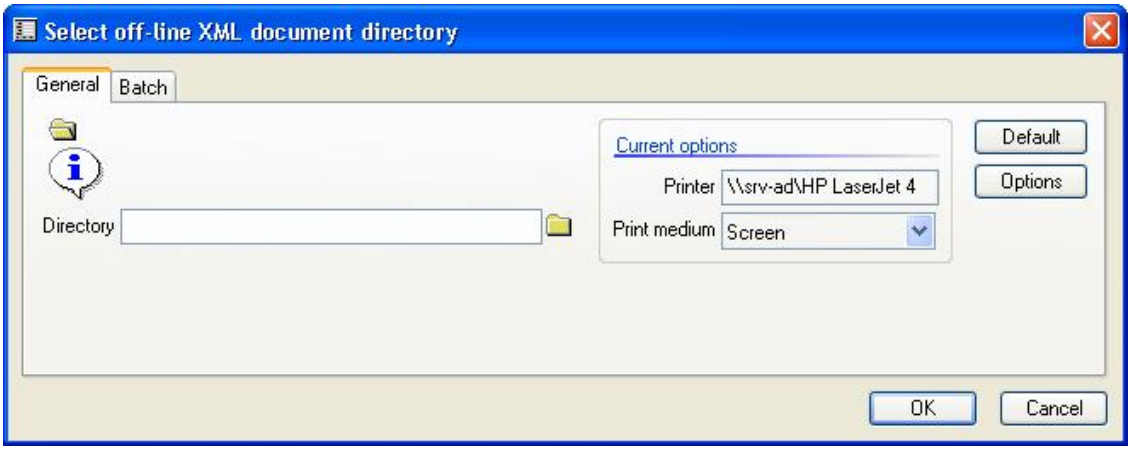
5.2.27 Import XML Documents

Import XML Documents can be found in the Periodic - Offline menu.



Offline created e-Con configurations (XML – files) can be imported and processed within this menu.

The tab Batch is the standard Batch functionality of Axapta and enabled the possibility to attach a batch process to the XML documents processing.



Directory:

Directory containing the XML configuration files to be processed. All files within this location will be processed and deleted from this location. After processing a document will be generated with information about status, created records and more

Printer

Selected printer.

Print medium

Selected output medium

Default Button:

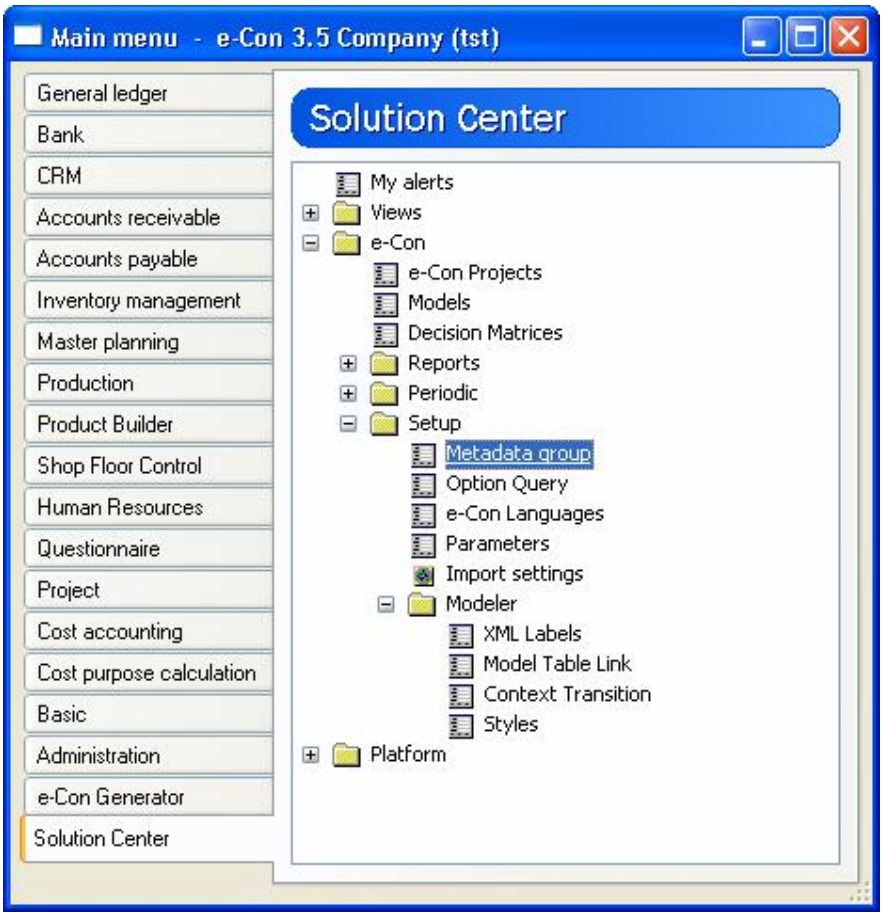
Clicking this button will change the printer and print medium to the defaults.

Options Button:

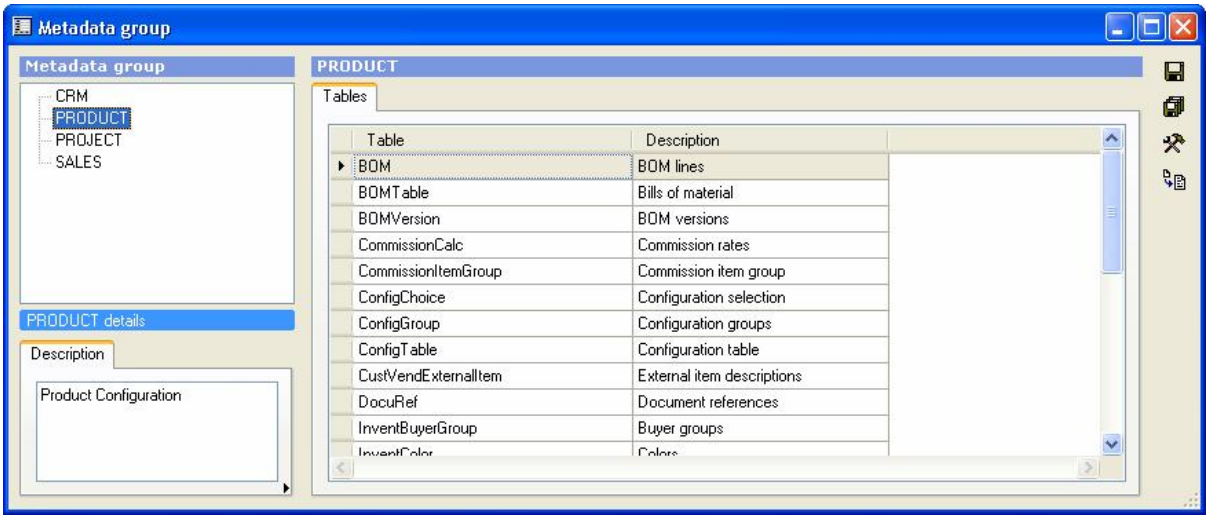
With this button printer and print medium can be set up and selected

5.2.28 Metadata group

The metadata groups can be found by going to the "e-Con" listing on the Main Menu, clicking "Setup", and then selecting "Metadata group" from the drop-down menu that appears.



Metadata groups are used in the 'external data wizard' of the e-Con studio. Groups and tables defined here will be shown up in the wizard and can be used in the e-Con data functions.



Product details - Description:
Description of the metadata group

Tables - Table
Table name being part of the metadata group.

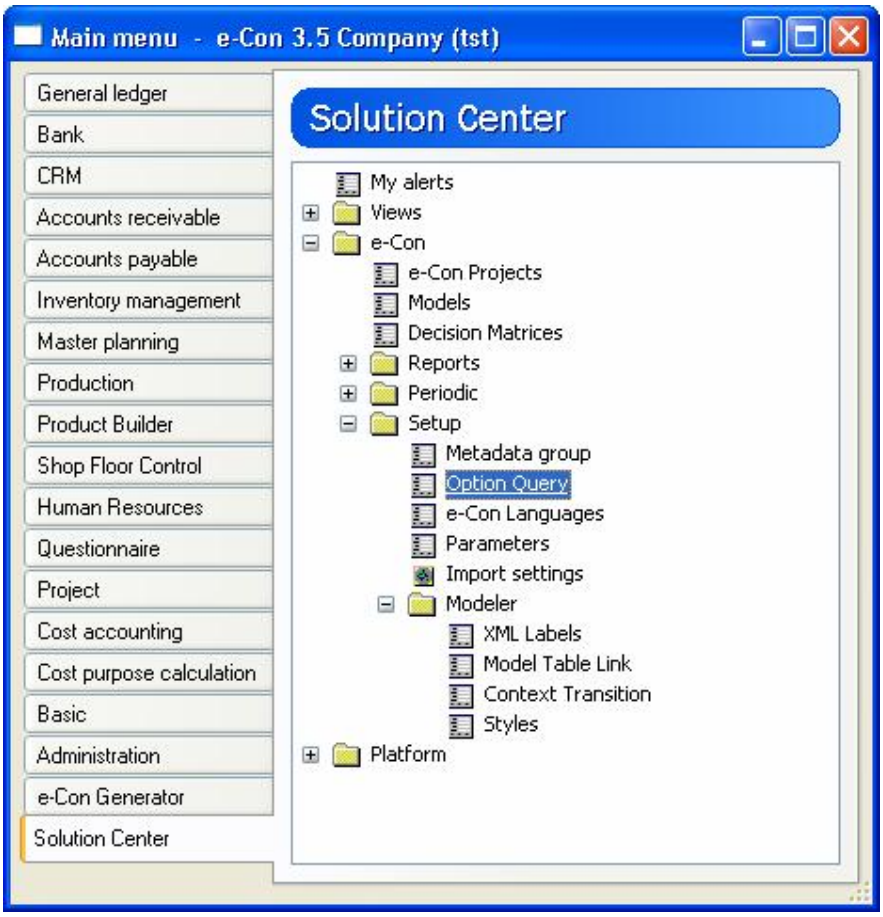
Tables - Description
Description of the table being part of the metadata group

Button – Query 
Start query to select multiple table at once

Button – Save & Generate 
Save and Generate the metadata XML source used by the 'external data wizard' of the studio.

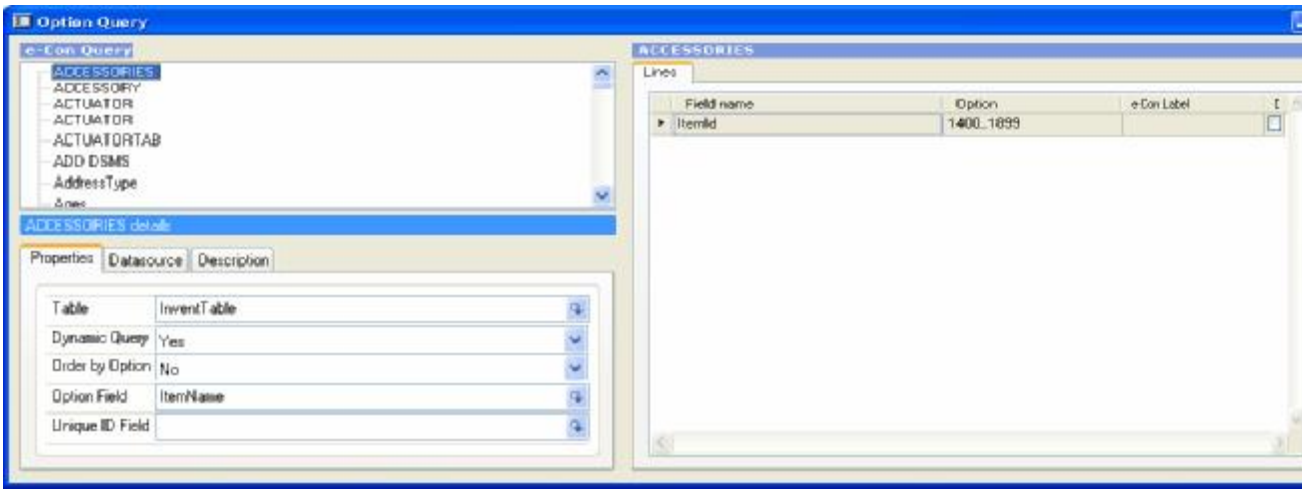
5.2.29 Option Query

The query form can be found by going to the "e-Con" listing on the Main Menu, clicking "Setup", and then selecting "Option Query" from the drop-down menu that appears.

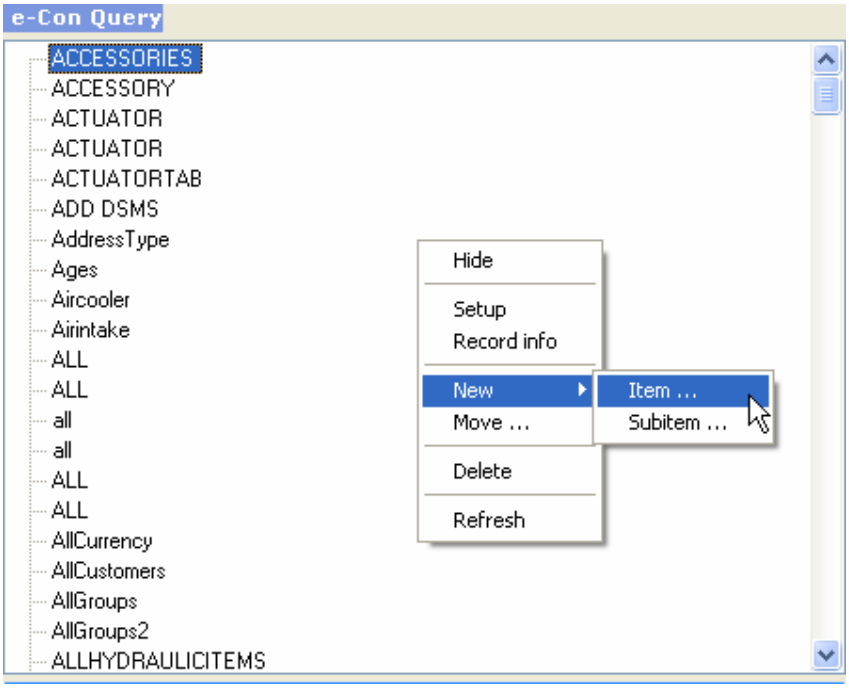


In the form that pops up, a query can be created. Queries are used to select information from Axapta tables. The end user will then be able to choose from the values selected at the Front End. There are several types of queries:

- "Query" (static), this query is executed upon generating the data structure
- "Option Query", this query is used to display options for additional properties that are not included in the Axapta table.
- "Dynamic Query", this query is activated during configuration by an 'add query' rule in the e-Con Studio.



5.2.30 Query tree



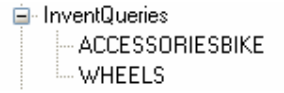
Tree displaying all queries.

New – Item

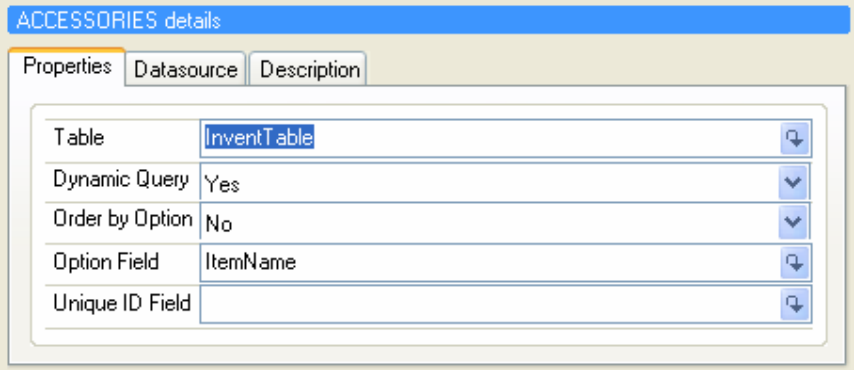
Creates a new query.

New –SubiItem

Creates a sub item for the selected item. Useful for grouping queries in the tree.



5.2.31 Properties tab of Query details



Table

Name of the table to which the query refers. If you want to make a selection based on, for example, the items, select InventTable here. If options does not exist in any Axapta table just leaf this field empty.

Dynamic Query

When this box is check-marked, the records that fit the result of the query will not be added to the XML document. The values are taken from the database only by means of rules. See section 6.6.3 for further information.

Order by Option Value

Only applicable for queries based on Axapta tables. When this box is check-marked, the order of the option list as shown in the drop-down box of the Front -end is based on the Option Field No. Without this box check-marked the order of the option list is based on the primary key of the table.

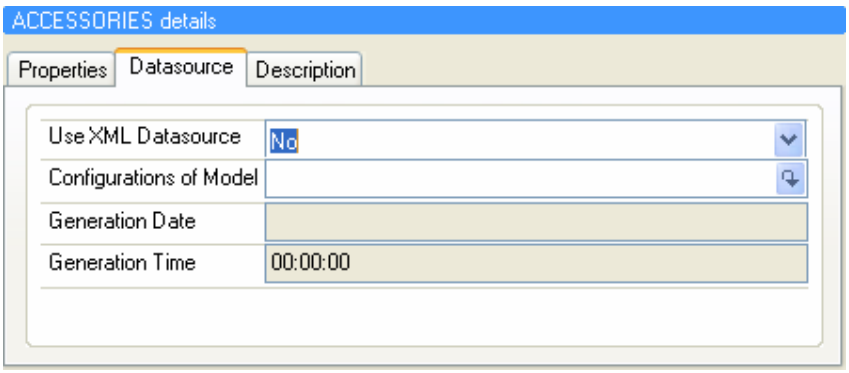
Option Field No.

This field is added to the XML document. It is used in the drop-down boxes of the Front End. In general, it is easier for the end user to choose between various descriptions than to make a selection from among item codes. The name entered here refers to the table field of the Table indicated after "Table Name"

Unique ID Field No.

This field indicates which field from the primary key acts as the key to generation the option. This field makes only sense for queries applied to table with a primary key that exists of multiple fields.

5.2.32 Datasource tab of Query details



Uses XML Data Source

Indicator if a XML Data Source is available from the query.

Configurations of Model

This field has to be filled up when a Data Source has to be build from former made configurations. Based on this model and the table name, a data source in XML format is build up. That data source is populated with the object id's and key values of the configurations of that particular model. Furthermore the data of the fields added in the option query lines are added as well.

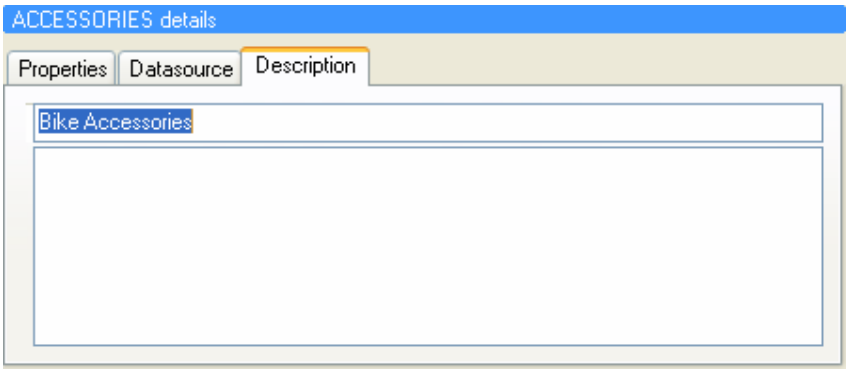
Generation Date

Last generation date of the data source

Generation Time

Last generation time of the data source

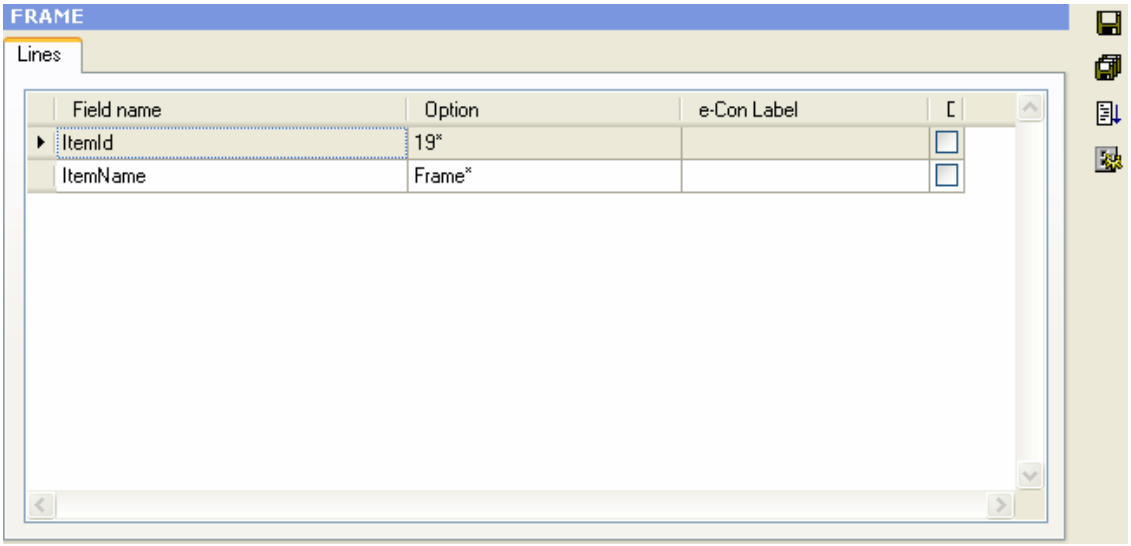
5.2.33 Description tab of Query details



Description

Description of the query.

5.2.34 Option Query Line Tab



Field Name

Name op the field out of the table the query is based on were a filter (option) must be set for to determine the outcome of the query.

Option

This is the actual query. You can use the standard Axapta filter signs, like 7*, 12..34 etc. to filter the record entered in the Filter Field. A query can consist of more than one line.

e-Con label

This field is only used for queries which are not applied to an Axapta table. In that case the description of the label selected here is used as the description of the option.

Default Value

If you want to propose a default value at the Front End, you can enter that default value here. In the case of a query based on a table: add a line where you type in the value and mark this value as default.

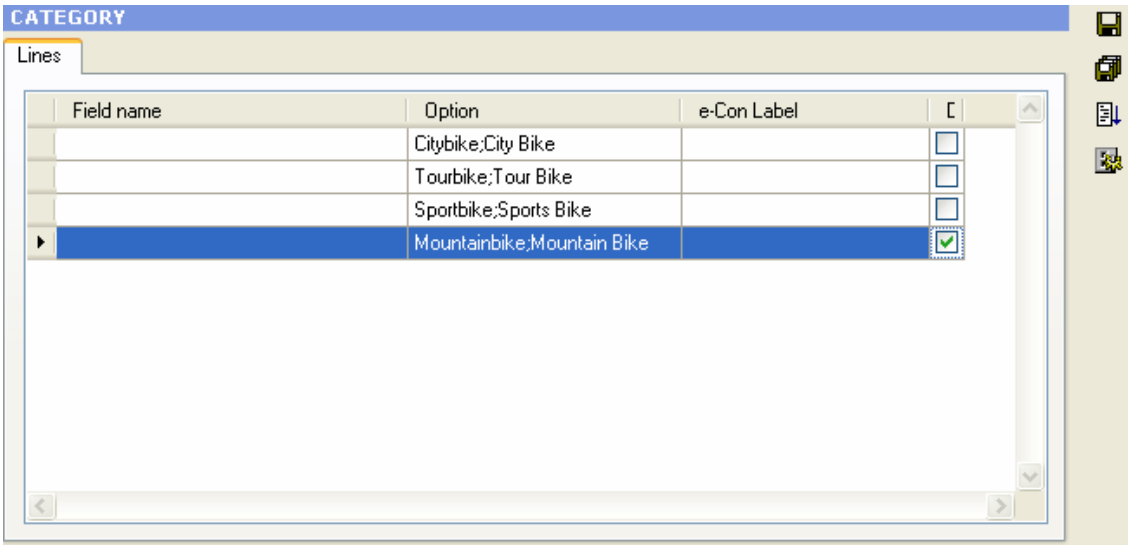
Button Run Query 

Clicking this button runs the query and will display the outcome of the query. Useful function for testing a query.

Button Generate Datasource 

it is also possible to store the outcome of a query in an external XML Data Source. It's even possible to generate an XML Data Source containing the configured objects of a Generic Model with user defined features. That makes it possible to check if a certain configuration already exists. Activating this button will create such an XML Data Source.

Example of a query used to create options with "Additional Property" member types for the Bike color.



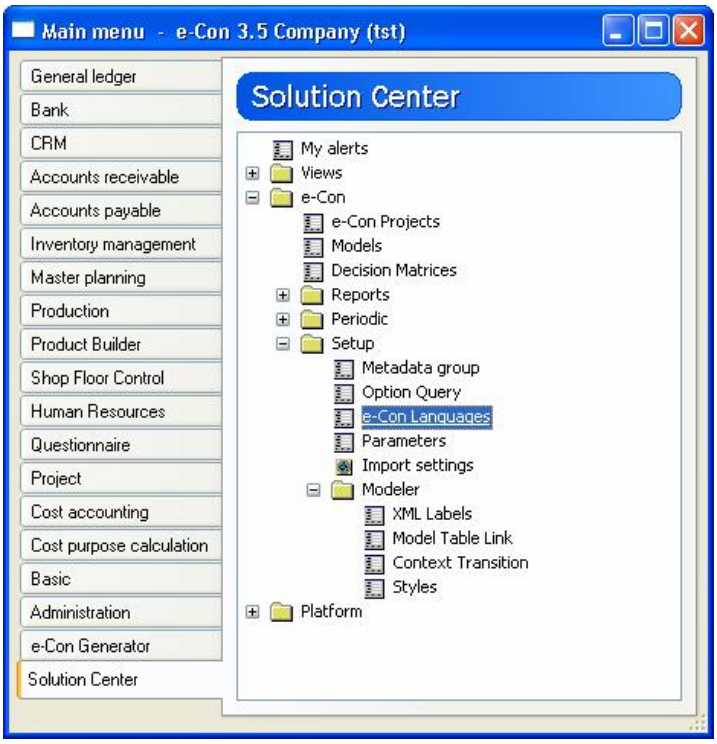
This query is not based on an Axapta table. In the Option field the option list is stated. When the member type is "Additional Property", the query is used as an Option field. Now the Table Name is empty. Use a different line for each option.

In the "Option" field, the value before the semicolon (";" otherwise known as the 'Option Description Delimiter') is chosen, but at the Front End, the value following the semicolon will be shown.

The end user has the choice between Pearl, Metallic and Chameleon. The default value is Pearl.

5.2.35 e-Con Languages

e-Con Languages can be found in the Set up menu.



Select every language you want to have available in the Front End here.

Language

Language id. Pick one of the languages from the list

Description

Description of the selected language

CultureCode

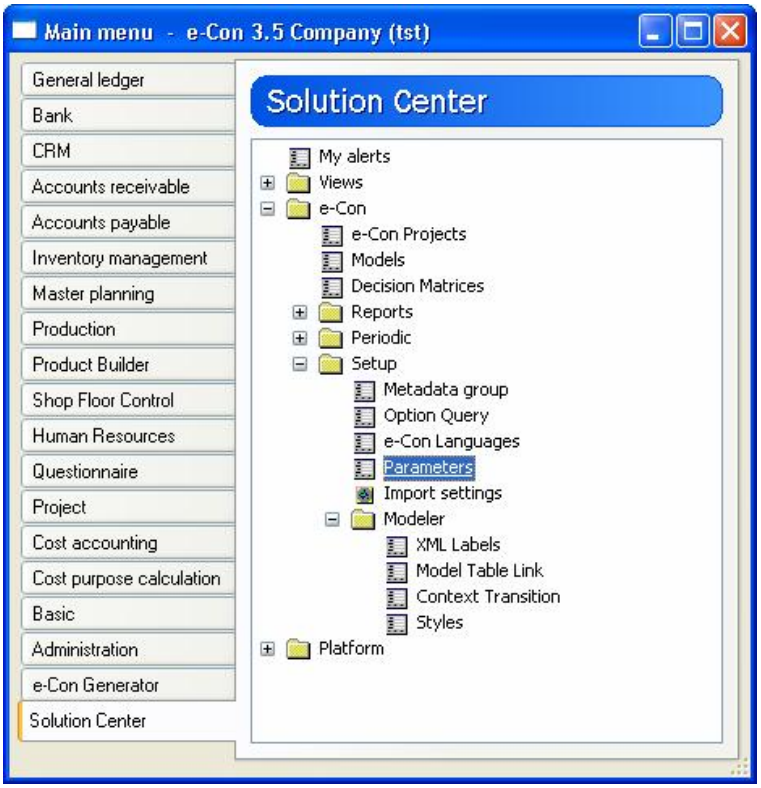
Select the corresponding windows culture name for the particular windows language here. Clicking the assist-edit button will auto pick the corresponding culture code.

Enabled

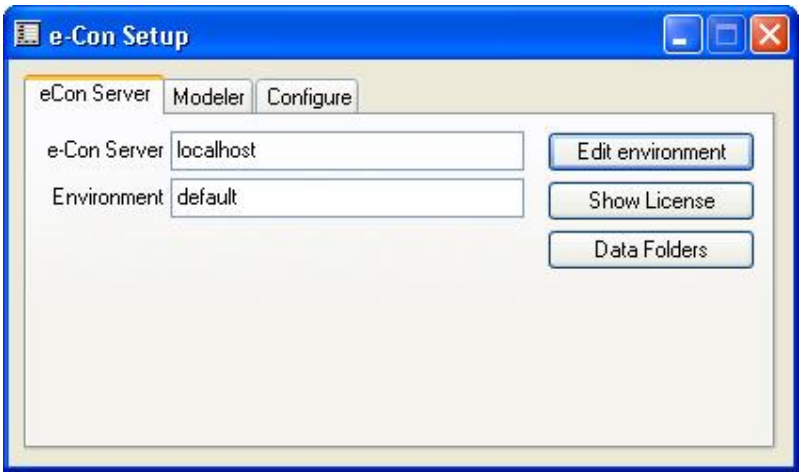
A check-mark in this field indicates this language is activated. When the language is enabled in this way, it can be selected when generating the data structure, or starting the Front End

5.2.36 Parameters

The parameters form can be found in the e-Con Main Menu under "Setup" – “Parameters”.



Before we can use e-Con, the setup data has to be entered. The location of files and user preferences are stored in the Setup form.



Tips & Tricks:

Default values can be entered in the e-Con Setup form: click on the "Initialize" button in the e-Con Setup form.

For a more detailed explanation of the e-Con Setup, see the Installation Manual on the product CD-ROM.

5.2.36.1 Fields of the eCon Server Tab

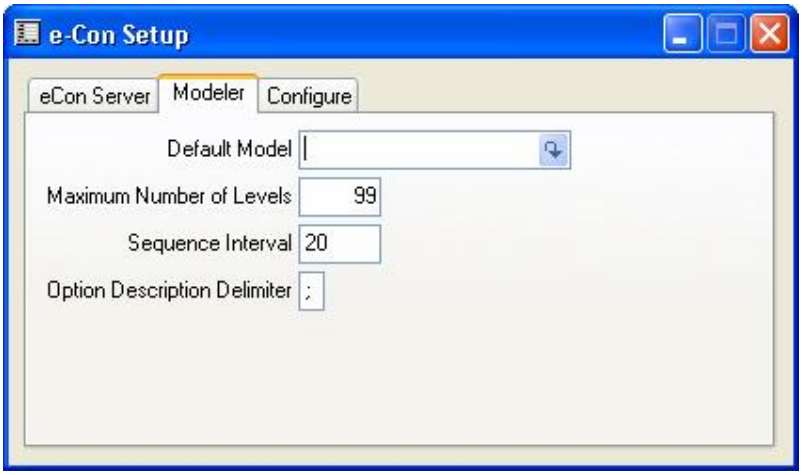
e-Con Server

Enter the name of the server on which e-Con is running.

Environment

The environment used. Per e-Con environment a set with settings are available.

5.2.36.2 Fields of the Modeler Tab



5.2.36.3 Default Model

The Entities of the model selected here are used as a template for new Entities. For example, when the default model has an Inventtable Entity with 5 members, and the first InventTable Entity is created in a new model, the whole InventTable Entity is copied from the default model. If the entity already exist in the model the entity is copied within the model. Filling this setup field will activate the copy function.

Maximum Number of Levels

To limit a loop in the model, you can define a maximum number of levels. When Axapta generates a data structure, it will stop if the number of levels in the document exceeds this value, and then it will generate an error.

Sequence Interval

The sequence of the entity members shown in the Front End can be determined and changed. Based on the Sequence Interval the sequence will be numbered. The sequence can be changed by filling manually the “Sequence” field with a number in the Entity Member form or by changing the sequence with Ctrl ↑ or Ctrl ↓ in the Model Setup Modeling tree.

Option Description Delimiter

Options are linked to the additional properties of Entities, via queries. The symbol entered here is used as a separator.

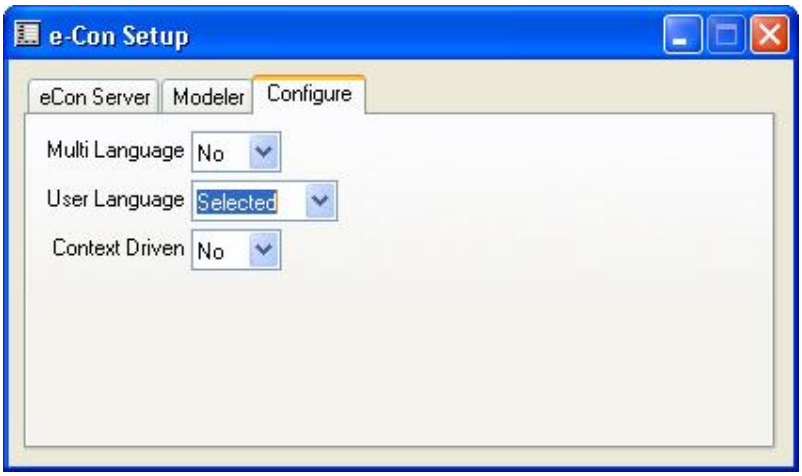
An additional property can have multiple options. For example, the "Color" additional property has the options "Red", "Blue" and "Yellow". These options are stored in a query (see section 3.3). In the "Option" field, the value displayed before the Option Description Delimiter is chosen, whereas the value after the Option Description Delimiter is shown in the Front End. Example: the Option Description Delimiter is a semicolon ";"

1;blue

2;red

3;yellow

Now the end user has the choice between "Blue", "Red" and "Yellow", but in a rule you can say: "If option is 1, then price is 10".



5.2.36.4 Fields of the Configure Tab

Multi Language

To activate the multilingual functionality. When “yes” is selected, the label translations of the e-Con labels, will be added to the model during the XML file generation.

User Language

Setting concerning the language selection for an e-Con model. There are four options:

- Default: The default model language is always the model language;
- Client: The Axapta client language is the model language;
- Document: The language of the customer of for example the quote or order e-Con is started from, is the language of the model;
- Selected: The model language is manual selected each time the model is started.

Context Driven

To activate the context driven functionality.

5.2.36.5 Button

Edit Environment

To open and edit the environment settings. The environment specified at the field “Environment” will be opened.

Show license

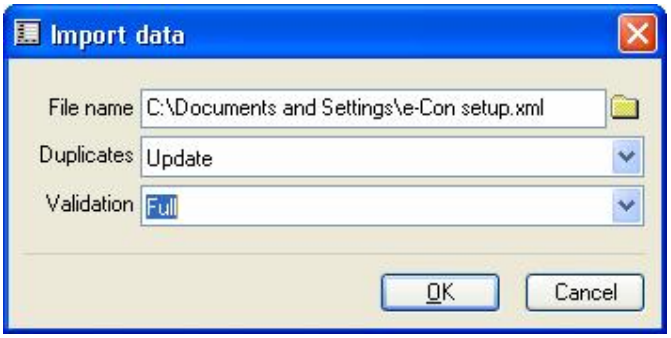
The XML license is shown. The XML license is available in the ...\e-Con\Web folder.

Data folders

To create the sub folder structure as required for e-Con in the data store as set up in the environment settings.

5.2.37 Import settings

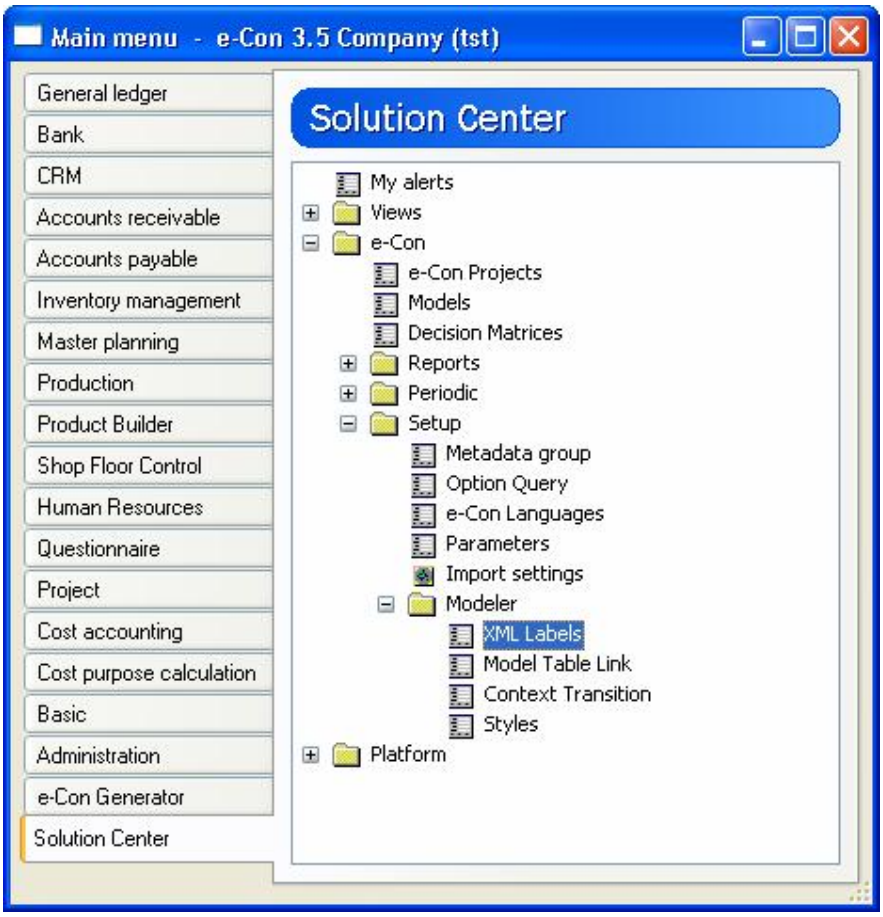
The "Import settings" form can be found by clicking on "Setup" in e-Con, and then selecting "Import Settings".



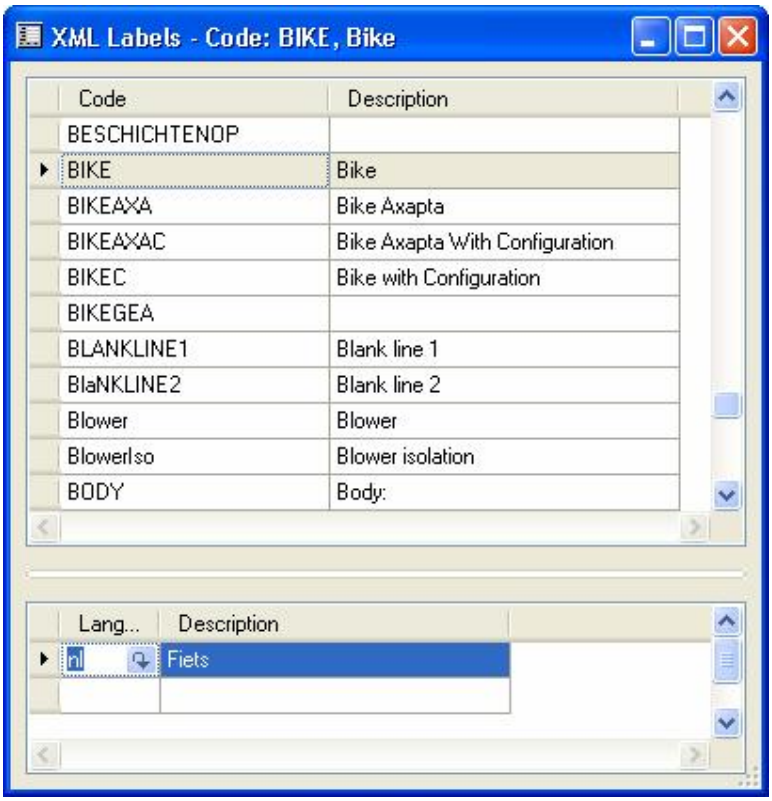
Data and/or settings can be imported here.

5.2.38 XML Labels

The "XML Labels" form can be found by clicking on "Setup" followed by “Modeler” in e-Con, and then selecting "XML Labels".



Labels are the names of Entities and members in the XML document, and can be defined in this form.



Code

The code for a Label. Choose a short name without spaces. This Code is used as an identification code (ID) when applying the rules in the e-Con Studio.

Description

Description of the Label. This Description is shown at the Front End. Use a meaningful name that will make sense to the end user.

Example: When the price of the bike (a field property) depends on the color of the bike (an additional property), you'll need to create two labels.

Code	**Description**
Color	Choose a color for the bike:
Price	The price for the bike will be:

You can then use a rule to give a value to the price. Such a rule might look like this: if "color" == "1" then "price" = 10. (Notice that we use the Code names in the rule, not the Descriptions.)

Language

In this part translations can be entered for the Labels in the languages needed. Descriptions of the Labels are displayed in the Front End in whichever language is selected when starting the configuration. For more information, see the section on Multilanguage Capabilities (section 6.12).

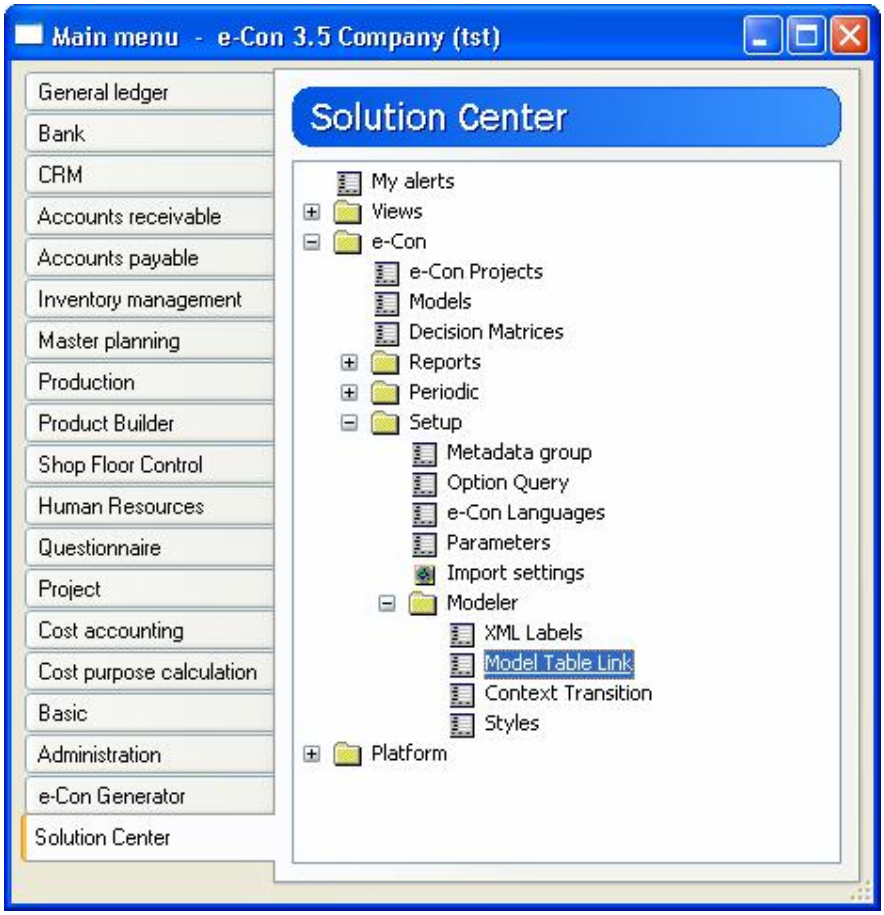
e-Con supports multiple languages. For practical purposes, this means that the end user, who answers the questions in the Front End, can see the questions, options and descriptions in the languages defined in the Generic Model.

Description

Translated description of the label

5.2.39 Model Table Link

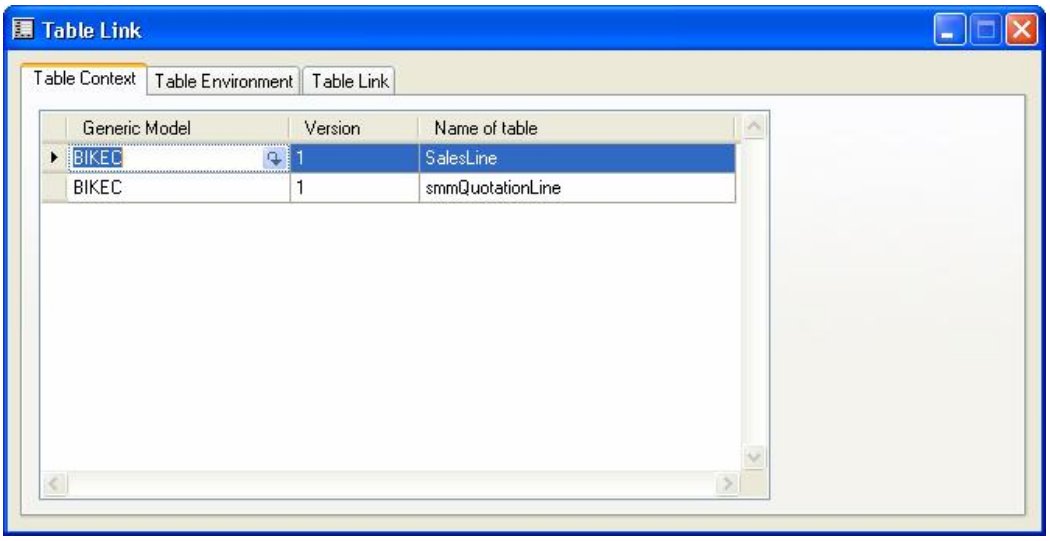
Model table link can be found in the Set up - Modeler menu.



The following functionality can be set up here:

- Table environment: Information of the Axapta Context can be passed to e-Con by means of arguments. These arguments can be read in e-Con with the function GetArg. The Axapta Context can for example be the Sales Line. The field custaccount can for example be passed to e-Con and known in e-Con by using the GetArg function to read this argument.
- Table link: Additional information can be populated in the Axapta Context after e-Con has returned to Axapta. The Axapta Context can for example be the sales line. Fields like quantity and config id can be populated by the e-Con model after e-Con returns at the sales line

5.2.39.1 Table Context Tab:



Generic Model

Select the Generic e-Con model where table link information must be added.

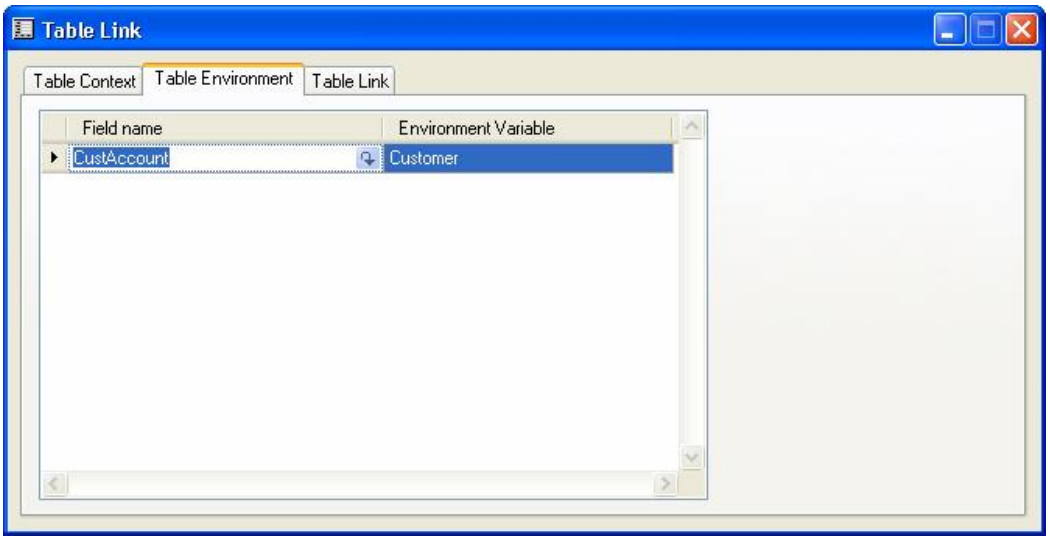
Version

Select the version of the generic e-Con model where table link information must be added.

Name of table

Select the table(s) where table link and table environment information must be added. Typically tables in this areas are ‘salesline’, ‘smmquotationline’.

5.2.39.2 Table Environment Tab:



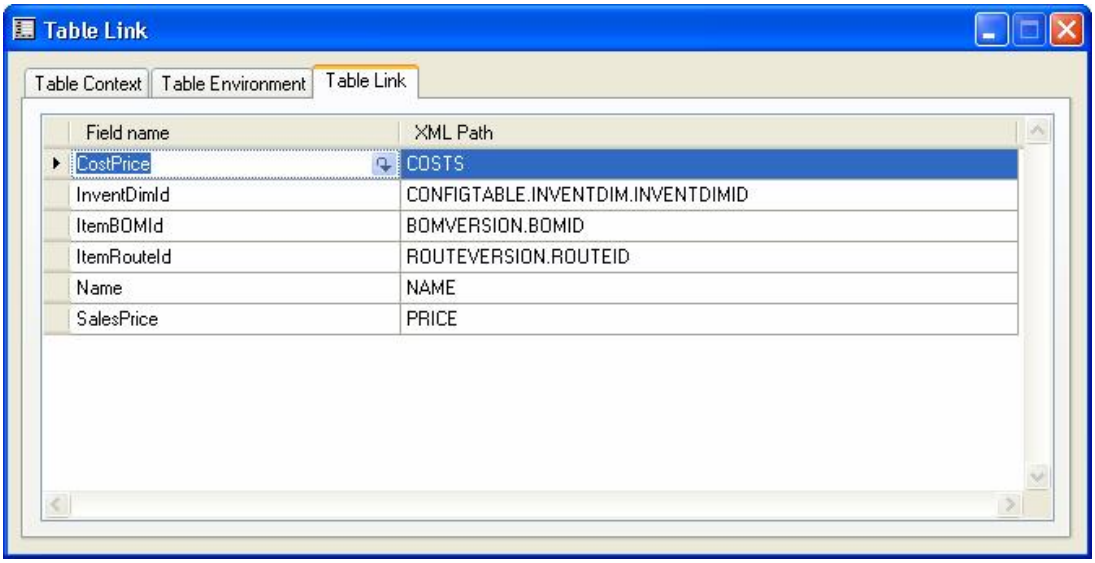
5.2.39.3 Field name

The field of the table which must be passed over to e-Con.

Environment Variable

The name of the variable (argument) used to store the value of the field. This variable can be read in e-Con with the GetArg() function. GetArg(Customer) will deliver the customer account of the sales line where e-Con is started.

5.2.39.4 Table Link Tab



Field Name

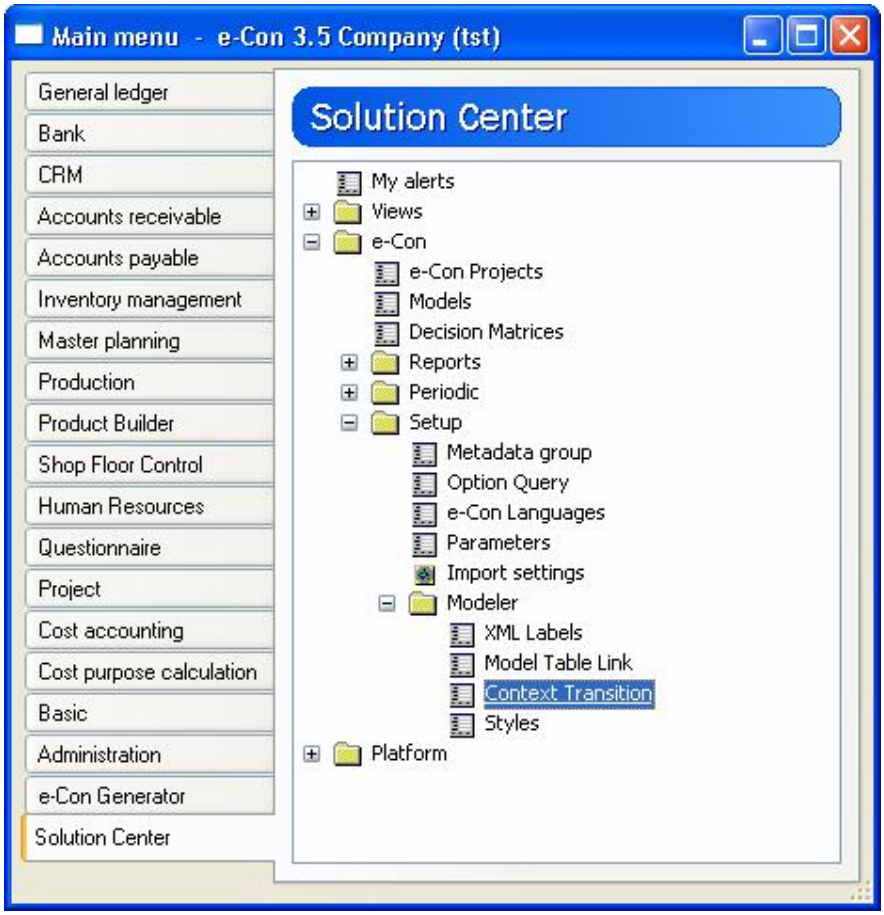
The field of the Axapta context (sales line in this case) to be populated by e-Con after processing of the e-Con model.

XML Path

The property or property path of the model which value must be stored in the field specified in 'Field Name'. The value of property 'COSTS' will be stored in the Axapta field 'CostPrice' of the Sales line.

5.2.40 Context Transition

Context Transition can be found in the Set up – Modeler menu.

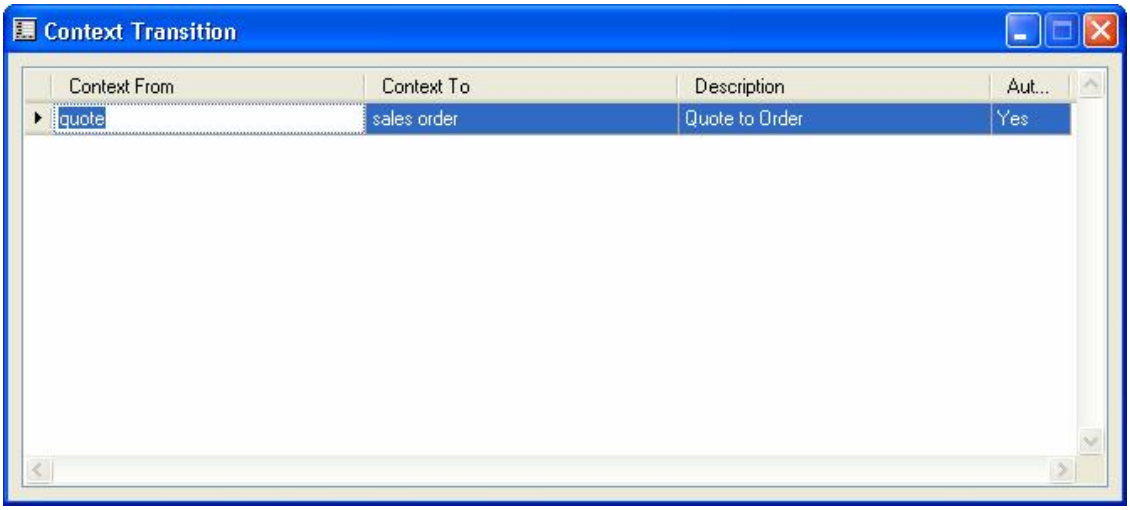


Within this menu context transition can be determined. A context is a business process context in Axapta. Two contexts are supported yet:

- Quote: Context when e-Con is launched from a CRM quote
- Sales order: Context when e-Con is launched from a sales order

With the context tranisition it is possible to launch e-Con in the back ground when a transition takes place from one to another context.

An example: When the context is quote the e-Con model will not create a Bill of Material and Routing. However, when the contex is sales order the Bill of Material and Routing are created. By setting up a context transition from quote to sales order, the e-Con model is recalculated in the back ground when the CRM Quote is converted into a sales order. During this recalculation e-Con will notice a change of context from quote to sales order and will create the Bill of Material and Routing.



Context From

The original context

Context To

The new context

Description

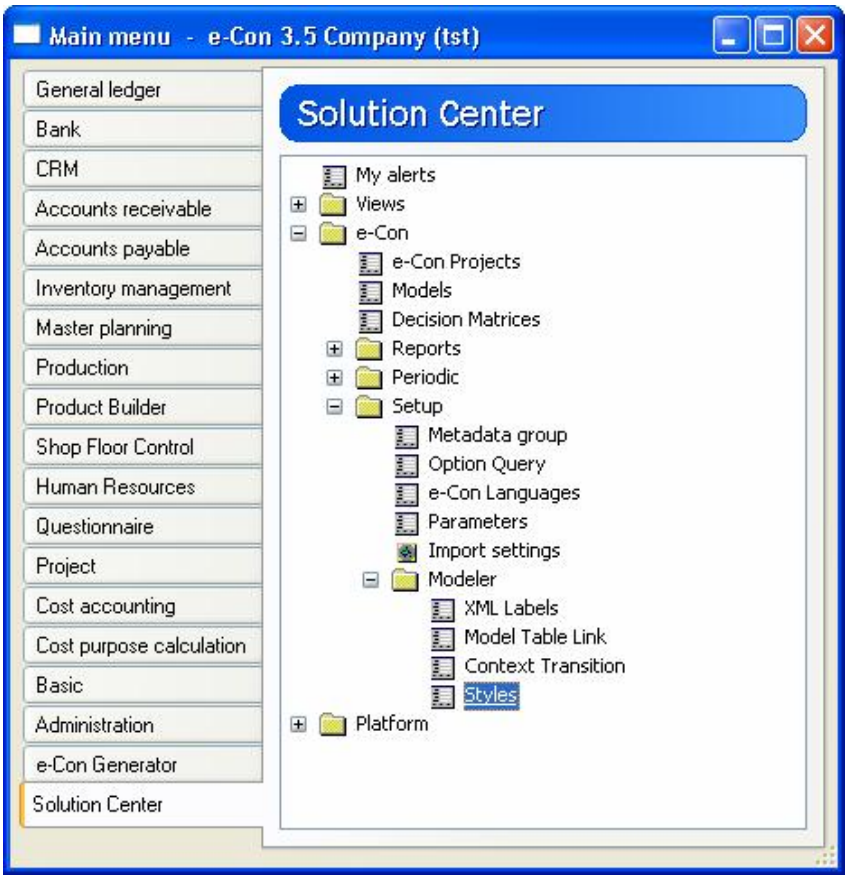
A description of the context transition

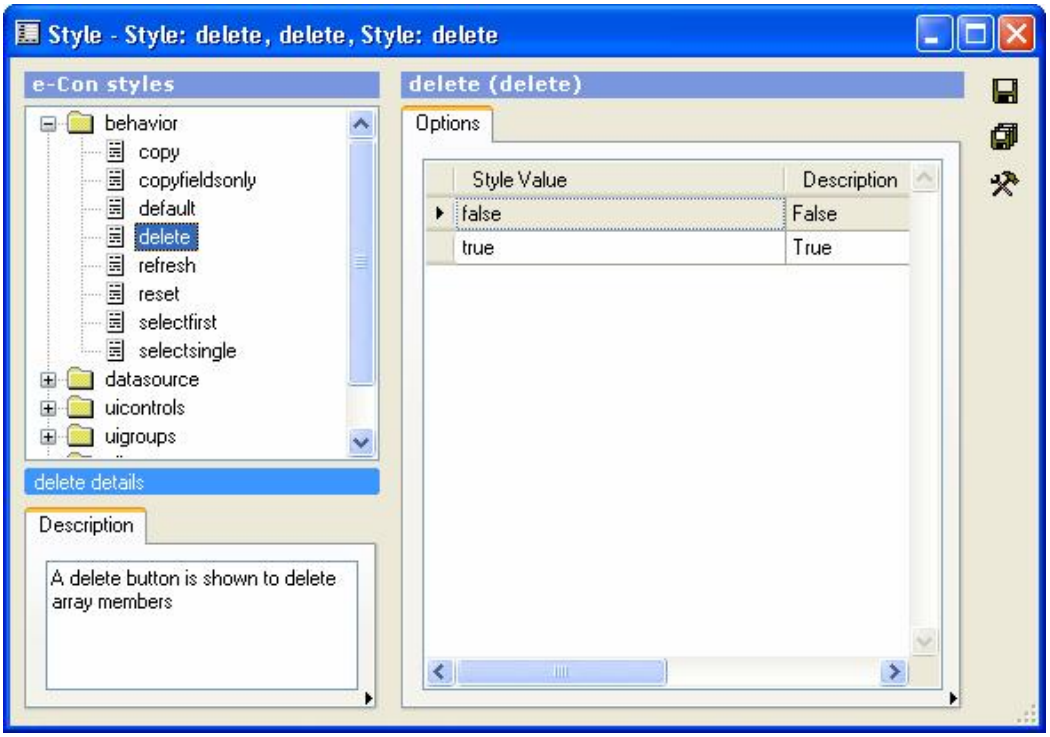
Automatic Processing

This box must be check-marked if you want to recalculater your e-Con model automatically during the context transition.

5.2.41 Styles

The Styles form can be found by going to "Setup" – “Modeler” in the e-Con menu, and selecting the "Styles" option.


Main menu - e-Con 3.5 Company (tst)
General ledger
Bank
CRM
Accounts receivable
Accounts payable
Inventory management
Master planning
Production
Product Builder
Shop Floor Control
Human Resources
Questionnaire
Project
Cost accounting
Cost purpose calculation
Basic
Administration
e-Con Generator
Solution Center
Solution Center
My alerts
Views
e-Con
e-Con Projects
Models
Decision Matrices
Reports
Periodic
Setup
Metadata group
Option Query
e-Con Languages
Parameters
Import settings
Modeler
XML Labels
Model Table Link
Context Transition
Styles
Platform


Style - Style: delete, delete, Style: delete
e-Con styles
behavior
copy
copyfieldsonly
default
delete
refresh
reset
selectfirst
selectsingle
datasource
uicontrols
uigroups
delete details
Description
A delete button is shown to delete array members
delete (delete)
Options
Style Value
Description
false
False
true
True

With styles special behavior can be added to properties. Special behavior added by styles are for examples:

- Copy and delete buttons for arrays
- Radio buttons
- Buttons
- Styles and colors for labels and fields
- Etc.

e-Con Styles pane

Overview of all styles. Styles are grouped in a folder structure here.

Description

Explanation of the style.

Options

Options available for the selected style

Buttons:

 Save current record

 Save all changes

 Initialize styles. All available styles and values are up loaded in Axapta when clicking the initialize button.

Chapter 6 e-Con Axapta Connector Data Structure

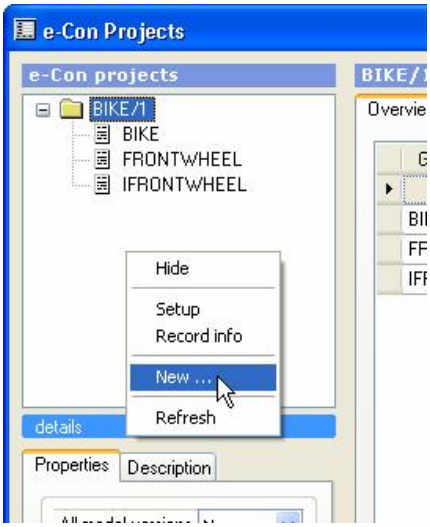
In this chapter, we explain the elements needed for setting up a model, and we explain the functionality of certain options.

6.1 e-Con Project

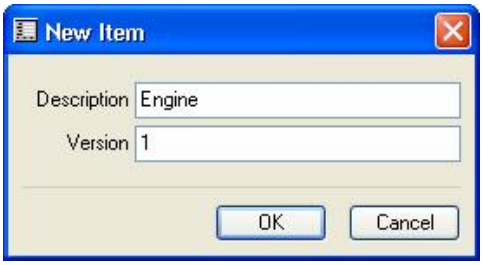
e-Con Projects are helping in the management and maintenance of the models. All e-Con Configuration Models within a project can be Build (compiled) in one simple action. The inbuilt project explorer offers a tree view of the project. Different tasks and actions can be started from here.

e-Con Projects are specially helpful in complex model structures build out of different sub models. All models can be tight together in a project.

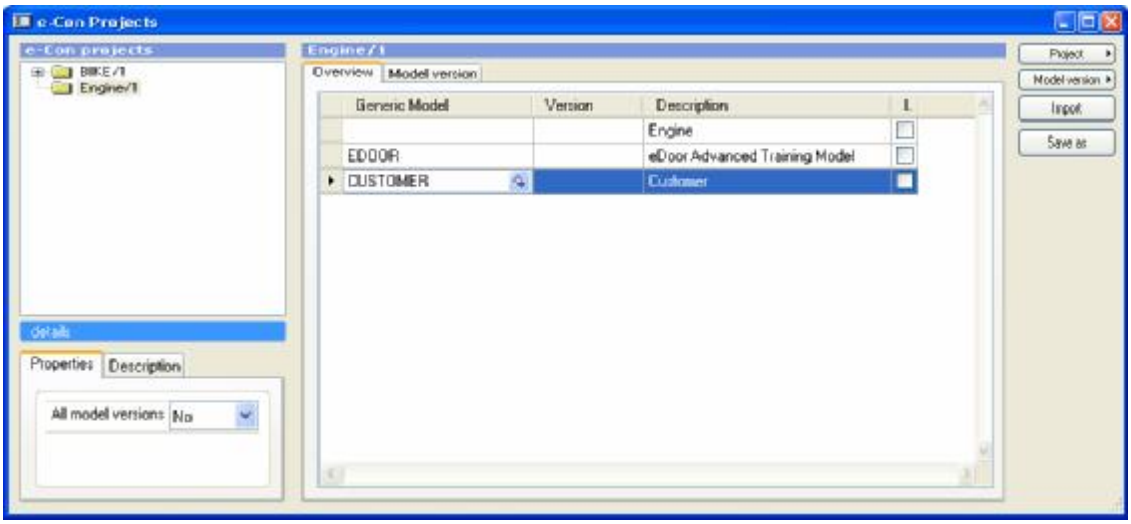
Add a project by right-mouse click in the e-Con Projects pane and select new.



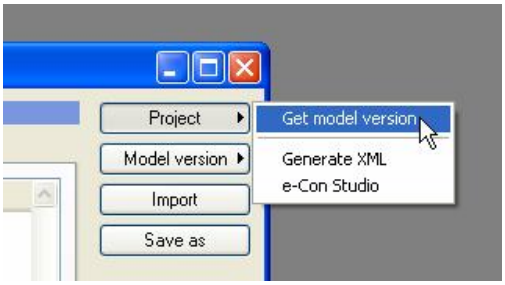
Enter a name and version and press OK:



The project is added. Next step is to add e-Con models to this project. This can be done by simply selecting models in the right pange, first create a new record and afterwards select a model.



Models can be added as well by selecting them with the Get model version function available under the project button.

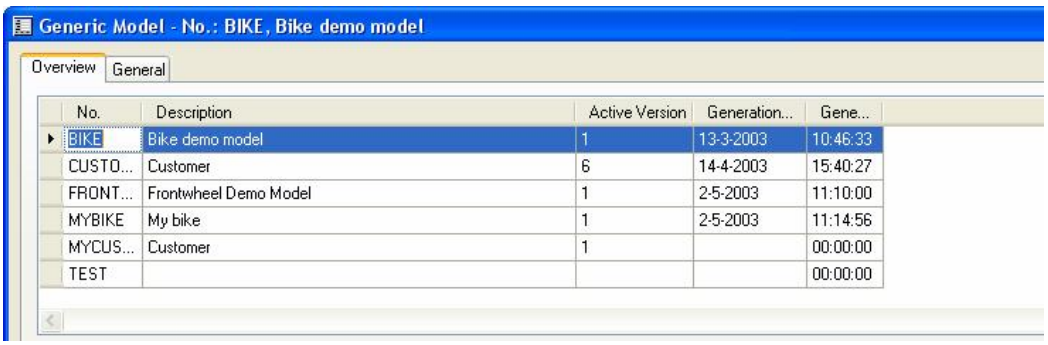


6.2 Various Data Structure Components

The foundation of a model lies within the data structure. In this section, we will explain the various components of the data structure that you will encounter while working with e-Con.

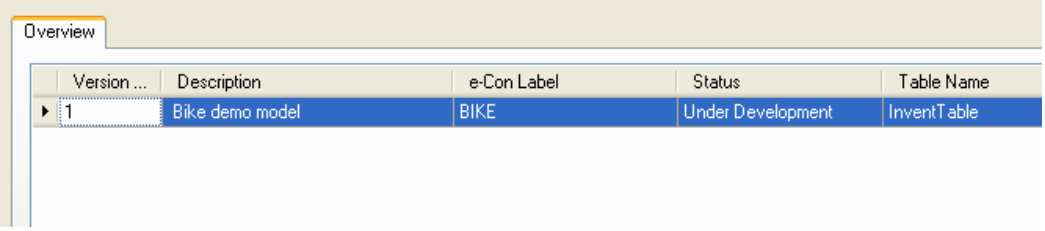
6.2.1 Model

Using the Generic Model form, you can lay the foundation for a new model. To do so, simply add a record with CTRL-N.



6.2.2 Version

You can insert a version name for the new model by going to the version lines.



You can make multiple versions from the same model. You can expand or change the structure of the model in the different versions. The changes will only visible in the Front End for the active version. To make a new version of a model, you must copy a model version and select the new version as the Active Version number. NOTE: Do this WITHOUT changing the number of the model itself. The description, however, can be changed in your new version.

This feature is very practical if you have a perfect working model, for example, and you want to test something in this model. Naturally, you don't want to risk messing up your perfect working model. Instead, you can copy this model to another version, and then do your testing on this newer version. If anything goes wrong, your original perfect working model remains unaffected.

Tips & Tricks: Always provide a clear description of the model, indicating the model's functionality. Then when you deal with different versions of the same model, you can use the description field to highlight any special functionalities which set that specific version apart from other versions.

6.2.3 Version Label

In the Generic Model Version List, you must enter a label for your version.

The label name you enter for this version will be used when generating an XML document. The XML document will be generated and saved under the name of the Version's label.

For example: in our BIKE Demo Model, we entered the name BIKE in the Label column. Upon generating the data structure, the XML document called "BIKE.XML" will be saved to disk. This XML document contains the data structure of your model.

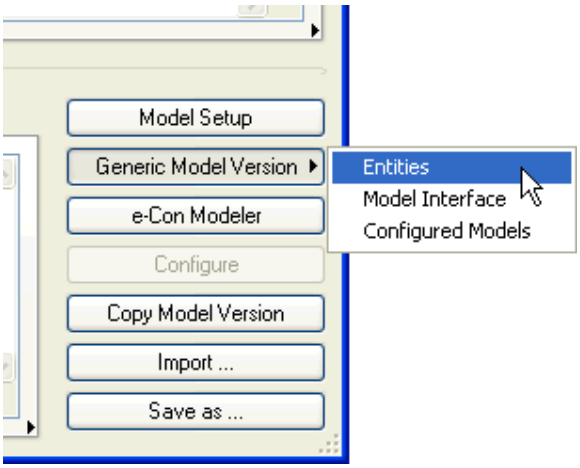
Tips & Tricks: You cannot use the same version label for different active versions.

6.2.4 Entities

You have to link a model to a specific table in Axapta. This linking is accomplished using the Entities. In effect, you are telling the model which table structure to use with all its fields, as well as what it is we are going to configure in e-Con, be it customers, items, or

something else. In the Bike Demo Model, a link (entity) has been made to the InventTable of the Axapta database.

Every Entity has a unique Label, which is used in the XML document. You can go to the Entities by clicking on the "Entities" button on the Generic Model form.



6.2.5 Entity Members

The Entity members are the same as the fields of a table, and make up what we call the data structure. Entity members are used to define the characteristics of the Entity. We distinguish among the following Member Types:

-Field Property: a link to a field of a Axapta table

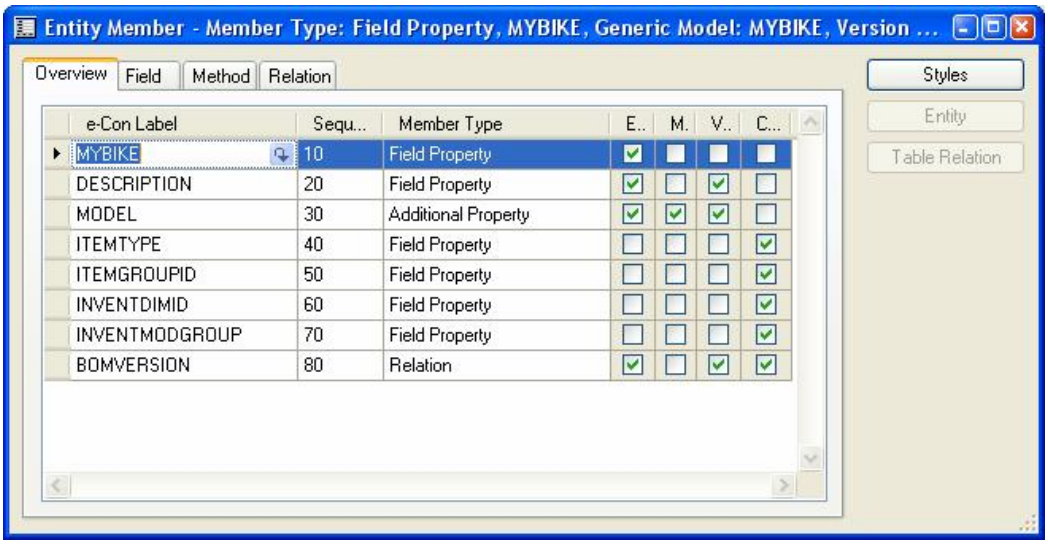
-Additional Property: a field that is not present in a Axapta database but is nonetheless needed in the Studio or the Front End

-Relation: used to designate a relationship to another Entity in the model (or other model). With relations the data structure is made, Item – BOMVersion – BOM lines.

-Display Method: a link to a display method of a Axapta table

Tips & Tricks: The Relation-"read only" is often used with a Dynamic Query. This provides a functionality for which you would normally have to use a connector to the database.

To go to the Entity form and from the Entities form, click on the "Entity Members" button.

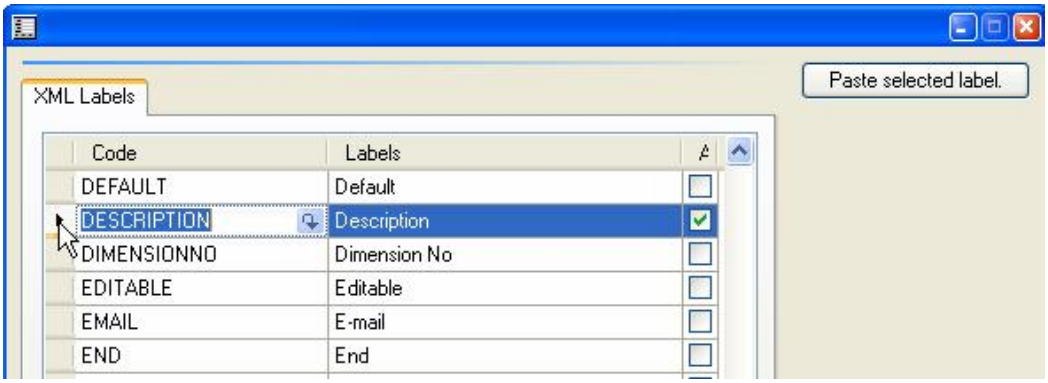


Every Entity member has also a unique Label. This Label is used in the e-Con Studio for compiling rules. End users, however, will see the "Description" in the Front End, which appears as a sort of question they must answer.

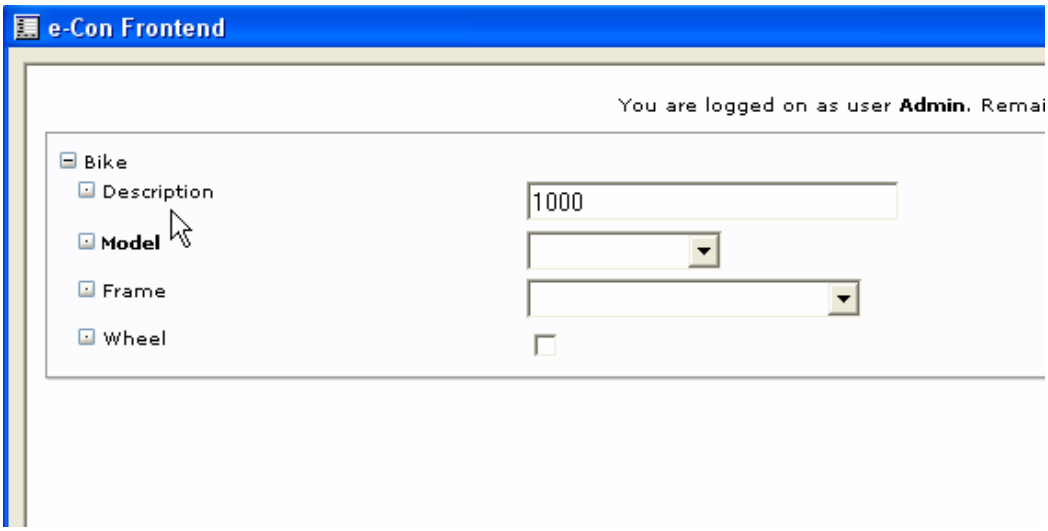
For example:

Label: DESCRIPTION

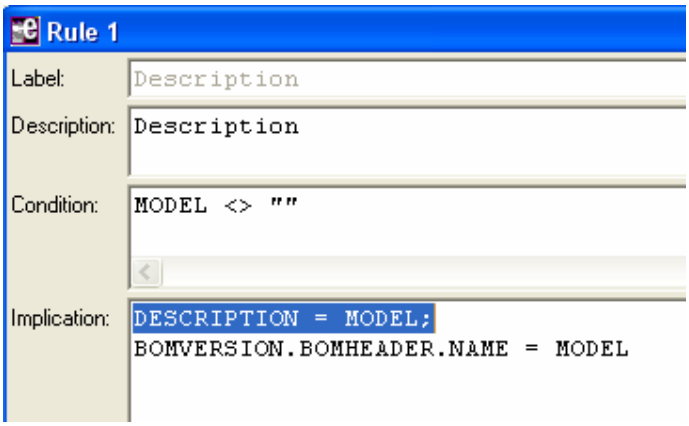
Description: Category



As you can see, it is the Description of the Label that appears in the Front End, as a sort of question to which the end user must provide an answer.

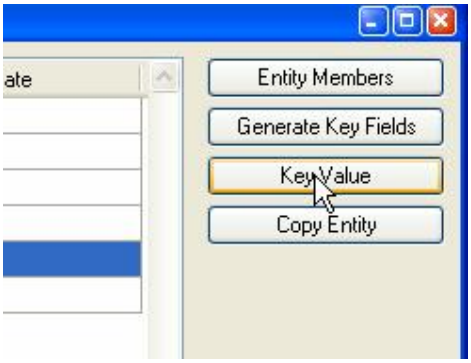


If you use the label DESCRIPTION in the rule engine, the Description of the label ("Description") will be displayed under the model structure (in section 1, below), but the Label name ("DESCRIPTION") (below) is used in the rule.

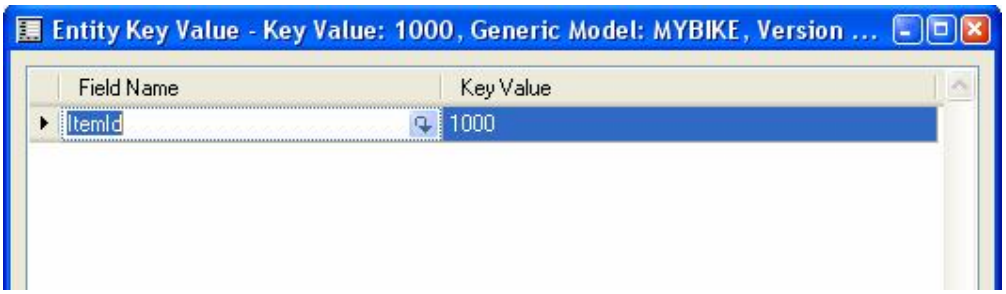


6.3 How to Define a Key Value

You can find the "Key Values" button on the Entity form.



A Key Value is a default table record (template) for the Entity that serves as the model for the Front End. The fields which are not entered in the Front End will be copied from the default table record you selected. The Key Value you define will also activate the Front End in certain sections of Axapta.



For example: In the Bike Demo Model, the item “e-Con Bike” is defined as a Key Value. This means that when you fill in a Sales Order and use the item “e-Con Bike” in the Sales Order Line, the Front End of the Bike Demo Model will be activated. See also chapter 4.4.

6.4 How to Change the Status of a Model

The status of a model can be any one of the following:

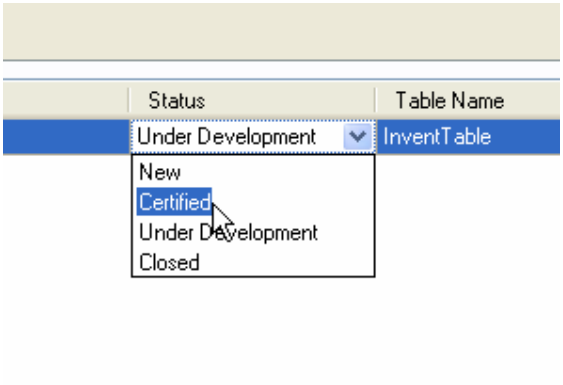
New: When a new model entered, the default Status is New. When a model has this Status "New", it cannot be used for configuration. Once the Status has been changed to "Certified", "Under development", or "Closed", the Status cannot be changed back to "New".

Under Development: This Status is used when making changes to the model; changes cannot be made if the Status is "Certified" or "Closed".

Certified: When the Status is "Certified", you can generate the data structure, or start up the Studio, or begin to configure.

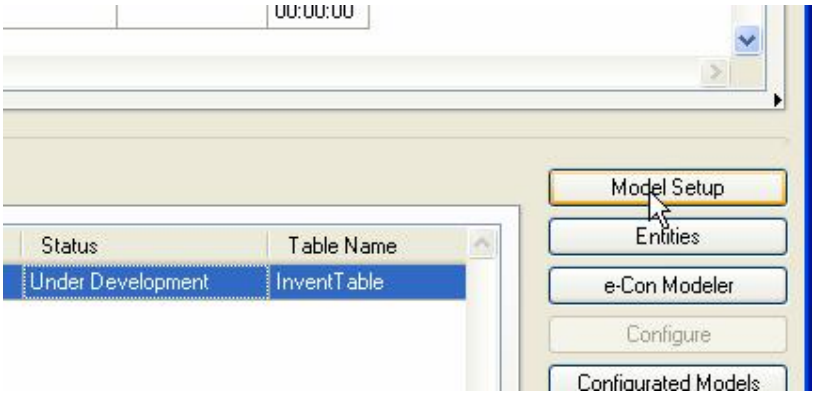
Closed: This Status is selected when a version has to be blocked for use. Use this Status for models that are obsolete.

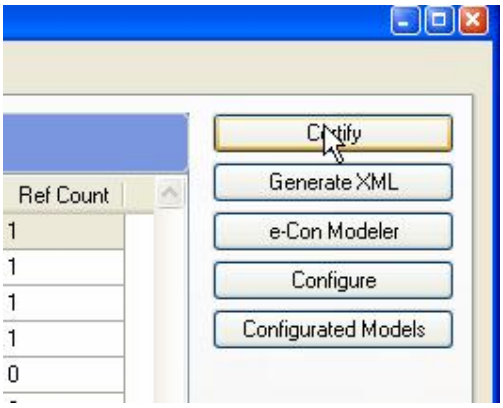
You can change the Status of a model by changing status on the version line.



In the Status column, you can click on the arrow to open a drop-down menu. Then you simply select the status you want, and it will appear in the Status column once you jump to another field.

In the e-Con Model form the version can also be certified. With the Model Setup button you can go to the e-Con Model form, that is only possible when the status is “Under Development”.





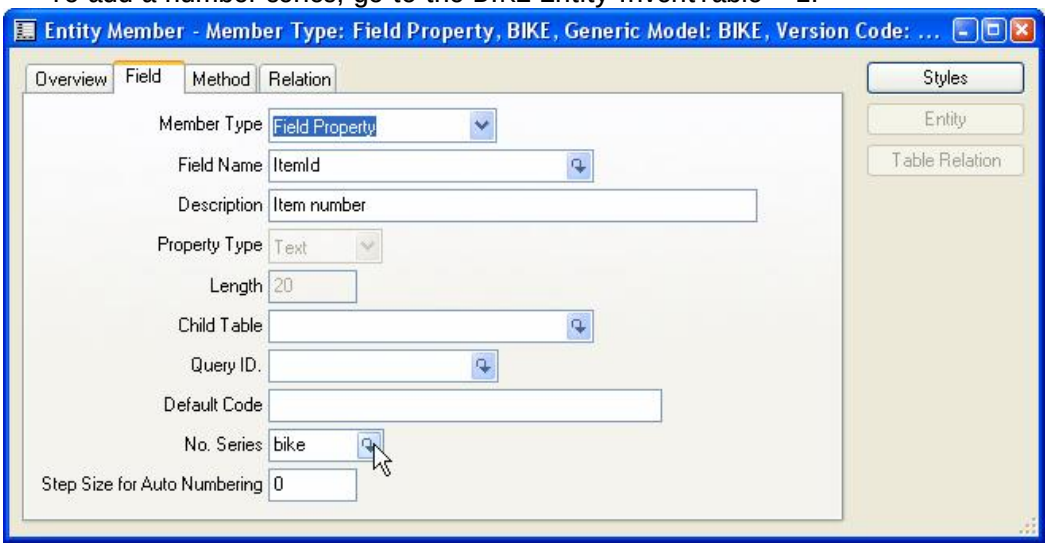
If there are problems with the data structure some error messages will be shown.

Read the errors and solve the problems.

6.5 How to Add a Number Series to Your Data Model

You can define a number series in e-Con, to be used for the Entity members you configure.

- To add a number series, go to the BIKE Entity InventTable – 1.



- Select the Field Property BIKE on the Entity Card. Go to the field labeled "No. Series" (Number Series).
- Use the lookup button to open the "No. Series" form.

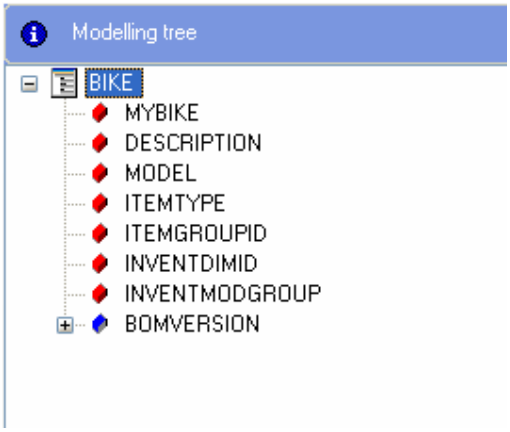
You can then select an existing Number Series from the list that appears.

If you designate the given Field Property as being invisible and not editable (by NOT check-marking these two fields), it will not be shown in the Front End. However, when the "Save" button is clicked, a number will be generated from the Number Series we defined for the Field Property and this will be saved in the Item table.

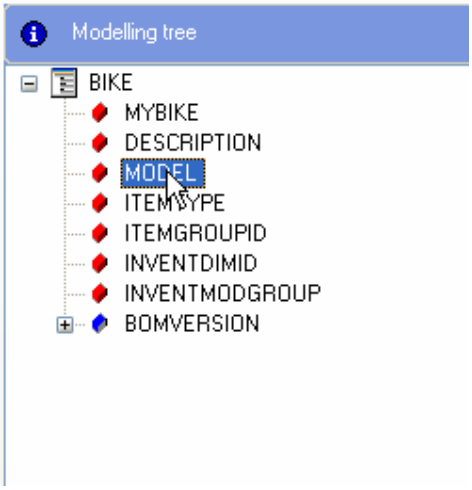
6.6 How to Change the Order of the Entity Members

e-Con allows you to change the order of Entity members for a particular Entity.

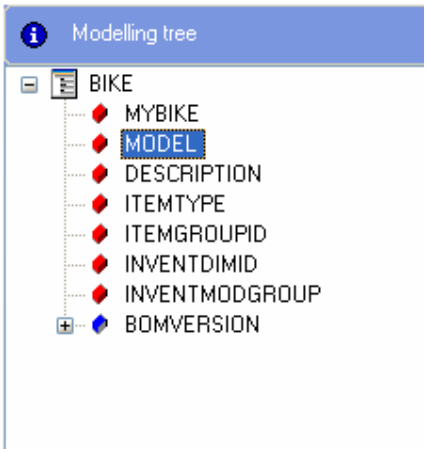
- Go to the e-Con Model form with the Model Setup button on the Generic Model form. Go to the Modeling tree.



- Select the entity member of which the order must be changed. (e.g. MODEL must be before DESCRIPTION)

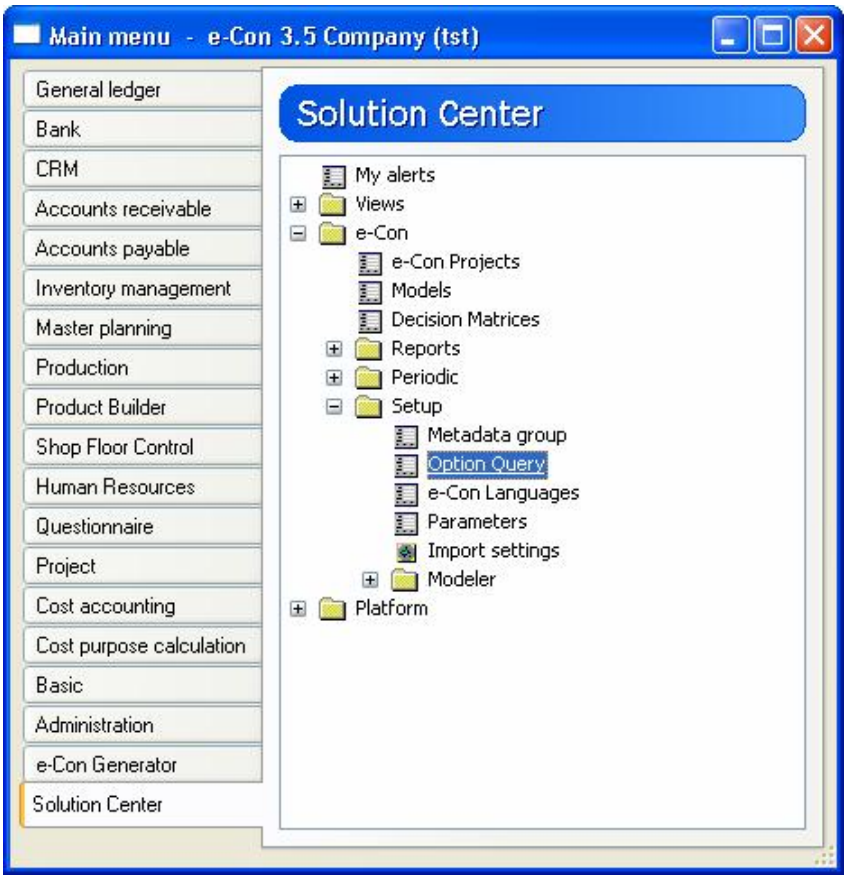


- With Ctrl ↑ place MODEL above DESCRIPTION.

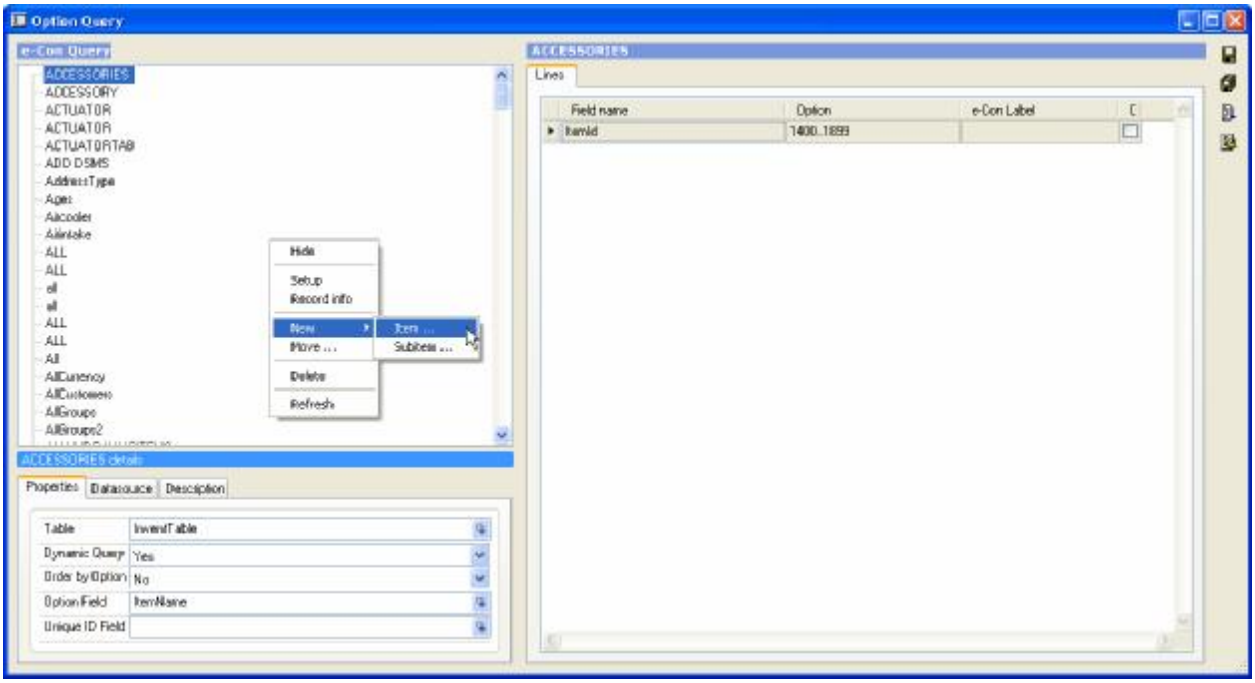


6.7 How to Define or Add a Query

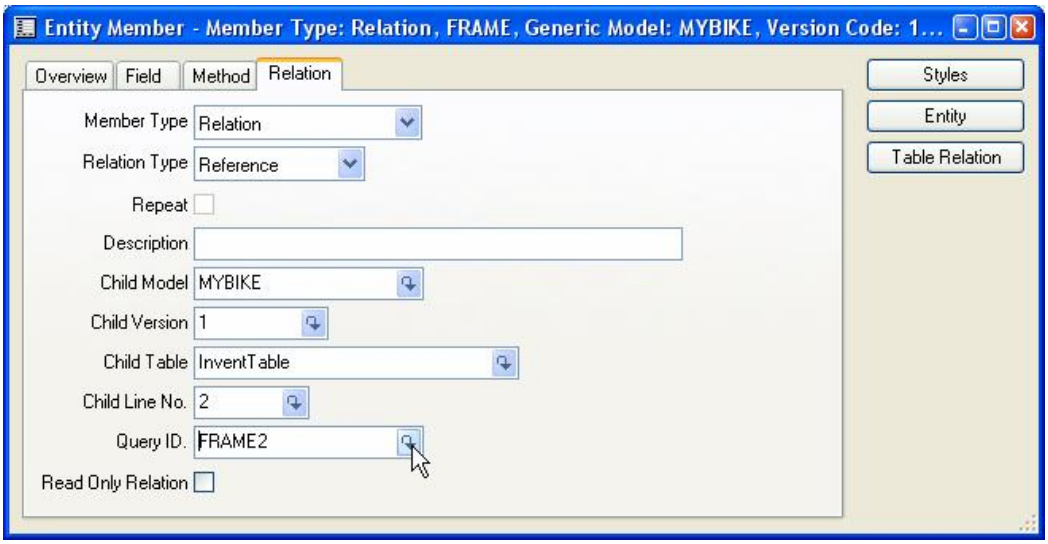
- Select "Option Query" in the e-Con Setup menu.



Using this form, a query can be created. Queries are used to select information from Axapta Tables or to provide selectable options which are not present in a Axapta Table. You can insert a query by selecting 'New item' or 'New subitem' from the context menu in the queries tree.



You can select an existing query by selecting the preferred query for the entity member. Go to the Entity Member form and select the query.

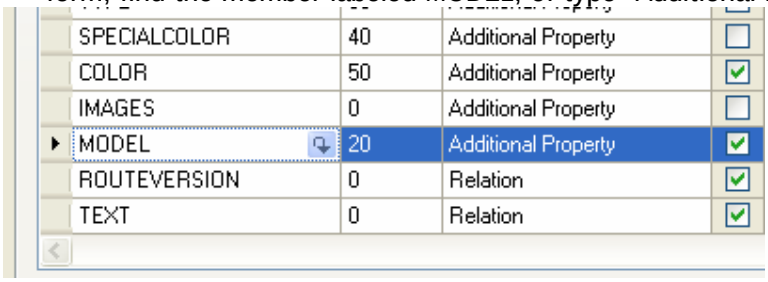


By selecting ‘Go to the maintable’ from the context menu of the ‘Query ID’ field the query table is selected.

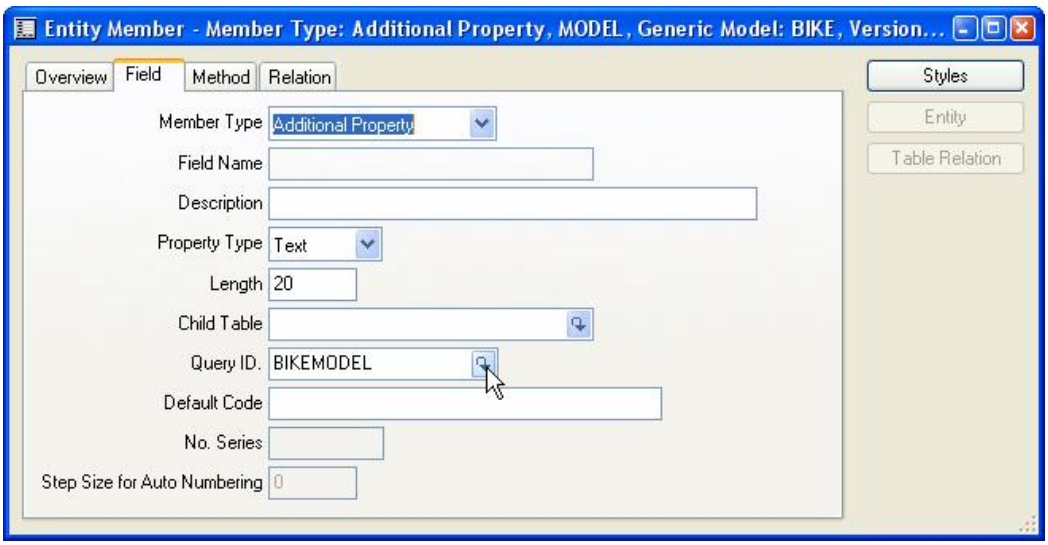
6.8 How to Define an Option List

In this section, we take you through the process of defining an Option List Query. In the Bike demo model, there are several different bicycle models for the end user to choose from. Let’s take a closer look at how this set of options, or "Option List" is built.

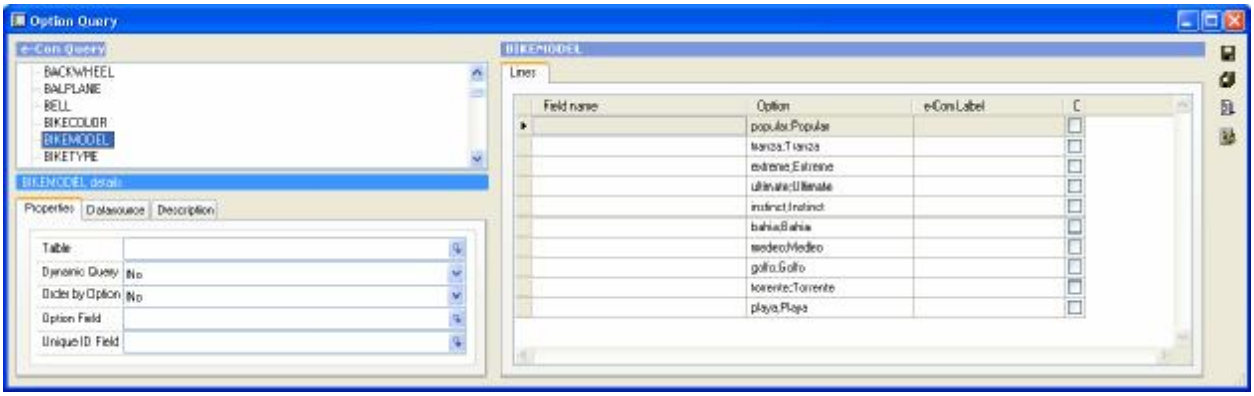
- First, select the "InventTable – 1" Entity on the Entities form, and then go to its Entity Members (by clicking on the "Entity Members" button). Once in the Entity Member form, find the member labeled MODEL, of type "Additional Property".



- On the Field tab you find the Query ID field with “BIKEMODEL” as query.



- To see the options of the “BIKEMODEL” query go to Setup - Option Query and select the query to look for.



- In the right pane, you can see how an Option Query is set up. There is no link to a table because the choices given are not available in a table. That's why Table Name and Field names are empty. In the "Option" column, the choices the end user can pick from are listed. The values in this column must be constructed in one of two ways:

a. popular;Popular

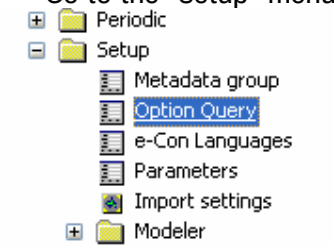
b. 1;Popular.

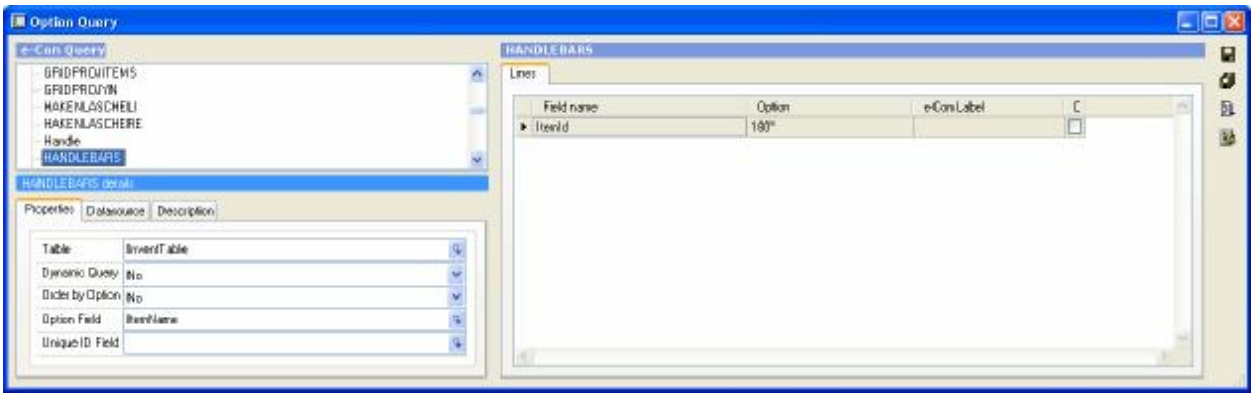
The first term is used in the Studio ("popular", if you use method a., "1", if you use method b.); the second will be displayed in the Front End. Using method b., for example, means "1" is used as a label in the Studio. This carries the advantage of a short name in the Studio, but method a. can be less confusing, since you can see directly what term it will display in the Front End.

6.9 How to Define a Query Related to a Axapta Table

In this section we will show how a query related to a Axapta table is defined.

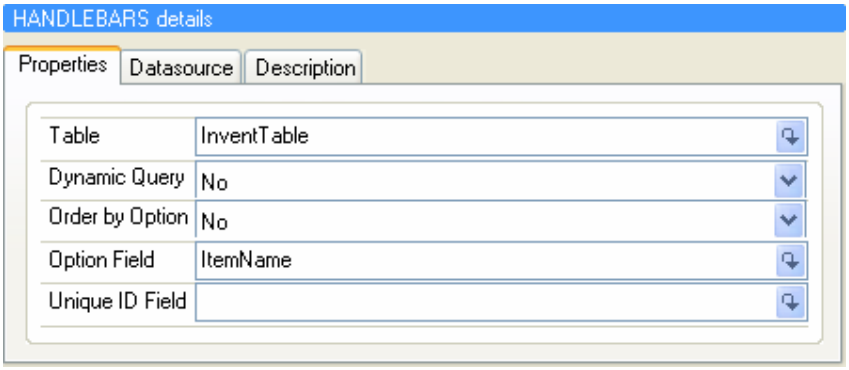
- Go to the "Setup" menu, and select the "Option Query" option.



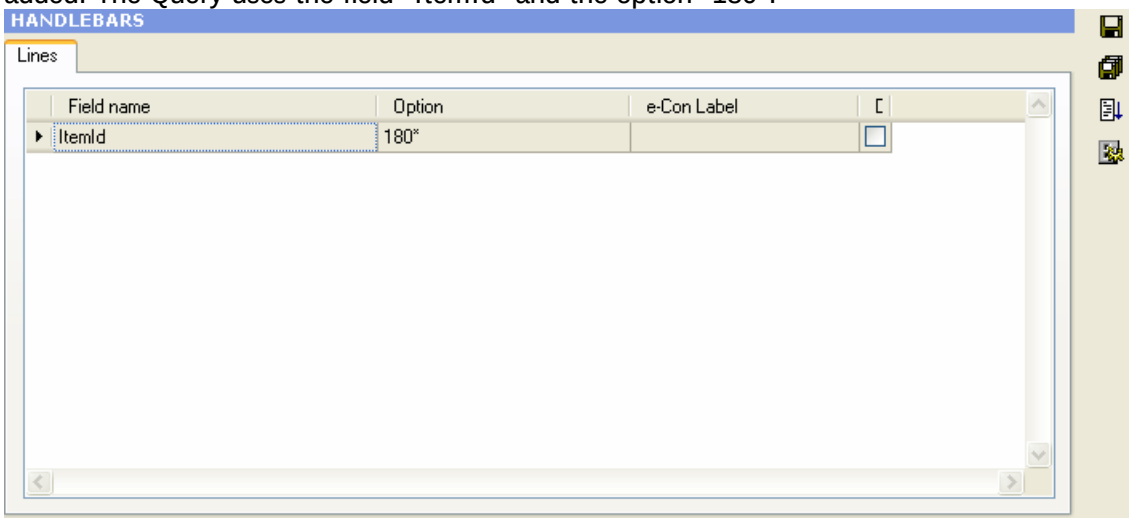


- Select Query ID "HANDLEBARS"

- At the properties tab of the query details you can see this Query is linked to Axapta InventTable, the Items table.



- If you link a Query to an Axapta table, you can use the fields which are available in this table to add an additional filter for the query. In the Field Name column, you select the fields from the table you want to use and in the Option field the filter is added. The Query uses the field "ItemId" and the option "180".

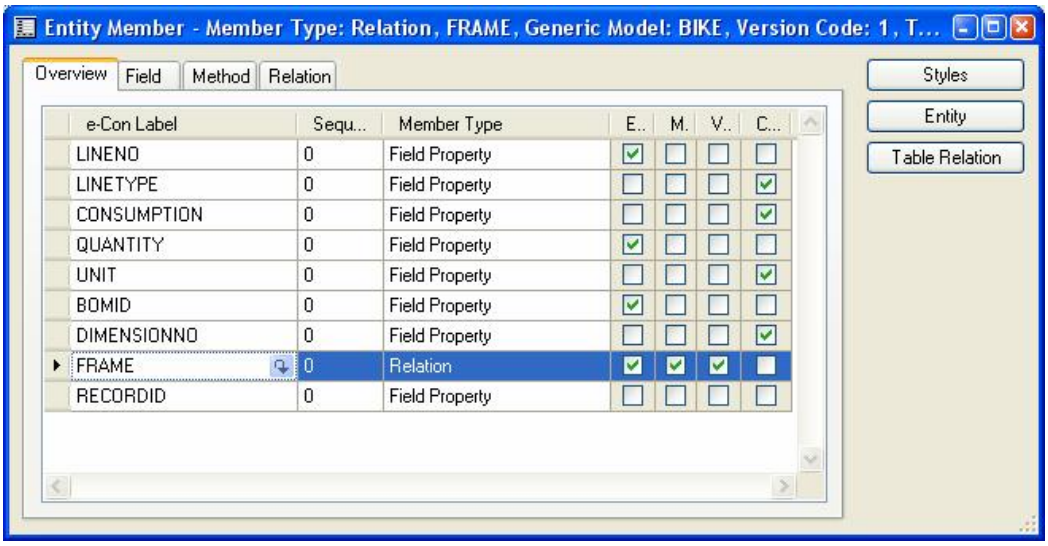


6.10 How to Define a Dynamic Query

A Dynamic Query is not generated in the data structure, but will be activated during configuration when triggered by an "Query" rule.

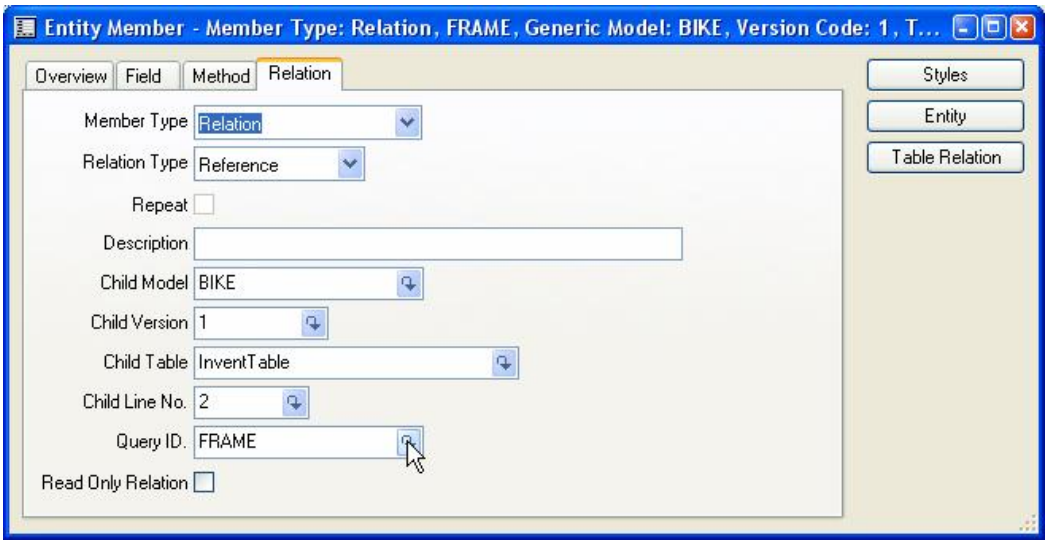
In this section, we will show how a Dynamic Query is built.

- Go to the Entity BOM – 1, and then go to the Entity Members (by clicking on the "Entity Members" button).



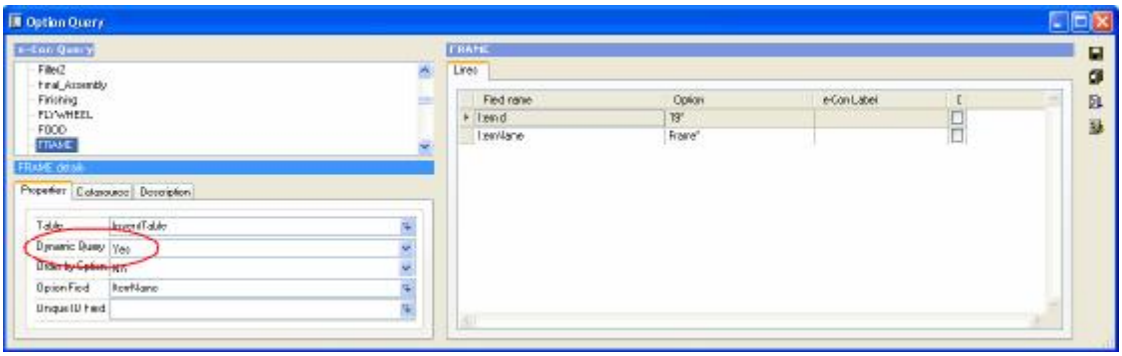
- Select the Entity member labeled "FRAME"

- On the “Relation” tab the query is added. In this case the FRAME query.



- To see the “FRAME” query go to Setup - Option Query and select the query to look for.

- Dynamic Query is set to Yes. So this is a dynamic query. All the filters are set the same as for not dynamic queries.

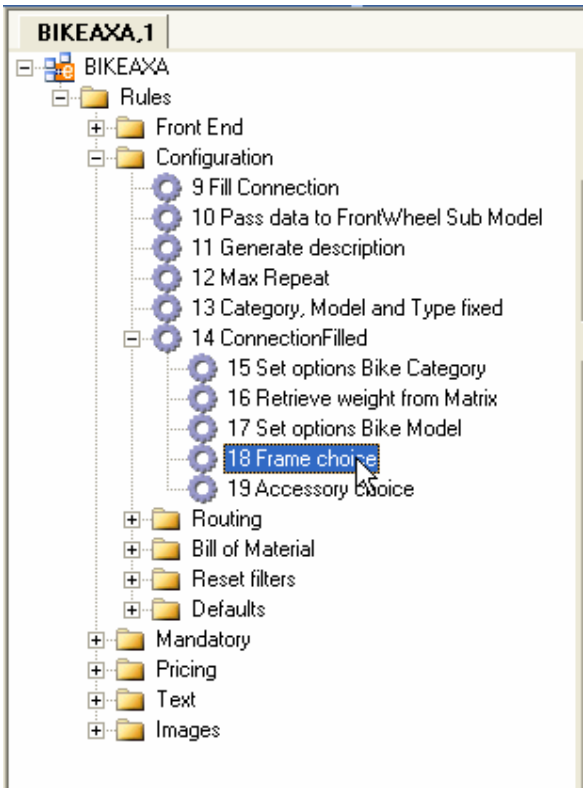


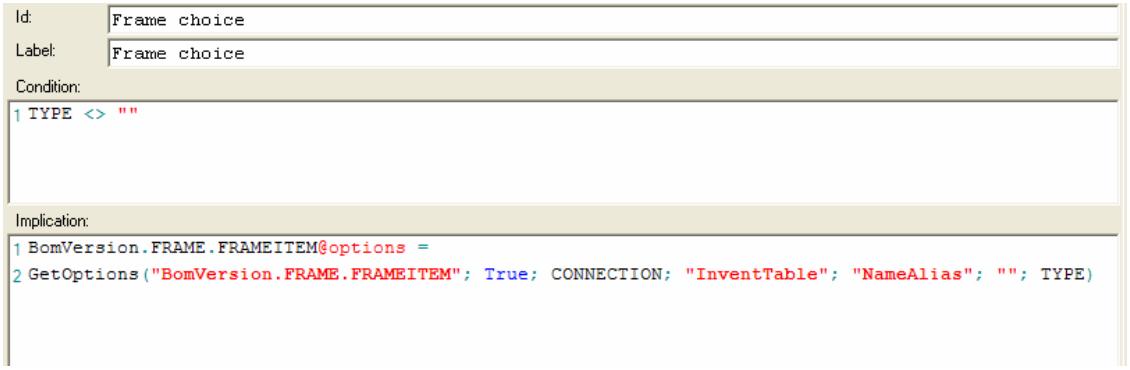
- All items with ItemId starts with 19 and ItemName starts with Frame are the result of this query.

This query looks like a standard Option Query. The only difference is that the field labeled "Dynamic Query" is yes. The rest of the dynamic section is provided for in the Studio rules.

- Go to the e-Con Studio

- Select "Rules", go to "Configuration" and select the rule "Frame choice"





- This rule activates the function GetOptions. This function read records from the "InventTable" using the filter "NameAlias" equals TYPE.

6.11 How to Add a Style

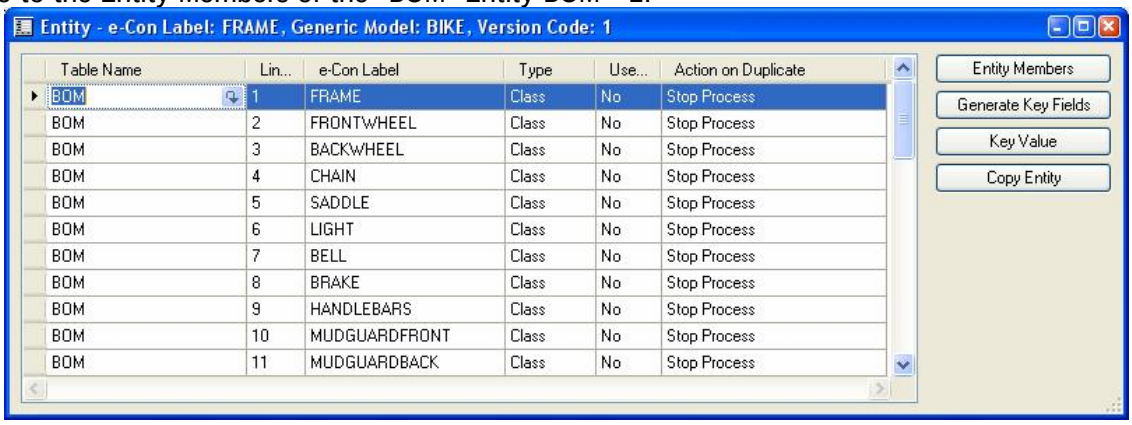
In this section, we will show you how to define a style. The various kinds of styles will be described.

6.11.1 Select First ("selectfirst")

The style labeled "selectfirst" will always select the first value available. The Value Description must be set to True in order to activate the "selectfirst" style. This style is

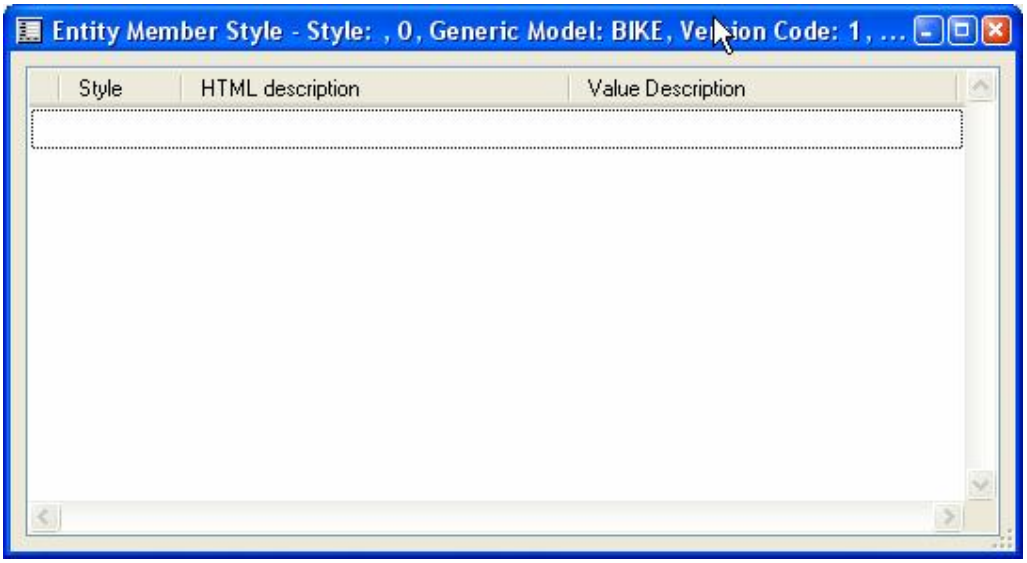
often used in combination with the "orderby" style. To demonstrate the functionality of the "selectfirst" Style, we will use the Bike Model.

- Go to the Entity Members of the "BOM" Entity BOM - 1.

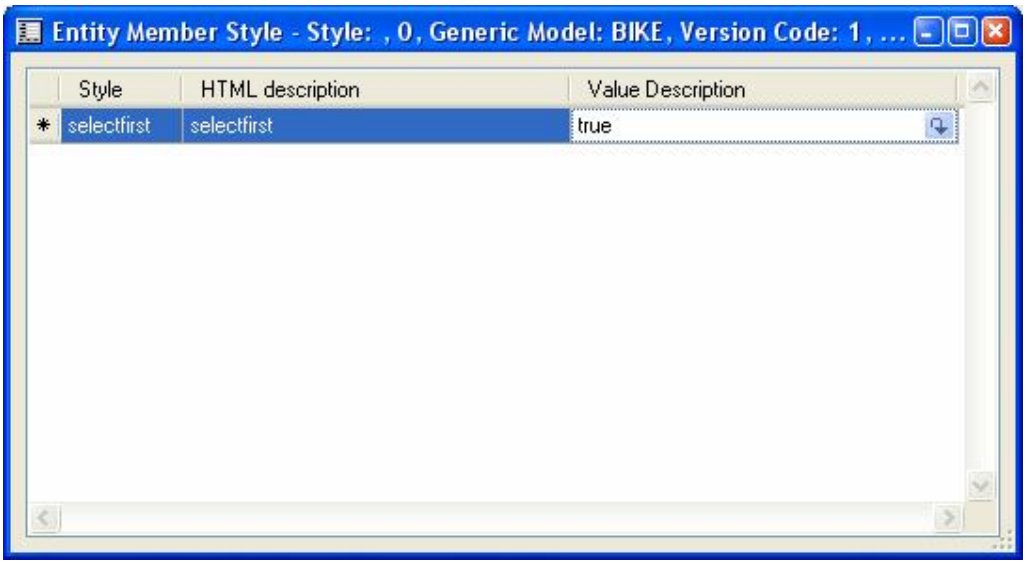


- Select the line for the Entity member FRAME (so that it becomes highlighted), and then click on the "Styles" button.

The following screen will appear:



Add the style with CTRL-N.



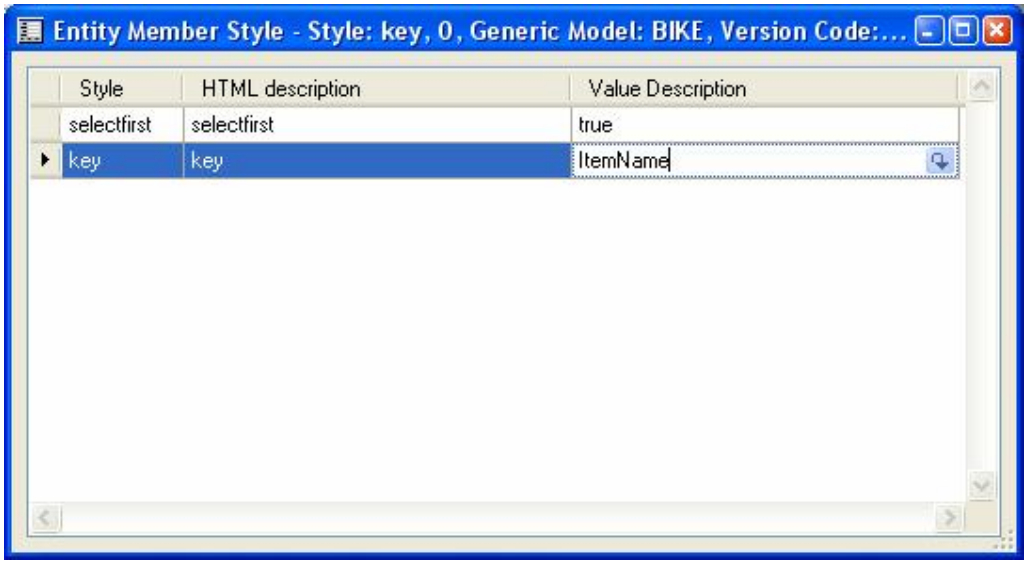
The "selectfirst" style is activated in the column "Value Description" by setting it to "true". The result of this style choice is that the first frame is selected.

6.11.2 Key (“key”)

The "key" style displays the information requested in a previously defined order.

To demonstrate the functionality of the "key" style, we will use the Bike Model once again. (NOTE: the "key" style is the same as the "orderby" style.)

- Select the "BOM" Entity BOM – 1, and go to its Entity Members
- To select the style for the Entity member labeled "FRAME", select the "Styles" button

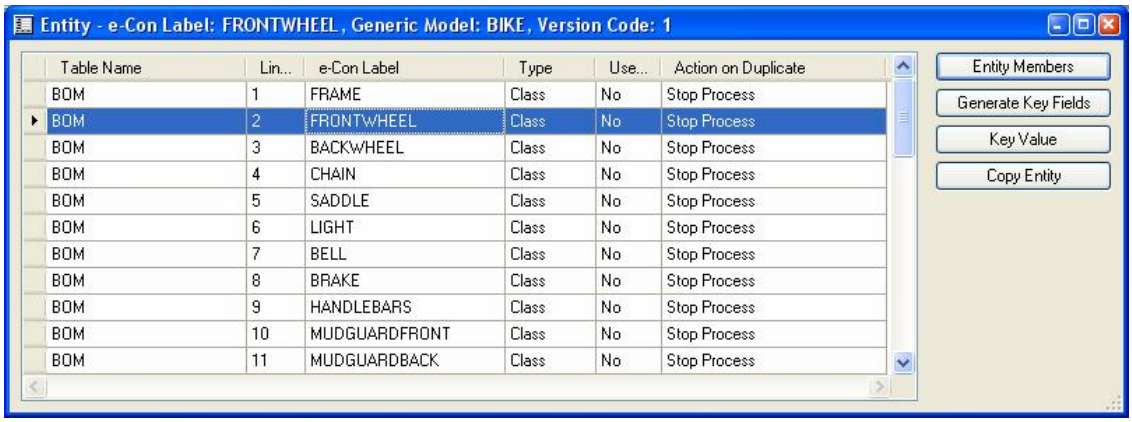


This "key" style is activated for field ItemName. In the column labeled "Value Description", you enter the field you wish the "key" style to have an effect upon. (NOTE: There is no lookup button in this column with which to find field numbers.)

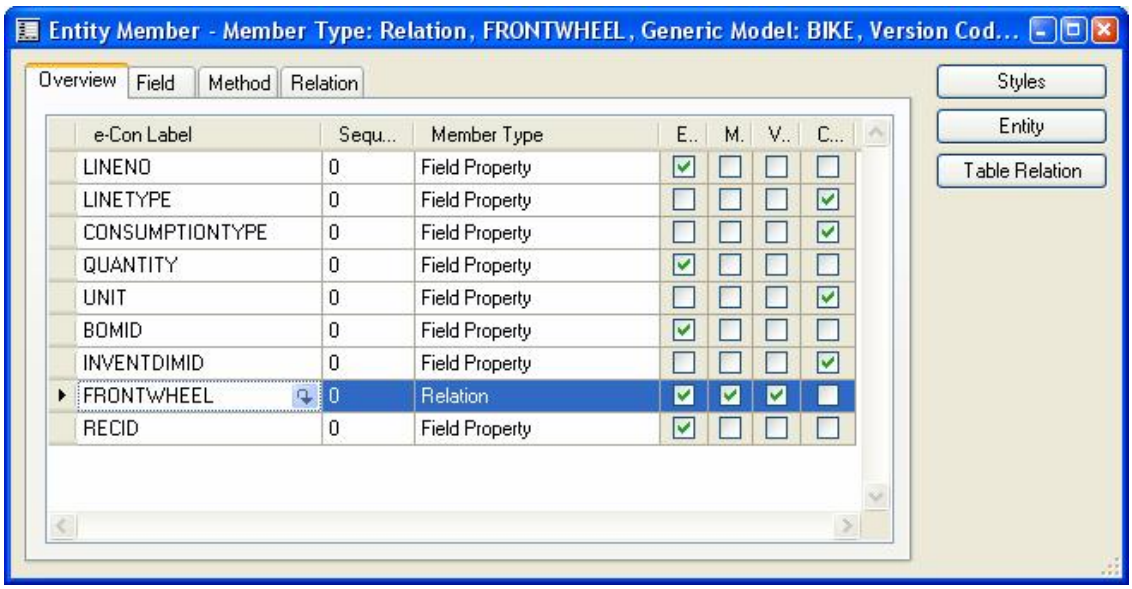
6.11.3 Auto-expand ("autoexpand")

The "autoexpand" style will automatically expand a relation which has been set by a rule, causing it to appear in expanded form in the Front End.

6.11.4 Go to the "BOM" Entity BOM-1 labeled "FRONTWHEEL", and go to its Entity Members

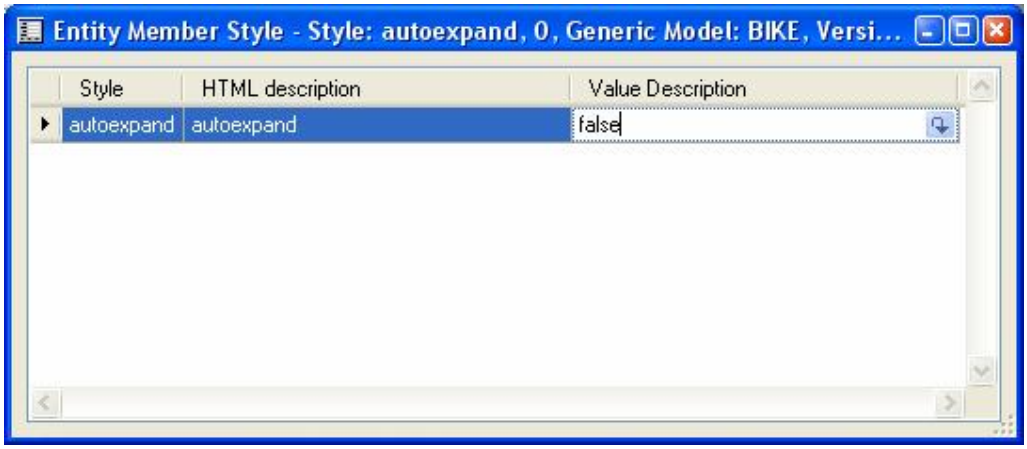


6.11.5 Select the Entity member labeled "FRONTWHEEL" (so that the whole row is highlighted)



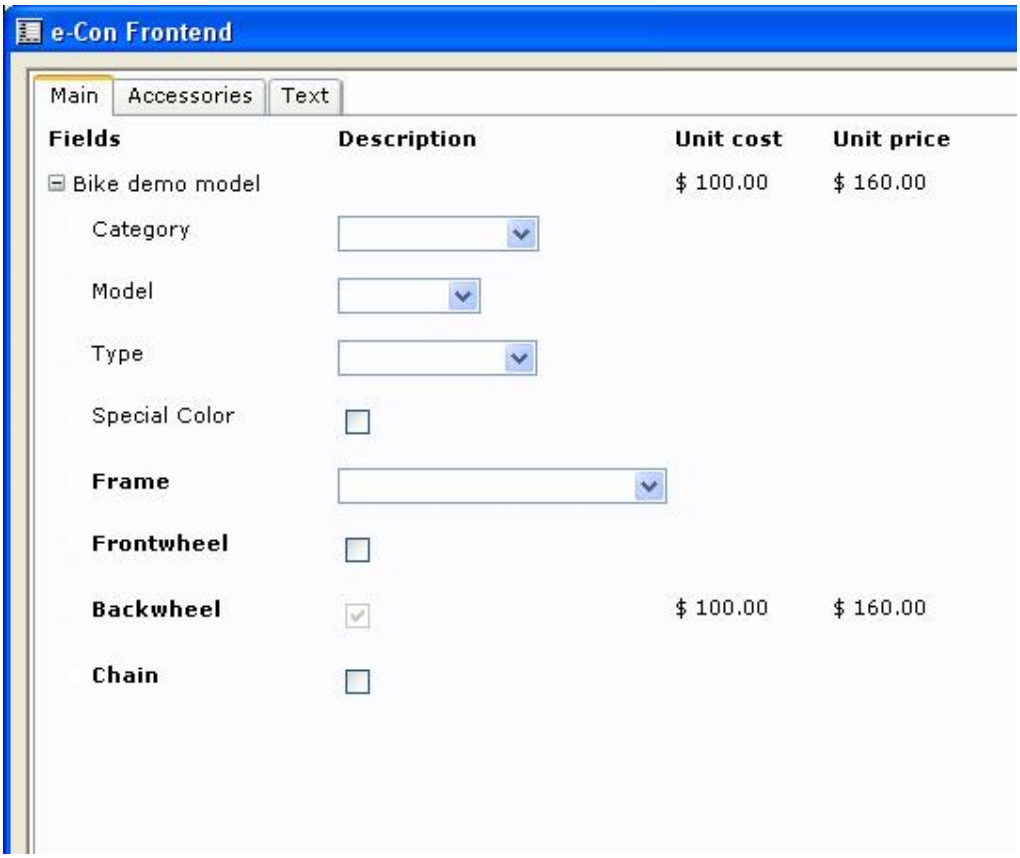
- Go to the Entity Member Styles form by clicking on the "Styles" button

The following screen will open:

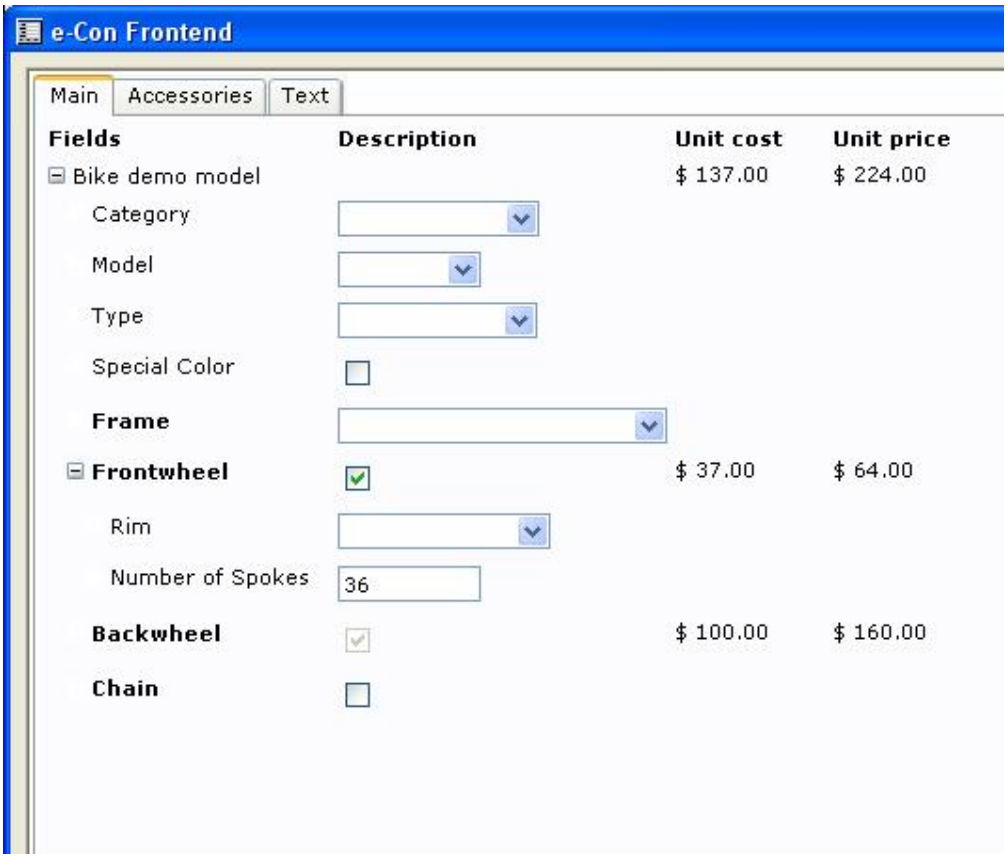


- Add the desired style by using the lookup button in the "Style" column. The value for an "autoexpand" style is true or false. In this case, we enter false.

The "autoexpand" style will automatically expand a relation which has been set by a rule, causing it to appear in expanded form in the Front End. In the case at hand, this results in an expansion of the Front Wheel options in the Front End.



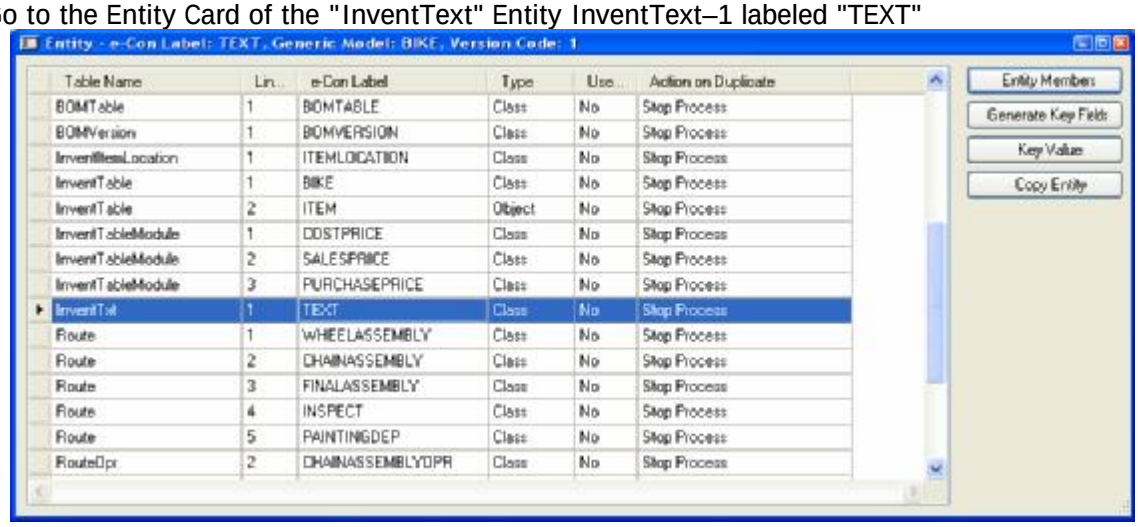
When the Front Wheel box is check-marked, the fields "Rim" and "Number of spokes" will also be shown.



6.11.6 Multi-line ("multiline")

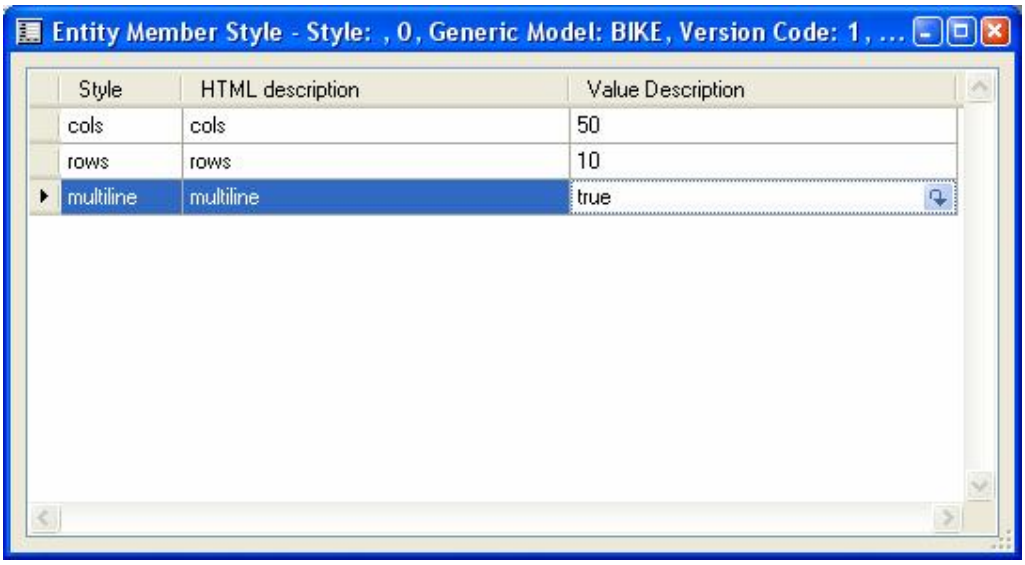
The "multiline" style option enables you to display an Entity member as a text box with several lines.

- Go to the Entity Card of the "InventText" Entity InventText–1 labeled "TEXT"



- Go to the Entity members.
- Select the Entity member labeled "TEXT" (so that the whole row is highlighted), and then click on the "Styles" button.

The following screen will appear:



There are 2 options for the "multiline" style which you can enter in the "Value Description" column.

- True: Entering "true" in the "Value Description" column will activate the "multiline" options
- Text: Entering "text" in the "Value Description" column will activate the "multiline" options and display them as a text box. The entered lines will be adjusted to fit the size of the field in the Axapta Table and saved in Axapta.

Tips & Tricks: This "multiline" style requires a field in Axapta with the capability of storing lines of text.

Below is the resulting text box which appears in the Front End:

6.11.7 Columns ("cols")

Both the "cols" style and the "rows" style are always used in combination with the "multiline" style. In the previous section we explained the "multiline" style. The "cols" style will determine the width of the multiline text box. In our example, the quantity entered in the "Value Description" field is 50, which indicates our text box is 50 characters wide (so in effect, you have 50 little 'columns').

6.11.8 Rows ("rows")

As described above, the "rows" style is used in combination with the "multiline" and "cols" styles. The "rows" style determines the height of the text box. In our example, the quantity entered in the "Value Description" field is 5, which indicates our text box is 5 rows (or lines) high.

6.11.9 Image Source ("imgsrc")

This style is only used in combination with images. With this style, the source of the images can be recorded. This "source" is the name of the image and the path to where the image file can be found. There are two ways to do this:

- By notating the complete path, like: "C:/My Images/Bike.jpg"
- By notating the path relative to e-Con;[yvo] the first part of the path then is always the path where e-Con is installed. For example: "./images/bike.jpg". (The complete path in that case would be "C:\eCon\images\bike.jpg".)

6.11.10 Image Visible ("imgvisible")

This style is only used in combination with images. With this style it is possible to make an image visible (or invisible). An image is always visible by default. To make the image invisible, assign a string with value "false" to this style. To make the image visible again, assign "true".

6.11.11 Image Position ("imgposition")

This style is only used in combination with images, and can have four values: “Left”, “Top”, “Right” and “Bottom”. With this style the position of the image is set. When “Left” or “Right” are chosen the image area will be shown at the left or right hand side of the Front-end without overlapping the questions and answers fields. With “Top” and “Bottom” the image area is shown above or under the Front-end.

6.11.12 Image Left ("imgleft")

This style is only used in combination with images. After determining where you want to have the image (right, left, bottom or top) on the Front-end with “imgposition” style, you can go further and determine where exactly you want the image to be positioned in relation to the left margin of the image area.

6.11.13 Image Top ("imgtop")

This style is only used in combination with images. After determining where you want to have the image (right, left, bottom or top) on the Front-end with “imgposition” style, you can go further and determine where exactly you want the image to be positioned in relation to the top margin of the image area.

6.11.14 Image Height ("imgheight")

This style is only used in combination with images. With this style, you can determine the height of the image. If you do not specify the Image Width (see below), this style will scale your image to the proper dimensions.

6.11.15 Image Width ("imgwidth")

This style is only used in combination with images. With this style, you can determine the width of the image. If you do not specify the Image Height (see above), this style will scale your image to the proper dimensions.

6.11.16 Template

With this style a key value of an entity can be changed. For instance the key value of the Bike Model is item e-Con Bike. Item fields that are not changed by configuration like Posting groups will be used from the key value item. With the Template style the item to be used as the key value can be changed so the configured item will be based on another item with for instance other Posting groups.

6.11.17 Default

With this style the behavior of the default attribute can be changed. When the value “init” is assigned to this style, and this style is applied to a member with the attribute @default, the default value is only used when the member does not have a value. In other words: The default is only set when the member is empty.

6.11.18 URL

When this style with the value “true” is applied to a member the label of this member will act as a hyperlink. Simple clicking this label will activate the hyperlink. When this style is applied to a member displayed in a column, the content of the column will also act as a hyperlink.

For example:

Applying the style URL to the member supplier, where supplier has the value http://wwwto-increase.com, will act in a hyperlink to the web site of To-Increase.

6.11.19 Zoom

With this style it is possible to make additional properties available in a pop-up screen in the Front-end. Using this style will add a button to the option. With a simple click on this button, a pop-up screen including all additional properties will appear. The user can then make its choice form this pop-up screen.

For more information see section “How to add a pop-up screen”.

6.11.20 ZoomColumn

This style is aways used in combination with the former style Zoom. The style “ZoomColumn” has to applied, with value “true”, to the properties that must me be visible in the pop-up screen. Only the properties from a reference with this still style will shown up in the pop-up screen.

For more information see section “How to add a pop-up screen”.

6.11.21 Notused

With this style the rules of the member are not executed when the input is changed in the Front End. If the rules are activated by another input change the rules are executed afterall. So with this style the execution of the rules can be delayed. This for improve of performance.

6.11.22 OptionInfo

In a pop-up screen it’s possible to show bullets for:

- the default option;
- the current selection;
- the former selection.

The color of the bullet indicates one the tree possibilities. With the style “OptionInfo” with value “true” assigned to a member or reference having options, this functionality will be

available. However it's mandatory to use the style "zoom" as well to show the pop-up screen.

For more information see section "How to add a pop-up screen".

6.11.23 Readonly

When this style is applied to a member this member is 'read-only'. That means that it's not possible to change the value.

When a field is read only it appears in light gray.

6.11.24 Radio

When this style is applied to a member with value "true" the options of the member are shown as radio buttons.

6.11.25 Reset

This style can be used to reset a member to a certain value by starting up the e-Con front end. Especially useful when a certain member has to be reset by reconfiguring.

The value added to this style becomes the member by starting up the eCon front end.

6.11.26 Customerselect

This style can be used add special functionality like a calendar, calculator and rich text editor to a member in e-Con. By applying this style an assist button is added to that member. Clicking this assist button lets pop up the calendar, calculator or rich text editor.

Apply the value "calendar" to the style to add the calendar, the value "calculator" to add the calculator and "editor" to have the rich text editor available.

For more information see section "How to add a Calculator, Calendar or Rich Text Editor to a member".

6.11.27 Initline

This style is useful when for example sales lines are added with eCon. The style init line triggers the "auto split key" functionality from Axapta. The value of the style is the initial line number from where with the aid of the "auto split key" functionality the first free sales line number is found.

6.11.28 Delete

With the aid of this style it's possible to delete a repeatable entity. When an entity is repeatable a button "new" automatically appears. With that button you can add new entities. However standard there isn't a button to delete a repeatable entity. By applying this style, with value 'true', to the repeatable entity an extra button is added to that member. Clicking this button will delete the particular entity.

6.11.29 Controlclass / Labelclass

With the aid of those styles it's possible to change the style of the 'questions' and 'answers' in the e-Con Front End. With the style 'Controlclass' the style of the 'answer' can be changed, the style 'Labelclass' is needed to change the style of the question.

Several styles are standard available, however it is also possible to add your own custom style to the e-Con style sheet. The name of the style has to be applied as value of the style 'Controlclass' or "Labelclass' to have that particular style applies to the member.

In the figure above various styles are applied to several members by means of the style

'Controlclass' and 'Labelclass'.

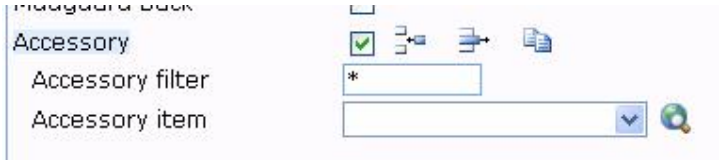
6.11.30 Format

With the aid of this style the format of an e-Con Field in the UI can be controlled in a way that texts or signs can be added in front of a field or behind a field. This can especially be used when a unit of measure must be displayed behind a field as shown in the figure below.

In the value assigned to the style "{0}" indicated the property. That means that when a certain character of text must added behind a field, the value must start with "{0}" followed with the character of text to shown. Displaying text in front of a field means that first the text followed by {0} must be keyed in.

6.11.31 Copy

With the aid of this style it's possible to copy a repeatable entity. When an entity is repeatable a button "new" automatically appears. With that button you can add new entities. However standard there isn't a button to copy repeatable entity. By applying this style, with value 'true', to the repeatable entity an extra button is added to that member. Clicking this button will copy the particular entity.



6.11.32 Select Single

The style labeled "selectsingle" will auto select the value when it's the one and only value left. In other words: When there is just one option left in a list this option will be selected automatically when this style is applied.

6.11.33 Columnwidth

With the aid of the style "columnwidth" the width of a column in the e-Con front end can be set up. This style will overwrite the width set up at the column. With this style the columnwidth can be dynamically set up in a rule. Make sure that properties exists at the top level of the model representing (equal to) the column id.

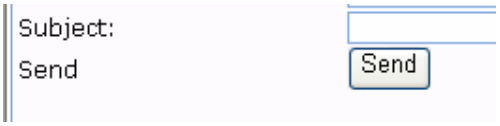
In the example the width of the column 'QUESTIONS" has a width of 200 at tab "General" and 100 at all other tabs.

Implication:

```
1 QUESTIONS@style.columnwidth =
2 if TAB == "General"
3 then 300
4 else 200
5 end if
```

6.11.34 Inlinebutton

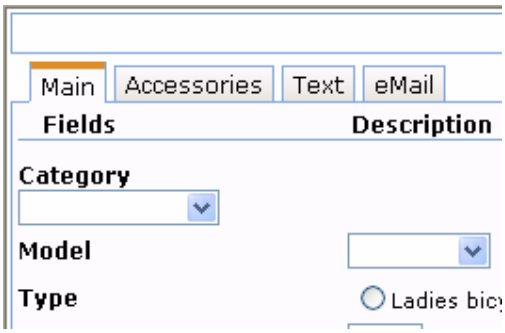
The style labeled "inlinebutton" will present a property as a button in the e-Con front end. In e-Con rules a tasks, like print or email, can be assigned to such a button.



6.11.35 Labelposition

With the aid of the style "labelposition" the position of the label of an e-Con UI field can be determined. The applicable values are:

- right: the label is shown right from the field (Default behavior)
- left: the label is shown left from the field
- bottom: the label is shown underneath the field
- top: the label is shown above the field
- none: the label is not shown at all



6.11.36 Valuebutton

The style labeled "valuebutton" will present a Boolean property as a button in the e-Con front end. Clicking the button will tokkle the value.

6.12 How to Include Another Data Structure

You can include or link to another model or data structure using e-Con, the so called sub models. To do so, a number of steps are required:

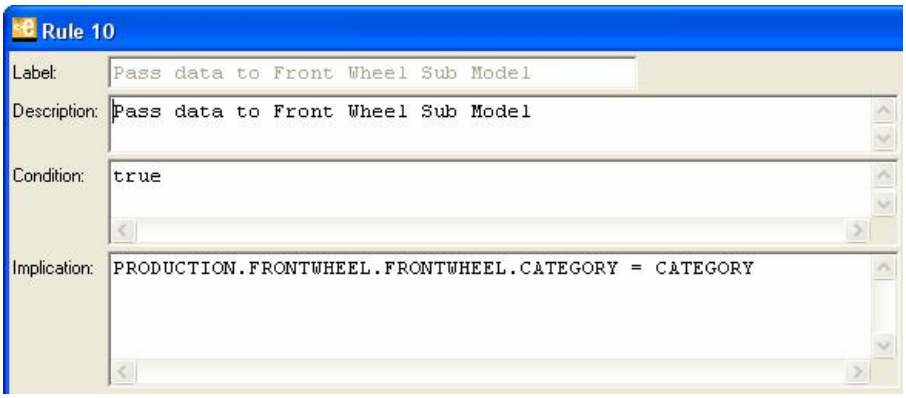
- Create an interface model
- Select the interface model for the to included sub model
- Set up the link to the sub model in the data structure of the main model.

6.12.1 Set up of an interface model

An interface model is needed for each sub model. The interface model is, as it is in the name, the interface or connector between a sub model and the main model. (main model is the model including the sub model).

All properties from the sub model used in the main model must be part of the interface model.

As an example the Bike demo model. In the bike demo model a sub model is available for the front wheel. In this front wheel the default type for the rim depends on the Category of the bike. Since the category is chosen in the main model Bike this information should passed down to the sub model front wheel. Here fore a property Category is added to the front wheel model and a rule is added in the bike model to pass the Category to the front wheel model.



Besides that category must be part of the interface model as well.

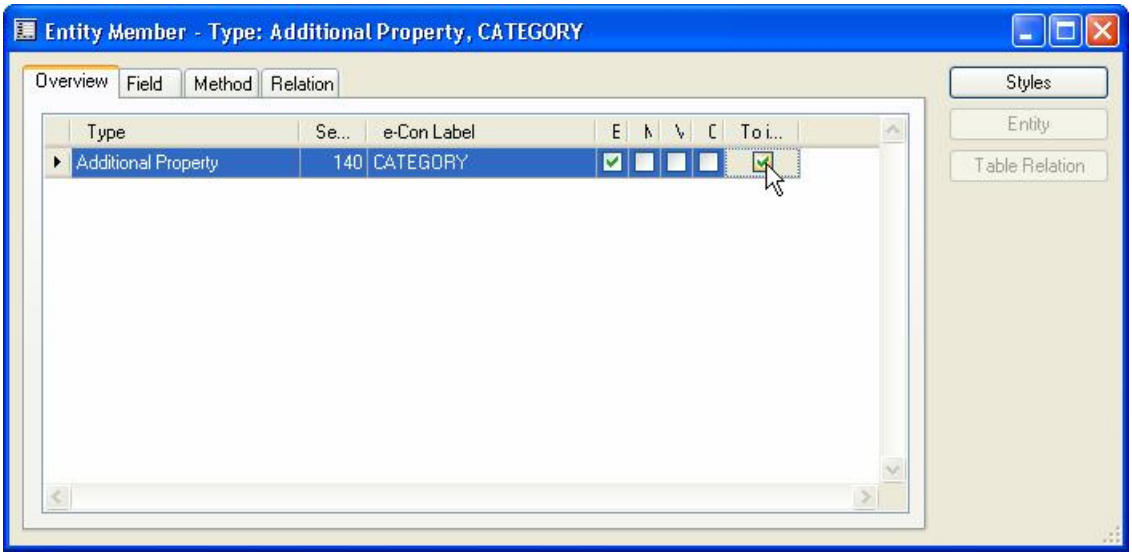
The type of the Rim of the front wheel is used in the Bike. A rule is available in the bike to generate text as specifications from the configured bike. Since the type of rim is part of these specifications as well information of the rim should be available in the bike model.

Two properties exists in the interface model for the front wheel:

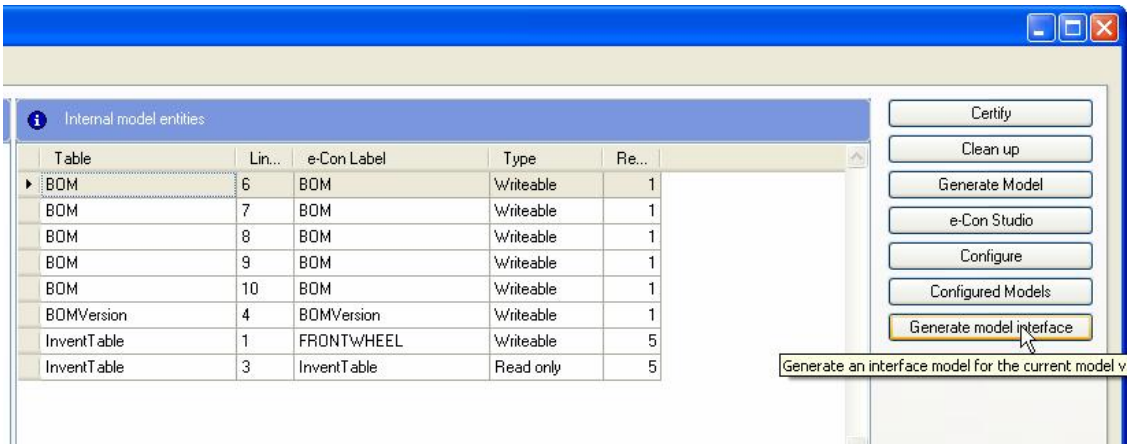
- Category
- Rim

To set up such a interface model the next steps must be performed:

- Open the submodel and check the field 'To interface' for every property that must be part of the interface. In the example for both properties 'Category' and 'Rim' this field is checked.

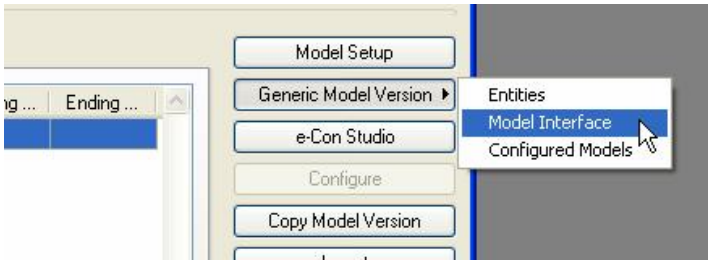


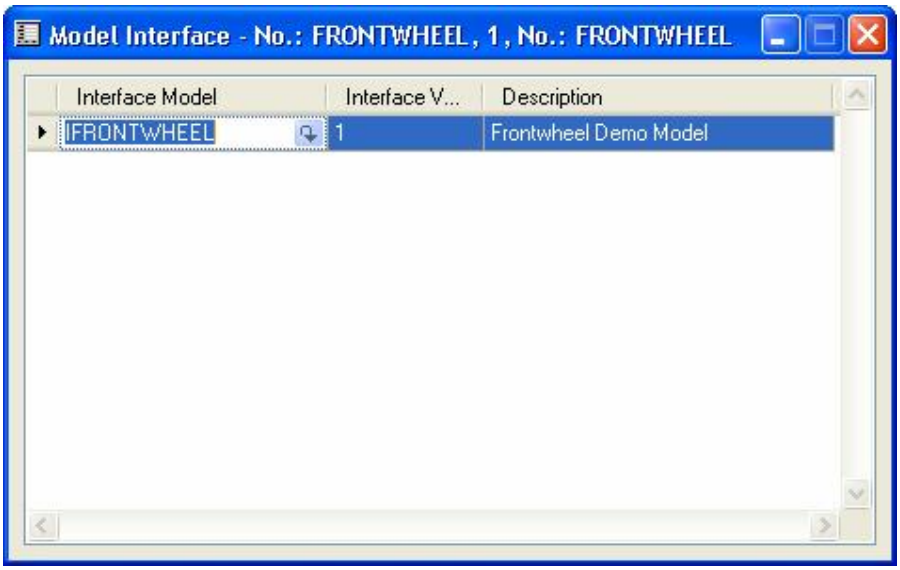
- Execute the function 'Generate model interface'.



- The system asks for name, description and label for the interface model that will be created after clicking OK. A new interface model is added in the Generic Model list. The interface model is added to the Frontwheel sub model.

- Check if the interface is added by selecting 'Model Interface' from the 'Generic Model Version' button.



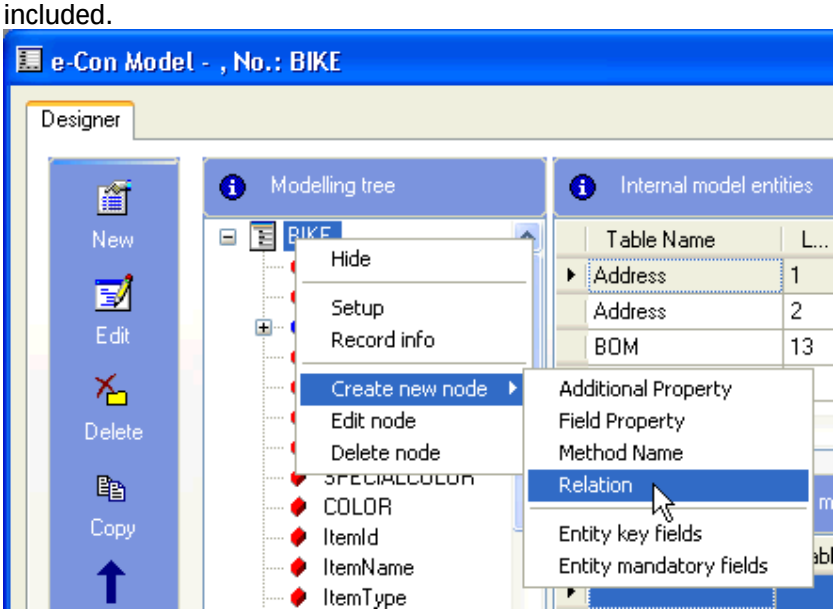


- Finally generate the data structure for the submodel ‘Frontwheel’

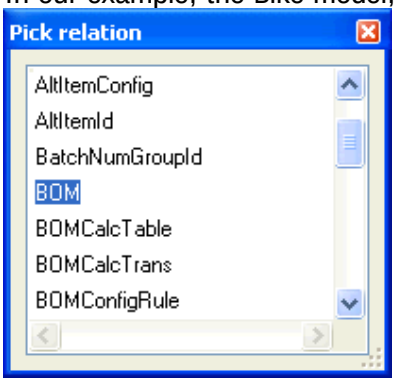
6.12.2 Set up the link to the sub model in the data structure of the main model.

The last step is to set up the link between the main model and the sub model, including a sub model in to the main model. This is done with the aid of a so called “Relation” member type.

- Add a Relation to the entity in the main model where the sub model must be included.

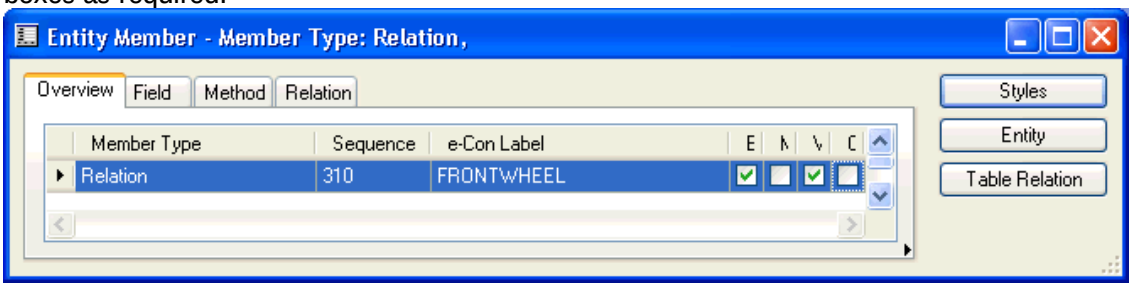


In our example, the Bike model, you must pick the relation ‘BOM’.

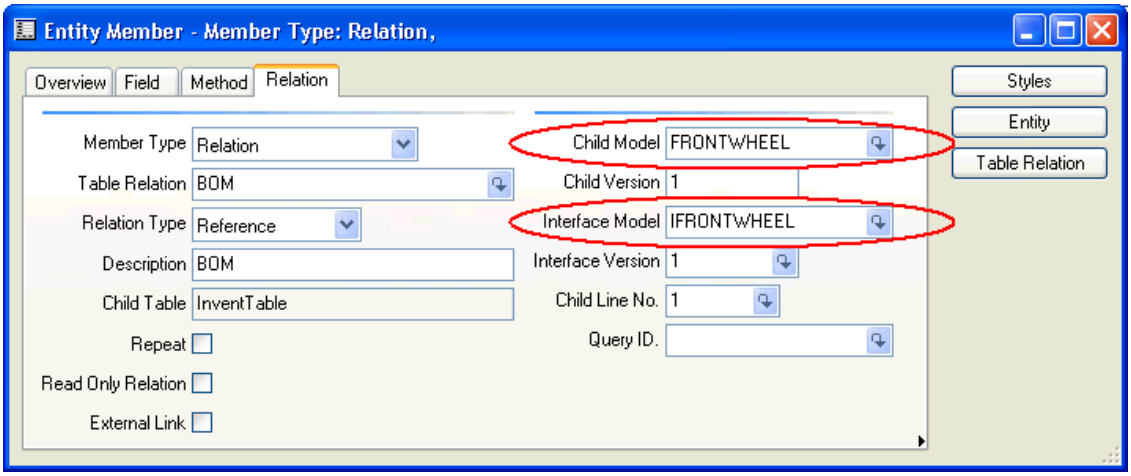


- Right-click the BOM entity and click ‘Edit node’. Change the e-Con Label to FRONTWHEEL and select the Editable, Mandatory, Visible and Constant check

boxes as required.



- On the Relation tab, select the cild model and the interface model. In our example, these are respectively the FRONTWHEEL model and the IFRONTWHEEL model.



- To generate the model, in the Generic Model form, click Model Setup.
- In the e-Con Model form, click Generate Model.

Tips & Tricks

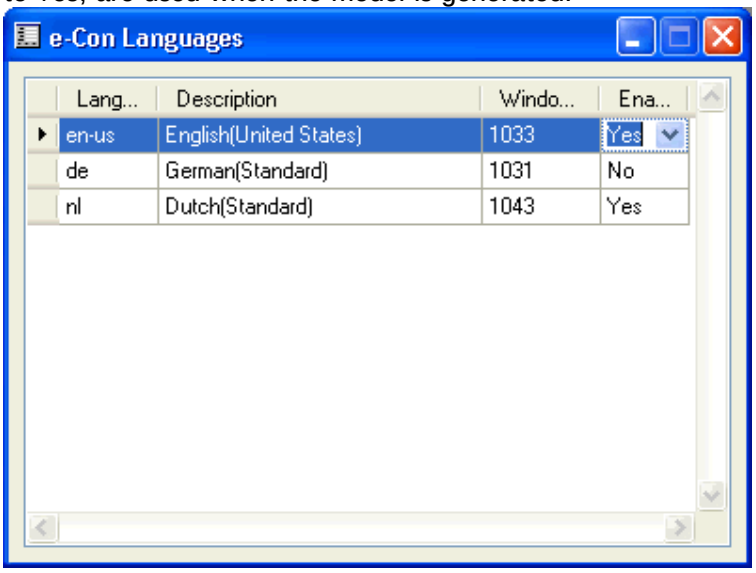
Always generate the Interface Model prior to the Sub Model and Main Model. When the main model is generated prior to the interface or sub model generation will fail.

There is also a SetInclude function to Include Another Model with a rule. See chapter 9 for more details

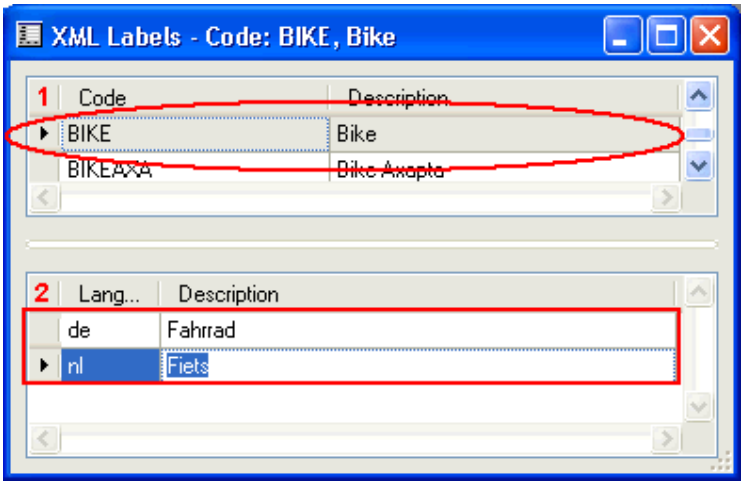
6.13 How to Make Your Model Multilanguage Compatible

You can setup e-Con and your models in such a way that you can use several languages. To make you e-Con models mulitlanguage compatible:

- In the e-Con menu, click Setup, e-Con Languages.
- In the e-Con Languages form, enter the languages that you want to use. You can enter label translations for all languages that you specify as e-Con languages. However, only the e-Con languages for which you set the value of the Enabled field to Yes, are used when the model is generated.

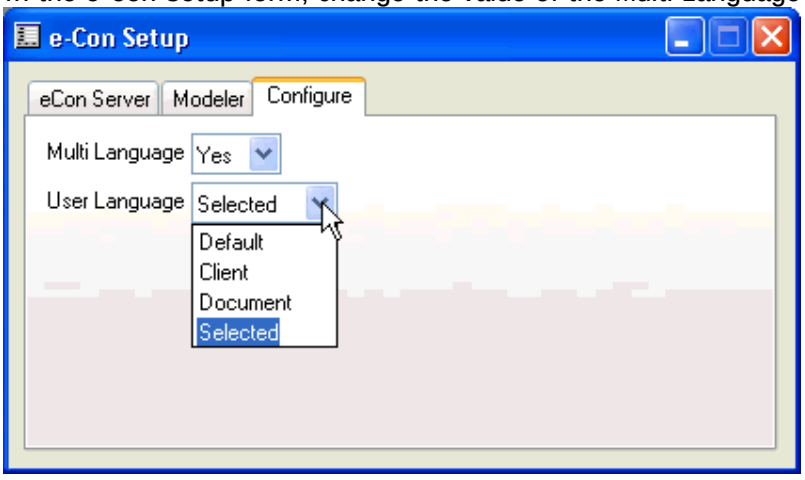


- Add label translations for the languages that you have specified as e-Con languages. To do so, in the e-Con menu, click Setup, XML Labels.
- In the XML Labels form, in the upper pane, select the label for to which you want to add translations.
- In the lower pane, enter the translations. You only need to add translations for the self-defined labels.



- To enable the use of the created translations, click Settings, Parameters.

- In the e-Con Setup form, change the value of the Multi Language field to Yes.



- In the User Language field, specify which language will be used if you enter configurations:
  - Default: the language as specified in the e-Con environment settings.
  - Client: the language of your computer.
  - Document: the customer's language as used in a spscific document.
  - Selected: in this case you get the option to select a language when you enter a configuration.

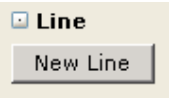
If you generate the model, the label translations are added to the generated model.

6.14 How to Work with Arrays

In this section, we explain how you can make Entities Repeatable, and how you can define arrays for these Repeatable Entities.

A Repeatable Entity can be very useful when you want to offer the end user the possibility of selecting from among an undetermined number of members.

For example, take the Bike demo model: you can configure a Bike using this demo model, but you add as much accessories as necassery. To deal with this, the Entity "Accessory" can be set up as a Repeatable Entity. That means that a button labeled "New ..." will be displayed in the Front End, so the end user can simply click this button for each new accessory needed.



6.14.1 How to Make Entities Repeatable

As explained above, e-Con allows you to designate some Entities as Repeatable Entities. This can be extremely useful when you do not know in advance how many Entities the end user will need.

The example given above was that of Sales Lines in the Sales demo model, but we can also look at the example of Accessories in the Bike demo model.

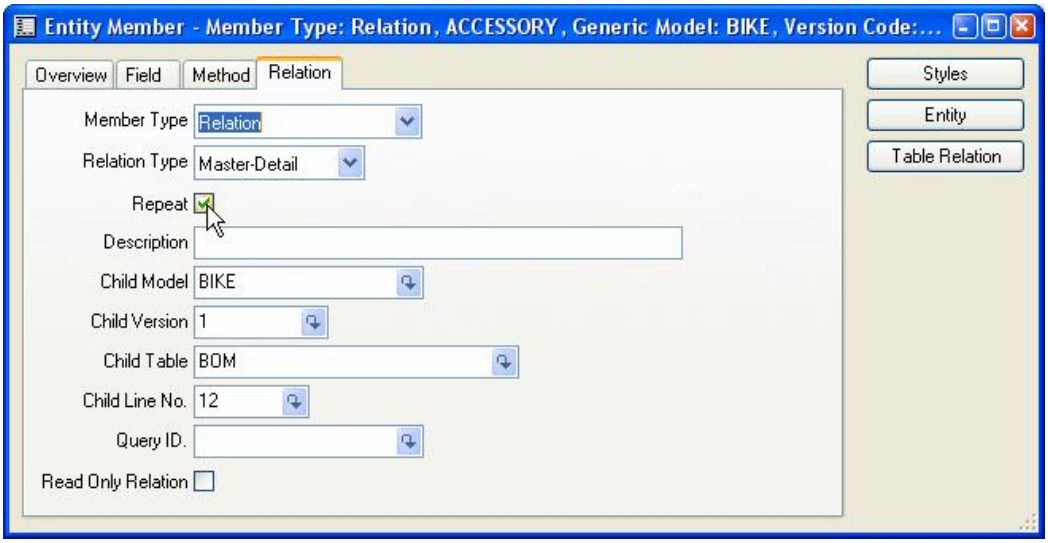
An Entity always causes a button to be displayed at the Front End. The button's label reads "New...." plus the label description, like "New Line", or "New Accessory". Clicking on this button adds a new Entity, which is visible at the Front End.

Designating an Entity as Repeatable is quite straightforward, but there are a few important rules to keep in mind:

1. An Entity is made Repeatable by putting a check-mark in the "Repeat" field in the relevant Entity's relation on the Entity Member form.
2. Only members of the type "Relation" can be made Repeatable; it's not possible to make a member of the type "Additional Property" or "Field Property" Repeatable.
3. Only relations of the type "Master-Detail" can be made Repeatable; it's not possible to make a relation of the type "Reference" Repeatable.

For example:

In the Bike demo model, the Entity member "Accessory"; has been designated as a Repeatable Entity. To do this, go to the Entity Member form for "BOMVersion" Entity BOMVERSION-1, and find the Entity member of type "Relation" that is labeled "Accessory". Simply check-mark the "Repeat" column for that member, and the corresponding BOM Line will be repeatable.



In short, the only step required for making an Entity into a Repeatable Entity: check-mark the "Repeat" field for the relevant Entity member.

6.14.2 How to Set Up Arrays

In the above section, we describe how Entities can be made into Repeatable Entities. Doing so will cause the Entity to be expanded with an index, that is, [1] for the first repeated Entity, [2] for the second, [3] for the third, and so on.

If we didn't have arrays, and you wanted to add a rule to a Repeatable Entity, then you would have to repeat this rule over and over again. For example, if you wanted to assign a default value to a member of a Repeatable Entity, the rule would look like this:

```
REPEAT[1].DEFAULT = "defaultvalue".
```

The rule would result in the member "DEFAULT" of the first Repeatable Entity "REPEAT" has a default value of "defaultvalue". But this rule would only apply to the first of the many repeated Entities. In order to carry this rule through and assign the desired default value to all the repeated Entities, you would have to add many more rules:

REPEAT[2].DEFAULT = "defaultvalue";

REPEAT[3].DEFAULT = "defaultvalue";

REPEAT[4].DEFAULT = "defaultvalue";

REPEAT[....].DEFAULT = "defaultvalue".

As you can see, this is not the best way to go about making rules. But arrays make this sort of repetitive rule-making unnecessary. Instead, you can define an array for a Repeatable Entity. Arrays make it possible to just create one rule for every member of the Repeatable Entity.

Solving the above problem with an array requires the following rule:

Index #n of REPEAT;

REPEAT[#n].DEFAULT = "defaultvalue"

In the first rule, the index is declared. In this example, the name of the index is "n". But it can be defined by the user.

The syntax has always to be:

Index #.. of <entity>

Enter the user-defined name of the array where the dots appear. Replace <entity> with the Repeatable Entity.

NOTE: if you select this Entity with the drag and drop functionality, the index [1] is added automatically. Just delete this index.

6.14.3 How to Control Arrays

It's possible to control an array by setting up how many Entities can be added. This effectively means designating the maximum number of members the array can have.

In the Bike demo model, for example, the maximum number of different accessories is three. After clicking twice on the button for a new one, this button simply disappears. At that point, it's impossible to add another New Accessory.

This functionality can be added to a rule by using the "@repeat" attribute.

Take a look at the bike example before reading further.

Implication:

```
1 index #n of BomVersion.ACCESSORY;
2
3 BomVersion.ACCESSORY(#n)@repeat = #n < 3
```

The first step is to define an array. (Explained previously.) Next, add the "@repeat" attribute to the Repeatable Entity. This attribute can either have the value "true" or "false". When the value is true, the "New.." button is shown in the Front End; when the

value is false, the "New.." button will disappear. The expression "#n < 3" delivers a value of "true" when the array is smaller then 3 members and "false" when the array has 3 or more members (thus making the button disappear in the Front End).

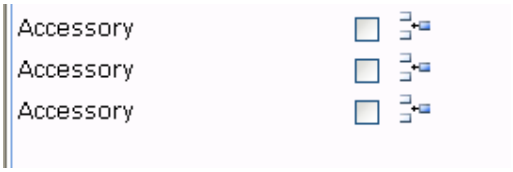
Furthermore it is possible to set the number of members of an array that will be automatically displayed in the Front-end. By adding the "@length" attribute to the repeatable Entity the number of repeatable entities can be set.

For example:

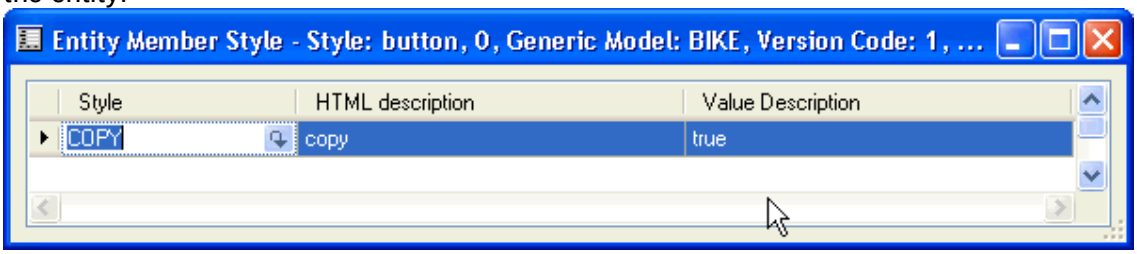
Implication:

```
1 BomVersion.ACCESSORY@length = 3
```

Thanks to the rule above, three members from the type accessory are automatically displayed in the Front-end:



Copying and deleting array elements in the UI is possible as well. By adding the style "copy" and / or delete with value "true" to the repeatable Entity a copy and / or delete button is displayed in the UI. Clicking this button will make a copy of the entity or delete the entity.



The "copy" and delete button in the UI:

6.14.4 Special array functions

There are special functions available for arrays. The following functions are available:

- Sort (Array;Key): Sort a dynamic array

- ApplyFunction(Array; Function; Field; Condition): Apply a function to all members of an array and return the result. Functions supported are Sum, Average, Concat, Min and Max. Conditions supported are “=” and “<>”.

In the following example the array “items” is sorted by the field “price”. The outcome is a list of items sorted by price.

Condition:

```
1 SortItems
```

Implication:

```
1 Items = Sort("ITEM"; "Price");
2 SortItems = false
```

In the following example the sum of all properties “Price” of the array “ITEM” is calculated where the value from “Quantity” is not “0”.

Implication:

```
1 Total = ApplyFunction("ITEM"; "Sum"; "Price"; "Quantity"; "<>"; 0)
```

6.15 How to Add Images to Your Model

With e-Con, it’s possible to show images in the Front End. Run the Bike demo model for a thorough understanding of the possibilities e-Con offers.

You can set up this functionality in a flexible way, and designate that different images be shown depending on certain members. It’s even possible to show more than one image in the Front End.

In this section, we will go through the steps of adding images to your model.

- Add a member at the highest level of your model and name it "IMAGES". This member will serve as the trigger for e-Con to show images in the Front End.

e-Con is structured in this way to optimize performance: When there are images in the model, e-Con must scan the XML structure twice – once for members and once to show images. The second check is only carried out if a member called "IMAGES" is present in the data structure. This streamlines performance if there are no images.

The next step is to add Business Rules in the Studio to set up the images. You can fully control your images with the help of Styles. A few styles are specially designed for use with images; these must be applied to a member. You could simply add this style to the new member "IMAGE", but it’s best to apply the styles to the related members.

For example, if you want to show an image of the Bike's Front Wheel, simply add the relevant styles to the member in the data structure where the Front Wheel is selected. This facilitates a clear overview.

Before displaying images in the Front End, space has to be reserved to show the images. This space is called the background. Use the style "Style.ImgSrc" to set up the background. Standard available is the file "transparant.gif". This file can be used as a background.

For Example:

IMAGES@style.imgsrc = "./images/transparant.gif";

IMAGES@style.imgposition = "right".

This will offer a transparent background space at the right hand side of the Front End.

The following styles are available to control the images:

- <memb>@Style.Imgsrc: With this style, the source of the images can be designated. This 'source' consists of the name of the image and the path where the image is stored. There are two ways to designate the source:
    - Notate the complete path, as follows: "C:/My Images/Bike.jpg"
    - Notate the relative path, the first part of the path always being the path where e-Con is installed. For example: "./images/bike.jpg". The complete path in this case would be "C:\eCon\images\bike.jpg".
- <memb>@Style.Imgvisible: With this style, it's possible to make an image visible (or invisible). An image is always visible by default. To make the image invisible, assign a string with value "false" to this style. To make the image visible again, assign "true".

- <memb>@Style.Imgposition: This style can have four values: "Left", "Top", "Right" and "Bottom". With the style the position of the image is set. When "Left" or "Right" are chosen the image area will be shown at the left or right hand side of the Front-end without overlapping the questions and answers fields. With "Top" and "Bottom" the image area is shown above or under the Front-end.

- <memb>@Style.Imgleft: After determining where you want to have the image (right, left, bottom or top) on the Front-end with "imgposition" style, you can go further and determine where exactly you want the image to be positioned in relation to the left margin of the image area.

- <memb>@Style.Imgtop: After determining where you want to have the image (right, left, bottom or top) on the Front-end with "imgposition" style, you can go further and determine where exactly you want the image to be positioned in relation to the top margin of the image area.

- <memb>@Style.Imgheight: With this style, you can determine the height of the image. If you do not specify the width, this value will scale your image to the correct dimensions.

- <memb>@Style.Imgwidth: With this style, you can determine the width of the image. If you do not specify the height, this value will scale your image to the correct dimensions.

The styles described above are only used for images.

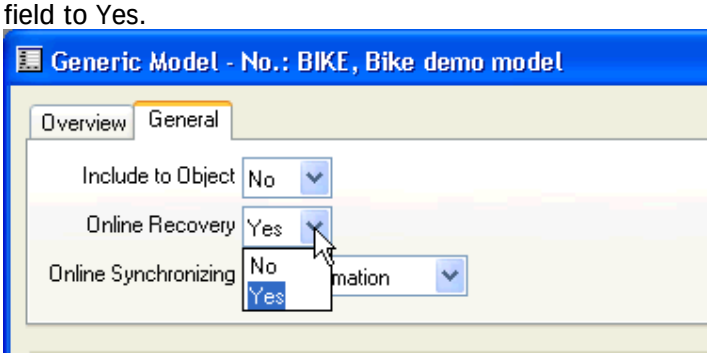
Tips & Tricks:

It's advisable to use logical names for the image files, especially when the images are to be displayed for particular members. In the Bike demo model, the image file names are based on the Category, Model and Type of the bike.

6.16 How to Set Up Recovery

If you enable online recovery, e-Con will show the errors if something goes wrong when you enter configurations. To enable recovery:

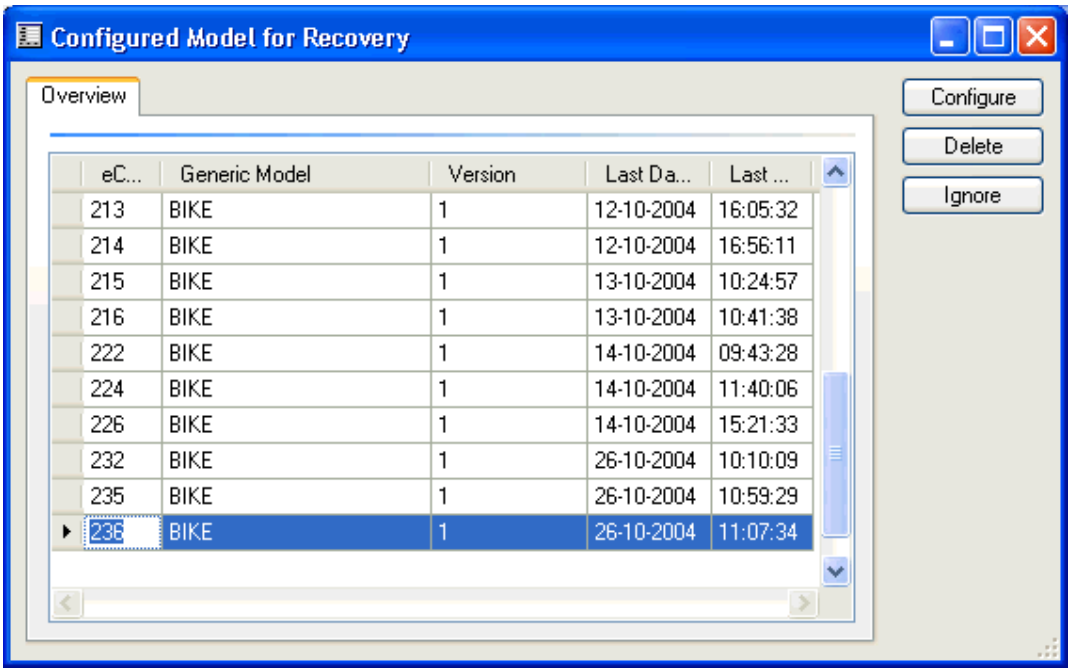
- In the e-Con menu, click Models.
- On the Generic Model form, on the General tab, set the value of the Online Recovery field to Yes.



As a result, errors are displayed when you save a configuration and something is wrong in the configuration.

If you again enter a configuration, first the Configured Model for Recovery form appears displaying all configurations, for which the following apply:

- The status is New.
- The Error check box is selected.
- The user who entered the configuration is the current user.
- The configuration's model is the current model.



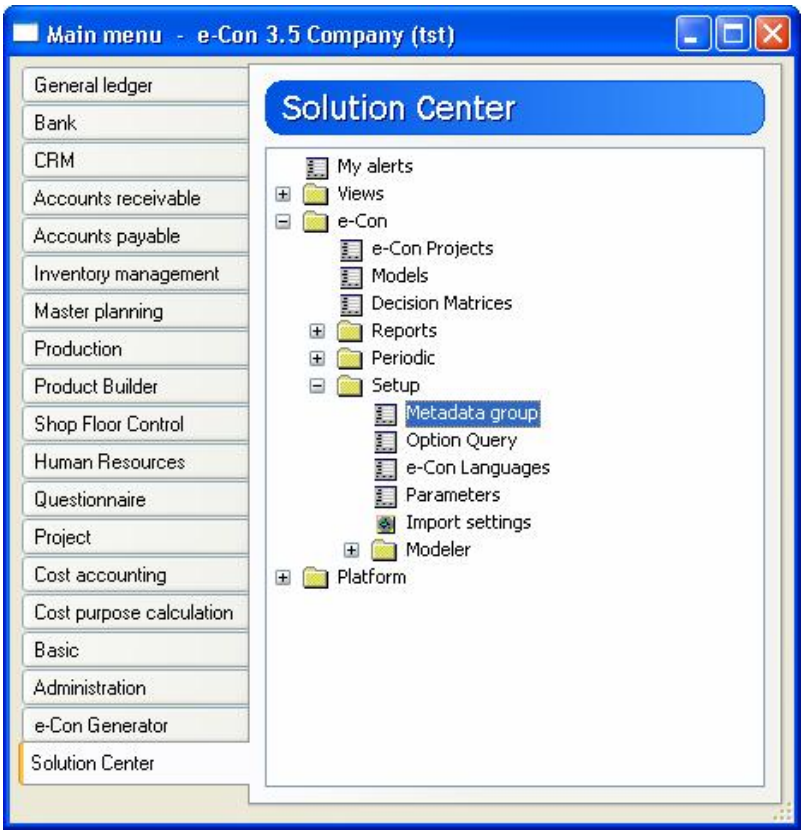
For each displayed configuration, you can decide whether you want to carry out one of the following actions:

- (Re)configure
- Delete
- Ignore

6.17 How to extend the Table Metadata Groups

In the Table Metadata Groups the tables to be available for the rule wizard are determined. The Metadata Groups are filled by default during the initialization of the e-Con setup. The most commonly used tables are available with the default Metadata. If other tables are needed the Table Metadata Groups should be extended.

- Go to Table Metadata Group in the e-Con setup.

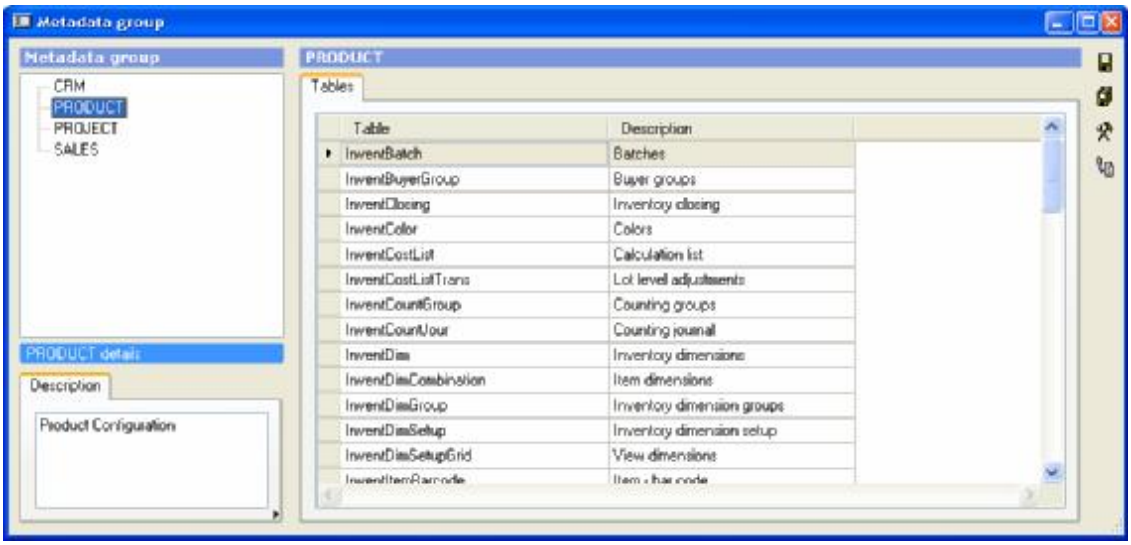


Following possibilities are available:

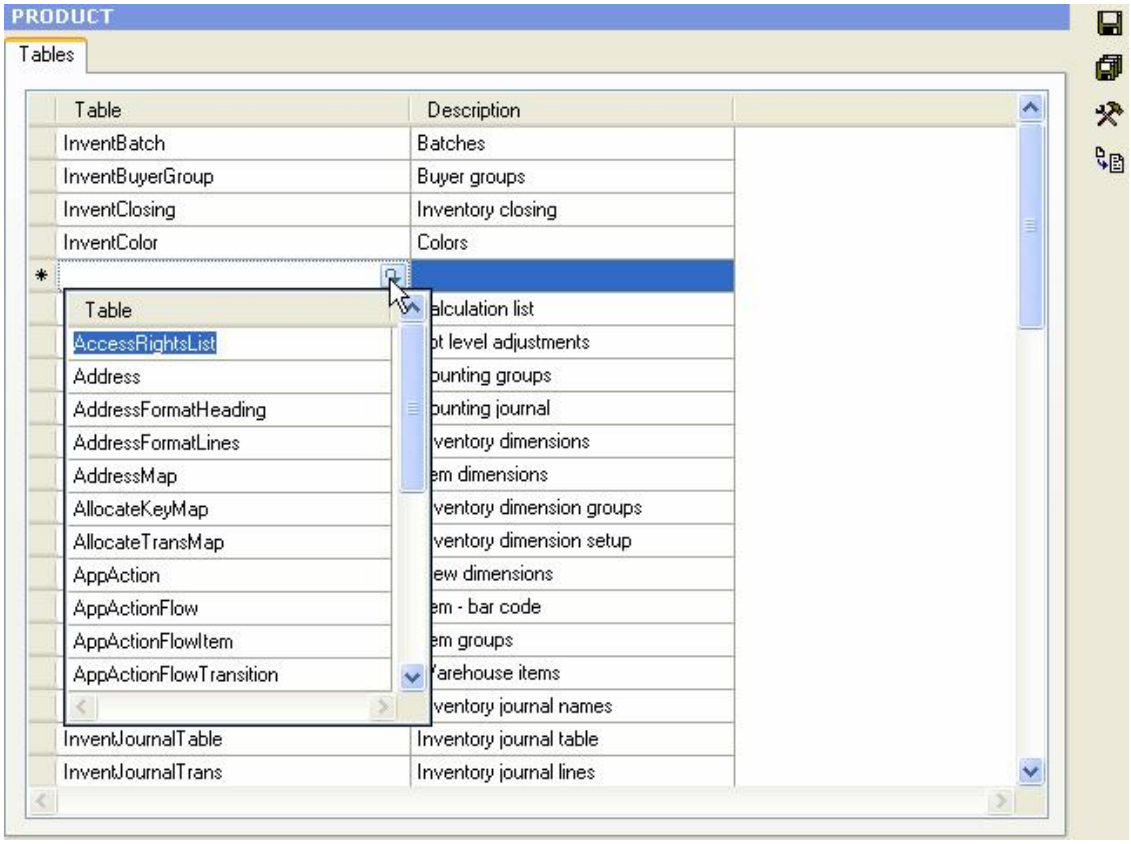
- Adding table to existing Table Group.
- Adding a new Metadata Group with members.

Adding table to existing Table Group

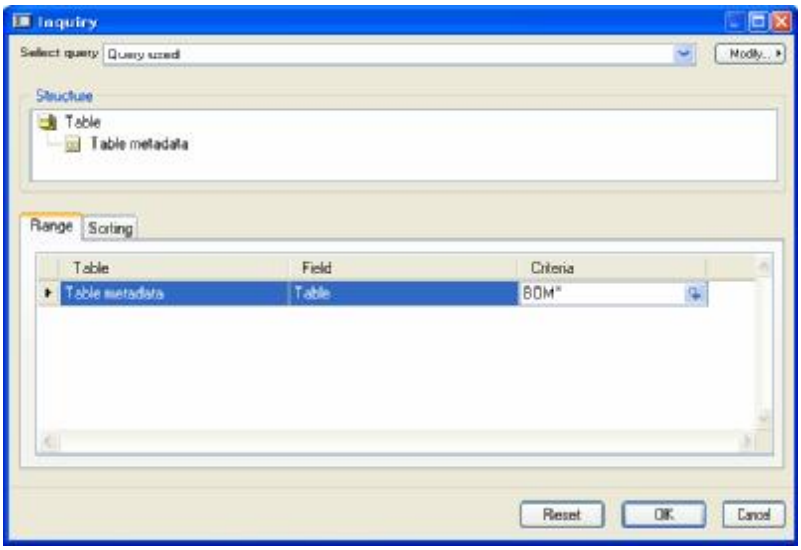
- Select the table group to be extended.



- Select the tables area of the group and press <Ctrl> N to insert a new record and select the table of your interest.



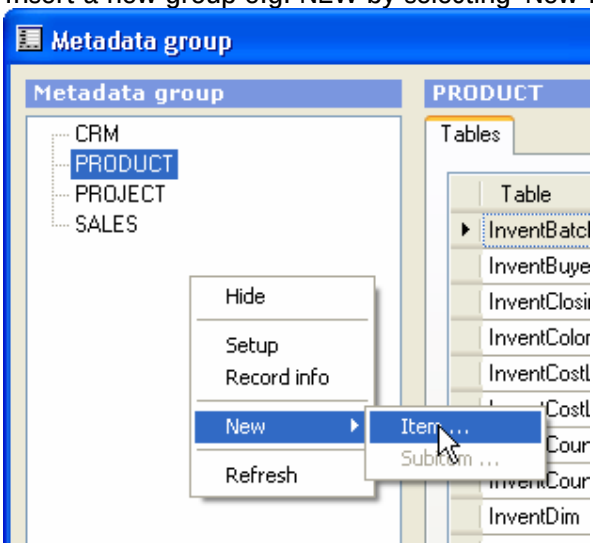
Another possibility is the uses of the  query mechanism to add a range of table at once to the meta data group.



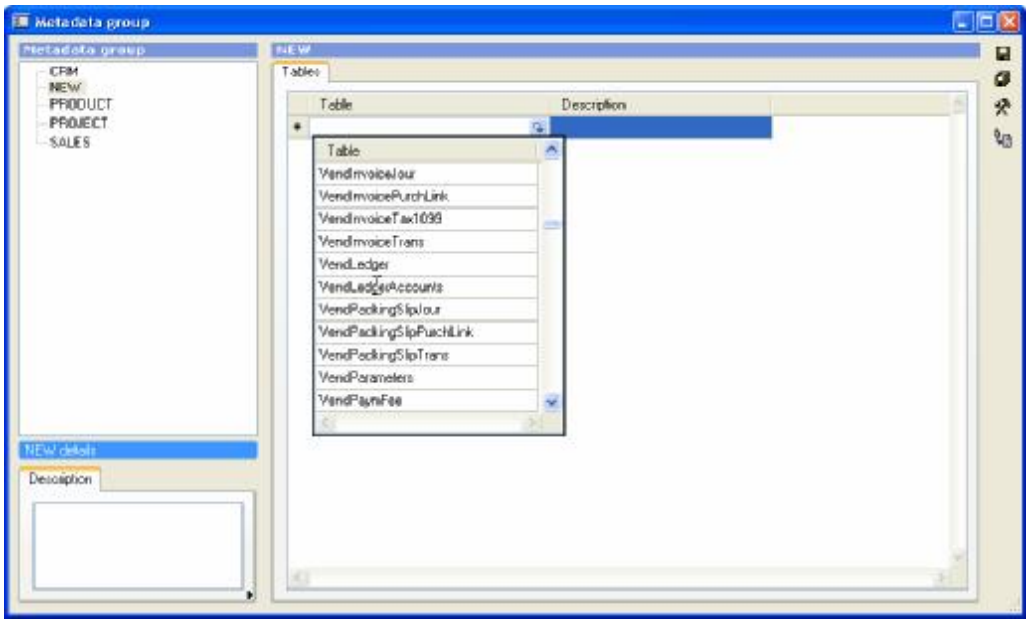
- Generate the Meta Data by clicking the save and generate button.  This will create the XML file to be used for the rule wizard.

Adding new Table Group with members

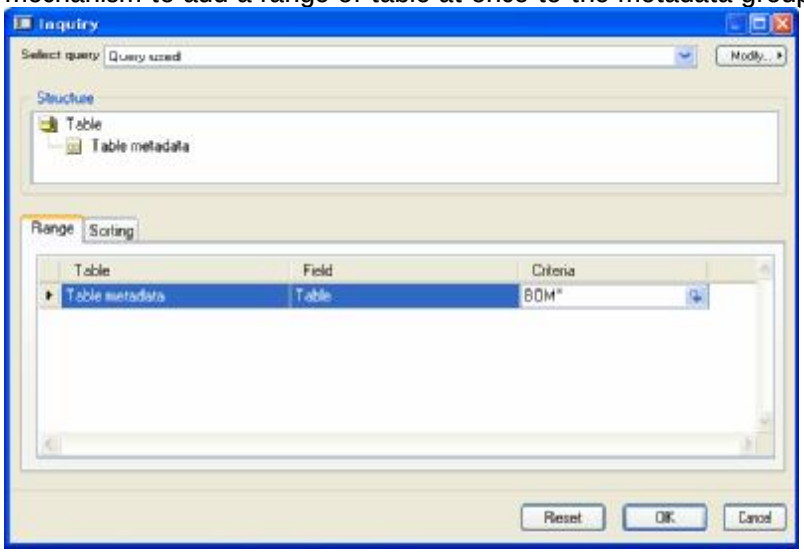
- Insert a new group e.g. NEW by selecting 'New Item' from the context menu.



- Select the Tables area of the new group. There are two ways of adding tables.
  - Adding table one by one: press <Ctrl> N to insert a new record and select the table of your interest.



- Adding tables by using a query: Click  to activate the query mechanism to add a range of table at once to the metadata group.



- Generate the Meta Data by clicking the save and generate button.  This will create the XML file to be used for the rule wizard.

Chapter 7 e-Con Axapta Connector Functions and Set-up

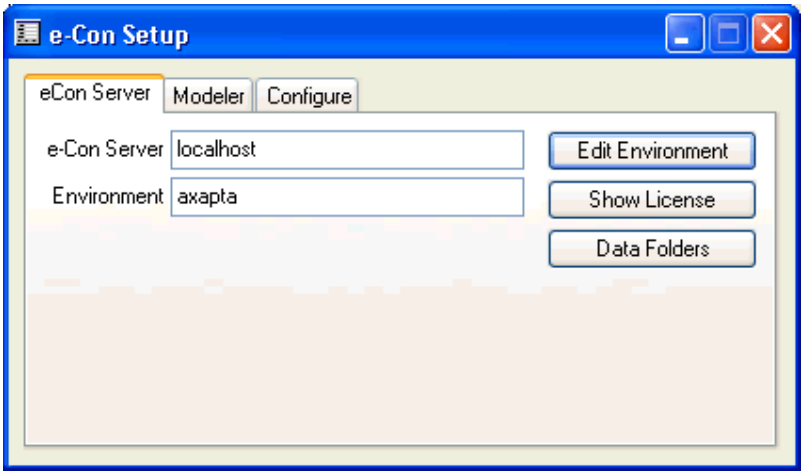
The purpose of this chapter is to provide you with information on general options and settings in e-Con. For a full description of all fields found on the "e-Con Setup" form, consult Chapter 5 or installation manual.

7.1 How to Set Up e-Con for Axapta

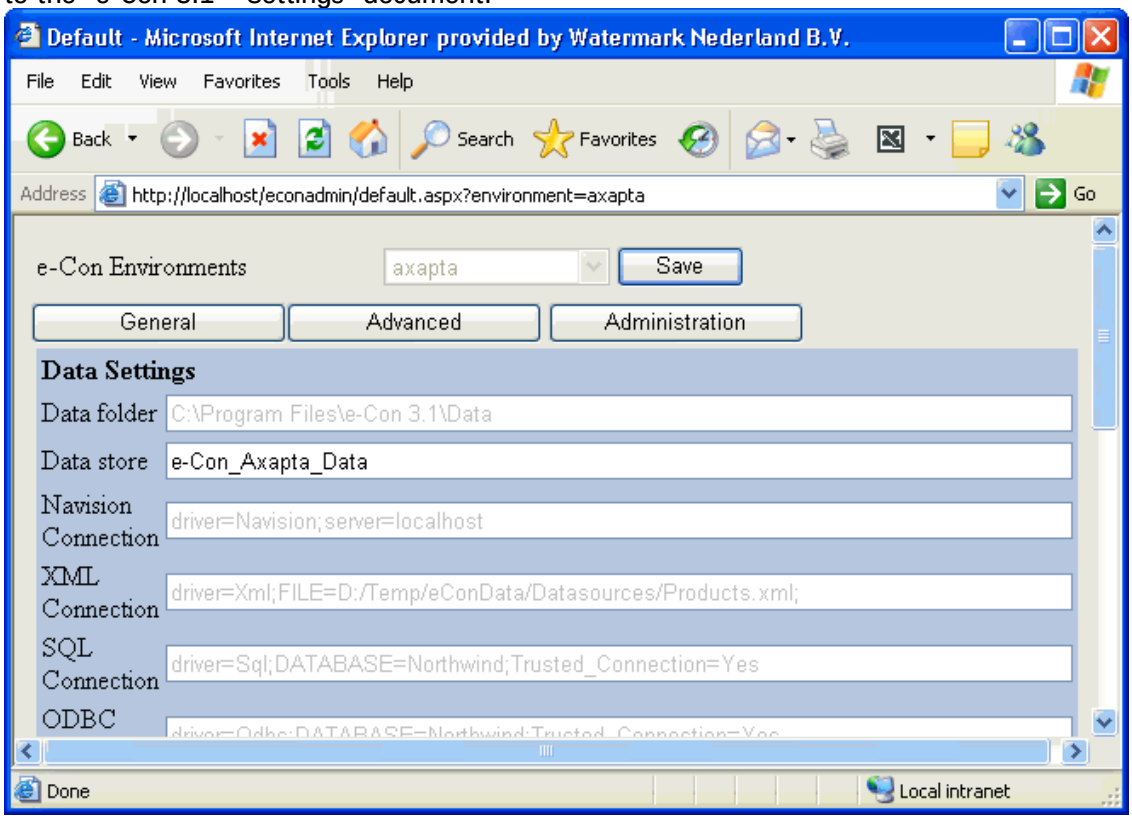
By going to "Setup" in e-Con's Main Menu and clicking on "Parameters" option, you can alter or check the default installation settings.

- Go to "Setup" in e-Con's Main Menu and click on the "Prameters" option.

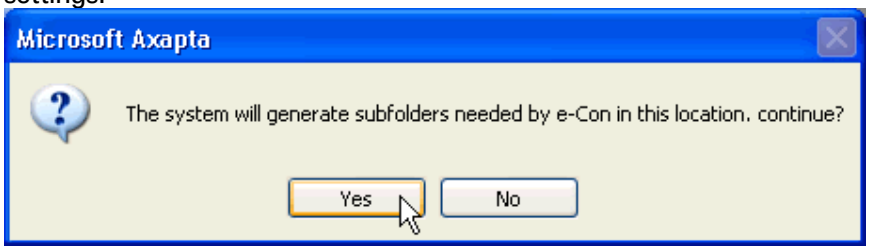
The following form will open:



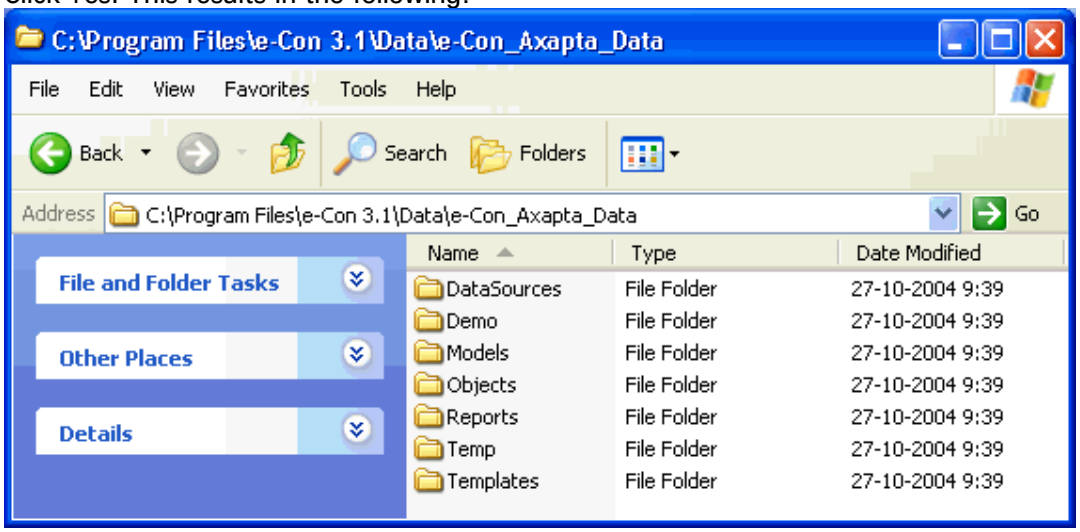
- Edit environment: Opens the e-Con Environment settings. For more nformation, refer to the “e-Con 3.1 – Settings” document.



- Show License: shows the XML license. Important is the name of the e-Con server machine. It must be in the e-Con Server field and equal to the name in the license.
- Data Folders: creates the folder structure that is needed by e-Con. The folder structure is created in the ‘Data store’ folder as specified in the e-Con Environment settings.



Click Yes. This results in the following:



7.2 How to Set Up the e-Con Application Server

For more information on setting up the e-Con Application Server, reefr to Chapter 11, “Setup in the e-Con Application Server” in the “e-Con 3.1 - Installation Manual for MBS-Axapta”.

7.3 How to Generate Meta Data

For more inforamtion on generating the meta data, refer to Chapter 10, “Generation of Meta Data” in the e-Con Application Server” in the “e-Con 3.1 - Installation Manual for MBS-Axapta”.

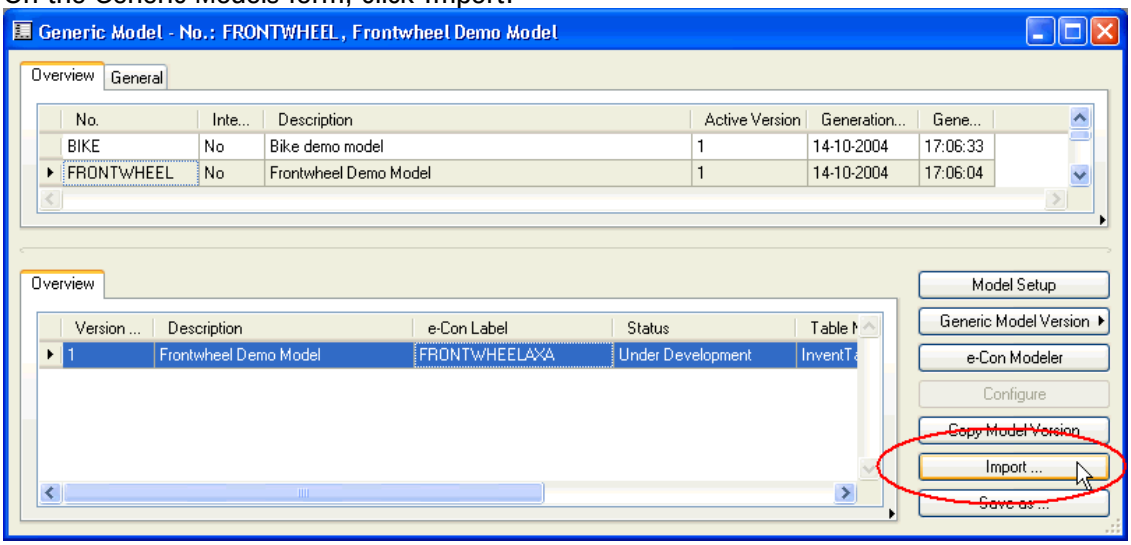
7.4 How to Import or Export Models

With e-Con, you can import and export models.

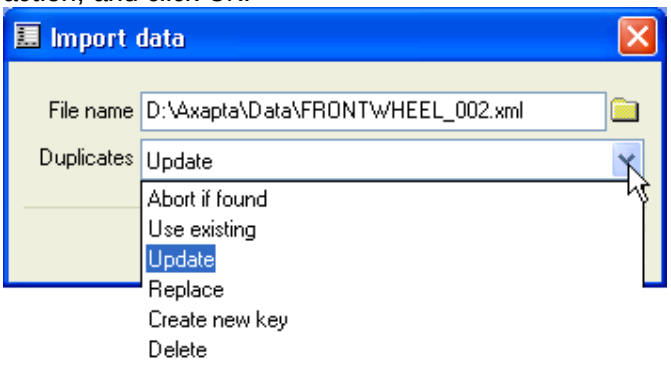
7.4.1 Model Import

To import a model:

- On the Generic Models form, click Import.



- On the Import data form, select the to-be-imported model's xml file, the duplicate action, and click OK.

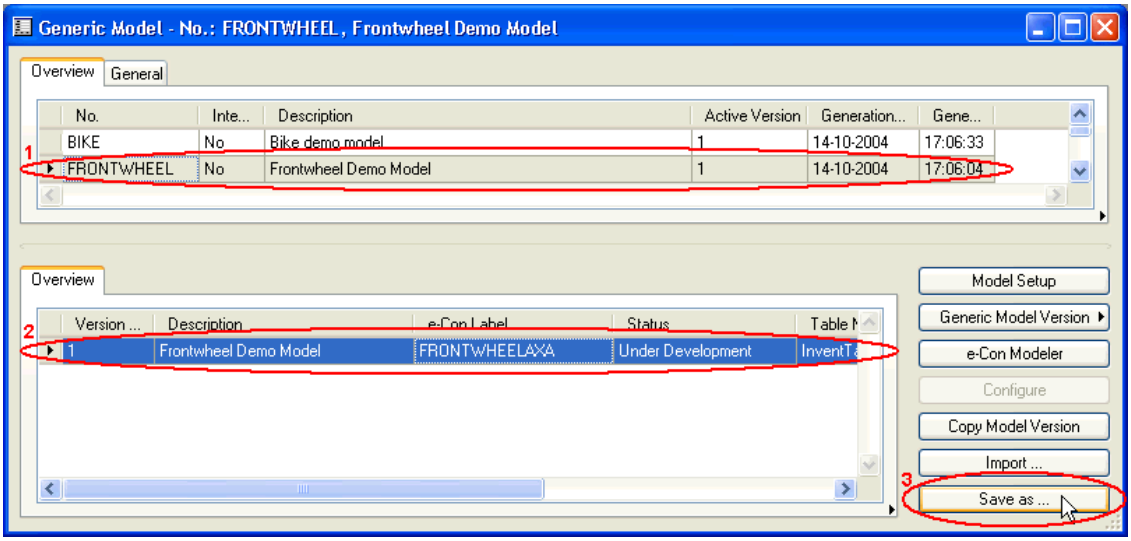


7.4.2 Model Export

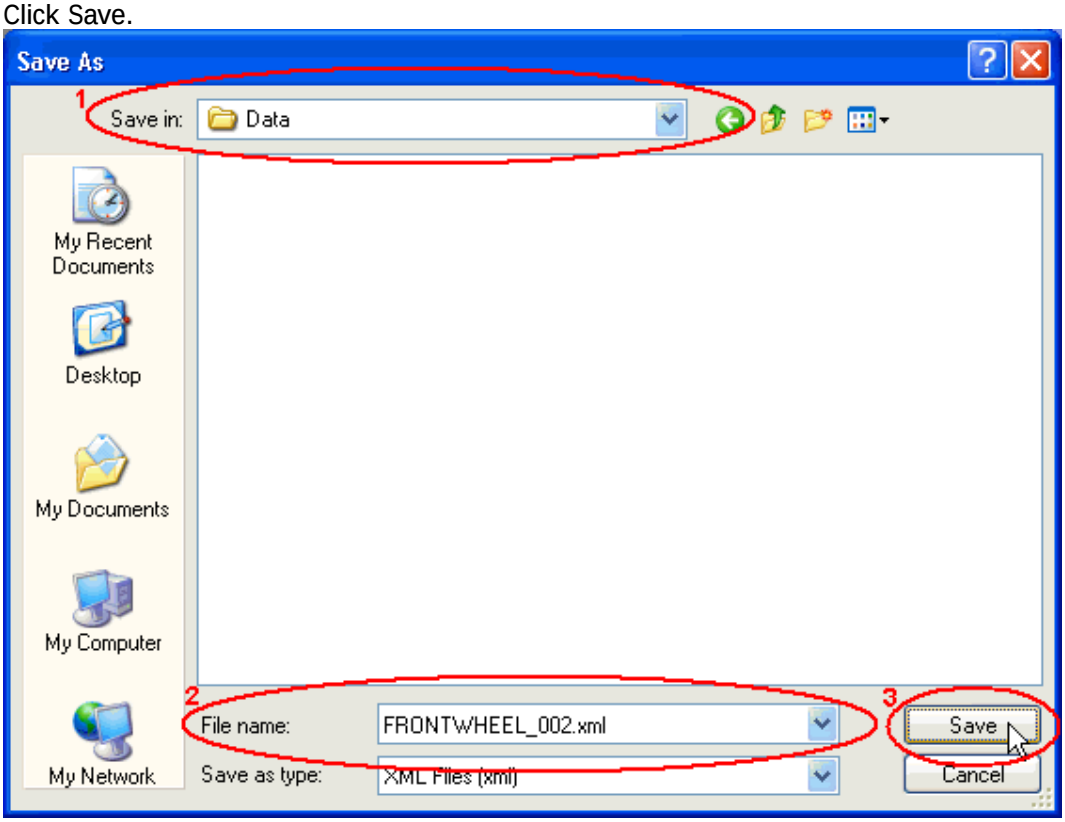
To export a model:

- In the Generic Model form, select a model.
- Select a version of the model.

- Click Save as.



- Specify the folder in which you want to store the model export file.
- Enter the file name.
- Click Save.

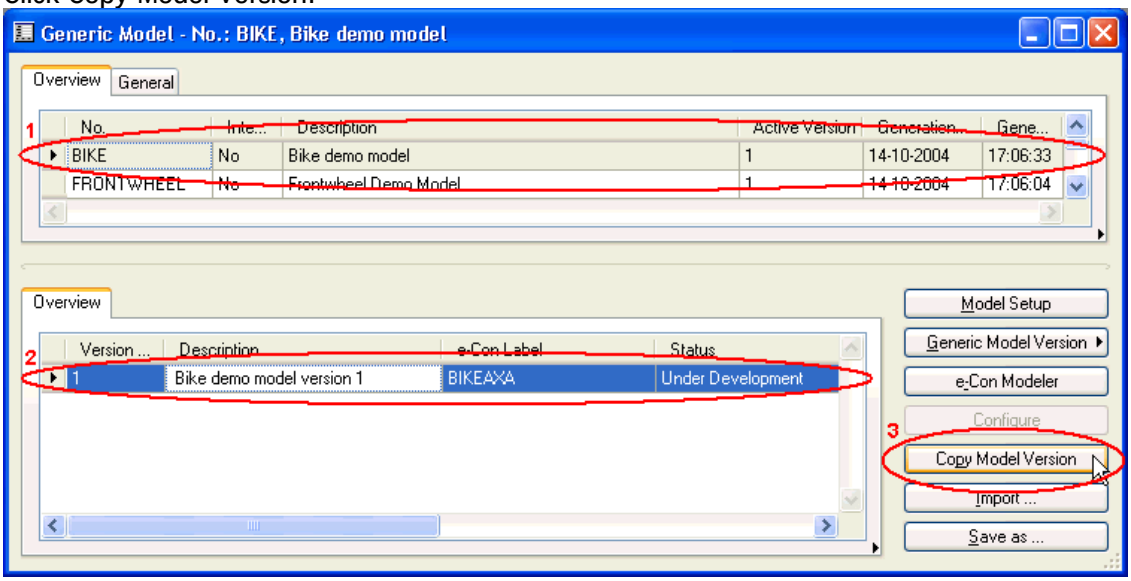


7.5 How to Copy a Model

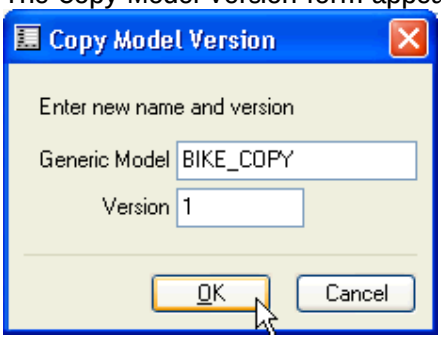
To copy a model:

- In the Generic Model form, select a model.

- Select a version of the model.
- Click Copy Model Version.



- The Copy Model Version form appears.



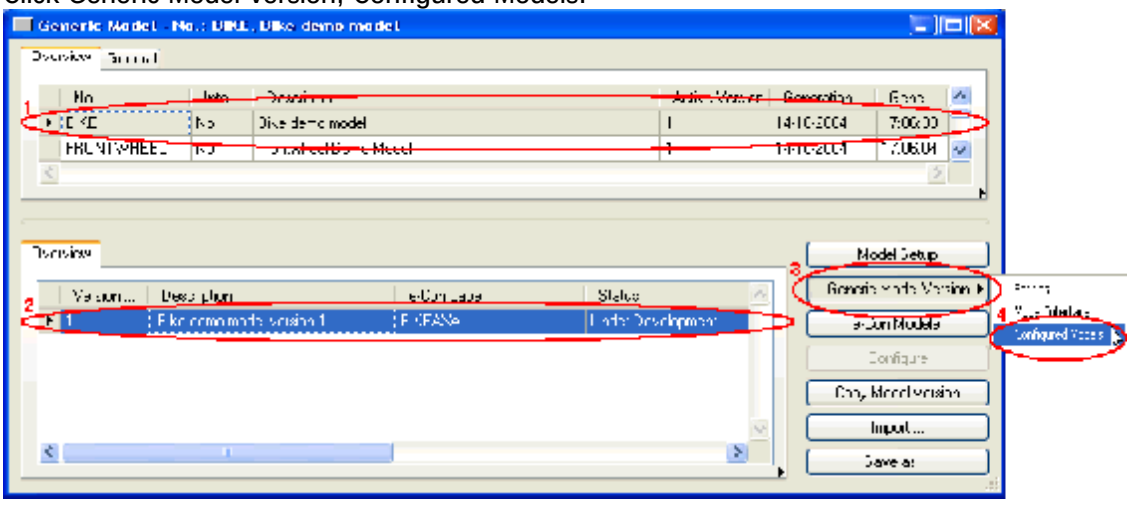
- Fill in the Generic Model and Version for the new (copied) model version. In the example shown, the BIKE model, version 1 will be copied to a new model with the name BIKE_COPY and Version Code 1. The contents of this new model will be an exact copy of the BIKE model.

7.6 How to View Your Configurations

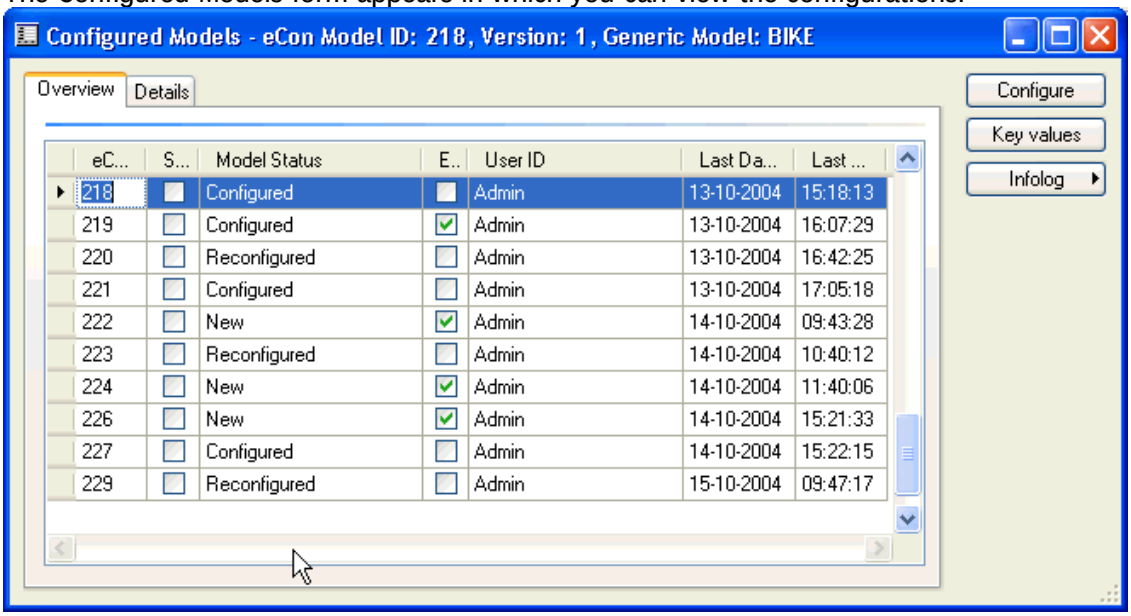
You can view the configurations that you created with e-Con.

- To view your configurations:
- In the Generic Model form, select a model.
- Select a version of the model.

- Click Generic Model version, Configured Models.



- The Configured Models form appears in which you can view the configurations.



e-Con Model ID
Automatically generated ID

Synchronize
If set to Yes before Configure the model will be synchronized during reconfiguration.

Model Status
Status of the model.

Error
Error messages will be displayed here.

User ID

User who generated the configuration.

Last Date Modified

Date when the latest changes were made to the model.

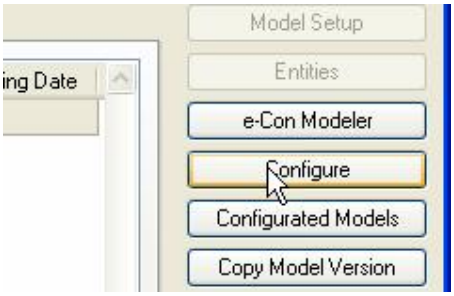
Last Time Modified

Time when the latest changes were made to the model.

Note: it is also possible to view the configurations from the e-Con Model form. Go there with the “Setup Model” button.

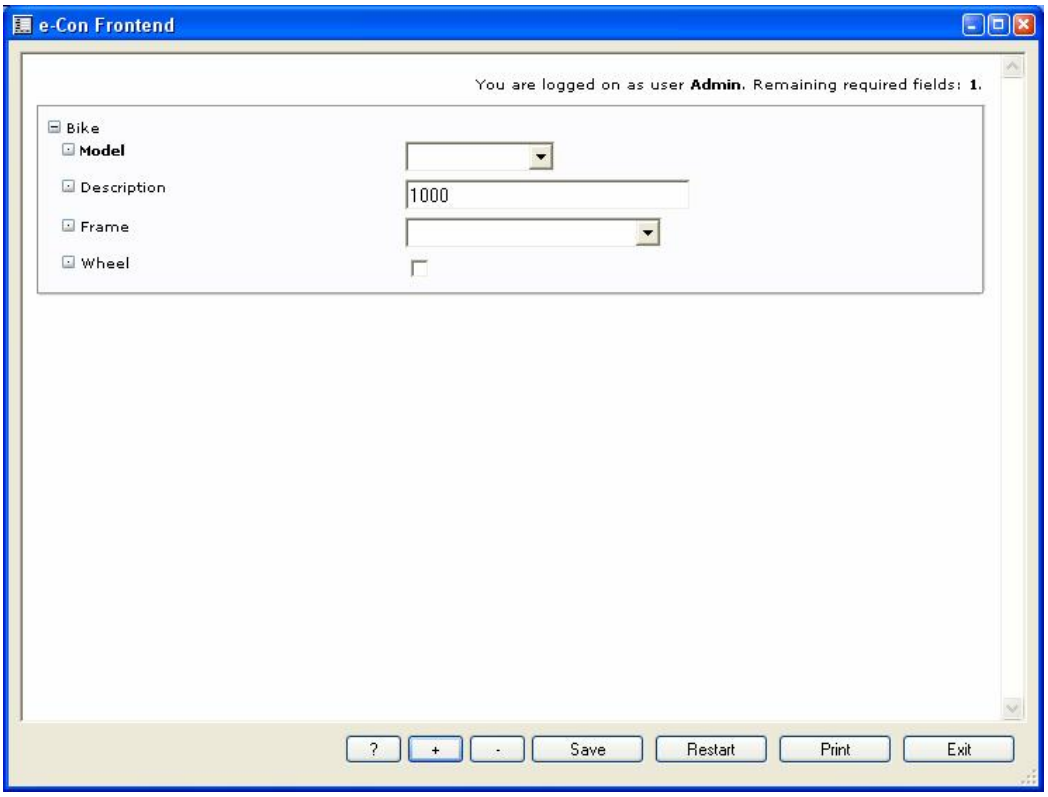
7.7 How to Start Up Your Model

- Select the desired model on the Generic Model form, the active version must be certified.



- Click the "Configure" button on the Generic Model.

The e-Con Front End will be activated:

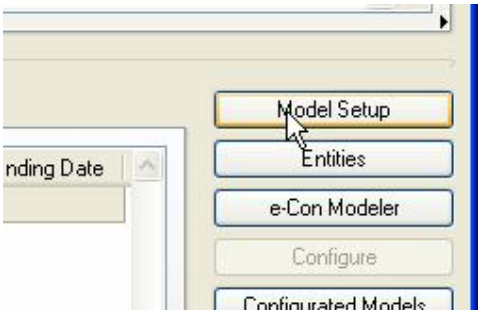


Note: it is also possible to start the model from the e-Con Model form. Go there with the "Setup Model" button.

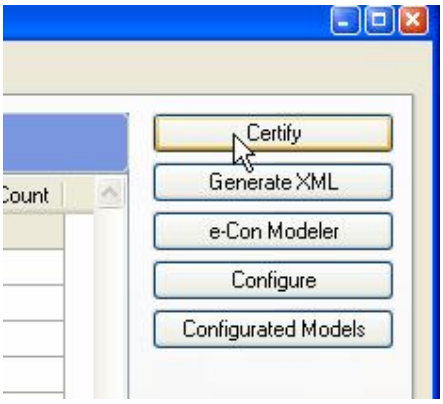
7.8 How to Generate XML

In order to view the last changes made to your model, XML must be generated based on the active version.

- Select the model on the Generic Model form for which you want to generate the XML. The active version must be "Under Development". Select the "Model Setup" button to go to the e-Con Model form.



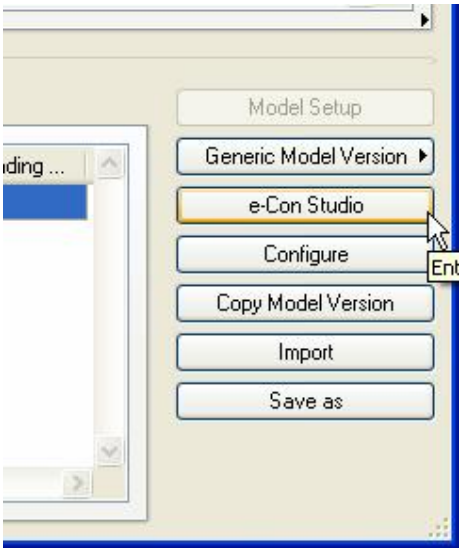
- Use the “Certify” and “Generate XML” buttons to certify the version and generate the XML based on the version.



7.9 How to Start Up the e-Con Studio

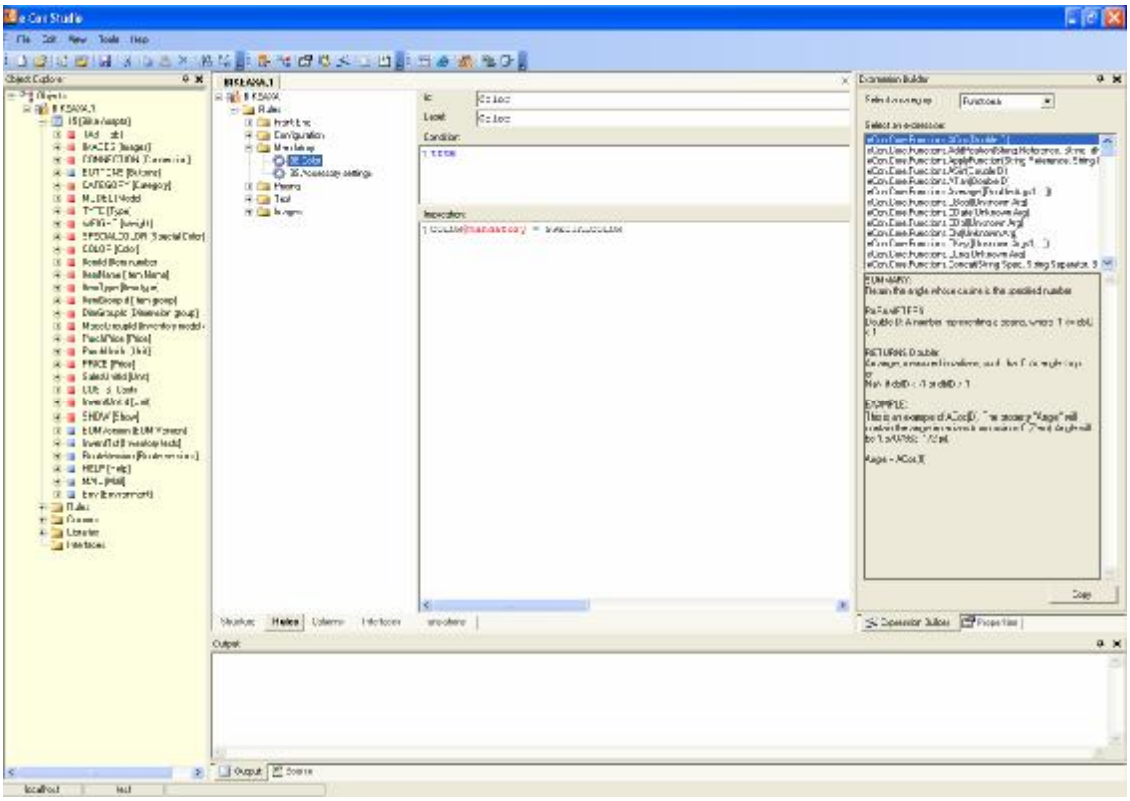
With the e-Con Studio, you can add rules, among other things, to the data structure.

- Select the desired model on the Generic Model form.



- Click the "e-Con Studio" button on the Generic Model form.

The e-Con Studio will then be started:



Note: it is also possible to startup the e-Con studio from the e-Con Model form. Go there with the “Setup Model” button.. Besides that the studio can also be started for a project. Select e-Con Projects and from there ‘e-Con studio’ which is available under the ‘Project’ button

7.10 How to Set Up Decision Matrices

This section describes the Decision Matrix functionality. An overview of this functionality is given, followed by instructions for setting up the matrix, and an explanation of the business rules needed.

7.10.1 Overview

You can use Decision Matrices to indicate an interdependency between member options in your model. In many cases, certain options are not possible for a given Model. "Options" in this case may mean other Options (like Color or Type) as well as Items.

It’s also possible to lay out such interdependencies in the business rules. However, these interdependencies may change frequently, and changing your business rules frequently is not ideal. In addition, it is often Sales Representatives, Assortment Control staff or Product Control employees who need to make these changes, and they are usually not authorized to alter the business rules.

So instead, e-Con makes use of easy-to-use Decision Matrices. By check-marking fields in these Matrices, employees can easily make the necessary changes in interdependencies

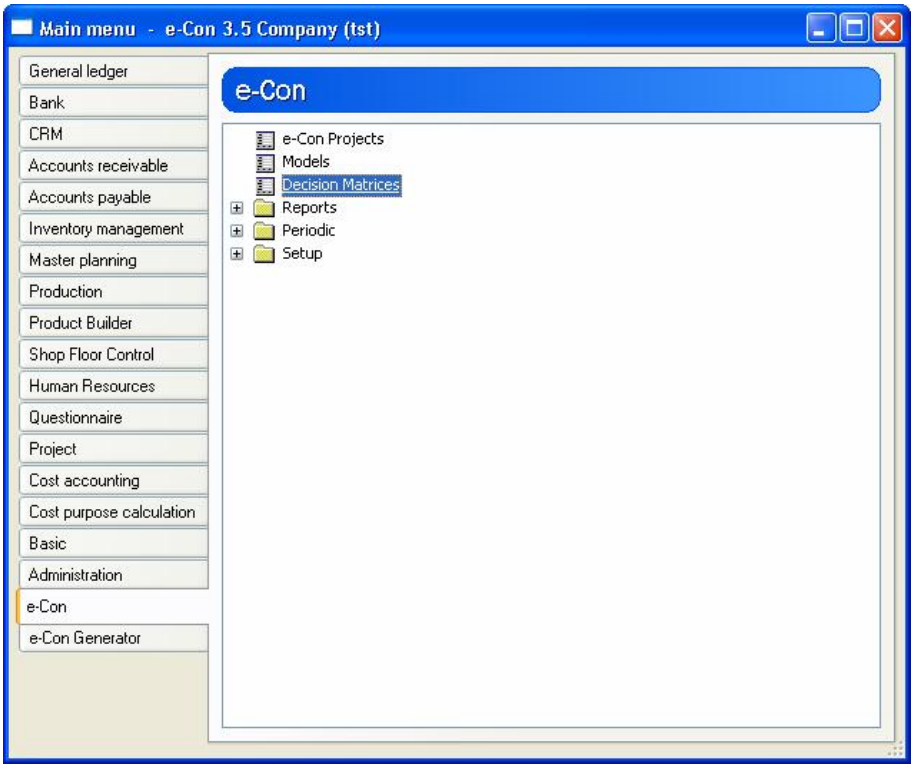
among Entity members. And they need only make use of a few existing business rules to consult the matrix.

7.10.2 Setting up the Matrix

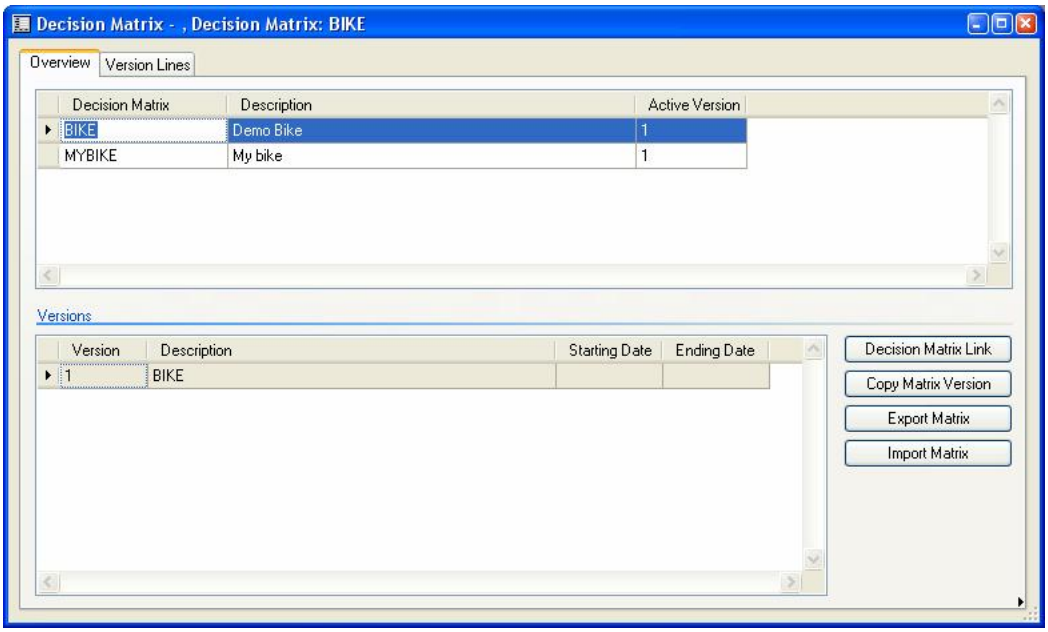
This section covers setting up a Decision Matrix in Axapta.

A decision matrix consists of two or more "Option sets" of which the options are dependent of each other. The Option sets are based on existing queries. These queries can be based on Axapta tables or not.

Setting up a Decision Matrix.

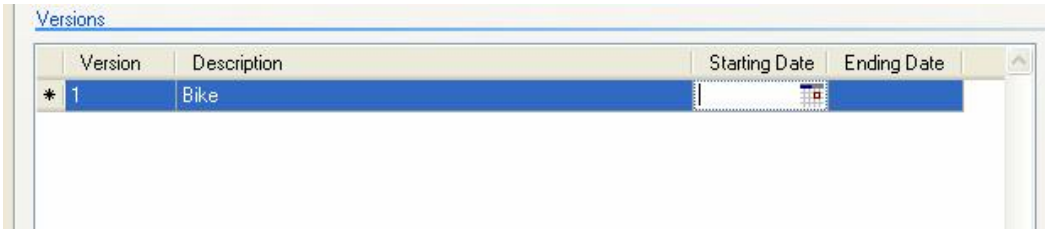


- First select "Decision Matrices" in the e-Con menu.



- The Decision Matrix form for setting up the Decision Matrices and Versions appears. Different Decision Matrices with version can be determined. Every Decision Matrix has an Active Version. The “Option sets” are determined as Version Lines per version.

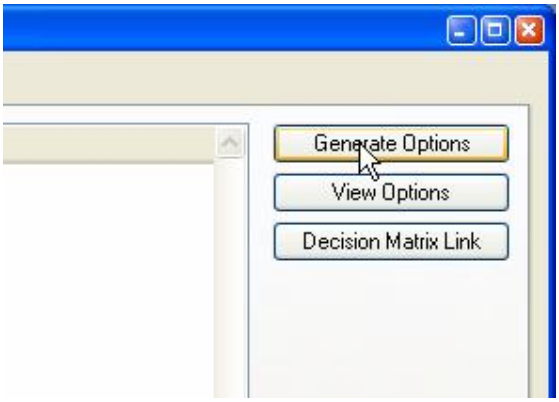
- Add a new line on the Overview tab with CTRL-N and fill a unique code for the Decision Matrix and a Description.



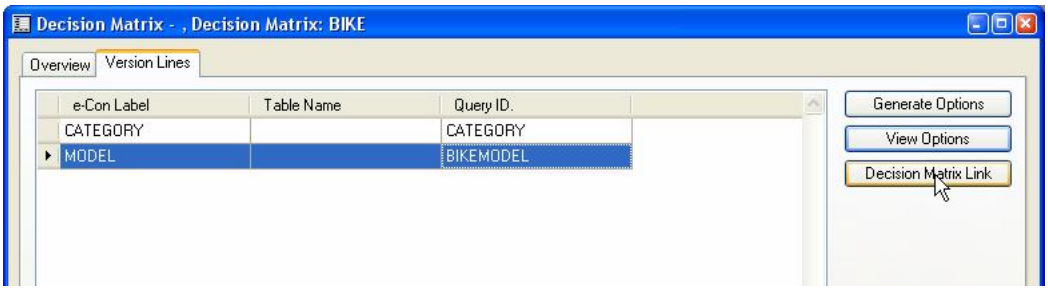
- Go to the Versions and add a new version.

- Select the new version as “Active Version” for the Decision Matrix.

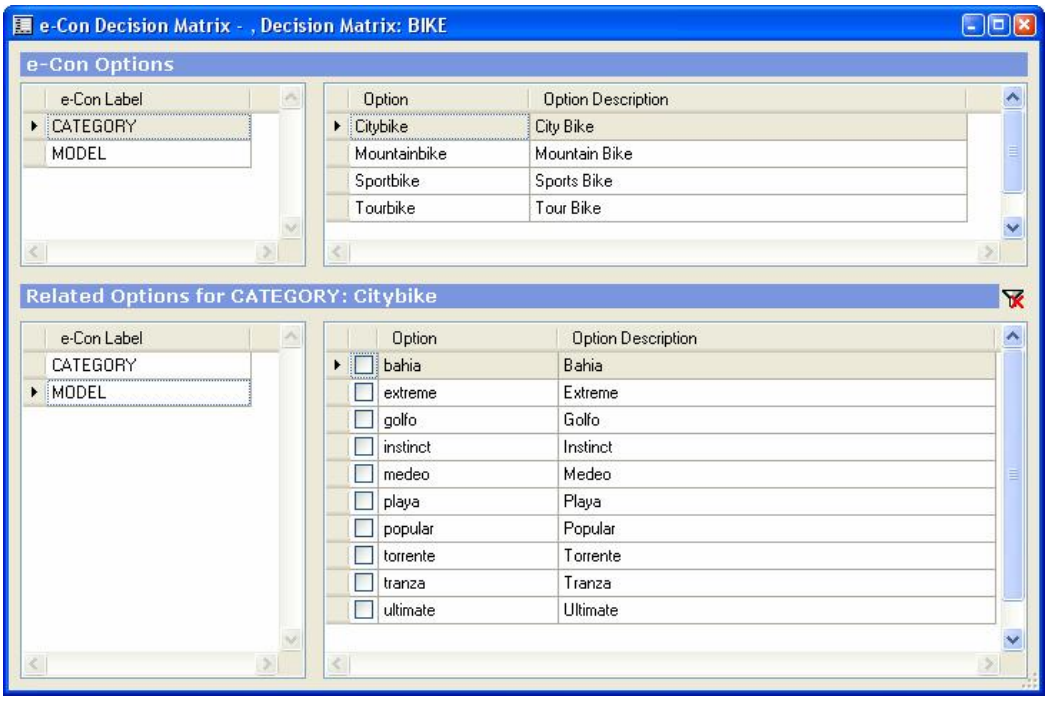
- Go to the Version Lines tab and add the Option sets. Add a new line with CTRL-N. Select the e-Con Label, Table Name (if the query is based on a Axapta table) and Query ID.



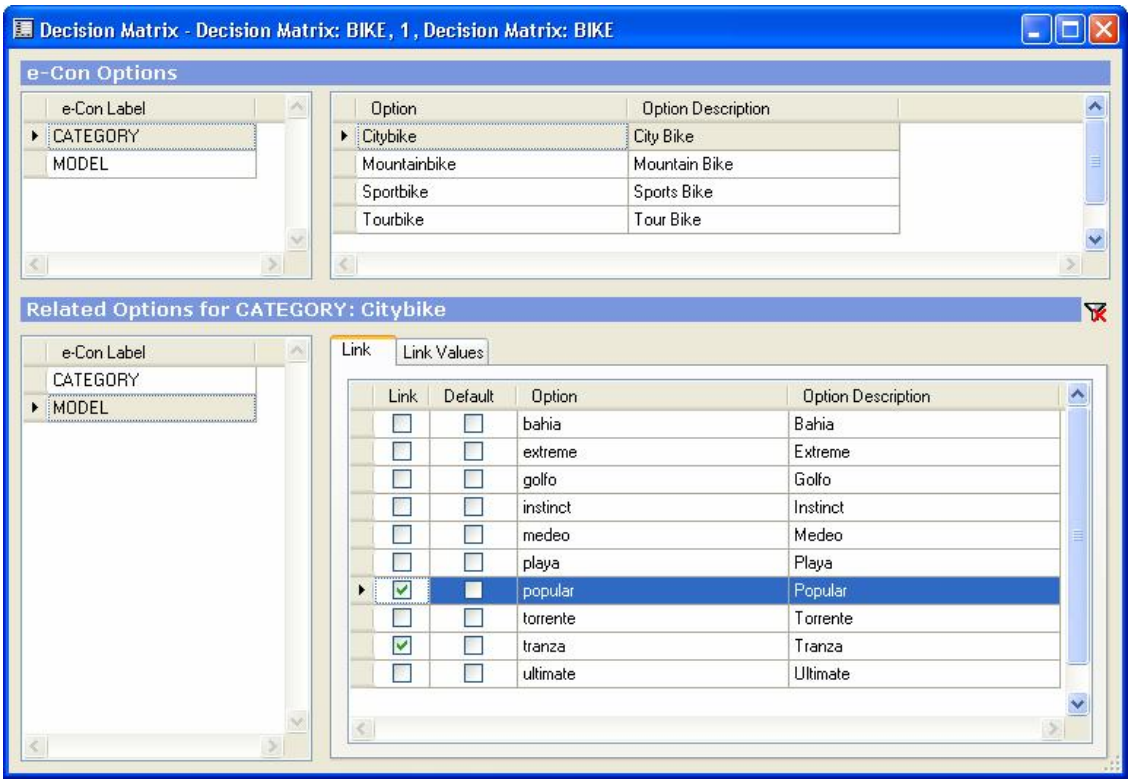
- Generate the options with the “Generate Options” button and view the options.



- Add a second option set e.g. MODEL. Generate the options and check them with View Options. Go with “Decision Matrix Link” to the form were the dependency between the options are set.



- With check-mark in the Boolean fields the relation between the options can be set. For instance Only models “popular” and “tranza” are possible for “Citybike”.



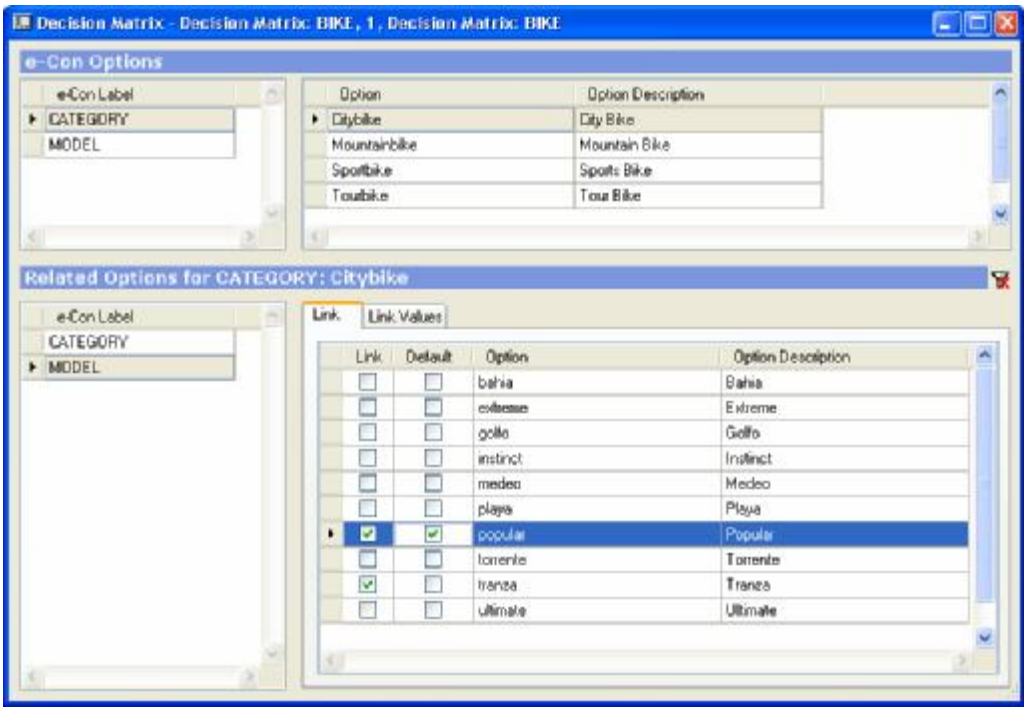
- Fill the Decision Matrix for all the Categories.

7.10.3 Checking the default option in the matrix.

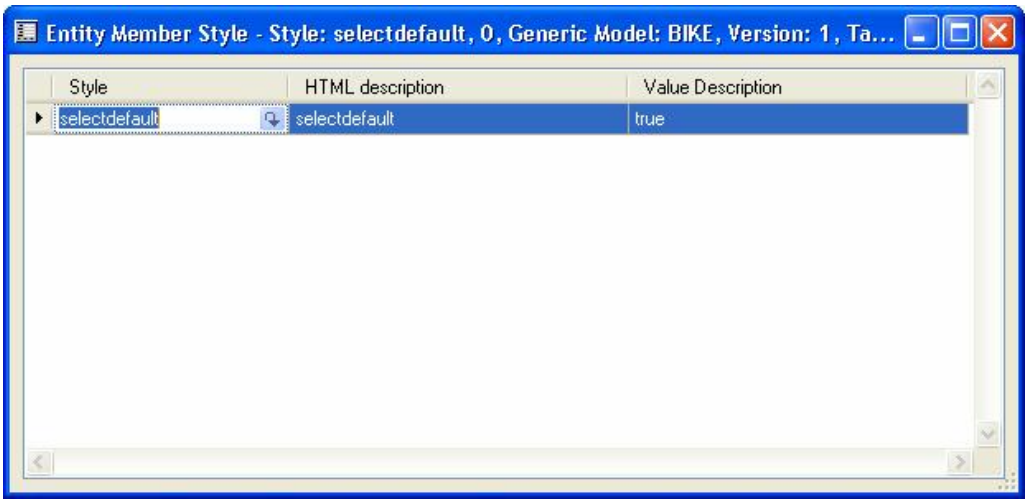
In the Decision Matrix valid options can be checked. Besides that the default option for a particular row can be checked as well.

Two steps are required to add this default behavior to the matrix functionality.

Select the matrix link and check for each row the default option. In this example popular is check marked as the default option for Category ‘Citybike’.



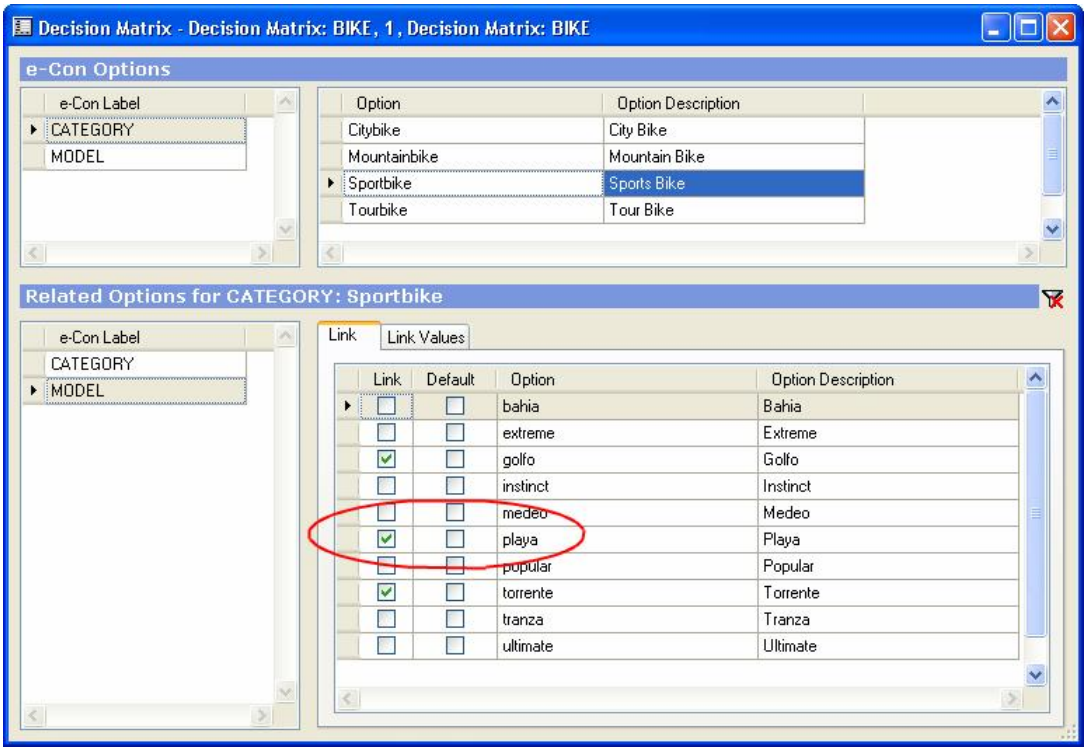
Add a new style to the property representing the Models. In this case the property model. The style to be added is: 'selectdefault' with value 'true'.



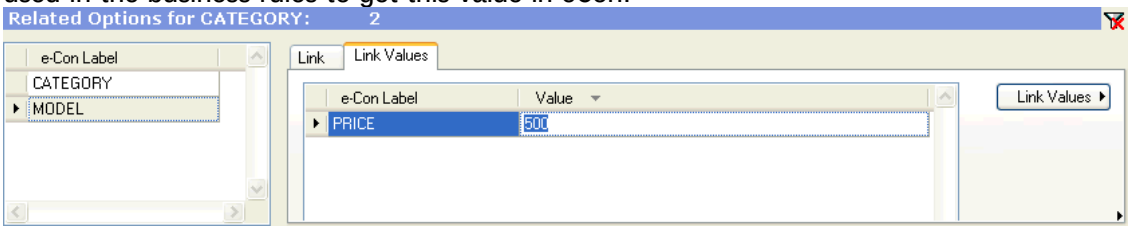
7.10.4 Adding Additional Data to a Selected Check Box in the Matrix

In this section we describe how additional data can be added to a checked field in the decision matrix. This functionality offers you the possibility to add data to options available in the decision matrix. As an example: It's possible to add the prices for all the monitors in combination with the computer model for. That means that the price for monitor vary depending on the computer model the monitor belongs to.

- Select the Decision Matrix as set up in the former step. And select the Matrix form as shown below.



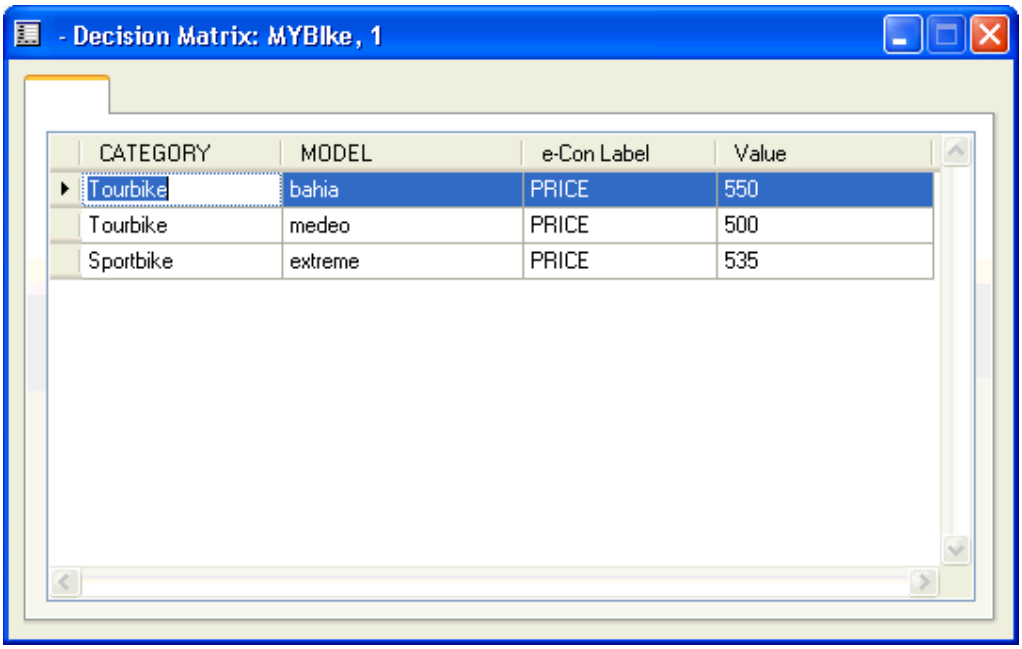
- Select a link for which a check box is selected.
- Open the Link Values tab.
- On the link values tab, enter a label and a value. This data is unique linked to the checked field in the matrix. The label of the value is important because that label is used in the business rules to get this value in eCon.



Tips & Tricks:

To assure that always the same labels are used for the additional data in a decision matrix, use the Link Values Defaults. The Defaults function can be found under the "Link Values" button on the Link Values tab of the e-Con Decision Matrix form. When additional data is entered, those defaults always appear.

There is an overview form available where additional data can be very easily entered. To open this form, on the Link values tab, click Link Values, Overview.



7.10.5 Add the Relevant Business Rules

Now that you've set up the Decision Matrix in Axapta, you will need to add the necessary business rules. These rules ensure that the options are retrieved from the Axapta database for those members that require it.

To set up the business rules, you will use the "@options" attribute and the following functions from the eCon.Data.Functions Library:

- GetMatrixColumns(<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>)
- GetMatrixRows(<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>)
- GetMatrixCell(<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>;<expr>)

The "GetMatrixRows" and "GetMatrixColumns" functions are both used for consulting the decision matrix to retrieve options.

The GetMatrixCell function is used to retrieve the additional data linked to a certain cell in the decision matrix.

- The "GetMatrixRows" function is used for retrieving options that are located in a row of the matrix. The variable value is then located along the x-axis of the Decision Matrix.
- The "GetMatrixColumns" function is used for retrieving options that are located in a row of the matrix. The variable value is then located along the y-axis of the Decision Matrix.

Below are two examples how to use those functions:

Implication:

```
1 CATEGORY@options =
2 GetMatrixColumns("CATEGORY"; true; CONNECTION; "BIKE"; "MODEL"; "CATEGORY"; "="; MODEL)
```

This rule establishes that when the end user selects a Model, the various Category options associated with that Model are retrieved from the Decision Matrix.

Implication:

```
1 MODEL@options =
2 GetMatrixRows("MODEL"; false; CONNECTION; "BIKE"; "MODEL"; "CATEGORY"; "="; CATEGORY)
```

This rule established just the opposite. When the end user selects a Category, the associated Models are retrieved from the Decision Matrix.

- The GetMatrixCell function is used to get information linked to a certain cell of the decision matrix, the so called additional data.

Below you will find an example how to use this function:

Implication:

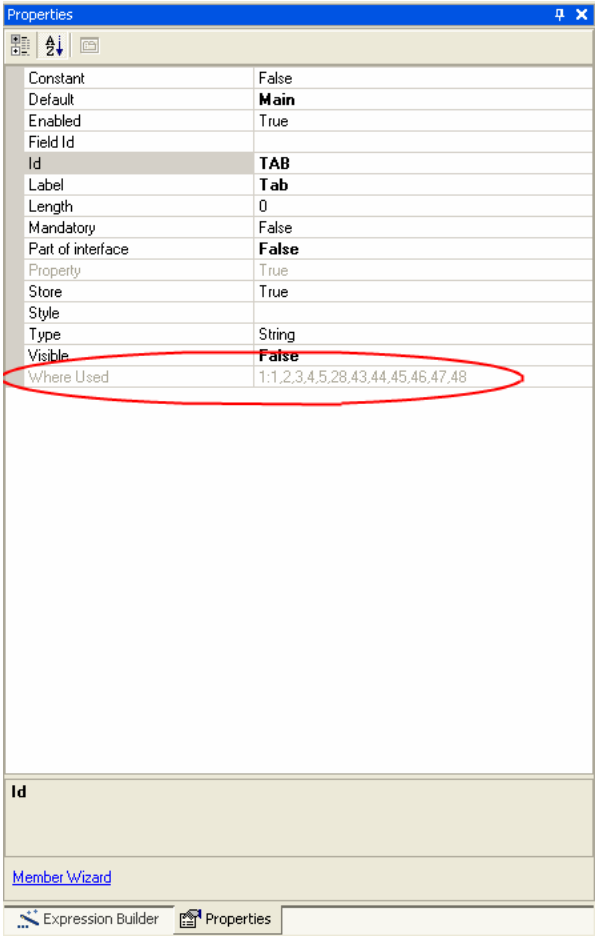
```
1 WEIGHT =
2 GetStringMatrixCell(CONNECTION; "BIKE" ; "Weight"; "CATEGORY" ; "="; CATEGORY; "MODEL" ; "="; MODEL);
```

In this rule the weight is retrieved from the Decision Matrix “Bike”. Based on the chosen Category and Model the right cell is consulted.

For more detailed information about these functions, see the document “Data Functions - Library Descriptions ”

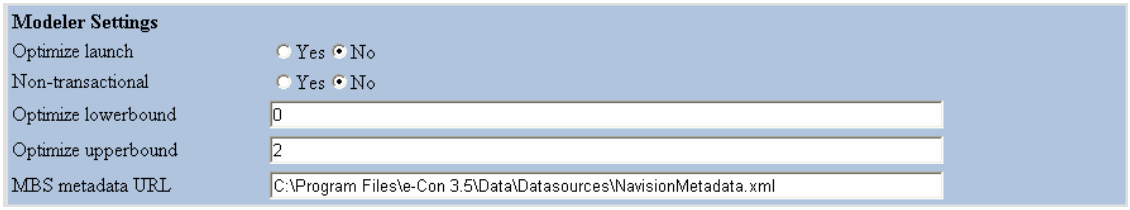
7.11 How to optimize a model

Optimization of the e-Con model means that the so called “Where Used” field from the members is calculated. Actually this means that is calculated in which rules a member is used. These rule numbers are stored in the so called “Where Used” field of the member.



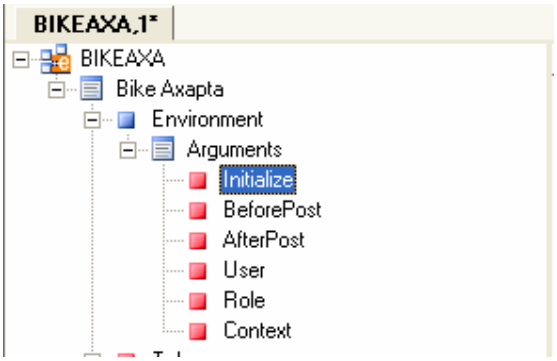
There are three fields in the e-Con settings that have effect on the optimization of the model:

- Optimize launch: With “Optimize launch” only rules with the Env.Initialize condition will be executed. This offers a faster start up of the model. See the next chapter.
- Optimize lowerbound: If there are repeatable entities in the Model (Arrays) the “Array Lowerbound” must be filled up. The value is always 0. This is the also the default value.
- Optimize upperbound: If there are repeatable entities in the Model (Arrays) the “Array upperbound” must be filled up as well. The value depends on the highest array index used in the model. The default value is 2, but if there are rules with an index higher then 2 this must be changed to that value. E.g if there is a rule containing the expression ‘Accessory[14] = 13, the “Array upperbound” must be 14.

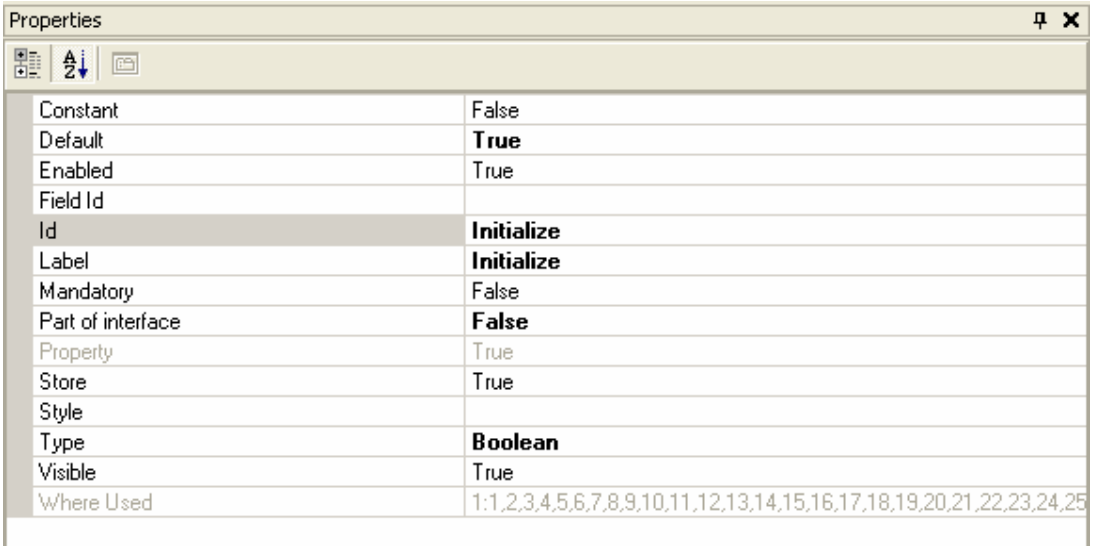


7.12 Optimize Launch detailed information

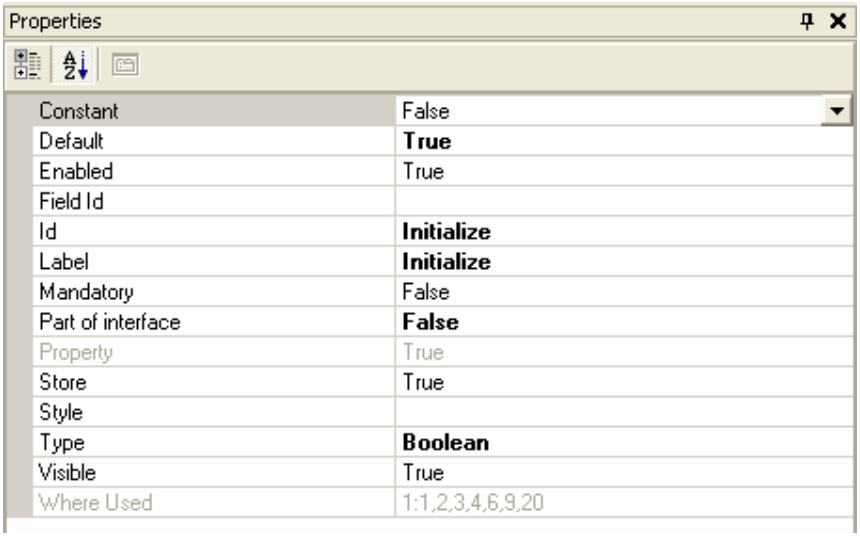
In the data structure is an Environment relation. One of the members is “Initialize”. With update the Initialize property can be opened. If the model is not using the optimized launch, all the rules are executed during launch (startup) of the model. When optimized, only the rules in the “Where Used” of “Initialize” will be executed.



In an optimized model without “Optimize launch” all rule numbers are in the “Where Used”.



In an optimized model with “Optimize Launch” only rules with the “Env.Initialize” condition are in the “Where Used”.



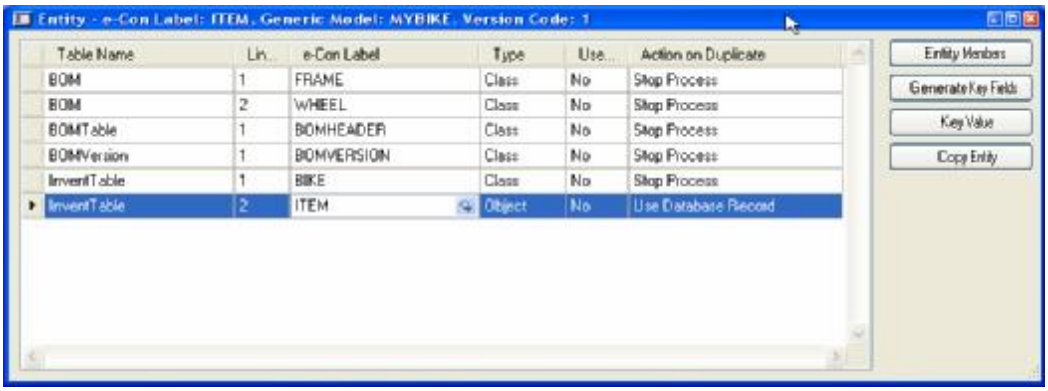
7.13 How to add a pop-up screen

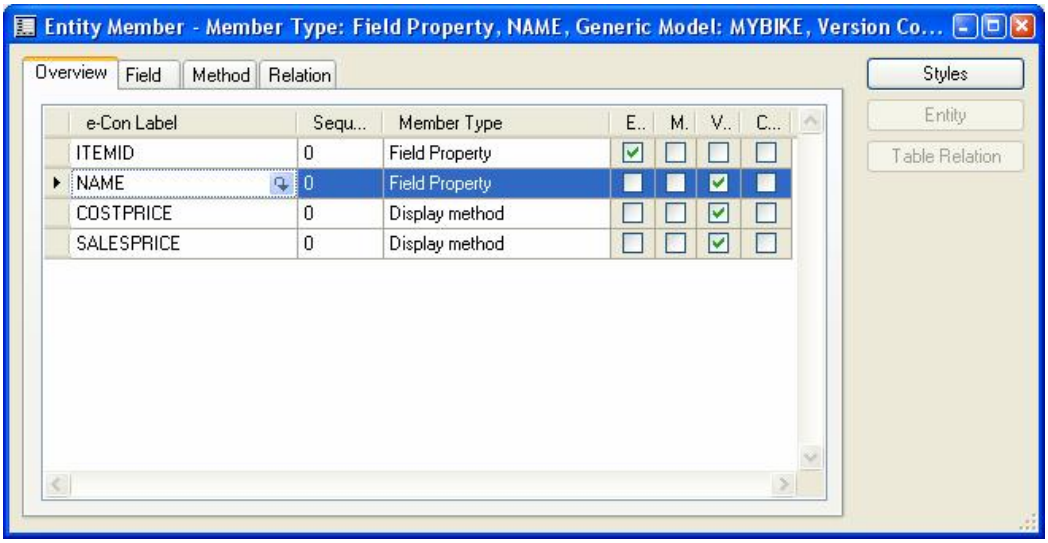
Sometimes more information is needed in order to select an option out of a drop-down-menu in the Front-end. For example, when a customer has to be selected it can be helpful to have more data shown than only the names or search name of this customer. The address, Zip code, City name, etc. might better facilitate the selection of the right customer.

In e-Con this additional information can be shown in a pop-up screen. In this pop- up screen, columns with additional information are displayed. An option can be selected by simply clicking it. The option’s value will automatically take over the member field.

The section below describes how to add this functionality:

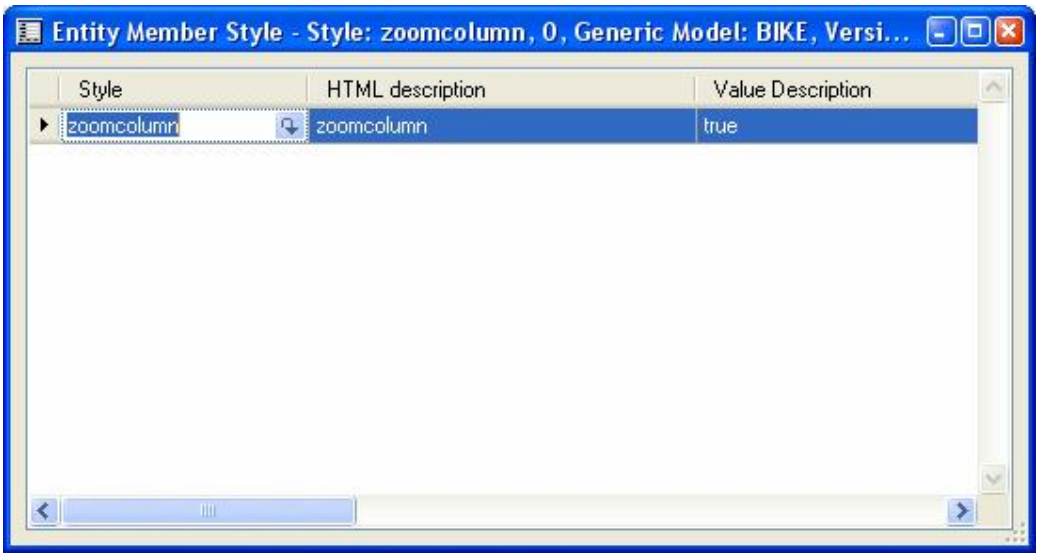
- First make sure that the additional information that has to be available in the pop-up screen is a property of the option member. For example the item name, item cost price and item sales price have to be properties of the option member item. To have more selection data.
- Before continuing a short description is given how to add such properties to a member. The frame selection will be used as an example:
    - Add new entity members to the InventTable-2 entity of the example of chapter 4. These are the properties to be shown in the pop-up screen.



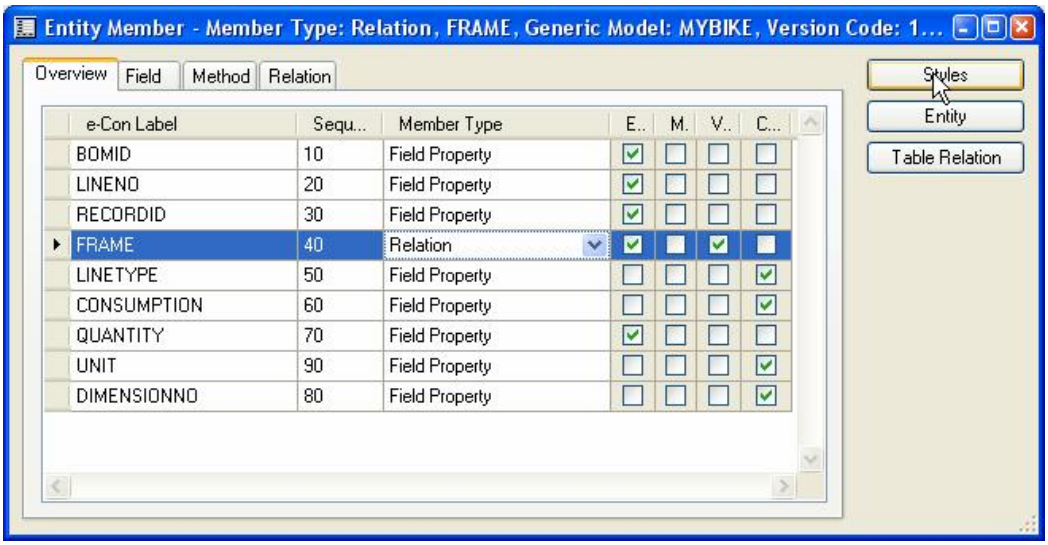


NAME	Field Name	ItemName
COSTPRICE	Display Method	CostPcsPrice
SALESPRICE	Display Method	SalesPcsPrice

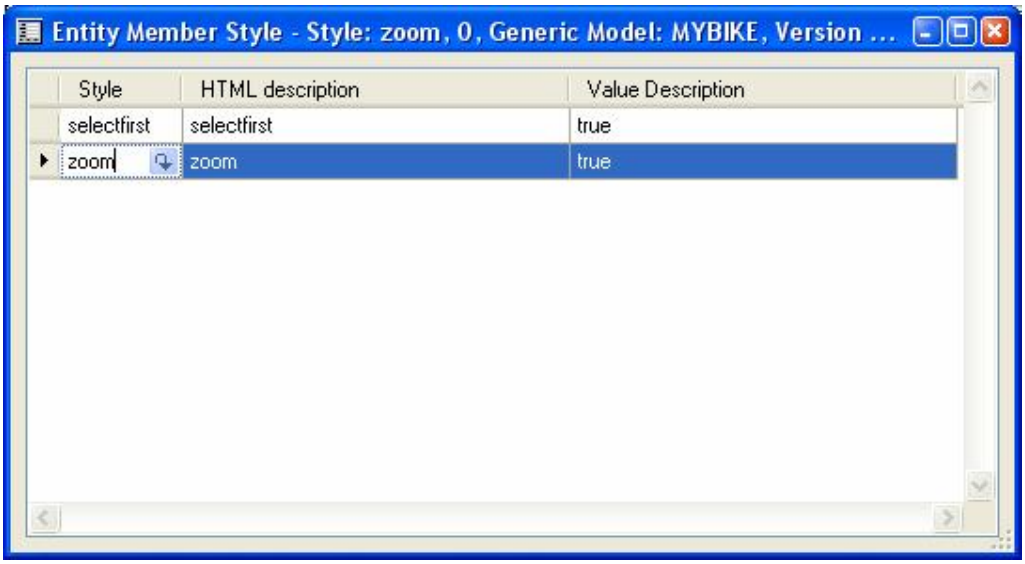
- To let those properties appear in the pop-up screen a style has to be added to all of them. Adding a style can be done in a business rule or in the data structure. The latter is described hereunder. Select the member for which the style has to apply
- Select styles in the member menu.
- Fill in the style ‘Zoomcolumn” with the value “true”.



- Last step is to add the style zoom with value “true”. This can be done in a business rule or in the data structure. The latter is described hereunder. Select the member for which the style has to apply, in our case the relation on the BOM-1 (FRAME) entity to the InventTable-2 entity.

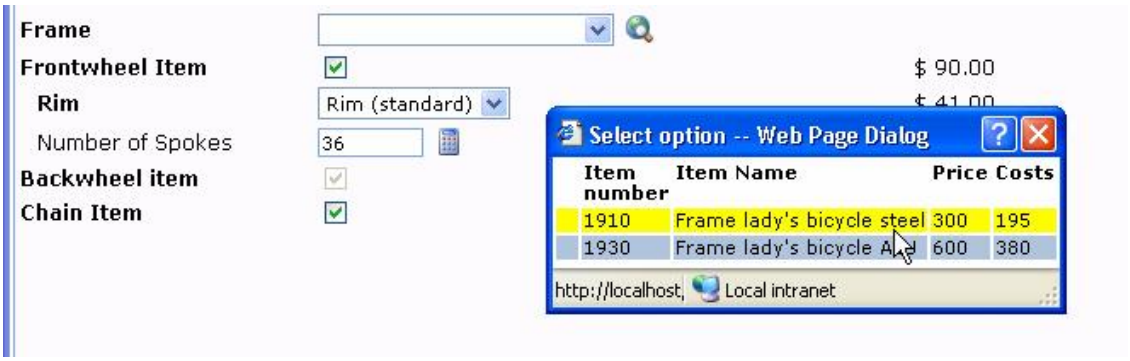


- Select styles in the member menu.
- Fill in the style zoom with the value “true”.



- Certify and generate the data structure and check the result.

An extra button is added to the member frame to start up the pop-up screen:



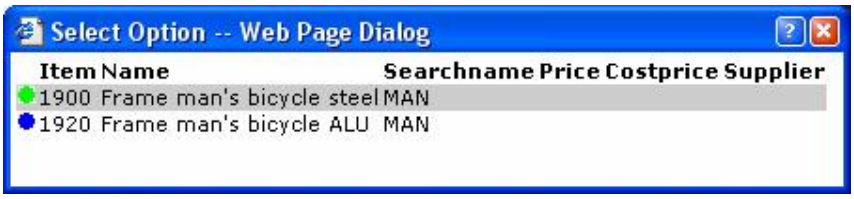
7.13.1 How to add to show my default, former and actual option in the pop up screen

Additional information can be shown in the pop up screen. Information about:

- Default Option
- Actual selected option
- Former selected option

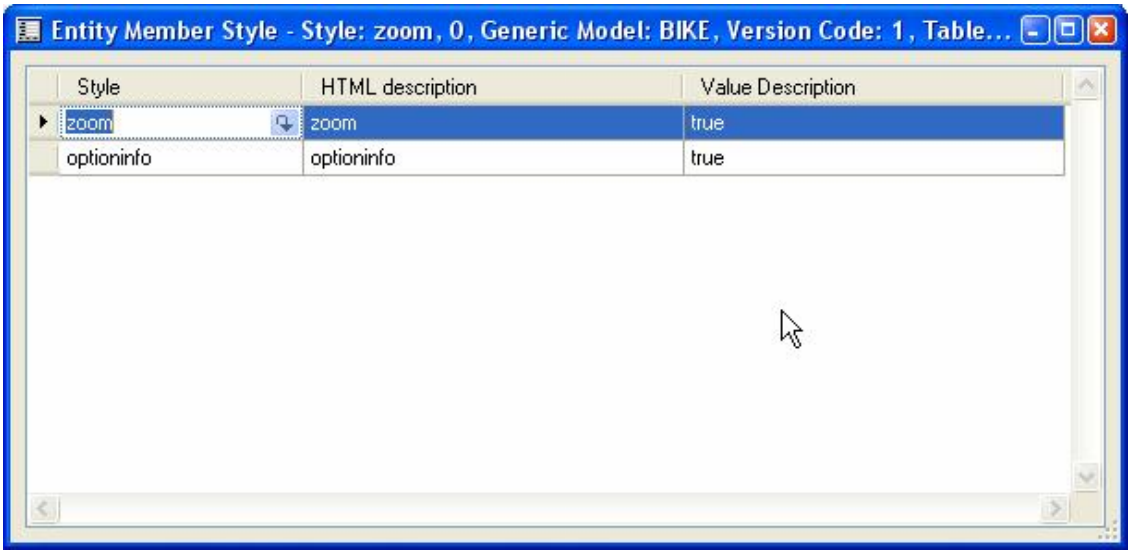
This information is shown as a colored bullet in front of the option. The default options have a yellow bullet, the actual selected options a blue one and the former selected option has a green bullet.

See the example below:



To add this functionality is not that difficult. It's all controlled by just one style. It's the style "Optioninfo" with value "true".

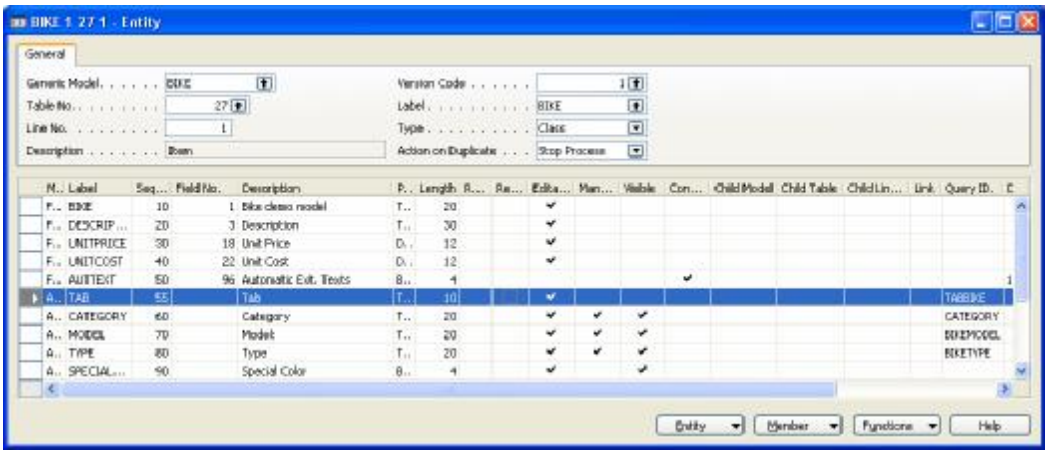
- Select the member containing the style zoom.
- Add the style "Optioninfo" with value "true".



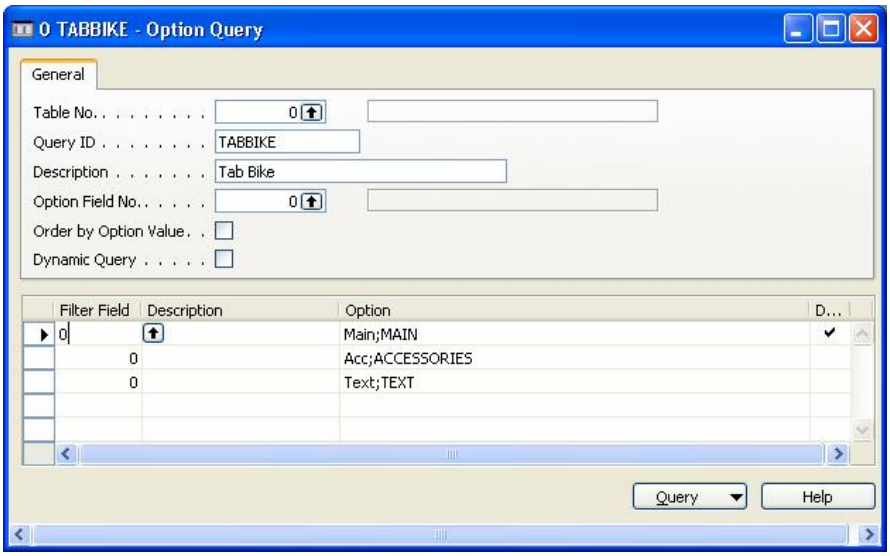
7.14 How to add Tabs in the Front End of my model

Sometimes when there are too many fields in the Front-end, it can be useful to present them in a more structured way. Classifying these fields into tabs can help us in such situations. Within each tab a logical grouping of fields can be displayed. This offers a better overview of all fields available in the model. The way of creating the tabs is described hereunder.

- First, add a new member with the label TAB. The member type is additional property.



- Add a query to that new member. The options of the query represent the tabs in the Front-end. In this example three tabs are chosen: Main, Accessories and Text.



- Certify and generate the data structure.
- The Last step is to add intelligence that will control which fields are to be displayed on which tab. Start the studio and add a new rule. See the example hereunder. In this example, the member Category is only visible within the tab “Main”.

Id: Tab == Main

Label: Tab == Main

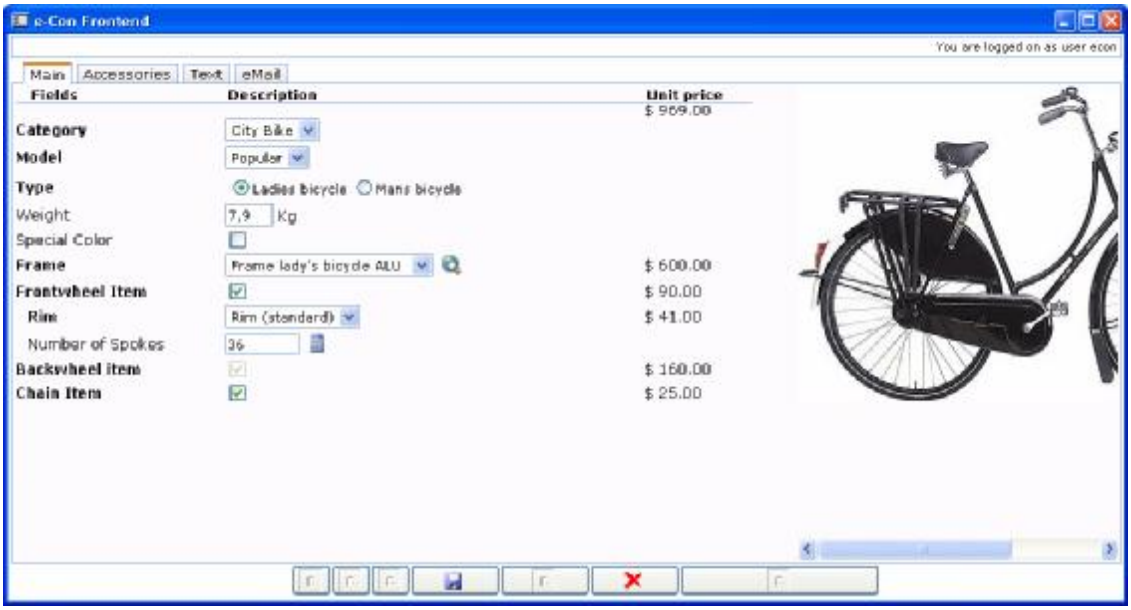
Condition:

```
1 True
```

Implication:

```
1 CATEGORY@visible = TAB == "Main";
```

This will result in a Front-end that looks like this:



Tips & Tricks:

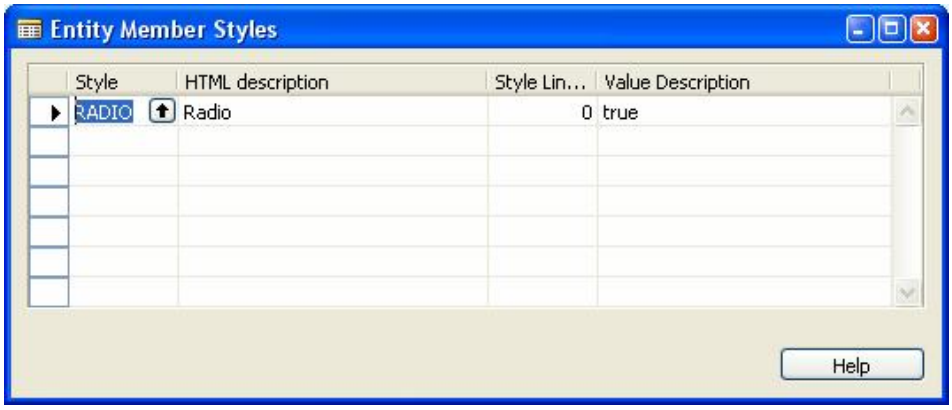
With the use of the attribute <memb>@option(<expr>)@visible it is possible to hide or show a tab in the front end. Simply fill in TAB as <memb> and the option that represents the Tab as <expr>.

7.15 How to add Radio Buttons

It's possible to display options as radio buttons instead as a drop down box in the e-Con front end. In this section is described how this functionality can be added.

- First select the member in the data structure used to be shown the radio button options.

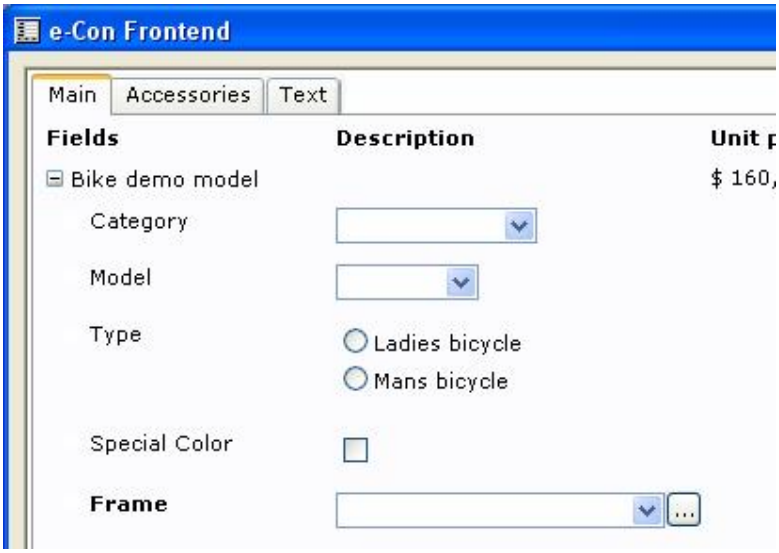
- Now a style has to be added. This can be done in a business rule or in the data structure. The latter is described hereunder. Select styles in the member



menu.

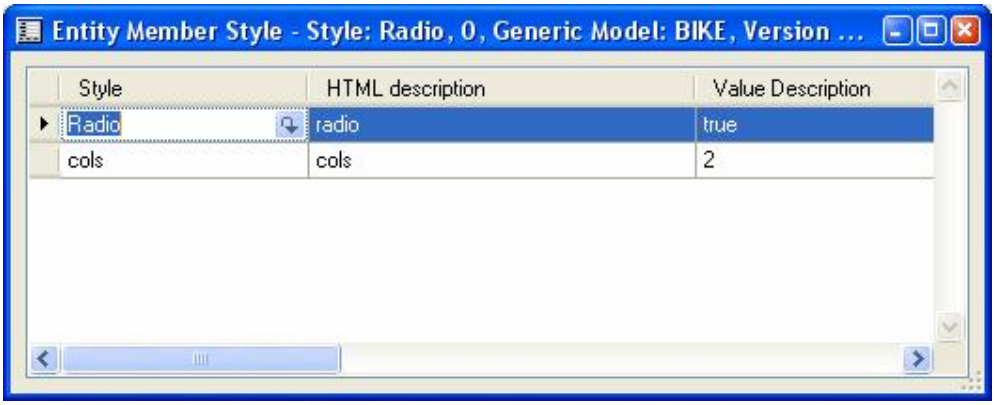
- Fill in the style 'RADIO' with the value description "true".
- Leave the screens and go to the Generic Model.
- Certify and generate the data structure and check the result. Leave the screens and go to the Generic Model.

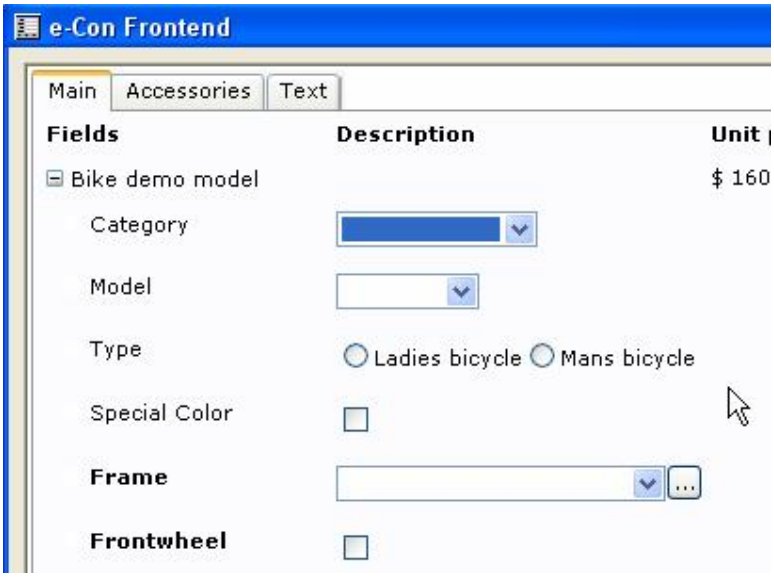
Test the result by configuring the model. As you the options will be displays as radio buttons. In our example the type of the bike is shown as radio buttons.



The buttons itself are shown under each other. It is also possible to changes this appearance. By adding the style Cols too, it's possible to set up how to display the radio buttons. The value of the style cols, a numeric value, determines the number of horizontal aligned radio buttons.

As an example below the outcome of the style ‘Radio’ with value ‘true’ and the style ‘Cols’ with value 2 added to the member ‘Type’ in the demo bike model.





7.16 How to add and change Buttons in the Front End of my model

In the e-Con front end several buttons are standard available depending on the content of the field “Buttons” in the e-Con Settings.

UI Settings	
Buttons	?+-PBRE

Each letter represents a button in the UI. The next letters and corresponding buttons are available:

?: How and Why ?

+: Expand all

-: All Collapse

P: Print

B: Process (Save of XML document at file system and process document in Attain)

R: Restart

E: Exit

S: Save (Save of XML document)

T: Save template and exit

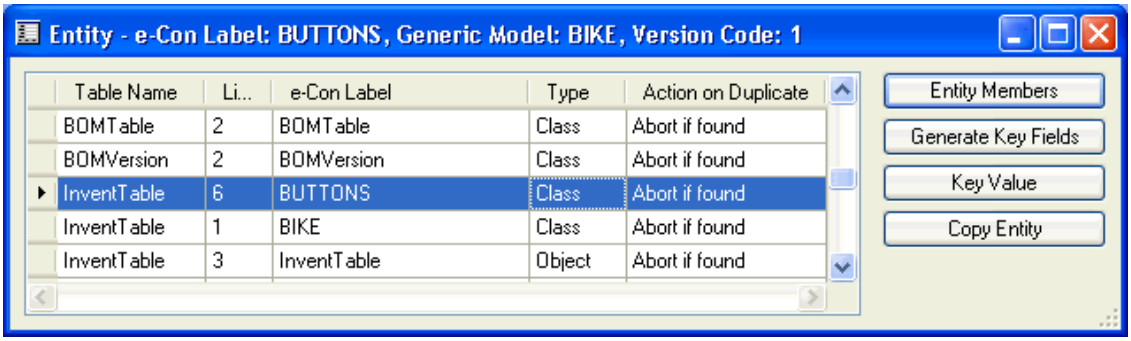
Q: Post

X: Post and exit

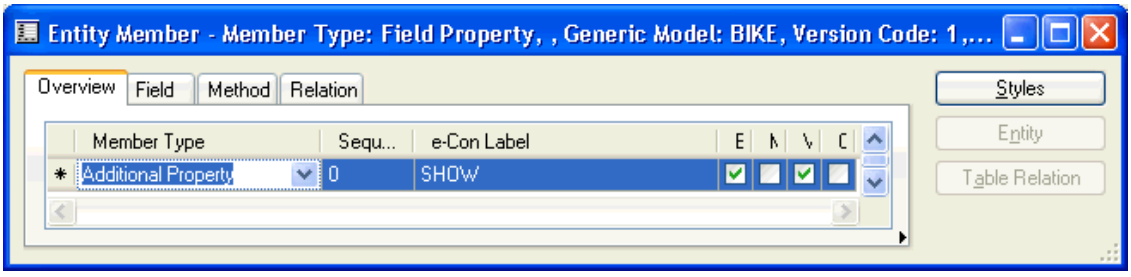
As mentioned those buttons are standard available in the UI. However it's possible to change the buttons. The ordering can be changed, buttons can be skipped, extra commands can be added to the buttons and new buttons can be added.

In this section is explained how buttons can be add and changed.

- First, add a new entity with the label BUTTONS. The entity table is not important so just select one.



- When e-Con find a entity with label buttons it knows that custom buttons will be available and therefore the standard buttons will not be shown. Just add those buttons as properties from the new entity buttons. Also add a property with the label "SHOW". This is a new button used later on.



The description of the label is the label shown at the button. The print button is in this case not added. To view the label's description, browse in the e-Con Label field.

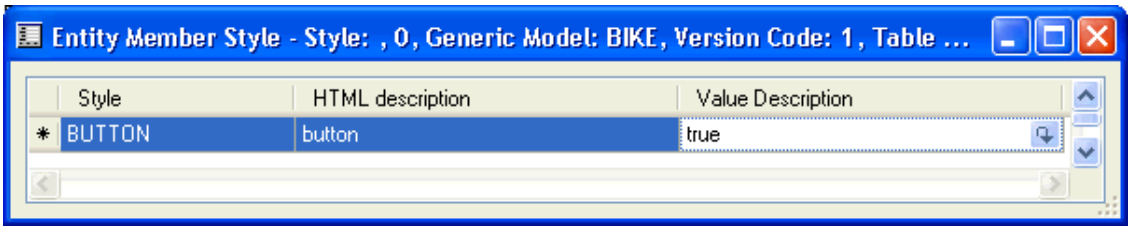
Make the properties visible when you want them to display this button.

Tips & Tricks:

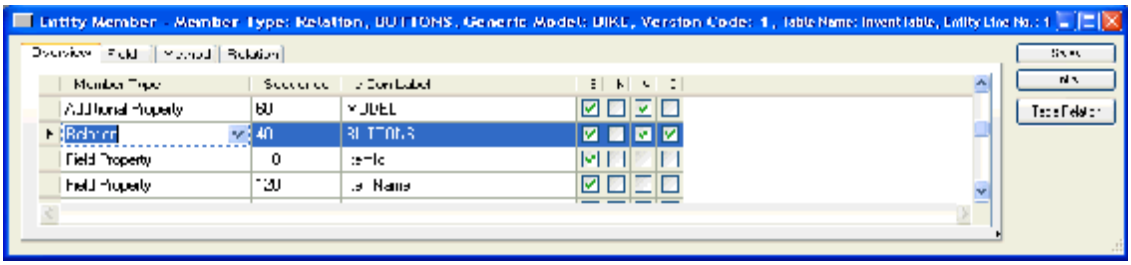
The sequence of the properties of the member button determines the order of the buttons in the front end. The property on top appears left the one at the bottom appears right.

Instead of checking the visible field up here it is also possible to make rules to let buttons appear. In that way you can let up appear under condtions.

- To let e-Con that those properties are buttons and not 'normal' fields add the style "button" with value "true" to each member that has to be displayed as a button. The styles can be found under the 'Styles' button.



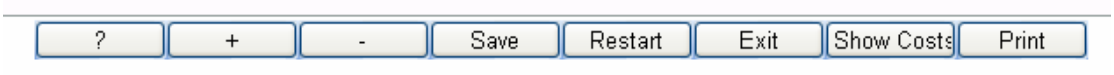
- What's left is to link this new entity to the root entity 'InventTable-1'. Add a new property from the type additional relation. Set up the relation between the entity 'InventTable-1' and 'InventTable-6'.



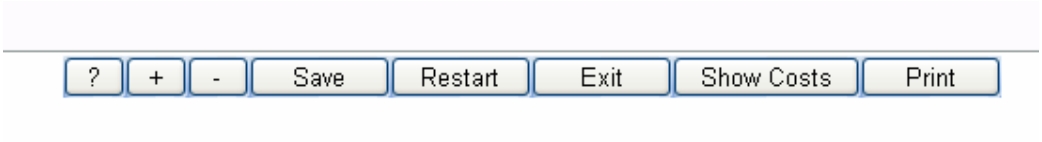
- Check constant and visible.

- Certify and generate the data structure.

- Check the result by configuring the model. As you will see the new buttons will be shown instead of the normal buttons.



- The sizing of the buttons can be better. The label "Show Costs" for instance is not shown completely. The size of the buttons can be controlled with the style "Cols". The value of the style determines the width of the button. Apply the stye "cols" with value "5" to the members "HOWWHY", "EXPANDALL" and "ALLCOLAPSED". Apply this style with value "15" to the member "SHOW". This will result in:



- The Last step is to add intelligence that will control what happens when a certain button is clicked. Maybe you have found out in the former step that even the standard button functionality isn't working anymore.

- Open the e-Con studio and add a new rule.

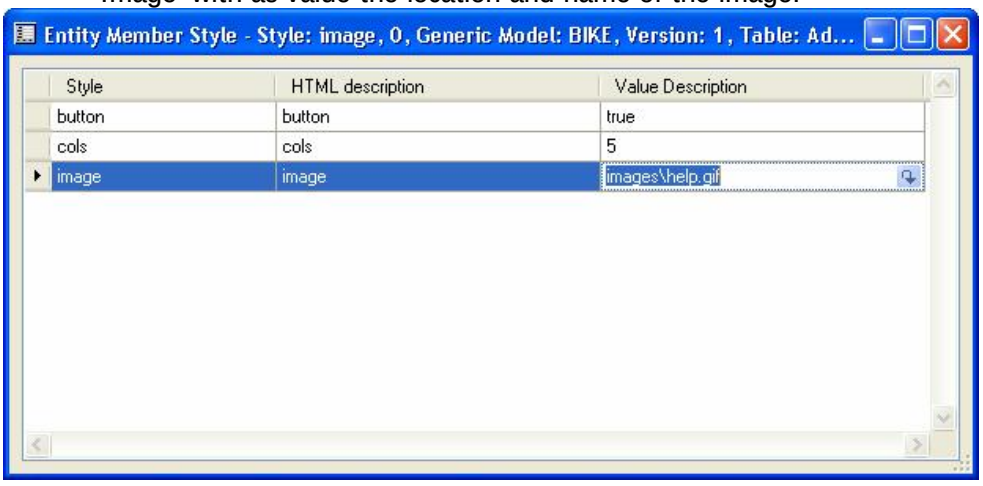
- The task that has to be executed by clicking a button has to be assigned as a string value to the particular member that represents the button. There are several standard tasks available to be executed. Besides that java scripts can be added to buttons as well.

For detailed information about tasks available, see the document "e-Con 3.5 Tasks Description".

7.16.1 Adding an image at a button

Instead of the labels of the button properties displayed at as button images can be shown up as well. To display an image at an button instead of the label add the following style to the button property:

- 'Image' with as value the location and name of the image.

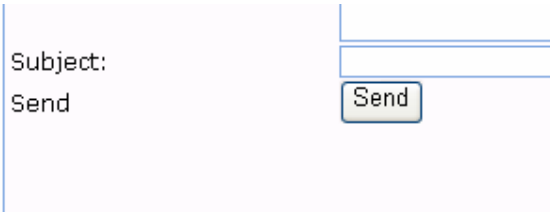


In above example the image help.gif is displayed at the button. This image is located in the folder 'image' available in the web folder of e-Con. The web folder is default "C:\Program Files\e-Con 3.5\Web\

7.16.2 Other buttons

Two other type of buttons exists as well. These buttons can be added by using the following styles:

- Inlinebutton with value true. The property is shown in the UI as a button. This button is shown in the UI inbetween the other fields.
- Valuebutton: This style can only be applied to Boolean fields. The Boolean field with this style is displayed as a button. Clicking the button wil tokkle the value of the Boolean field.



7.17 How to add a Calendar, Calculator or Rich Text editor to a member

To easily select a date for a field in the e-Con front end it's possible to add a calendar to that field. With the aid of the calendar a date can be selected that will taken over in the field itself.

It's also possible to add a calculator to a certain field. With the aid of the calculator a certain value can be calculated. That value will be taken over in the field in the e-Con front end when you leave the calculator and return to the e-Con front end.

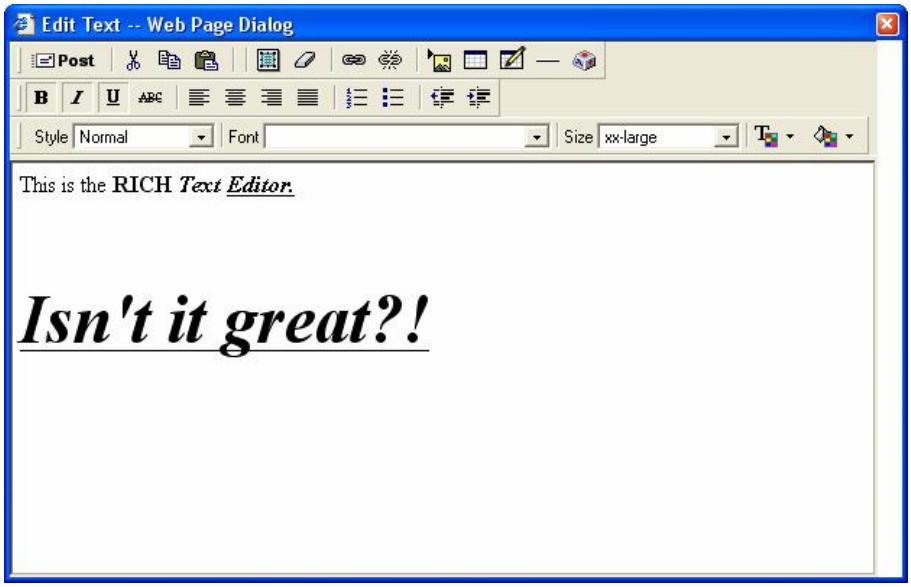
Further more a rich text editor can also be added to a member. In the rich text editor text can be added. A lot of text editor functions, as bold, italic and underlined fonts, bullets and numbers etc. is offered in that editor. By posting this text the editor will be left and the text is taken over in the member.

In this section is described how this functionality can be added to an e-Con model.

- First select the member in the data structure used to be shown the calendar or calculator.
- Now a style has to be added. This can be done in a business rule or in the data structure. The latter is described hereunder. Select styles in the member menu.
- Fill in the style "customselect" with the value:
  - "calendar" for adding a calendar
  - "calculator" for adding a calculator
  - "editor" for adding a rich text editor.
- Leave the screens and go to the Generic Model Card.
- Certify and generate the data structure and check the result Leave the screens and go to the Generic Model Card.

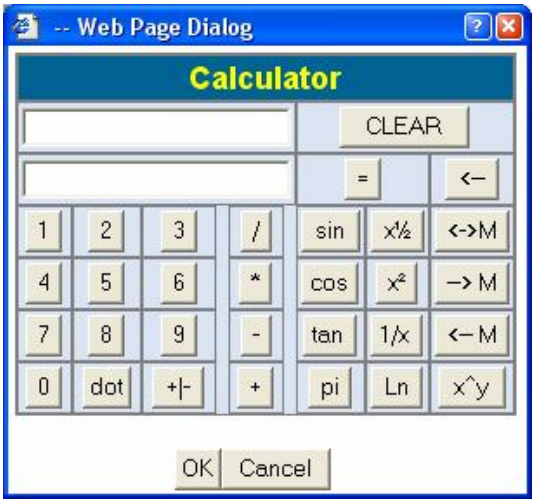
Test the result by configuring the model. As you will notice an assist button is added to the member. By clicking this assist button the calendar, calculator or text editor will appear.

Example: the rich text editor. By clicking the post button the editor is closed and the text is added to the e-Con member.



Example: the calendar. By clicking the OK button the calendar is closed and the selected date is taken over to the e-Con member.

Example: the calculator. By clicking the OK button the calculator is closed and the calculated value is taken over to the e-Con member.



7.18 How to change the style of the questions and fields in the Front End

Normally the questions and answers of e-Con appear in a certain standard font and style in the Front End. Only the mandatory fields appear in bold.

However it is also possible to use your own font's and styles and add to the labels and fields appearing in the front end. Other fonts and styles can be added to the style sheet of e-Con (e-Con.css) and can be applied to a member due to a style.

Use the style "controlclass" and "labelclass" to apply styles to both the label and the field (Question and Answer).

A few styles are standard available in the e-Con.css:

- eConMemberItalic Font style italic is applied
- eConMemberBold Font weight bold is applied
- eConMemberSmall Font size extra small is applied
- eConMemberLarge Font size large is applies
- eConMemberStrikeThrough The text decoration line-trough is applies
- eConMemberRight The alignment of the text is right
- eConMemberCenter The alignment of the text is center
- eConHighPriority The background color red, the font colour yellow
- and the font weight bold are all applied
- eConMediumPriority The background color forest green and the font colour yellow is applied
- eConLowPriority The background colour light grey and the font colour yellow is applied

If more styles are needed or your own style you have to define them in the style sheet from e-Con (e-Con.css)

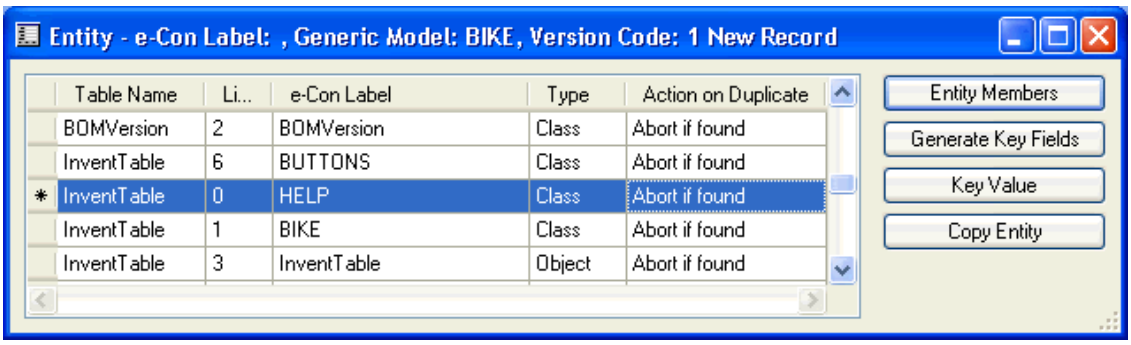
7.19 How to add help text to fields in your model

Within e-Con it's possible to add help text for your model. This help texts can be added for each field in the UI. There are two possibilities to display those help texts:

- As tool tips: Moving the cursor over the field in the e-Con UI will display the help text as a tool tip. (This kind of help text is added to the demo model BIKE)
- As a separate button displayed at the right side of the field. The user has to click on this button the have the help text popped up.

To add these help texts the next steps should be performed:

- First, add a new entity with the label HELP. The entity table is not important so just select one.

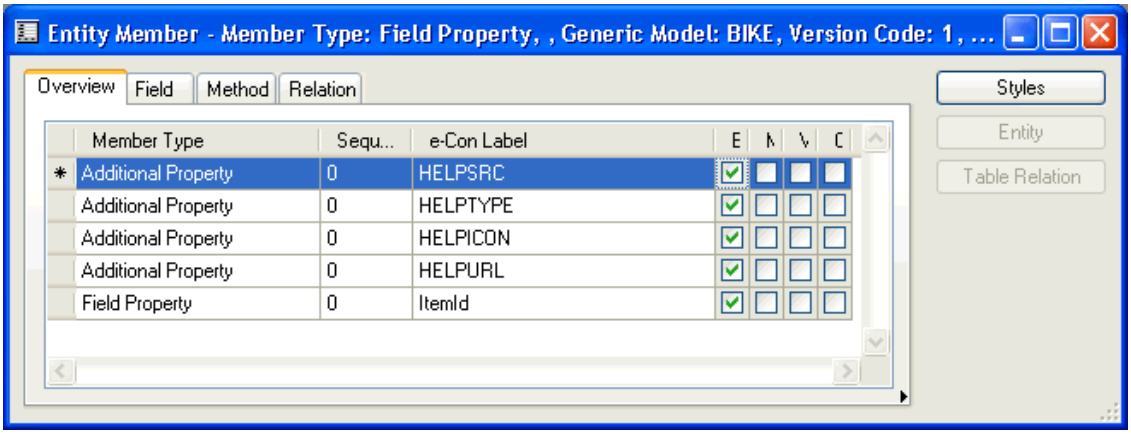


- Generate key fields for this entity and add the next members:

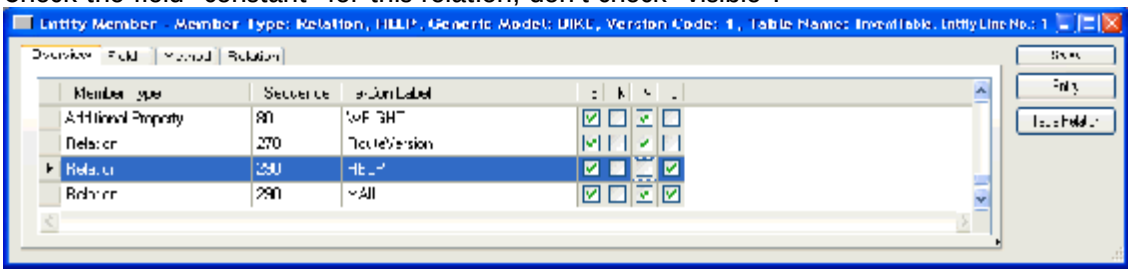
- HELPURL
- HELPICON
- HELPTYPE
- HELPSRC

Those members are required to set up the help texts. Later on, values are assigned to those members by means of the e-Con rules.

- The names of the labels comes very precise since e-Con search for properties with these names to determine the settings for the help functionality.



- What's left is to link this new entity to the root entity 'InventTable-1'. Add a new property from the type Relation. Set up the relation between the entity 'InventTable-1' and 'InventTable-7'. The label of this new relational property **must** be "HELP". Here as well, e-Con is searching for a reference "Help" to determine help settings. Check the field "constant" for this relation, don't check "visible".



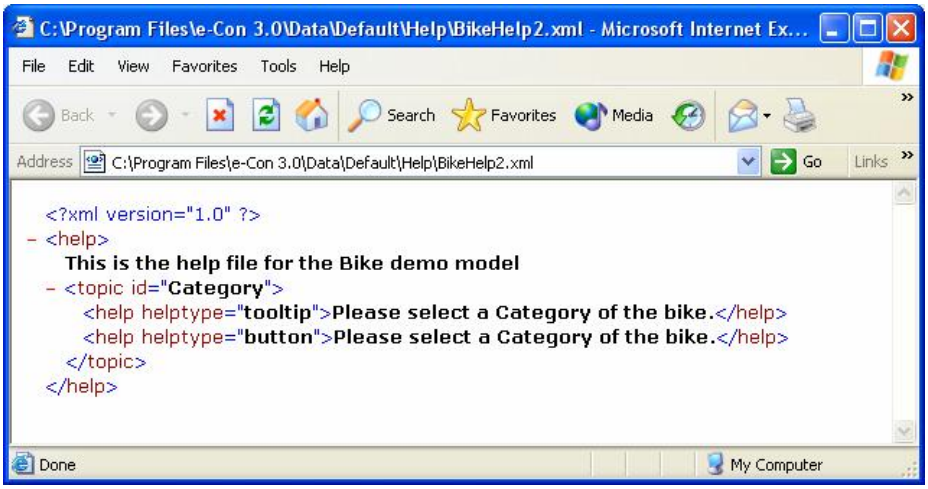
- Generate the data structure and open the studio.
- In the studio add a rule to assign values to the properties of the "help" entity. These values determines the settings and behavior of the help functionality. The next values must be applied to these properties:
    - HELPURL: Determines the name of the help file. The location of the help file is the folder "Help" as available in the e-Con data folder. The file as specified here must be available here. The set up of this help file will be described later on in this section.
    - HELPICON: The icon and location oft this file to be displayed as help button. The standard installed icon is "help.gif" and is located in the "images" folder from the "e-Con 3.0/Web" folder.

- HELPTYPE: Two values are allowed here: “tooltip” and “button”. By choosing “tooltip” the help will be displayed as tool tips from the particular fields in the e-Con UI. The value “button” will add a help button at the right site of the fields in the e-Con UI.
- HELPSRC: The value of this member must always be “external”. This means that the help must be retrieved from an external source.

Tips & Tricks: It’s also possible to assign the values in the data structure. Simply enter the values in the “default code” fields from the properties of the help entity.

The final required step is to set up the help file itself. Advised it to use the help file from the bike demo model as an example. This file can be found in the .”.\e-Con 3.0\Data\Default\Help” folder. The help file is a XML file itself with a certain predicted structured.

Let’s use the next help file as an example to explain how to build up this file:



- Add a <topic id = “”> tag for each field (property) in the UI where help is needed. In above example help is needed for the field Category. Important notice: In the help

file the field must be start with a Capitol letter. e.g.: Category, Production.Frame.Frame.

- Add at least one <help helptype=""> tag. The content of the tag determines if the text following by the tag is for displaying on a tooltip or for displaying under a button. However if the help must be shown as tooltip or button is set up in the model.

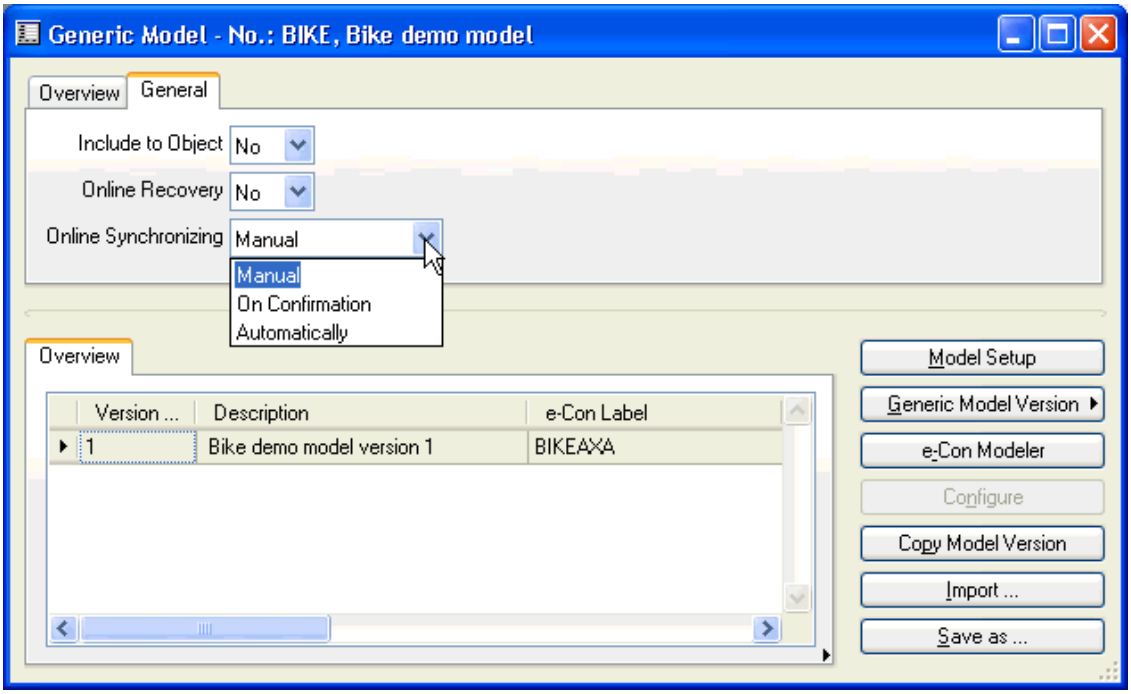
- Add the help text it self in between the <help helptype=""> and the </help> tags.

7.20 How to synchronise my configurations against the latest model

During the life cycle of a model many changes, upgrades and extensions of a model take place. Sometimes it's required to update existing configurations with these latest model changes, sometimes it's not. Such an update is called "Synchronization" in e-Con terms. There are different ways how synchronisation can take place. The section explains the different possibilities in this area.

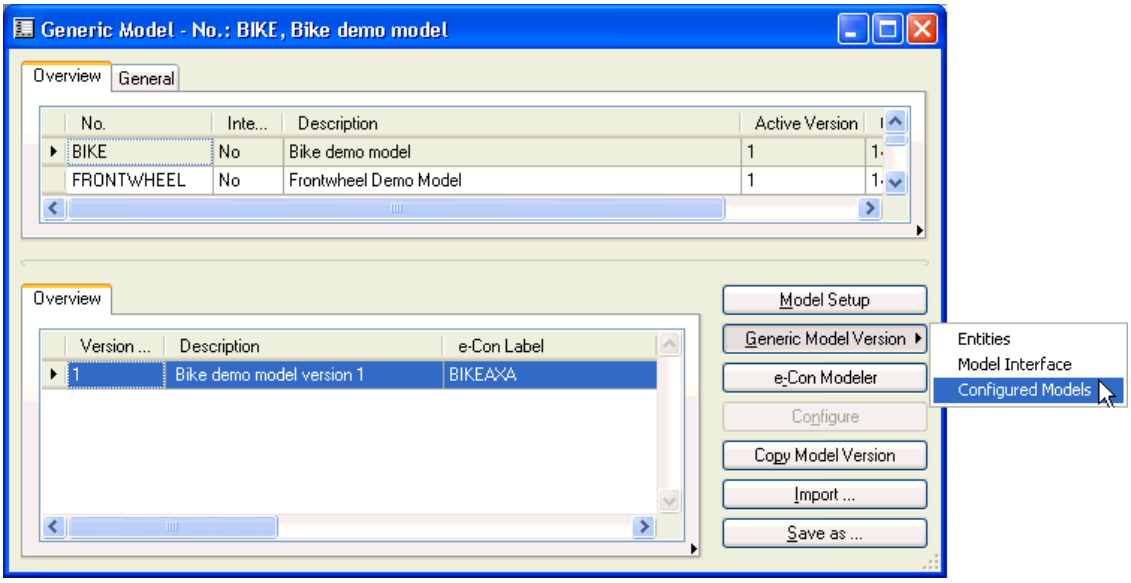
A controlled synchronisation can only take place when changes are done in a controlled way via versions. Changes made to a model without adding a new version are **always** synchronized during reconfiguration. In other words, changes made to the active version are always applied to an existing configuration during the reconfigure process.

When changes are done via a versioning procedure, each change or set of changes is a new model version, a certain set up of synchronization can be done. This set up can primarily be done on the generic model card. At the tab page "XML" a field "online synchronization" exists. This field determines how the model will act when the version of the configuration differ from the active version of the model.

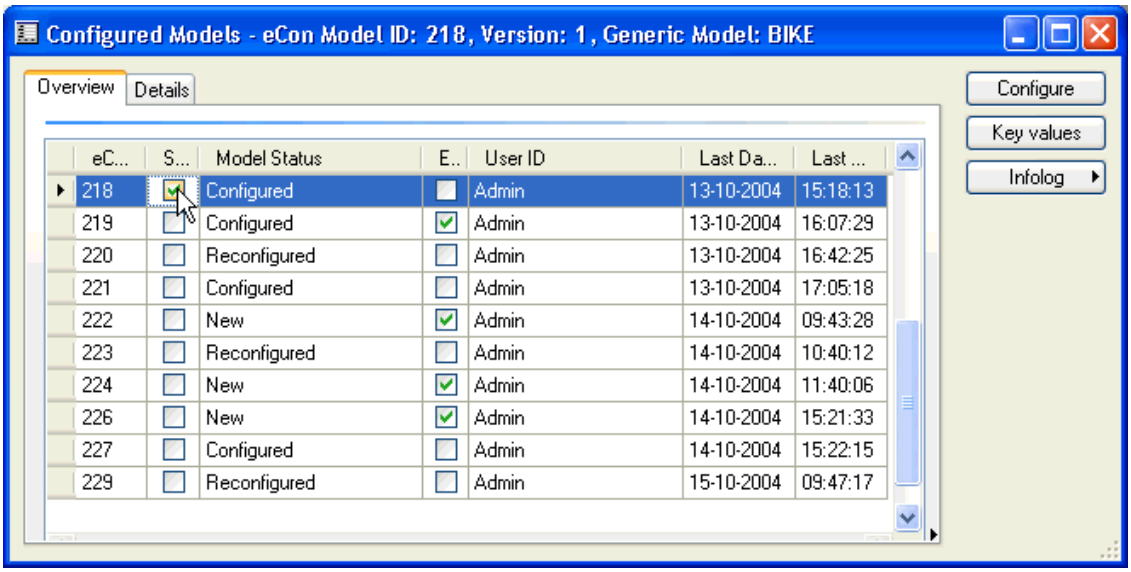


Three possibilities exists here:

- Manual: The configuration is only synchronized with the active version of the model when the field “Synchronize” for that particular configuration is true. This “Synchronize” field is located in the “Configured Models” table. This table can be displayed by the function “Configurations” from the “Functions” menu available at the Generic Model Card.



- The field “Synchronize” can be checked here for each configuration needed to update.



On Confirmation: Each time, when e-Con detects a difference between the configuration version and active version of the model, is asked if the configuration must be synchronized with the active version of the model. The field “Synchronize” does not play any role in this case.

- Automatically: When e-Con detects a difference between the configuration version and active version of the model synchronization takes automatically place without any user interaction. The field “Synchronize” does not play any role in this case.

7.21 How to add e-Mail functionality to my model.

It’s possible to set up e-mail functionality in your model that allows you to send the configuration as an attached document to a certain e-mail address.

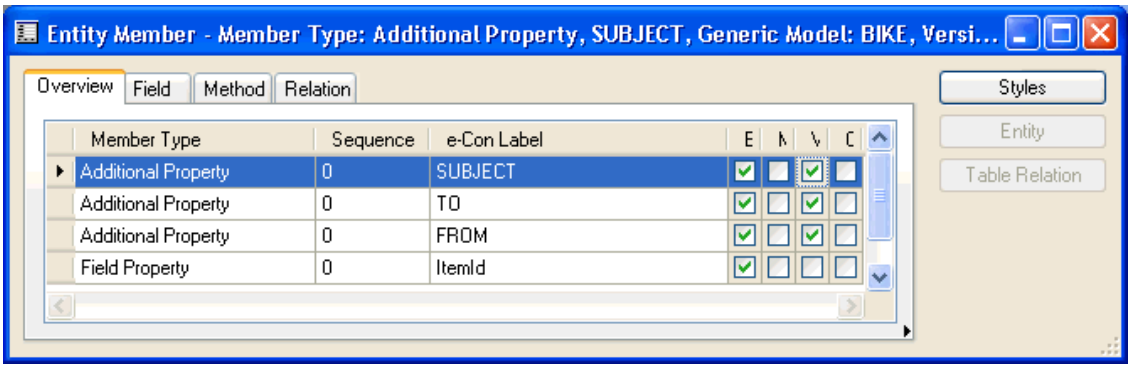
From, To, Subject and Body can be part of the message. In this section is described how to set up such functionality.

In this example three properties are added to allow the user to enter the ‘To’, the address of the e-mail, ‘Subject’, subject of the e-mail, and ‘Body’ the actual message.

The ‘From’, from who’s the e-mail, will be part of the settings. An extra setting will be added here.

Finally a rule is added to add the e-mail task to a button.

- First, add three properties to your model which represents the ‘From’, ‘Subject’ and ‘To’ fields required to send an e-mail.



- Add a new button ‘Send’. See section 7.17 for more details about adding custom buttons.

- Certify the model and Generate the Data structure.

- Open the e-Con Studio.

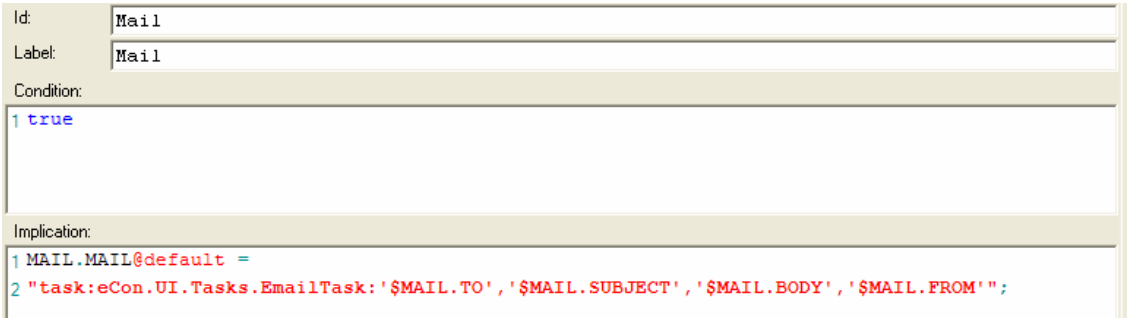
- Add a new rule ‘Mail’. In this rule the task for sending an e-mail is assigned to the custom mail button.

  The task to be used is:

  ‘task:eCon.UI.Tasks.EmailTask’

  This task can have a maximum from four additional expressions, ‘To’, ‘Subject’, ‘Body’ and ‘From’. When one or more from these expressions are missing e-Con expect them in the e-Con settings and will retrieve them from there.

  In our example only the ‘From’ is retrieved from the settings. Therefore the second expression is empty.

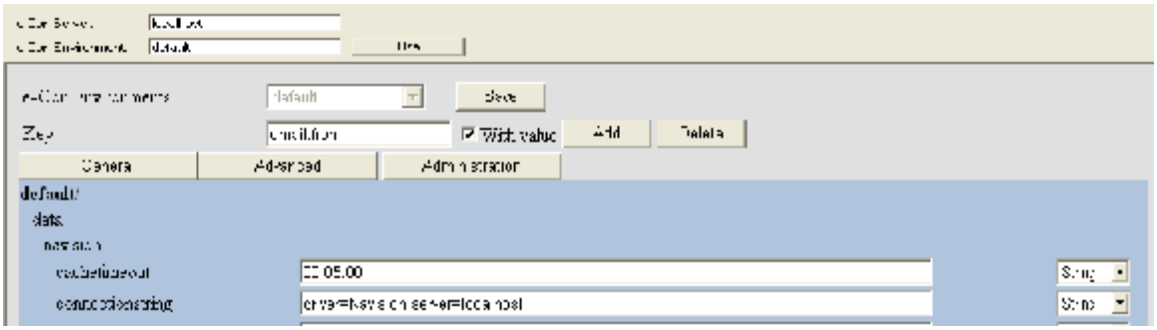


Notice that the members used in this string have a $ as prefix. It's an indication for e-Con that the part of string following on $ is a member of the model.

- Since 'From' is retrieve from the settings it must be added to the settings of this environment. Select 'Options..' from the toolbar.

- Select the 'Administration' tab page from the settings.

- Add a new key 'email.from' by entering this key name in the field 'Key' on top of the screen. Clicking 'Add' will add the key to the settings.



- Select the new key and enter the e-Mial address to be used as the 'From' address.

- Save the changed settings by clicking the 'save' button, leaf the settings.

- Finally build the model.

- Before the result can be checked, the settings from the e-mail host must be available as well. Select the 'email host' field in the e-Con settings. Fill this field with the name of the e-mail (SMTP) relay host (typically the e-Con web server that also runs the SMTP server).

- Now the result can be checked.

7.22 How to save my model without leaving the UI (posting)

The general task to save and process a configuration in Axapta is the process task. (represented by a 'B' in the Button set up. However another possibility exists as well to save and process a configuration: the so called 'Post' action. Posting a configuration means that a configuration is saved at the file system and the XML document is posted to Axapta. The UI is still open after this action (leave without exit). This can be especially useful when multiple configurations must be made with almost the same specifications. After posting a configuration the change can be done, posting, changing, posting etc.

To set up the post functionality few actions must take place:

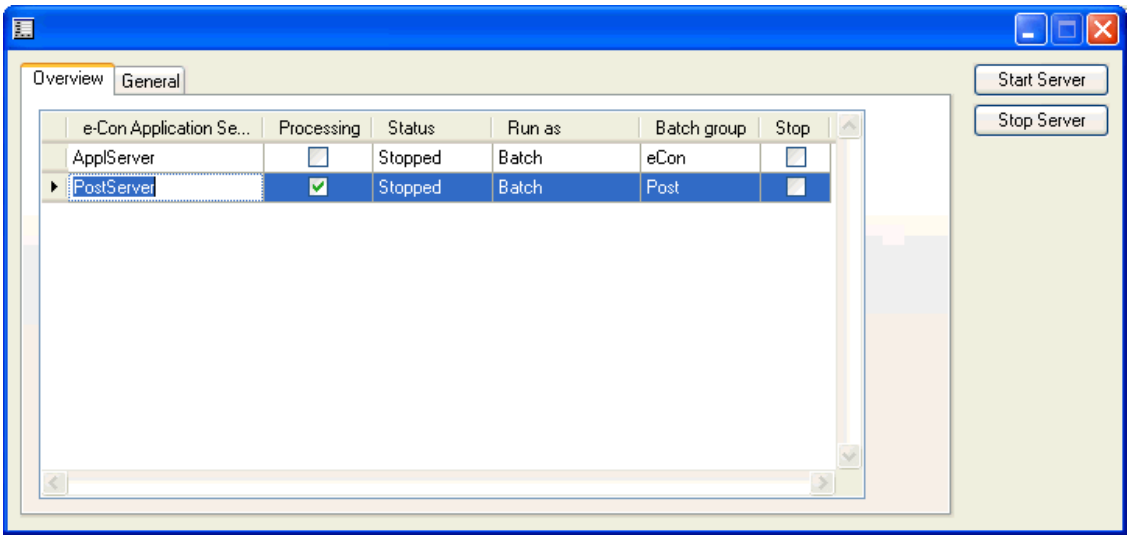
- Installation from another eCon Application Server. Since the posted configurations are picked up by the eCon Application Server and afterwards processed a separate eCon Application Server is needed here.
- Adding from the additional eCon Application Server to the eCon Application Server Handling table from the e-Con Set Up menu.
- Adding a post button to the UI.
- Adding a few rules to control the assignment of the right object id's.

Installation from another eCon Application Server.

For information concerning installation of a second eCon Application Server see the installation manual from MBS Axapta.

Setting up the eCon Application Server Handling table.

- In the e-Con menu, click Periodic, e-Con Application Server.
- Enter a new line and fill up a name for this Application Server. Check the field 'Document Processing'. This indicates that this Server is used for the processing of posted XML documents.



Adding a Post button to the UI.

- In the e-Con menu, click Setup, Parameters. The e-Con Setup form appears.
- Click 'Edit Environment'.
- Add the letter 'Q' or 'X' to the buttons field. Every character here represents a button. The 'Q' represents the Post button the 'X' represents the Post and Exit button.

Adding rules to control the object ID.

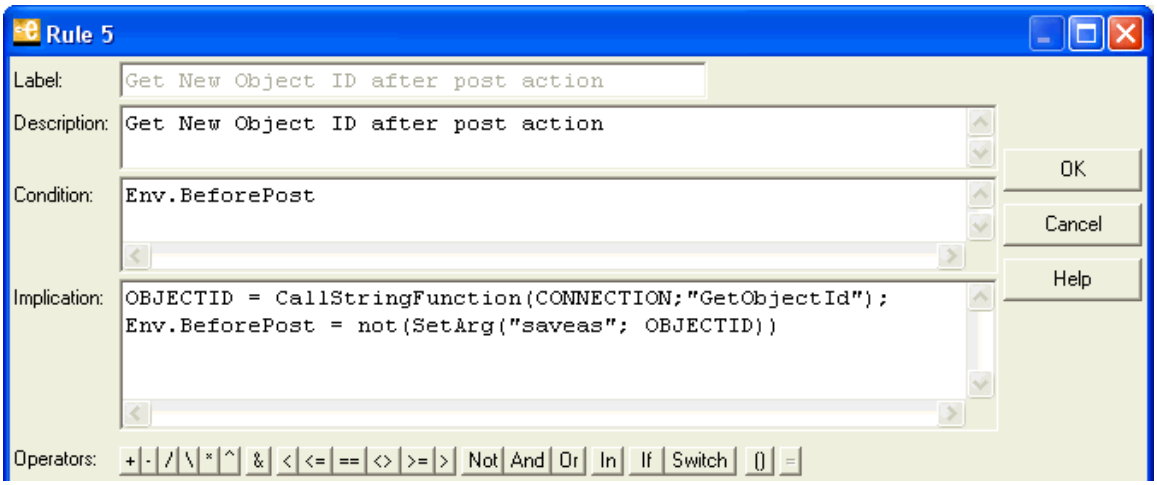
The object ID is the number which the configuration and XML file becomes after the save or post action. There are some special functions to control the number (object id) to assign to the posted configuration.

Furthermore two environment variable exist 'Env.BeforePost' and 'Env.AfterPost'. The 'BeforePost' becomes true just before the post action, the 'AfterPost' becomes true after the post action. These environment variables allows to trigger and execute rules just before and or after the posting from a configuration.

- Enter a rule to retrieve an new object number. This can be done with the function CallStringFunction(CONNECTION;"GetObjectId")

- Enter a rule to assign this Object number to the document to be saved. This can be done with the function SetArg("saveas"; OBJECTID)

Example from the Sales Order Demo model.


Rule 5
Label:
Get New Object ID after post action
Description:
Get New Object ID after post action
Condition:
Env.BeforePost
Implication:
OBJECTID = CallStringFunction(CONNECTION;"GetObjectId");
Env.BeforePost = not(SetArg("saveas"; OBJECTID))
OK
Cancel
Help
Operators:
+ - / \ * ^ & < <= == <> >= > Not And Or In If Switch () =

Chapter 8 e-Con Studio

For a detailed description of the e-Con studio see the document “e-Con 3.5 – e-Con studio - User manual”.

Chapter 9 e-Con Tasks

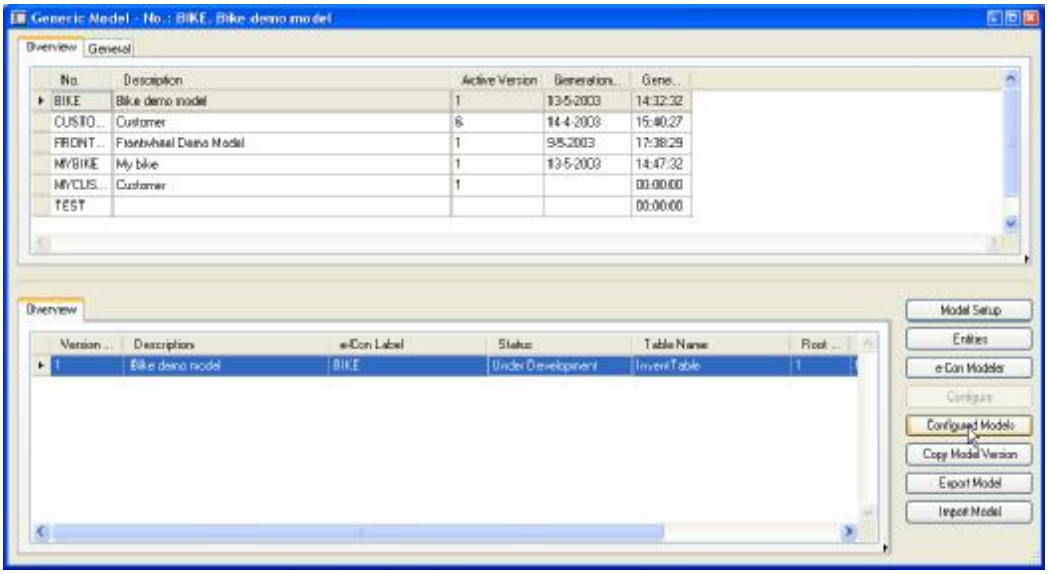
For a detailed description of the e-Con tasks see the document “e-Con 3.5 – Tasks Description”

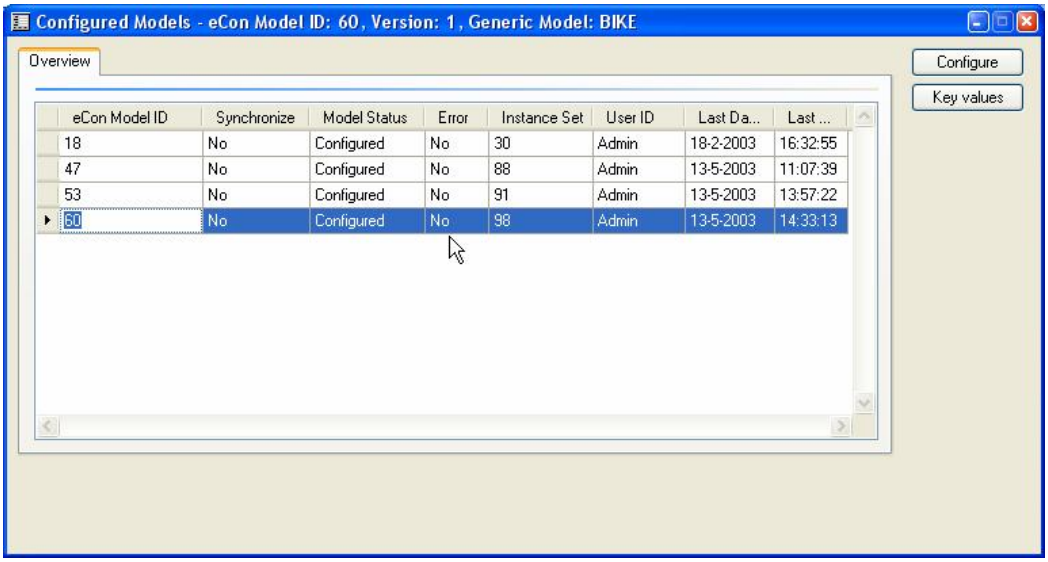
Chapter 10 Troubleshooting

This chapter explains how to address error messages. First, we discuss the quickest ways to address commonly occurring error messages, like "There are errors in the rules." Unfortunately, sometimes other sorts of errors also occur, like "Loop detected" or simply a model not performing as you expected. In the second section of this chapter, we will outline a procedure for addressing these other types of errors.

10.1 How to Solve Errors in the Data structure after configuration

When there are errors it is shown in the Configured Models list.





If there is a error Yes will be in the Error column.

10.2 Common Procedure for Resolving Errors

In this section, we will outline a procedure for addressing errors. Sometimes the error message "Loop detected" appears or a model simply does not perform as you expected. It's often difficult to determine what's causing the problem, especially when you're dealing with a large or detailed model.

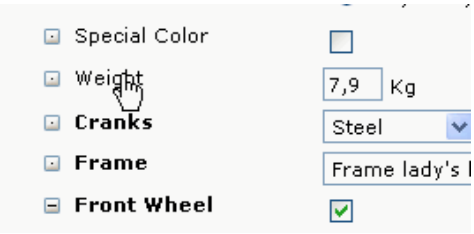
Tips & Tricks:

The error message "Loop detected" occurs when e-Con finds a loop in the rules. For example: The implication "A = A + 1" will cause this error message to appear. The rules engine is declarative, which means that every time "A" is changed, this rule will be executed. But executing this rule causes A to be changed, causing the rule to be executed, causing a change in A…and so on. e-Con detects this loop in the rules and provides the error message "Loop detected".

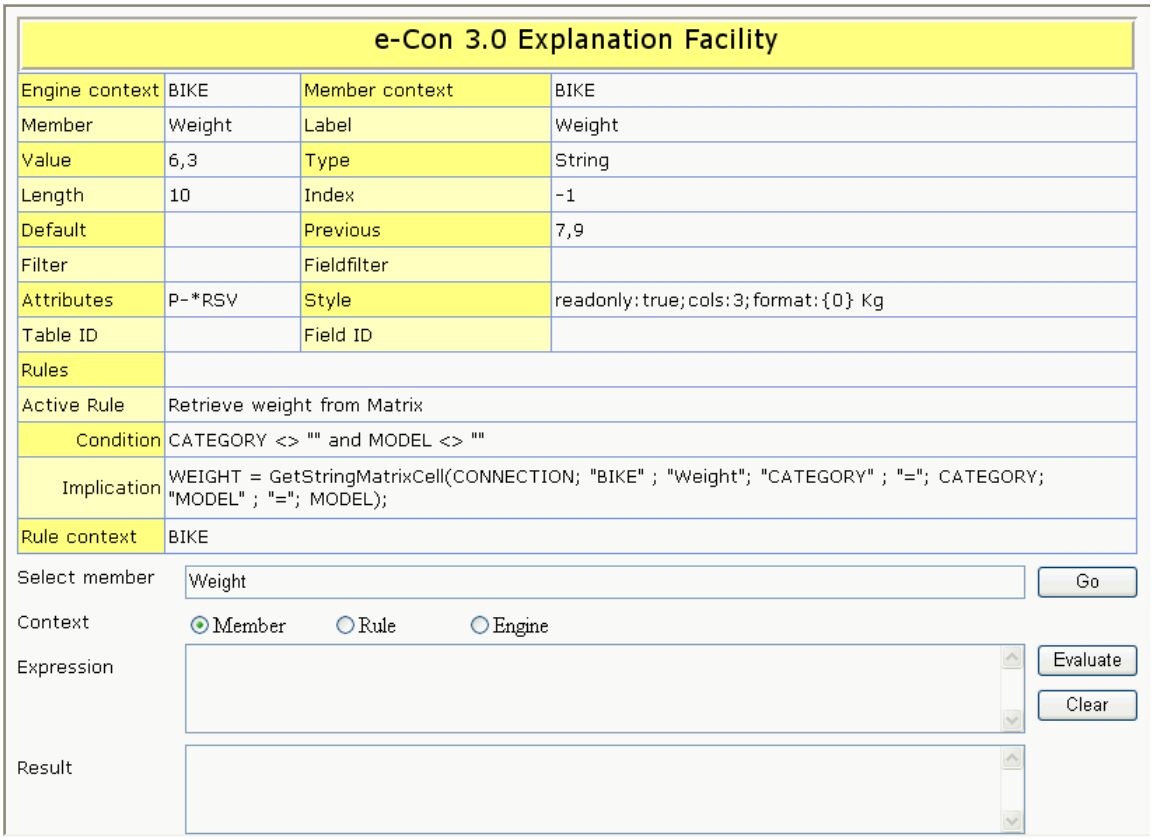
There is a helpful function to understand what happens in the model. Sometimes your model does not give an expected outcome or behavior. With the aid of the "?" button (How and Why) in the Front-end you can check how a certain value is calculated. You can also understand which rule(s) the member takes its value from.

How it works:

When the "?" button is clicked the cursor changed to a hand. This hand can be moved to a field in the front end.



A right mouse click at the field in the UI offers you the next screen:



In this screen the next fields are available:

Engine Context:

The parent model running.

Member Context:

The (sub)model the member is located.

Member

The member of the model the How & Why function is applied to. Actually the member from the UI field selected.

Label

The label from the selected member.

Value

The value from the selected member

Type

Type from the selected member

Length

Length from the selected member

Index

Index from the selected member. Actually this only make sense when the member is part of an array (so called repeatable entity). When the member is not part of an array the value is always '-1'.

Default

The default value from the selected member. When there is no default value set for the member this field is empty.

Previous

The previous value from the selected member .

Filter

The Filter applied to the member.

Attributes

The attributes applied to the member.

Style

The styles applied to the selected member.

Table id

The table id from the related table from Axapta (Only applicable for field properties)

Field Id

The eventually field id from the related field from Axapta (Only applicable for field properties)

Rules

Where used from the member. The where used are the rule numbers where the member is used in.

Active Rule

The rule(s) description involved in the calculation from this field

Condition

The condition from the active rule

Implication

The implication from the active rule

Rule Context

The (sub) model in which the rule is executed.

Select member

The member analyzed. It's possible to key in a member from the model to be analyzed as well.

Context

Three possibilities to select from:

- Member:
- Rule:
- Engine:

Expression

A rule expression can be entered here.

Result

The result from the above typed expression. This is calculated after clicking the activate button

Button 'Go'

To analyze the member entered in the 'Select Member' field.

Button 'Evaluate'

To calculate the expression as keyed in the 'Expression' field

Button 'Clear'

To clear the 'how and why content'. All fields will be blanked.

Member Model Path:

The path in the data structure to the selected member.

Rule Model Path:

The location of the rule where the value derives from. This is especially useful when there are sub models. This path locate in which model the rule is available.

Condition:

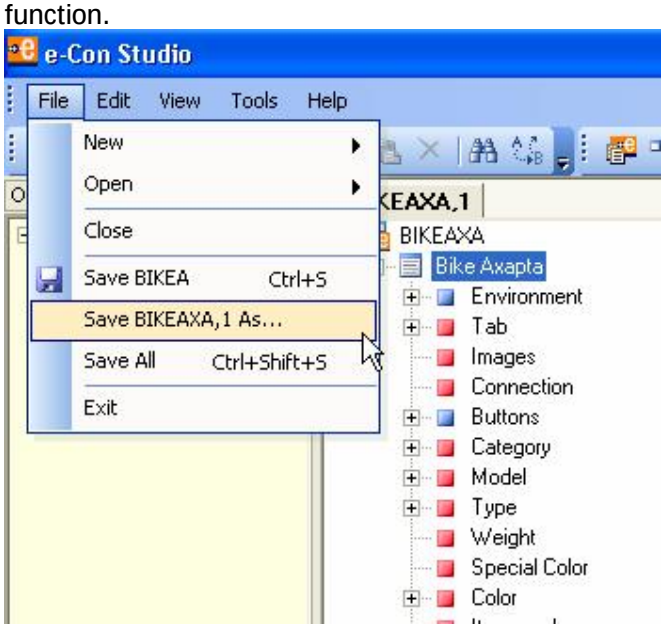
The condition of the rule where the value derives from.

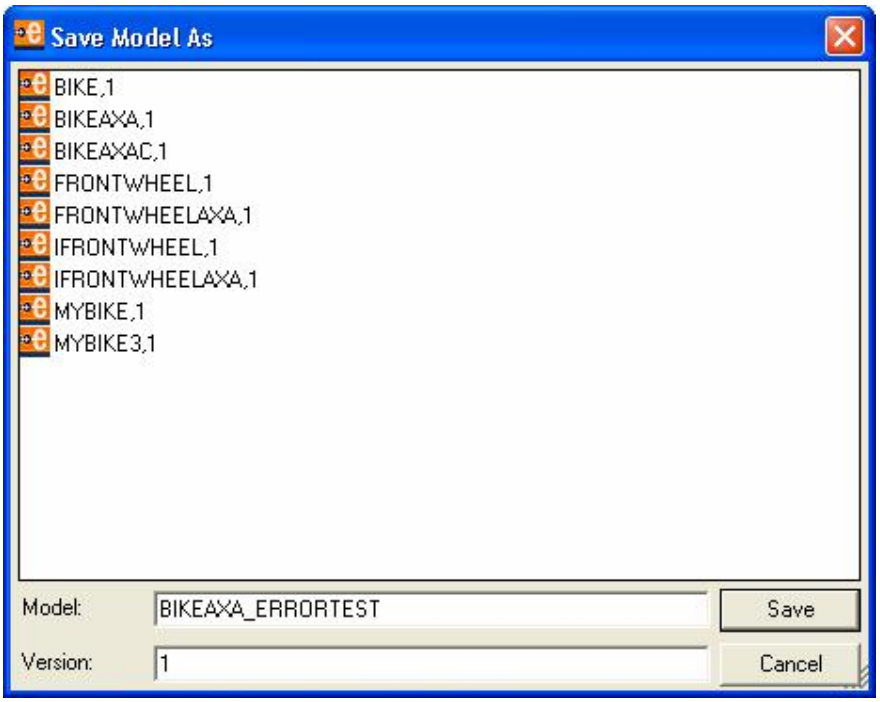
Implication:

The implication of the rule where the value derives from.

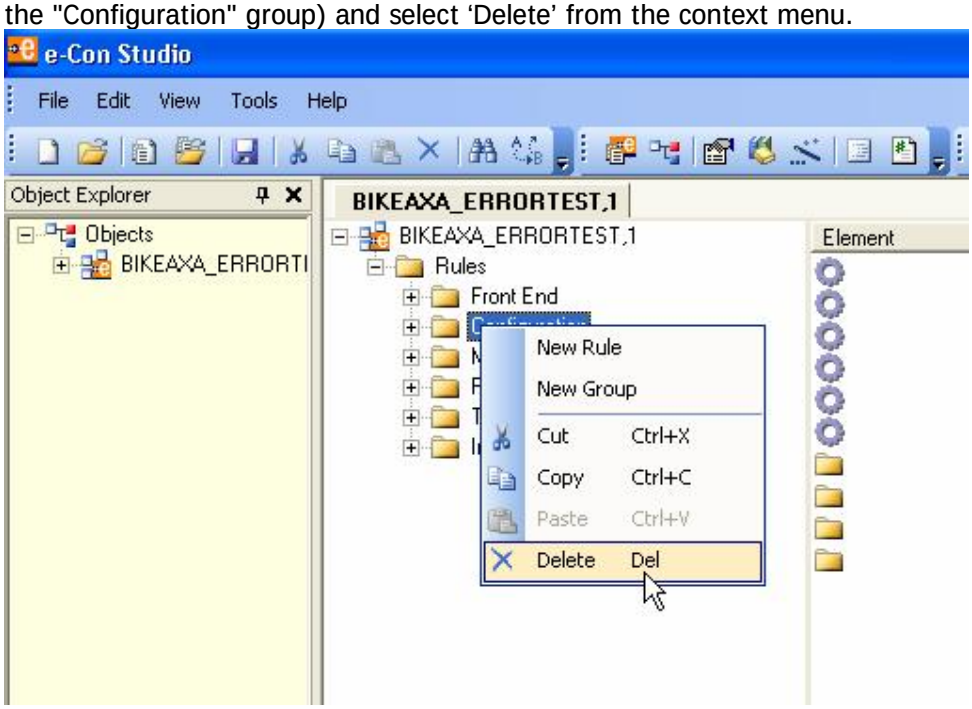
Below, we provide step-by-step instructions for the quickest way to find the cause of error messages. The Bike demo model is used in our example.

1. Start the e-Con Studio in the relevant model.
2. Save your model to another file. For example, a file called "BIKEAXA_ERRORTEST". To do this, go to the "File" menu at the top of the Studio, and select the "Save ..., As..." function.





3. Now that we have made a copy of our model and its rules, we can use this copy to search for the rule causing the error message.
4. Delete the first rule group in your model. To do this, select the group (in our example, the "Configuration" group) and select 'Delete' from the context menu.



5. Check the results by using the "Browse" function to view your model. You will find the "Browse" button in the tool bar of the e-Con Sudio.

Now there are two possible scenarios:

   a. The error message still occurs. Now you can be certain that the rule causing the error is *not* located in the deleted group (here, the "Configuration" group). Continue with step 6.
   b. There are no errors found. Now you can be certain that the rule causing the error *is indeed* located in the deleted group (here, the "Configuration" group). Proceed to step 7.

6. Delete the next group of rules. Again, check the results by using the "Browse" function to view your model. Continue with subsequent groups until the error message no longer appears. At that point, you can be certain that the rule causing the error is located in the last group you deleted.
7. Go to the "File" menu and select the "Open" function.
8. Select the original BIKEAXA model and repeat steps 1 and 2. This ends up in a BIKEAXA_ERRORTEST model which is an exact copy of the BIKEAXA model
9. Now that we know in which Group the rule causing the error is found, the next step is to identify which particular Rule is causing the error. This is done in much the same way as identifying the group.
10. Click on the relevant group in the folder tree to expand it.
11. Delete the first rule in the group and check the results by using the "Browse" function to view your model.
    Again, there are two possible scenarios:
    a. The error message still occurs. Now you can be certain that the rule causing the error *cannot* be the rule you have deleted. Continue with step 13.
    b. There are no errors found. Now you can be certain that the rule causing the error *is indeed* the one you just deleted. Proceed to step 14.
12. Delete the next rule. Again, check the results by using the "Browse" function to view your model. Continue with subsequent rules until the error message no longer appears. At that point, you can be certain that the rule causing the error is the last rule you deleted.
13. Now that we know which Rule is causing the error, we can return to our original model (not the copy). To do so, go to the "File" menu and select the "Open" function. Select the original model (in our example, this would be the file named "BIKE.XML", and NOT the copy we made for testing purposes called "BIKE_ERRORTEST.XML").
14. The final step is to select the rule you now know contains the error and figure out what the problem is.

Tips & Tricks:

Sometimes it's impossible to select an option for a member. You can select an option, but the field will be cleared once you leave the field of the member. This is almost always caused by using a double equal sign in a rule instead of a single equal sign. In effect, you

have made an assignment or a default assignment for the member. The single equal sign should be used to indicate an assignment, while the double one indicates a comparison.

10.3 List with most common syntax errors

The most common error descriptions that may appear during compilation of the e-Con rules are listed in the table below. Possible causes are also provided.

Error Description	**Possible Cause**
Implication expected	You added an unnecessary semicolon ";" after the last implication in the rule.
Variable expected	You forgot to enter a variable in, for example, an assignment.
"(" expected	You forgot an open parenthesis "(".
")" expected	You forgot a closed parenthesis ")".
";" expected	You forgot to add a semicolon ";" after an implication, in spite of the fact that there is another implication in the rule.
Undefined member .. (the member concerned is noted instead of the dots):	You made a typo in the member name, or the member has been changed, or the member is no longer available in the data structure.
Undefined attribute .. (the attribute concerned is noted instead of the dots):	You made a typo in the attribute name.
Member of attribute .. is undefined (the attribute concerned is noted instead of the dots):	You made a typp in the member of the relevant attribute or the member is changed or no longer available in the data structure.
Undefined function .. (the function concerned is noted instead of the dots):	You made a type mistake in the function.
Unexpected symbol:	You made a typo in the rule.
"else" expected:	The if-then-else structure isn't set up correctly; you most likely forgot the "else" statement.
"then" expected:	The if-then-else structure isn't set up correctly; you most likely forgot the "then" statement.
"end" expected:	The if-then-else structure isn't set up correctly; you most likely forgot the "end if" at the end.
"if" expected:	The if-then-else structure isn't set up correctly; you most likely forgot the "end if" at the end.

"#" expected:	You forgot the number sign "#" in the declaration of the array.
Index variable "#.." not declared (the variable concerned is noted instead of the dots):.	You used an array in an implication, but forgot to define this array in the implication concerned.
Attribute .. is read-only (the attribute concerned is noted instead of the dots):	Some attributes are read-only. In a rule, an assignment is applied to a member with a read-only attribute.

10.4 List with most common .Net compile errors

The most common error descriptions that may appear during compilation of the e-Con model to the .Net run time assembly are listed in the table below. Possible causes are also provided.

Compile Error Description	**Possible Cause**
Cannot implicitly convert type 'int' to 'string'	A type mismatch between two properties. Probably a comparison or assignment is done between properties from an unequal type. (This error can, of course, also happen for other data types)
Type of conditional expression can't be determined because there is no implicit conversion between 'int' and 'bool'	A type mismatch between a function and property or value. Mostly this error occurs when the MsgBox(<string>) function is used. MsgBox("String") <> 1 instead from only MsgBox("String") will solve this error.
Cannot implicitly convert type 'eCon.Base.EConCollection' to 'string'	A type mismatch between two properties, a string value can not be assigned are compared with a property used to store options. (<member>@options = "string" or <member> = {} both will generated this compiler error)
The best overloaded method match for 'eCon.Data.Functions.GetOptions(eCon.Interfaces.IEConClass, string, bool, string, string, params object[])' has some invalid arguments	The data type from one of the arguments of the function isn't correct. The number of the argument is shown in the next compile error.
Argument '3': cannot convert from 'string' to 'bool'	This message follows always the former messages. This error indicates that the data type from the argument from the function is not valid.
Operator '>' cannot be applied to	Operators can not be applied to a string

operands of type 'string' and 'string'	value. Convert the string first to a double or long before comparing can take place.
Unreachable code detected	It's just a warning and it can be ignored.
Could not write to output file '....' The process cannot access the file because it is being used by another process. '	This mainly happens when the interface model or a sub model is generated. The dll from the interface is still used after ending the configuration process. Close and reopen the Attain client to solve this problem.
'...Interface0' does not contain a definition for '<property>'	The "<property>" as specified in the error message, is a property from a sub model and used in the current model. This property is missing in the interface for that particular model.

Chapter 11 Glossary

The main terms used are outlined below.

Generic Model

The highest level of a generic model.

Label

Name of the Version, Entity or Entity member that is used in the XML document. The description entered for the label is also displayed in the Front End. The label entered for the version is used as the filename of the XML model file that is created.

Entity

Generic section of the data structure of the model. Entities are used to indicate in which Axapta table e-Con is to insert records. An Entity consists of a table number and a line number.

Entity Type

The type of Entity:

- *Class*
- *Object*
- *Both*

Class

Generic section of model. Records are created for this type of Entity.

Object

The lowest level of the model. Records are retrieved from the database for this type of Entity.

Entity Member

Member types of an Entity:

- *Field property*
- *Additional property*
- *Relation*

Field Property

This type of member is linked to a field in the Axapta table. This type is used if you want to change or type data into a Axapta table field.

Additional Property

This type of member is not linked to any field in Axapta. This type is used frequently for additional questions and variables.

Relation

Link between two Entities. This type of member is required in order to create a relationship between two Entities.

Additional Relation

Special link between two Entities. The linked Entity is not stored in Axapta when the configured object is saved.

Relation Type

Type of relation between two Entities:

- *Reference*
- *Master/Detail*

Reference

A relationship of many to one (n:1). For example, many items can be related to a BOM (Bill of Materials).

Master/Detail

A relationship of one to many (1:n). A BOM line can be related to just one BOM Header.

Relation Link

Primary key fields that are linked in both related tables (Entities). These are required in order to establish a relationship in a relational database.

Key Field

Unique field of a record that is always required when a new record is inserted in a table.

Key Value

A Key Value can be assigned to an Entity and will act as a template for the Entity. All the fields that aren't automatically entered by the Entity will be retrieved from the Key Value. Using the Key Value of the Root Entity, the model can be started up from different processes in Axapta.

Root Entity

Highest Entity that is linked to the generic model.

Query

There are two types of queries:

- *Selection from a Axapta table.*
- *Option list for an additional property.*

Style
Styles can be added to a member, influencing what it does.

Context
The business context or department from which e-Con is activated, e.g. Production.

Decision matrix
Matrix to set up interdependencies between member options.