Defining Associations in Business Object Builder eXpert

Summary
Associations create relationships within the nodes of a BO or between different BOs. This tutorial introduces the most important types of associations and explains how to define them using the Business Object Builder eXpert (BOBX) tool.

Level of complexity: Expert
Time required for completion: 3 hours

Author: Thea Hillenbrand
Company: SAP AG
Created on: June 20, 2014
### TABLE OF CONTENTS

**BEFORE YOU START** ................................................................................................................................. 4
Objectives .................................................................................................................................................................. 4
Prerequisites ........................................................................................................................................................... 5
Systems, Releases, and Authorizations .................................................................................................................. 5
Knowledge ............................................................................................................................................................. 5

**WHY DO WE NEED ASSOCIATIONS?** ......................................................................................................... 6

**GROUNDWORK** ............................................................................................................................................. 6
Sales Quote BO ....................................................................................................................................................... 6
ROOT Node Data Structure ..................................................................................................................................... 6
ITEM Node Data Structure ..................................................................................................................................... 7
Determination .......................................................................................................................................................... 7
Create Sample Data Using the BO Test Tool .......................................................................................................... 7

**Business Partner BO** ...................................................................................................................................... 7
ROOT Node Data Structure ..................................................................................................................................... 8
ADDRESS Node Data Structure .......................................................................................................................... 8
Create Sample Data Using the BO Test Tool .......................................................................................................... 9

**SPECIALIZATION** .......................................................................................................................................... 9
Procedure ............................................................................................................................................................... 9
Groundwork ............................................................................................................................................................ 9
Create the Association ........................................................................................................................................... 10
Enter the Association Header Data ...................................................................................................................... 10
Define the Association Key Binding ................................................................................................................... 11
Define the Association Condition Binding ......................................................................................................... 12
Save and Generate ............................................................................................................................................... 14
Test the Association .............................................................................................................................................. 14

**PARAMETRIZED SPECIALIZATION** ............................................................................................................. 16
Procedure ............................................................................................................................................................... 16
Define the Association ........................................................................................................................................... 17
Test the Association .............................................................................................................................................. 20

**FOREIGN KEY ASSOCIATION** .................................................................................................................... 22
Prerequisites .......................................................................................................................................................... 22
Procedure ............................................................................................................................................................... 22
Create a Foreign Key Association ......................................................................................................................... 23
Define the Binding ............................................................................................................................................... 24
Test the Foreign Key Association ........................................................................................................................ 25

**FOREIGN KEY ASSOCIATION – CROSS BO** ............................................................................................. 26
Prerequisites .......................................................................................................................................................... 26
Procedure ............................................................................................................................................................... 26
Create the Representation Node ......................................................................................................................... 28
Create the Foreign Key Association .................................................................................................................... 29
Define the Association Binding .......................................................................................................................... 30
Test the Foreign Key Association ........................................................................................................................ 31

**IMPLEMENTED ASSOCIATIONS** .................................................................................................................... 33
Prerequisites .......................................................................................................................................................... 33
ROOT Node Attributes ......................................................................................................................................... 33
Product Name Attributes .................................................................................................................................. 34
BEFORE YOU START

Business Object Processing Framework (BOPF) supports out-of-the-box composition (parent-child) relationships. Creating the Business Object (BO) tree structure provides the composition and the corresponding reverse associations TO_ROOT and TO_PARENT so that you can navigate within the composition tree back and forth. Other relationships have to be defined explicitly. The respective BOPF entity is the association. There are different kinds of relationships -- basically the foreign key association and the specialization. Associations can be modeled or implemented. This tutorial describes how to create the modeled associations using Business Object Builder eXpert. We advise using our Eclipse tool because it is more intuitive. But as long as it is not available in your system, this guide will help you to work with BOBX.

Objectives

By the end of this tutorial, you will
- Understand the different types of associations
- Understand why you should define explicit associations
- Be able to create specializations
- Be able to create foreign key relationships
- Understand the advantages of modeled associations

The tutorial uses the same example as in the Getting Started with Business Object Framework. As we are creating a foreign key relationship, we need a second BO, the BUSINESS_PARTNER BO. When creating this association we will use the Alternative Key concept, so you should have an understanding of the corresponding tutorial. Don’t worry about the complexity of the overview -- we will go through the different associations step by step.

The SALES_QUOTE BO consists of the ROOT and ITEM nodes, the ROOT node containing header data such as the sales quote ID, the related business partner, status, total amount, and administrative data. The ITEM node contains position data such as the product ID, quantity, and amounts.
The BUSINESS_PARTNER BO consists of the ROOT and ADDRESS node. The ROOT node contains header data such as the company name, legal form, and a reference to the main address. The ADDRESS contains the city, postal code, email address, and so on. A company may have different addresses so this is a 1:0…n relationship.

In the sales quote we will create a foreign key relationship to the business partner. In the business partner we will create a relationship between the reference to the main address in the root node and the address details. So we have two different kind of foreign key relationships; the first one is a cross BO association that enables navigation from one BO to another, and the second one is an inner BO association enabling navigation from one node of the BO to a certain instance of another node of the same BO.

Besides the foreign key relationship, we will create two specializations from the SALES_QUOTE ROOT node to special ITEM instances. A specialization is comparable to a composition, where the result is filtered by conditions that are defined at design time. In the first one, the condition for the specialization is fixed in the definition: we navigate to all positions free of charge. In the second one, the user can set the condition using parameters: we navigate to all positions that exceed an amount entered by the user.

**Prerequisites**

In order to be able to perform the tutorial, the following prerequisites have to be fulfilled.

**Systems, Releases, and Authorizations**

- BOBX is part of the Business Suite Foundation Layer and, therefore, included in the following SAP Business Suite releases:
  - SAP Business Suite EHP6, all SP
  - SAP Business Suite EHP7, all SP
- To implement this example, your SAP user requires the developer authorization profile (S_DEVELOP authorization object)

**Knowledge**

- Basic knowledge in ABAP OO
- Experience with DDIC tools
- Experience with some of the BOB tutorials
- Experience with the tutorial *Getting started with BOBX*
- Experience with the tutorial *Alternative keys in BOPF*
- Optional: *Creating and Implementing Determinations in Business Object Builder*
WHY DO WE NEED ASSOCIATIONS?
In the ABAP Data Dictionary you can define foreign key relationships. This definition is used in some contexts, like checks in SAPGUI or value help. But already without these features the definition is very useful if you want to navigate in the data: you see the relationship. This helps you to understand the data structures and their dependencies.
This argument also applies to BOPF. You can model even more types of relationships. So knowledge about the application is made explicit and visible. But this is not all. BOPF also offers runtime support for the association models that allows navigating along association paths with the help of APIs. So your definition and the BOPF runtime navigate you through the data.

GROUNDWORK
To create the associations we need two business objects. You can create them using the tutorial *Getting started with BOBX* as a template. The first BO is the SALES_QUOTE BO with a ROOT and ITEM node. The second BO is the BUSINESS_PARTNER BO with a ROOT and ADDRESS node. In the following chapters we describe the minimum data structure for these nodes. You can use the data elements if they are already available in your system. If not, you can use the predefined data type feature in the Data Dictionary.

Sales Quote BO

![Z_SALES_QUOTE Diagram](image)

Create the sales quote BO using the tutorials *Getting started with BOBX*. It is also possible to create the BO with the transaction BOB – but the first tutorial describes the mechanics of the BOBX tool, so that you are already familiar with the tool when you start creating the associations.

ROOT Node Data Structure

The following table describes the data structure of the ROOT node of the sales quote BO.

<table>
<thead>
<tr>
<th>Component</th>
<th>Typing Method</th>
<th>Component Type</th>
<th>Data Type</th>
<th>Length</th>
<th>Decimals</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUOTE_ID</td>
<td>Types</td>
<td>/BOBF/EPM_SQ_ID</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>
QUOTE_STATUS | Types | /BOBF/EPM_SQ_STATUS | CHAR | 2 | 0
BP_ID | Types | ZEPM_BP_ID | CHAR | 10 | 0

Note: If the data element /BOBF/EPM_BP_ID is available in your system, don’t use it. Our data element refers to a value help on our business partner table.

ITEM Node Data Structure

The following table describes the data structure of the ITEM node of the sales quote BO.

<table>
<thead>
<tr>
<th>Component</th>
<th>Typing Method</th>
<th>Component Type</th>
<th>Data Type</th>
<th>Length</th>
<th>Decimals</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM_POS</td>
<td>Types</td>
<td>SNWD_SO_ITEM_POS</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>PRODUCT_ID</td>
<td>Types</td>
<td>/BOBF/EPM_PRODUCT_ID</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>QUANTITY</td>
<td>Types</td>
<td>SNWD_QUANTITY</td>
<td>QUAN</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>QUANTITY_UNIT</td>
<td>Types</td>
<td>SNWD_QUANTITY_UNIT</td>
<td>UNIT</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CURRENCY_CODE</td>
<td>Types</td>
<td>SNWD_CURR_CODE</td>
<td>CUKY</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>GROSS_AMOUNT</td>
<td>Types</td>
<td>SNWD_TTL_GROSS_AMOUNT</td>
<td>CURR</td>
<td>15</td>
<td>2</td>
</tr>
</tbody>
</table>

Take care to create the relationship between QUANTITY and QUANTITY_UNIT as well as AMOUNT and CURRENCY_CODE in the Data Dictionary on the respective tab.

Determination

For test purposes it is sufficient to enter some amounts in the BO Test tool BOBT. If you want to calculate the amount based on the quantity and product price, you can do this using the tutorial Creating and Implementing Determinations in Business Object Builder.

Create Sample Data Using the BO Test Tool

To test the associations later on, we need test data. Create some sales quotes with positions. In the ROOT node you have to enter a business partner ID. Use an ID that you will create later on in the next step. In the position we need products, for example, advertising material, which is free of charge (amount = 0), and also products with different amounts in one sales quote. This data will be used to test the specializations.

Note: if you used the data element /BOBF/EPM_BP_ID in the ROOT node definition, take care not to use the value help feature in the BO test tool. The value help associated to the data element points to our business partner table and not to your one.

Business Partner BO

Create the business partner BO using the tutorial Getting started with BOBX. It is also possible to create the BO with the transaction BOB – but the first tutorial describes the mechanics of the BOBX tool, so that you get used to them; this will be helpful when starting with the associations.
ROOT Node Data Structure

The following table describes the data structure of the ROOT node of the business partner BO.

<table>
<thead>
<tr>
<th>Component</th>
<th>Typing Method</th>
<th>Component Type</th>
<th>Data Type</th>
<th>Length</th>
<th>Decimals</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP_ID</td>
<td>Types</td>
<td>ZEPM_BP_ID</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>COMPANY_NAME</td>
<td>Types</td>
<td>SNWD_COMPANY_NAME</td>
<td>CHAR</td>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>MAIN_ADDRESS</td>
<td>Types</td>
<td>/BOBF/CONF_KEY</td>
<td>RAW</td>
<td>16</td>
<td>0</td>
</tr>
</tbody>
</table>

ADDRESS Node Data Structure

The following table describes the data structure of the ADDRESS node of the business partner BO.

<table>
<thead>
<tr>
<th>Component</th>
<th>Typing Method</th>
<th>Component Type</th>
<th>Data Type</th>
<th>Length</th>
<th>Decimals</th>
</tr>
</thead>
<tbody>
<tr>
<td>CITY</td>
<td>Types</td>
<td>SNWD_CITY</td>
<td>CHAR</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>POSTAL_CODE</td>
<td>Types</td>
<td>SNWD_POSTAL_CODE</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>STREET</td>
<td>Types</td>
<td>SNWD_STREET</td>
<td>CHAR</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>BUILDING</td>
<td>Types</td>
<td>SNWD_BUILDING</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>COUNTRY</td>
<td>Types</td>
<td>SNWD_COUNTRY</td>
<td>CHAR</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>EMAIL_ADDRESS</td>
<td>Types</td>
<td>SNWD_EMAIL_ADDRESS</td>
<td>CHAR</td>
<td>255</td>
<td>0</td>
</tr>
</tbody>
</table>

In the sales quote BO we want to reference the business partner ID and not the node key, which is a GUID. We need to define an alternative key for the business partner ID BP_ID. This is described in the tutorial How to Create Alternative Keys with BOBX,
Create Sample Data Using the BO Test Tool

Now we have to create the business partners for the business partner IDs that we used in the sales quote BO. These ones will be used to test the cross-BO foreign key association. Start the transaction BOBT and create these instances. Create several addresses for each business partner. Take care to enter the key (a GUID) of one of these addresses in the MAIN_ADDRESS field of the ROOT node. This will be used for the BO internal foreign key association.

Result

You have now created the BOs and test data we need to start our actual tasks. In the next step we will create a specialization association.

SPECIALIZATION

In different contexts and UIs our end user wants to display all items of a certain sales quote that are related to advertising material. Of course you can implement this relationship for every context again and again, with the risk of inconsistent implementations. In BOPF we are going to define a specialization association. A specialization ‘specializes’ the composition (parent-children) relationship.

In a real application advertising material is a product category. To simplify our example we refrain from introducing a third BO representing a product with the attribute product category. Instead we are just filtering the items with the total amount of zero, assuming that advertising material is free of charge and that all other products are not for free.

Fig. 4: Specialization Association

Procedure

Groundwork

Before we start with the definition of the association we have to do some more groundwork. Association configurations can use fixed values that are defined as constants of an ABAP interface. So for the TO_ITEMS_FREE_OF_CHARGE association we have to define a constant in an interface for the comparison of the amount. Launch SE80 or Eclipse to define the ABAP interface ZIF_SALES_QUOTE_C. Define the constants SC_ITEM_FREE_OF_CHARGE and activate the interface.
interface ZIF_SALES_QUOTE_C

   public .

   constants SC_ITEM_FREE_OF_CHARGE type SNWD_TTL_GROSS_AMOUNT value '0'.

endinterface.

Create the Association

Finally we can start defining the specialization association on the ROOT node of the sales quote BO. Launch the Business Object Detail Browser for the sales quote BO. Open the Node Elements and the ROOT folders and select Create Association from the context menu of the Associations folder.

![](image)

Fig. 5: Create Association via context menu of Associations folder

Enter the Association Header Data

You can enter the header data of the association on the next screen. Enter the name of the association TO_ITEMS_FREE_OF_CHARGE and a description. We are using the prefix TO_ for association names, but you are free to follow this convention.

As it is possible to add any number of advertising materials or none at all in a sales quote, we select the multiplicity 1:0…n.

The node that we want to associate to the ROOT node is the ITEM node. So select ITEM in the dropdown list box. ITEM is the target node of the association.

Next we have to define the resolving node. The resolving node defines whether the source or the target node of the association has the information of the resulting target node instances. For example, a foreign key association has to be defined as source-resolved because the target node persists the key of the target node instance. Composition and specialization associations need to be defined as target-resolved because the framework can derive the target instances keys on the target node with the help of the parent key attribute. Therefore we select Resolve by Target Node for the association TO.Items_FREE_OF_CHARGE.

We have finished defining the association header data.
Define the Association Key Binding

We will define how the ROOT and ITEM nodes are related in maintaining the association binding. Switch to the Association Binding tab.

The first step is to relate the keys to one another: the child node always contains the foreign key to its parent node. We use this information to create the first condition. Select Specialization from the dropdown list box button Binding Pattern in the toolbar. The tool creates already the foreign key relationship to bind the instances of the children to the correct parent instances.
As we are using an attribute of the ROOT node – the ROOT node key – for the binding definition, the tool selects **Source Node Attribute** as **Association Binding Category**. We always define conditions using the target node. This has been defined in the header tab of the association. But the values for the binding can originate from different sources. With the Association Binding Category we specify the origin.

**Define the Association Condition Binding**

Finally we are going to define the condition ‘GROSS_AMOUT equals 0’. Create a new binding with the **Create** button.

We bind the GROSS_AMOUT to a constant value. So select **Constant Value** in the dropdown list box of the Association Binding Category.
The condition is going to use the gross amount of the ITEM node. Therefore, select \textit{GROSS\_AMOUNT} in the dropdown list box of the \textit{Target Node Attribute}.

Now enter the interface containing the constant \textit{ZIF\_SALES\_QUOTE\_C} in the column \textit{Constants Interface}.

The \textit{Relational Operator} is \textit{EQ} by default. We accept the operator. Last but not least, select the constant \textit{SC\_ITEM\_FREE\_OF\_CHARGE} in the dropdown list box of the column \textit{Binding Value From}. 
We have completed the definition of the association.

**Save and Generate**

Save and generate the definition. You will find details about the Generate function in the tutorial *Getting Started with BOBX*. Basically, the generation executes a consistency check of the model and regenerates the constants interface of the BO.

The generation log shows what the function did.

**Test the Association**

We are going to execute some tests in the BO test tool – BOBT. Launch the transaction BOBT or press the test button in the menu bar. The function opens the test tool in a separate window. Select the sales quote instance you created in the chapter Groundwork.

If you open the dropdown list box of the associations, the new association TO_ITEMS_FREE_OF_CHARGE is displayed beside the ITEM association.
To get an overview on the items, we select first the ITEM association. In our example we have created one position with advertising material.

We go back with the back button and select the TO_ITEMS_FREE_OF_CHARGE association. This time we see only the position containing the advertising material.

We execute a final check and select a sales quote instance without advertising material. We first display the items by following the ITEM association.
If we go back using the back button and select the association TO_ITEMS_FREE_OF_CHARGE, the list is empty.

Result

In this chapter we have defined a specialization association for the sales quote. It was a 1:0…n relationship, resolved by the target node, meaning that the condition is defined for target node attributes. In the binding section we first had to relate the keys. In the case of specializations that are special parent-child relationships, we bound the PARENT_KEY attribute of the child to the KEY attribute of the parent. Then we formulated the condition for the specialization by binding the attribute GROSS_AMOUNT of the child node to a constant defined in an ABAP interface. The model has been saved and generated and we could test it. The new association is visible in the BO test tool as association.

In the next chapter we are going to define a parameterized specialization, that is, an association where the condition values depend on user input.

PARAMETRIZED SPECIALIZATION

In this chapter we want to provide the possibility for the user to filter the positions of a sales quote that exceed an amount the user enters. In technical terms, we provide a parameterized specialization.

The parameterized specialization is a specialization association, that is, it refines the parent-child relationship. In our case we define an association on the ROOT node to the ITEM node once again.
Fig. 23: Parameterized Specialization

We have to execute the same steps as in the previous chapter. In this chapter we avoid repetition and focus on the differences.

**Procedure**

Launch the transaction BOBX and open the *Business Object Detail Browser* for the BO SALES QUOTE. Switch to edit mode with the toggle *button* in the toolbar. As we want to create an association from the ROOT node to the special ITEMS, we open the *Node Elements* folder for the ROOT node and select the function *Create Association* in the right mouse menu.

**Define the Association**

The basic settings are the same as in the previous chapter: we associate the ITEM to the ROOT node in a cardinality of 1:0...n. The condition is defined on the target – the ITEM – node. In addition, we need to define the input parameters for the condition. This can be done by creating a Data Dictionary structure and referencing them in the association definition. So, enter the name of your structure ZSQ_C_TO_ITEMS_AMOUNT in the input field *Filter Structure* and navigate to the Data Dictionary with double-click.
Defining Associations in Business Object Builder eXpert

We want the user to enter the minimum amount for the ITEMS to be displayed. The elements, AMOUNT, and CURRENCY_CODE must be contained in the parameter structure. Define these components using the EPM data elements or your own ones. Don’t forget to maintain the Currency Fields relationship in the last tab. After you have entered the enhancement category you can activate the definition and navigate back to the Business Object Detail Browser.

Back in the BO editor, switch to the Association Binding tab. Again we start the definition by selecting the binding pattern Specialization in the dropdown list box. This creates an entry in the table with binding the keys of the parent and child node.
We create a new entry with the New button and select Association Parameter Attribute as Association Binding Category in the dropdown list box.

![Fig. 27: Select the Association Parameter Attribute](image)

In the column Target Node Attribute the attributes of the ITEM node are provided. Select the GROSS_AMOUNT attribute.

![Fig. 28: Select the attribute GROSS_AMOUNT of the ITEM node](image)

In the next step we enter the Relational Operator GE as we want to provide all items that exceed a certain amount. Last but not least, we create the binding to the parameter structure component AMOUNT created some steps previously.

![Fig. 29: Bind it to the component AMOUNT of the DDIC structure](image)
Repeat this step to bind the CURRENCY_CODE of the ITEM node to the CURRENCY_CODE of the parameter structure. Use the EQ operator for the comparison.

![Association Binding](image)

We have completed the definition. Save and generate the BO.

**Test the Association**

As a last step, don’t forget to test the association in the BO test tool. Launch the transaction BOBT or press the Test button in the menu bar. Select a sales quote instance with several items. Verify that the new association is displayed in the association menu button.

![Association in Menu Button](image)

To get an idea of the values, you should select the ITEM association first.

![Display All Items](image)

In our example we have several items below 1000 € and some exceeding this value. So 1000 would be a good test. Go back to the ROOT node using the back button and now select the association TO_ITEMS_HIGH_AMOUNT. This one pops up a screen to enter the values for the amount and currency.
Enter the value 1000. You are not obliged to enter values for all parameters; just click the checkbox to ignore the component. Neglecting the currency code is not a good idea, but in our example all positions are in the same currency, so it does not really matter for our test.

Note: If you do not enter a value or mark the ignore component button, the tool will search for the currency code space.

The conditions maintained in the association definition are joined with AND. So the result list displays only the items that exceed 1000 €.

We execute a second test and enter the limit 4000.

The result list is empty, as expected.
Result

We have completed the creation and testing of a parameterized specialization. In principle, the definition is the same as in the specialization case. You only add a parameter structure allowing the user to parameterize the conditions.

FOREIGN KEY ASSOCIATION

You certainly know foreign key relationship from the ABAP Data Dictionary: in an ABAP table definition you keep a reference to another table where you find the details of the referenced object. In BOPF this is the same. The only difference is that a BOPF model already consists of a composition tree. You can define foreign key relationships within the composition tree or define one for the node of a different BO. As the definition is different from a BOBX tool perspective, we normally narrow the term foreign key association to the inner BO foreign key association, and the one to a different BO is often simply called Cross BO Association.

In this first chapter dealing with foreign key relationships, we will focus on the inner BO association.

We didn’t find a canonical example for this kind of relationship in the Sales Quote BO. So we will switch to the Business Partner BO. The Business Partner BO has a ROOT node with the business partner name, legal form, and so on. A business partner may have several offices at different places, that is, he/she has more than one address. So we have a child node ADDRESS with a city, street, email, and some more attributes.

Prerequisites

You have created the BUSINESS_PARTNER BO as described in the chapter Business Partner BO.

Procedure

The address of the head office of a business partner is a special one. We want to make it visible in our BO model. There are different ways to mark it. For example, you can introduce an address type on the ADDRESS Node and create a specialization association with the cardinality 1:0…1. However, in our example we choose a different approach: on the ROOT node, we create an attribute MAIN_ADDRESS that refers to one of the addresses, and we define an inner BO foreign key association.
Create a Foreign Key Association

Launch the Business Object Detail Browser and open the Business Partner BO in edit mode. We are going to create an association from the ROOT to the ADDRESS node. So open the folders Node Elements and ROOT. Create an association with the right mouse menu.

The foreign key relationship is of cardinality 1:0…1 and it is resolved by information based on the ROOT node with the attribute MAIN_ADDRESS, that is, the source node. So select Resolve by Source Node as Resolving Node.
Define the Binding

Switch to the Association Binding tab to define the binding. After you have chosen the definition Resolve by Source Node, the dropdown menu of the Binding Patterns provides the definition Foreign key for MAIN_ADDRESS. Select this entry.

The tool creates the binding. The Association Binding Category is Target Node Attribute as the value for the binding is the key of the ADDRESS node, the target node. The Source Node Attribute MAIN_ADDRESS is equal to the Binding Value From KEY.
Save and generate the definition and you are done. Now we are going to test the new association.

Test the Foreign Key Association

Launch the BO test tool with the transaction BOBT or the test button in the BO editor. Select a business partner instance with several addresses. In the association menu, the new association is displayed.

But first we display the details of the ROOT node. The MAIN_ADDRESS refers to the key of a certain ADDRESS instance.

We follow the ADDRESS association that displays all addresses of the business partner ITelO. The key of the first one in Rome corresponds to the MAIN_ADDRESS in the ROOT node.
Fig. 43: The Address in Rome is the MAIN_ADDRESS

We go back using the back button. If we follow the association TO_MAIN_ADDRESS, the one in Rome is displayed.

Fig. 44: Result list of the Association TO_MAIN_ADDRESS

Result

We have created an inner BO foreign key association and tested it. In the next step we are going to create a foreign key association cross BO – a Cross BO Association.

FOREIGN KEY ASSOCIATION – CROSS BO

Prerequisites

To execute an example of this step, we need the SALES_QUOTE BO as defined in the chapter Sales Quote BO. Make sure that you have created the attribute BP_ID in the ROOT node structure. This attribute will refer to the Business Partner BO defined in the chapter Business Partner BO. The business partner ID BP_ID must be defined as an alternative key.
Procedure

Based on the ROOT node attribute BP_ID in the sales quote, we are going to create a cross-BO foreign key relationship to the ROOT node of the Business Partner BO.

To create a cross-BO association, we need a representation of the referenced node in the source BO; in other words, we need a representation node in the sales quote BO. So the picture above shows an abstract of the implementation details. More precisely, we are going to create the following model: we add a representation of the business partner in the Sales Quote BO and we create a foreign key association to the representation node.

This is some work to do. So let’s start.
Create the Representation Node

Open the Sales Quote BO in the Business Object Detail Browser in the edit mode. We want to create a child node of the ROOT so open the Node Structure folder. In the right mouse menu, select Create Subnode and then Business Object Representation Node.

Enter the node name and the name of the referenced Business Object. In our case this is the BUSINESS_PARTNER BO. Press enter to refresh the UI. You can then select one of the nodes of the Business Partner BO in the dropdown list box of the input field Ref. Node. Select ROOT.

Save and generate the definition. The representation node of the Business Partner BO is displayed as a child of the ROOT node of the SALES_QUOTE. To indicate that this is a representation node, the icon is different.
Create the Foreign Key Association

Finally we can create the foreign key association. Open the Node Elements and the ROOT folders. Create an association with the right mouse menu function.

Enter a name and description. As in the previous chapter, the cardinality of the association is 1:0…1. The dropdown list box for the Associated Node displays also the representation node for the business partner. Select this entry.
Fig. 51: Representation node is displayed

As with the inner BO foreign key association, the resolving mode is *Resolve by Source Node*.

**Define the Association Binding**

Switch to the *Association Binding* tab. According to our header data, it is only possible to create an attribute binding.

**Fig. 53: The Association Attribute Binding is the only possible Binding Pattern**

Create a new binding with the *New* button. Select *Cross-BO Binding* as *Association Binding Category*. 
Select the BP_ID as Source Node Attribute. The relational operator is always EQ and cannot be changed.

Select the BP_ID as Binding Value From.

Now that we have defined the binding, we have completed the cross-BO association. Save and generate the BO model. Then continue with testing.

**Test the Foreign Key Association**

Launch the BO test tool with transaction BOBT or the test button. Select one sales quote instance. In our example, the selected sales quote references the business partner 9000000025.
Defining Associations in Business Object Builder eXpert

Fig. 57: Select a Sales Quote in the BO test tool

The association menu button displays the newly created association. Select it.

Fig. 58: Select the Association TO_BUSINESS_PARTNER

The details of the ROOT node of the business partner are displayed.

Fig. 59: Navigation to the Referenced Business Partner

And you can continue to navigate within the business partner or go back with the back button.

Fig. 60: You can Navigate within the Business Partner

**Result**

We have defined a navigation path from one BO to another. This has been done simply by defining associations, without any code.
IMPLEMENTED ASSOCIATIONS

Modeled associations are very convenient as they work without code. But there is another reason for using modeled associations: the metadata can be used to push down UI requests to the database. We provide this feature in 7.40 SP08 /SAP_BS_FND 7.47 SP06 in combination with Floorplan Manager (FPM) or Gateway services.

Whatever the case, there are limitations for modeling. If the binding condition is too complex, it is more convenient to code it. Also, the tool has limitations. For example, we do not support the use of system variables like SY-LANGU in the binding definition. The elimination of this limitation is on our backlog. For the time being, you have to use the implemented association feature to navigate to a text in the logon language.

The good news is that we provide a reuse function.

In this chapter we will show you how to use the reuse function. We have to switch again to another example because there are no natural texts -- either in the sales quote or in the business partner. We are going to use the product. If you want to implement this example, you have to do some more ground work. You have to create the Product BO.

The Product BO consists of the nodes ROOT and PRODUCT_NAME. The ROOT node carries header data like product id, price, and currency. The PRODUCT_NAME node is language-dependent and has a name and a description element. We are going to define an association providing the texts in the logon language.

Prerequisites

If you want to re-implement this example, you have to create the Product BO with the following attributes.

**ROOT Node Attributes**

<table>
<thead>
<tr>
<th>Component</th>
<th>Typing Method</th>
<th>Component Type</th>
<th>Data Type</th>
<th>Length</th>
<th>Decimals</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCT_ID</td>
<td>Types</td>
<td>/BOBF/EPM_PRODUCT_ID</td>
<td>CHAR</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>TAX_TARIF_CODE</td>
<td>Types</td>
<td>SNWD_PRODUCT_TAX_TARIF_CODE</td>
<td>INT1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CURRENCY_CODE</td>
<td>Types</td>
<td>SNWD_CURR_CODE</td>
<td>CUKY</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>PRICE</td>
<td>Types</td>
<td>SNWD_UNIT_PRICE</td>
<td>CURR</td>
<td>15</td>
<td>2</td>
</tr>
</tbody>
</table>

Fig. 61: Structure of the Product BO
Product Name Attributes

<table>
<thead>
<tr>
<th>Component</th>
<th>Typing Method</th>
<th>Component Type</th>
<th>Data Type</th>
<th>Length</th>
<th>Decimals</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANGUAGE_CODE</td>
<td>Types</td>
<td>LANGU</td>
<td>LANG</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>NAME</td>
<td>Types</td>
<td>SNWD_DESC</td>
<td>CHAR</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Types</td>
<td>SNWD_DESC</td>
<td>CHAR</td>
<td>255</td>
<td>0</td>
</tr>
</tbody>
</table>

Save and generate the BO. To test the association later on, create a product with texts in different languages.

Procedure

As the standard navigation to the text node is language-dependent, we do not create a new association, but redefine the composition relationship.

Fig. 62: Redefinition of the Composition

Define the Text Association

Open the Product BO in edit mode. Open the Node Elements, ROOT and Associations folders. Select the composition relationship PRODUCT_NAME that was created automatically when you defined the corresponding child of the ROOT node.
We are going to replace the framework implementation with an implementation of the BOPF reuse library. The class /BOBF/CL_LIB_C_LANGUAGE selects the target node of the association by SY-LANGU if no language has been specified. The filter structure /BOBF/S_LIB_C_LANGUAGE allows you to parameterize the association with a language code.

The following screenshot shows the structure definition in the Data Dictionary of /BOBF/S_LIB_C_LANGUAGE. It contains only the component LANGUAGE_CODE.
Finally you have to bind the language code of the filter structure to the language code of the PRODUCT_NAME node. So switch to the Association Binding tab in the BO editor.

Use the Category Association Parameter Attribute, like in the case of parameterized specialization. Actually, this association is a parameterized composition. Bind the LANGUAGE_CODE of the PRODUCT_NAME node to the LANGUAGE_CODE of the filter structure. Save and generate the definition and you are done.

Test the Association

Launch the BO test tool and select a product with different texts. Select the association PRODUCT_NAME.

You will see a popup for entering the language code. If you ignore the component LANGUAGE_CODE, the class selects the texts in all languages.
If you enter a language code, the class selects the text in the required language.

Result

We are at the end of this tutorial. We have learned how to define the most important types of associations using the tool BOBX. The Eclipse tool is more intuitive, but only available in EHP7 SP6.

Now, have fun creating a net of BOs in expanding this example by introducing a supplier in the Product BO that refers to the BUSINESS_PARTNER, or by establishing a cross-BO association between the SALES_QUOTE and the product. In the test tool you can navigate back and forth.

The consumption of associations will be covered in a different tutorial.