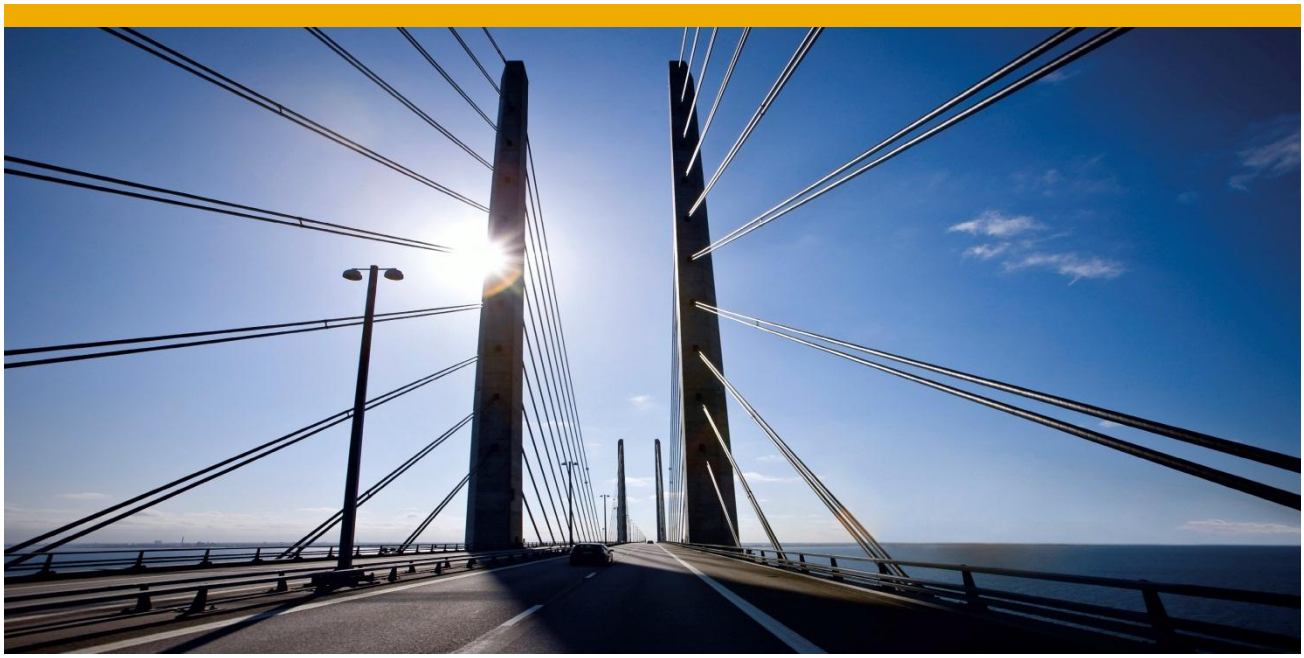

Creating and Implementing queries with Business Object Builder

Summary

This tutorial shows how to create queries to select instances from the database. The tutorial uses the Business Object Builder (BOB).

Level of complexity:	**Beginner**
Time required for completion:	**10 minutes**

Author:	**Thea Hillenbrand**
Company:	**SAP AG**
Created on:	**20 January 2014**

TABLE OF CONTENTS

BEFORE YOU START

In many scenarios, there are canonical views of a business object type that are of interest to more than one consumer - for example, a list of all instances, or a list of the instances I have to process or have processed in the last 4 weeks, and so on.
BOPF supports this creation of views with the query entity. Bear in mind that queries always select the data from the database and do not consider uncommitted changes that are currently in the buffer of the framework.

Objectives

By the end of this tutorial, you will be able to

- Create a simple query returning all instances of a node according to selection parameters
- Create a more complex query requiring some calculation
- Test the new features of the business object

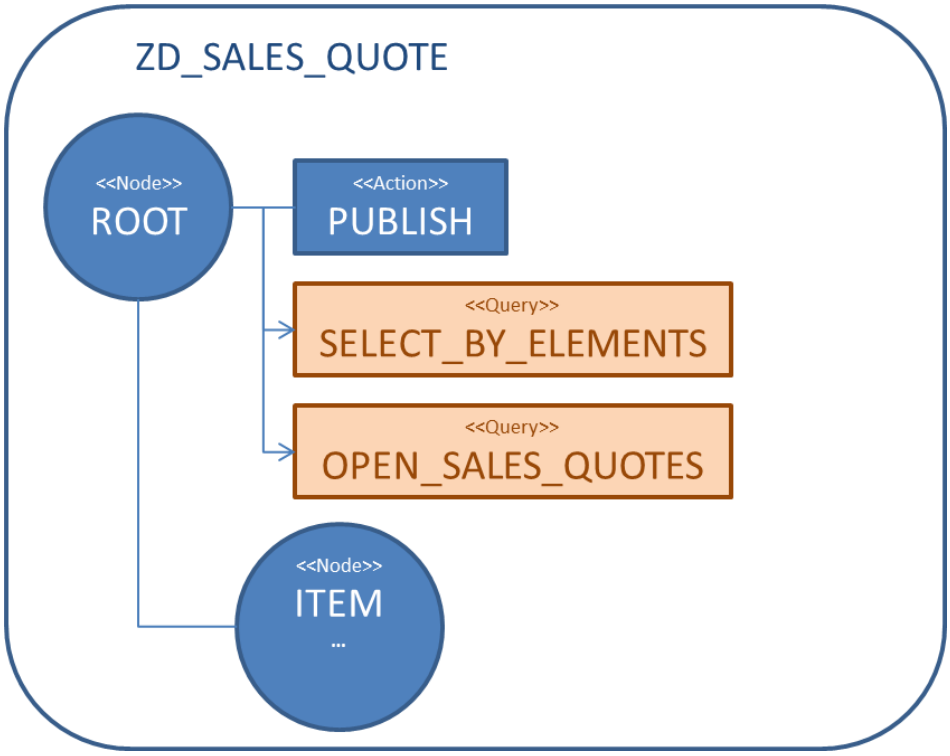

Fig. 1: Structure of the Business Object to be enhanced

The tutorial starts with the business object created in the 'Getting Started with Business Object Processing Framework'. It consists of the ROOT node with minimal header information, like QUOTE_ID, and the ITEM node with position data like PRODUCT_ID, quantity and price information. The goal is to create a query (**SELECT_BY_ELEMENTS**) that returns all instances of the root node of the business object ZD_SALES_QUOTE according to selection parameters. This query is provided by the framework generically. The second query (***OPEN_SALES_QUOTES***) is a custom-defined query and returns the open sales quotes for a work list.

Prerequisites

To be able to perform the tutorial, make sure the following prerequisites are fulfilled.

Systems, Releases, and Authorizations

- BOPF is part of the Business Suite Foundation Layer and, therefore, included in the following SAP Business Suite releases:
  - SAP Business Suite EHP5, SP11
  - SAP Business Suite EHP6, SP05
  - SAP Business Suite EHP7, all SP
- To create queries, your SAP user requires the developer authorization profile (S_DEVELOP authorization object).
- To implement this example, you need the business object created in the tutorial “Getting started with Business Object Processing Framework”.

Knowledge

- Basic knowledge in ABAP OO
- Experience with DDIC tools.

CREATE A SIMPLE QUERY (NODE ATTRIBUTE QUERY)

In this step, you will enhance the ROOT node of the Business Object (BO) “*SALES_QUOTE*”. This Business Object follows the semantics of the sales quote based on the NetWeaver Enterprise Procurement Model (EPM).

We will create the simple query SELECT_BY_ELEMENTS, returning all instances of the root node according to selection parameters. Each attribute of the root node can be used as a selection parameter.

Procedure

Launch the Business Object Builder (BOB)

Transaction BOB provides the design time for custom business objects as well as business object enhancements.

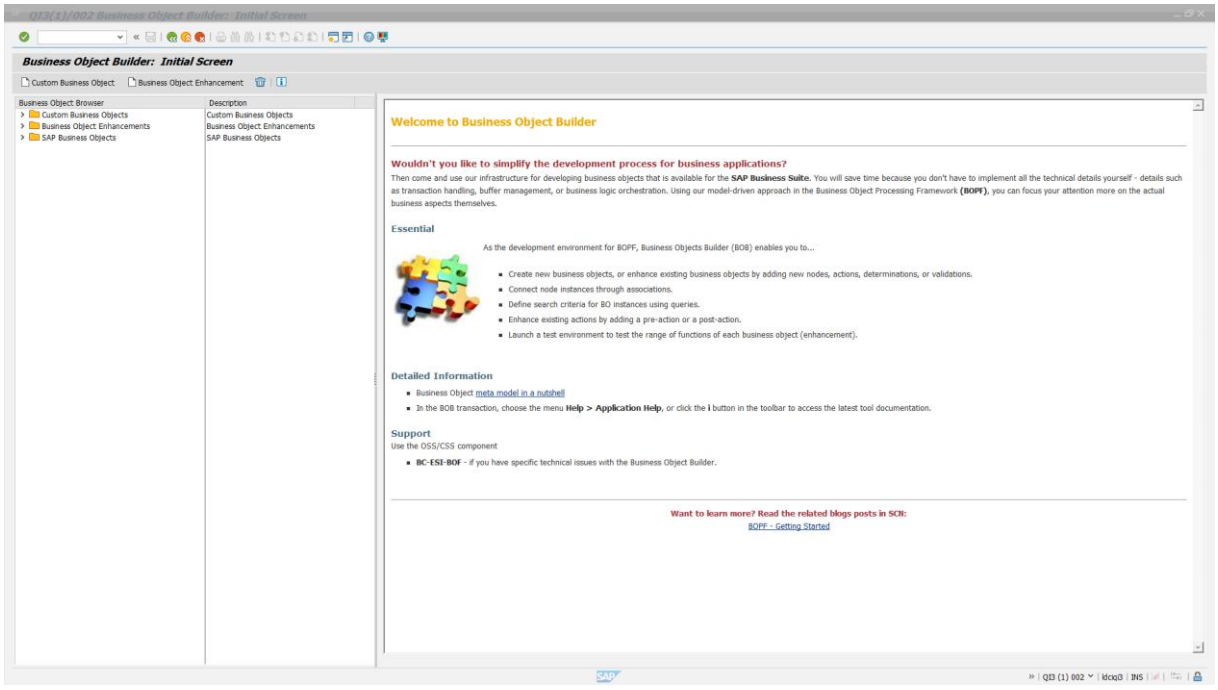

Fig. 2: Welcome page of the BOB transaction in a SAP customer system

On the left side of the initial screen you see three categories of BOs that are currently available in your system:

- Custom Business Objects: BOs created by the customer in their system
- SAP Business Objects: Extensible BOs that are delivered by SAP
- Business Object Enhancement: BOs that are enhancements to an existing business object.

Select the Business Object ***ZD_SALES_QUOTE***.

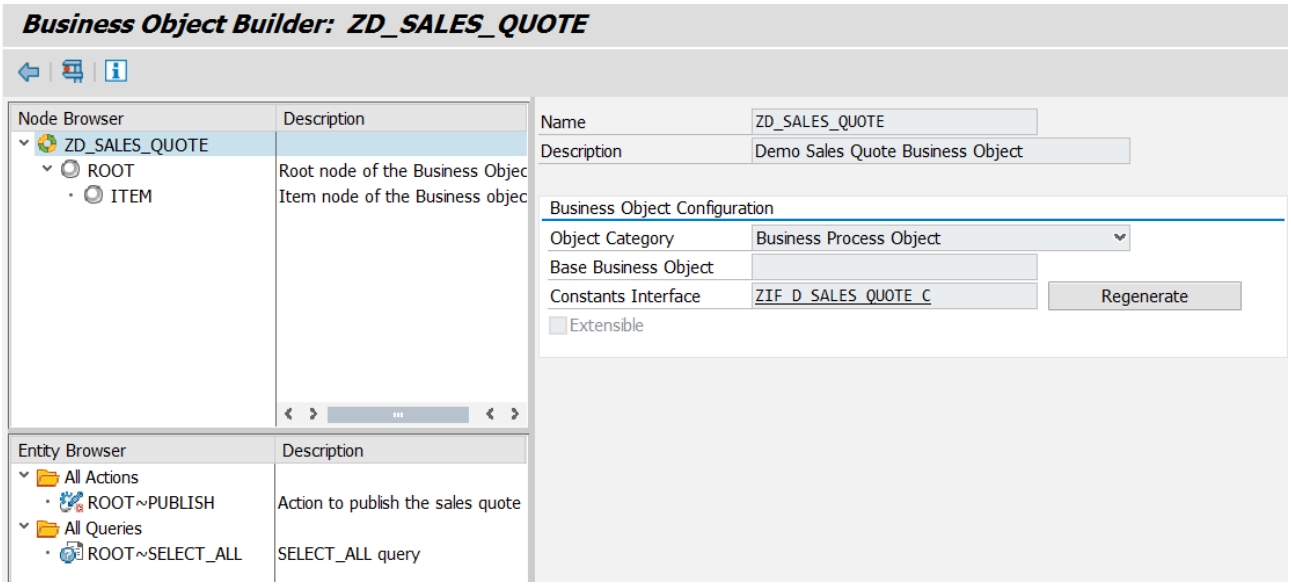

Fig. 3: SALES_QUOTE resulting from 'Getting started with Business Object Processing Framework'

Select the node *ROOT*. In the context menu you will find the action **Create Query**, which launches the Query Creation Wizard.

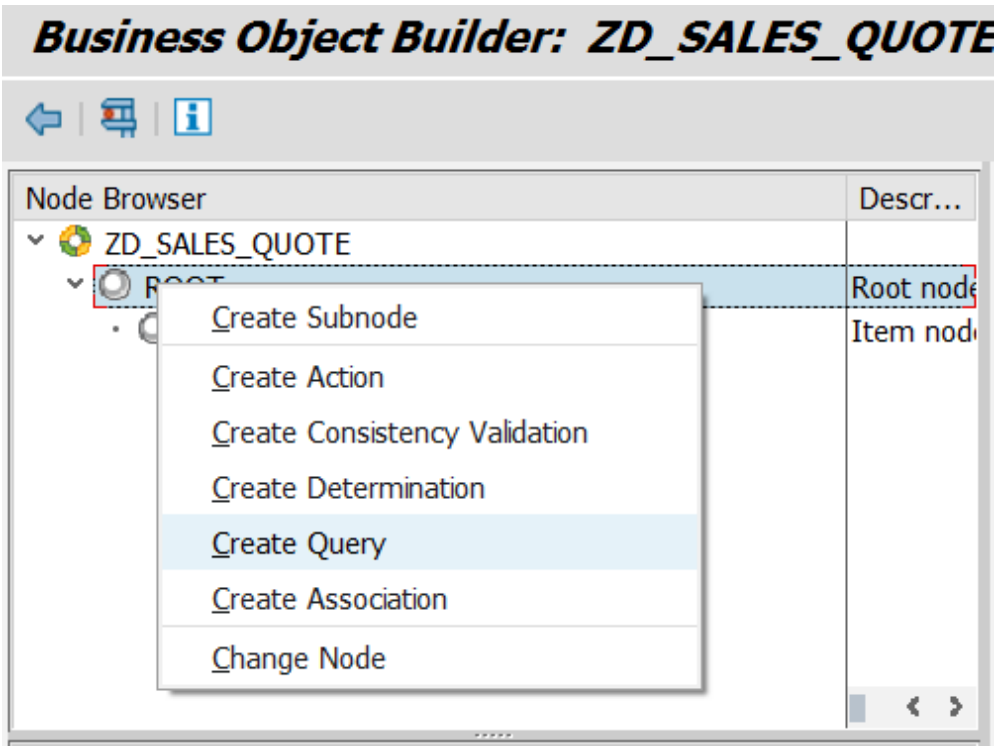

Fig. 4: Launch creation wizard

Start the wizard and enter the query name and description.

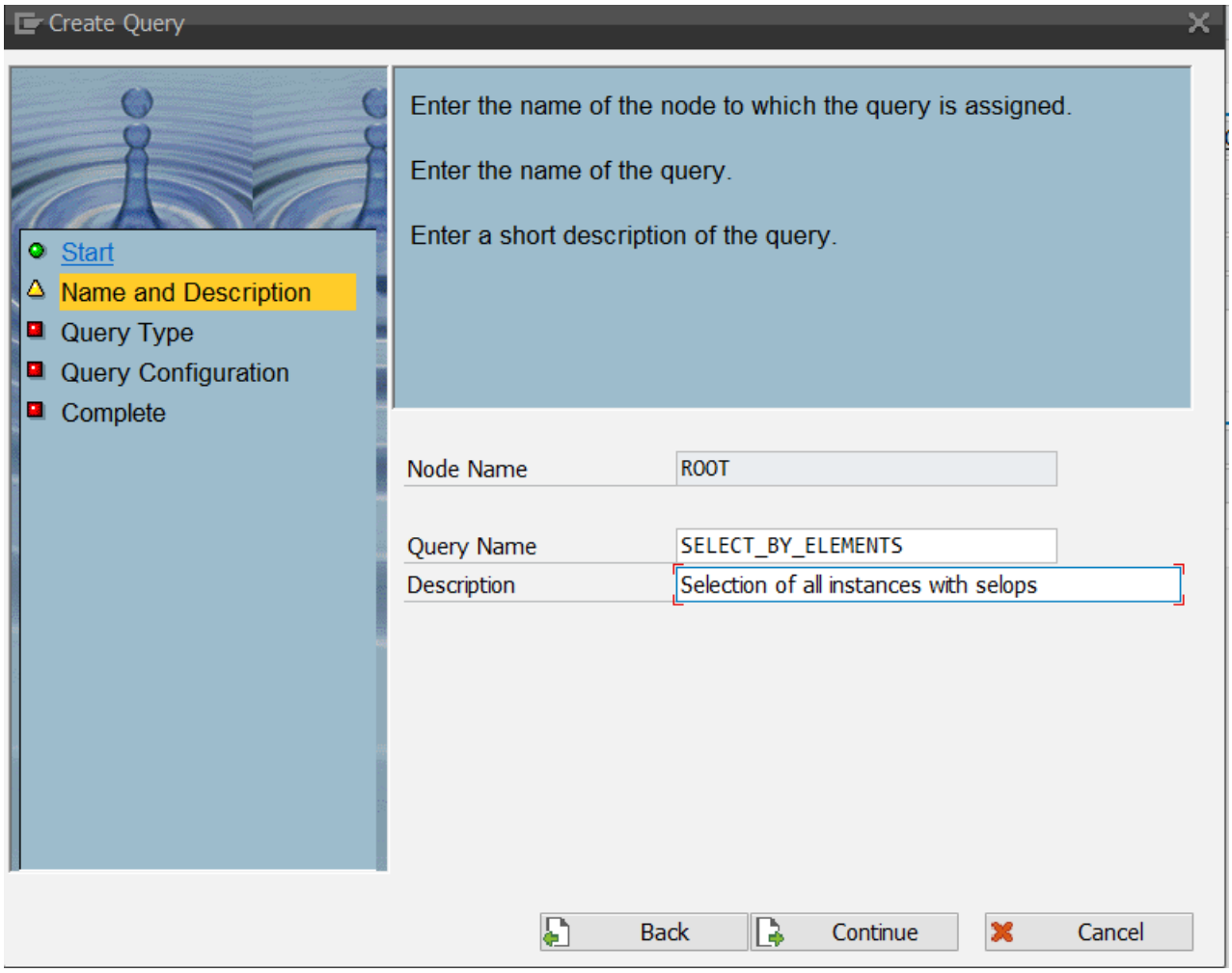

Fig. 5: Enter the Query name and description

Continue with the next page. Select the query type *Node Attribute Query.* This type returns all instances of the root node according to the selection parameters. Each attribute can be used as a selection parameter.

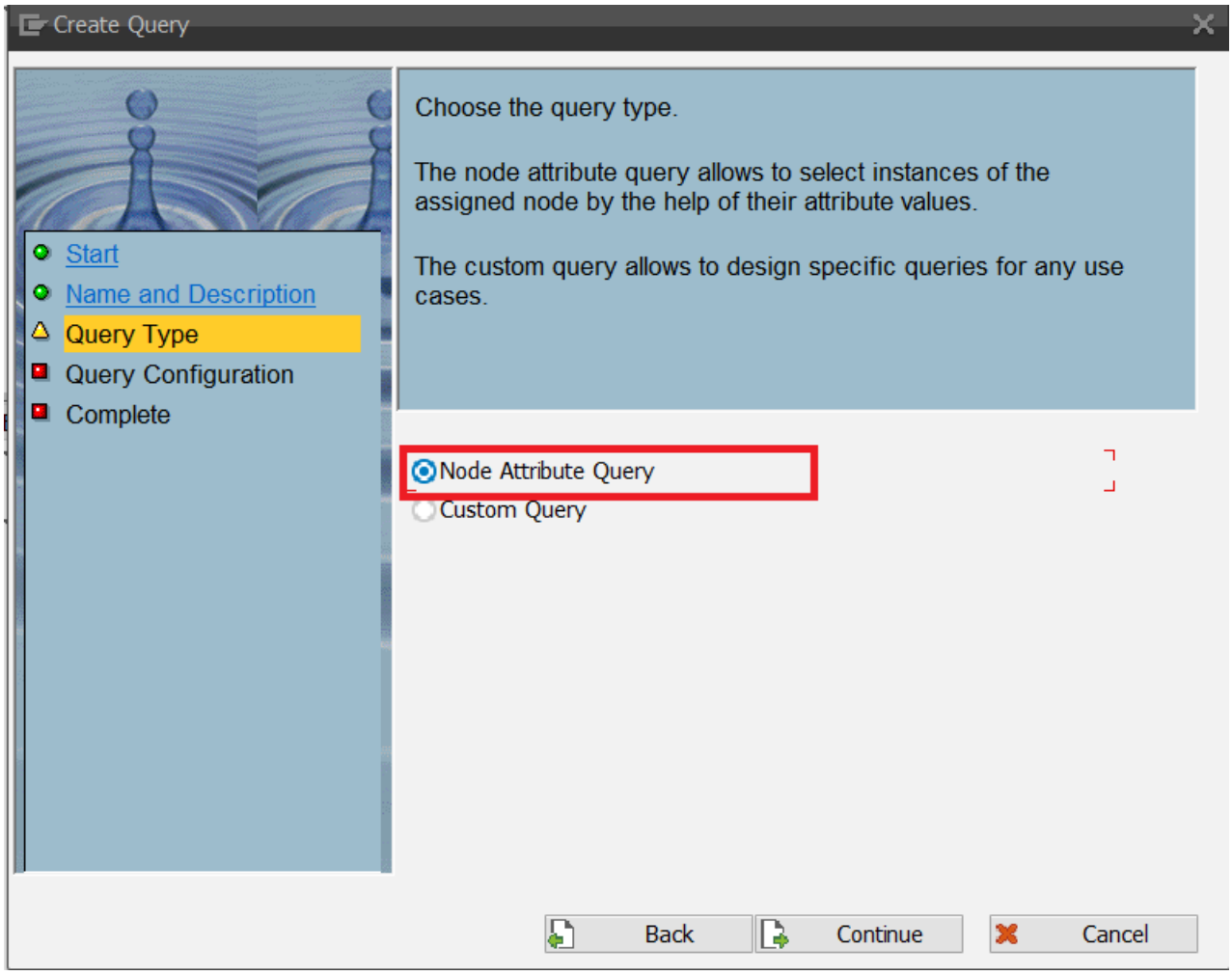

Fig. 6: select the query type - node attribute query

Continue with the next step and complete the wizard. You have completed the query definition and can now test it using the BO test tool

Launch the Test Tool

Launch the test tool with the **Test** button. The new query is visible in the load folder and can be executed.

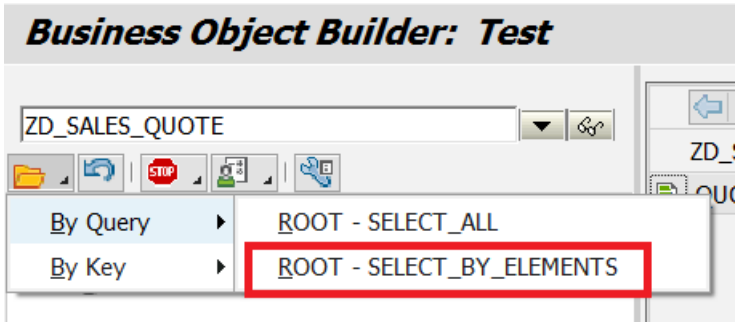

Fig. 7: the new query is visible in the test tool

The tool creates a selection screen where you can enter your selection parameters. We want to select only the sales quotes with status ‘published’, so we enter **P** in the attribute *QUOTE_STATUS*.

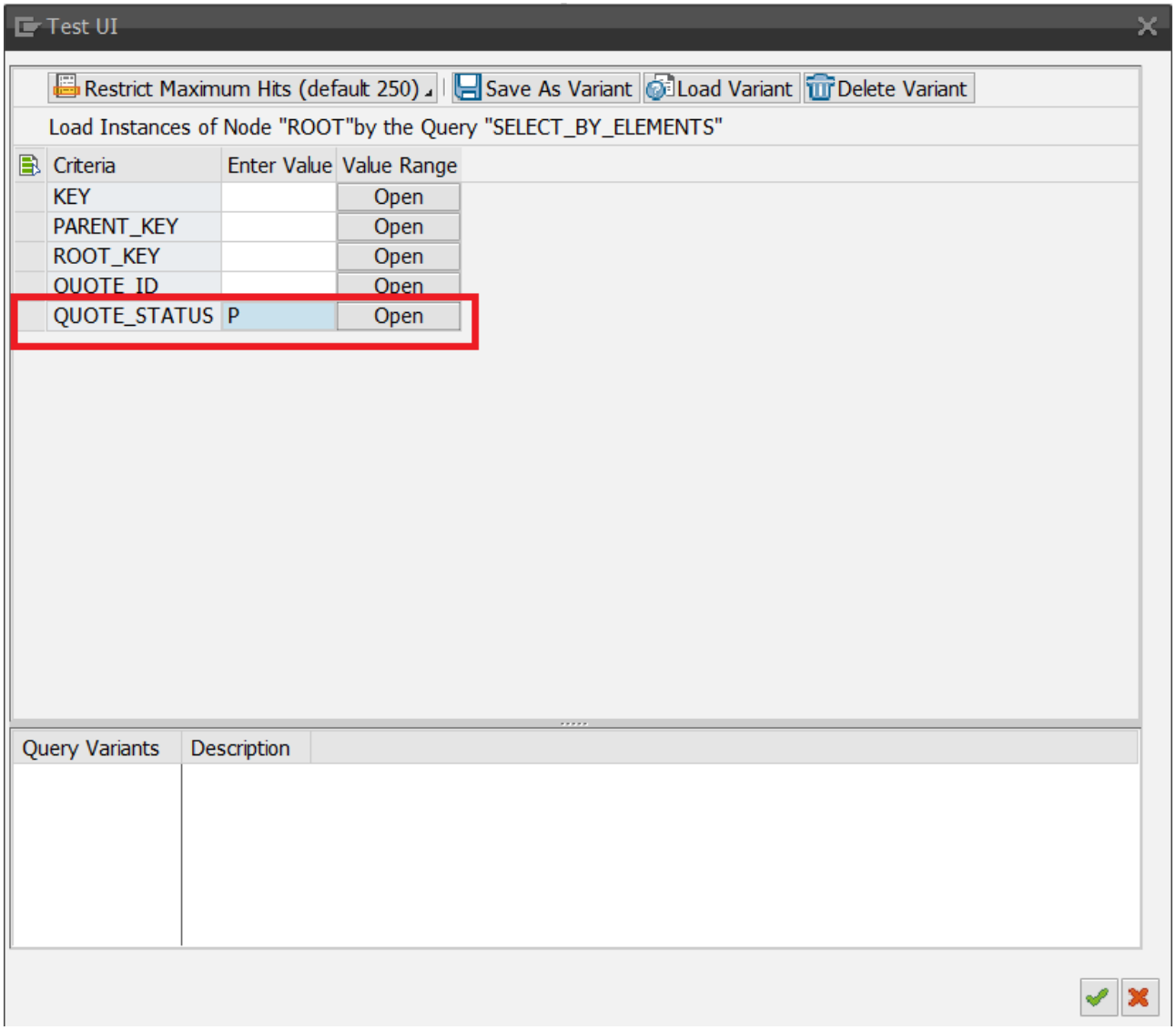

Fig. 8: Enter selection parameters

Execute the query by choosing the enter button; you get only the published instances.

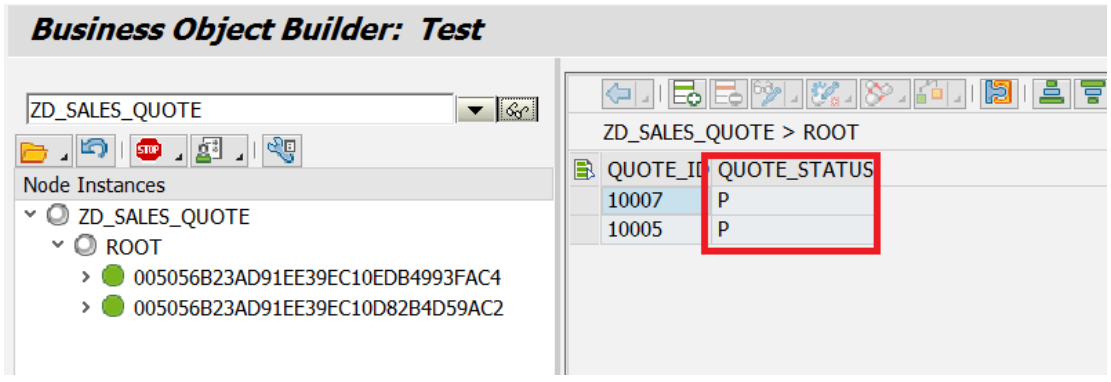

Fig. 9: Query result list

Result

Simple queries that return the values of a node can be created without any line of code, simply by means of a few clicks. In the next chapter we will learn how to create more complex queries.

CREATE A CUSTOM QUERY

There are many reasons for implementing a query on your own. The simplest one is that you want to execute an authorization check. In EHP 7 we have integrated the authority checks into the BO model and the runtime. In older releases you have to implement them on your own. But the example we will implement now step by step is a different one. We will create a query returning the work list of open sales quotes – open means all sales quotes that are not published.

Prerequisites

Open BOB and launch the Query Creation Wizard on the root node of the Business Object *ZD_SALES_QUOTE* – as in the previous chapter.

Procedure

Continue the Wizard to Create Queries

Enter the query name ***OPEN_SALES_QUOTES*** and the description.

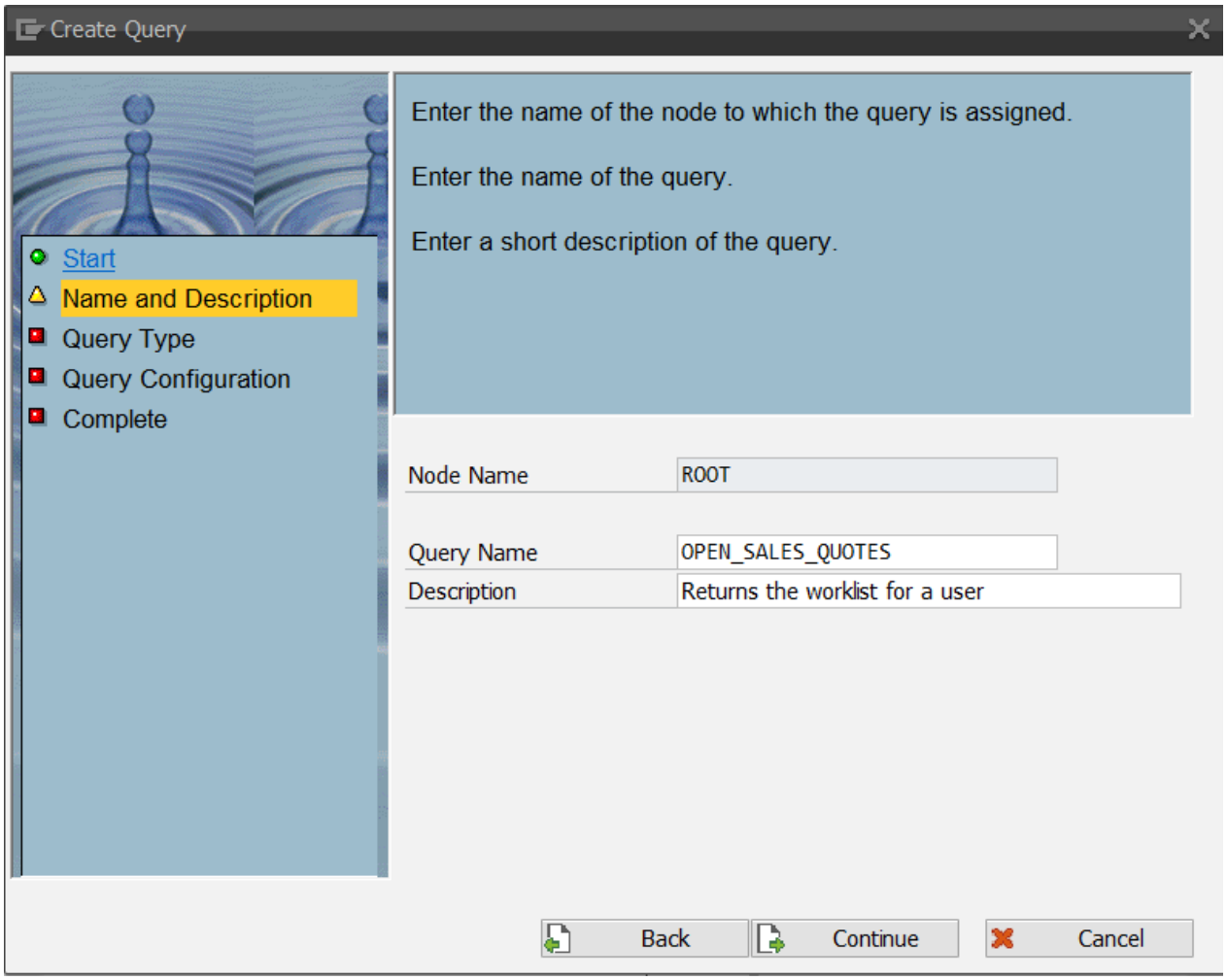

Fig. 10: Enter the Query name and description

Continue with the next page. Select the query type *Custom Query.* This query type allows you to create use case-specific queries.

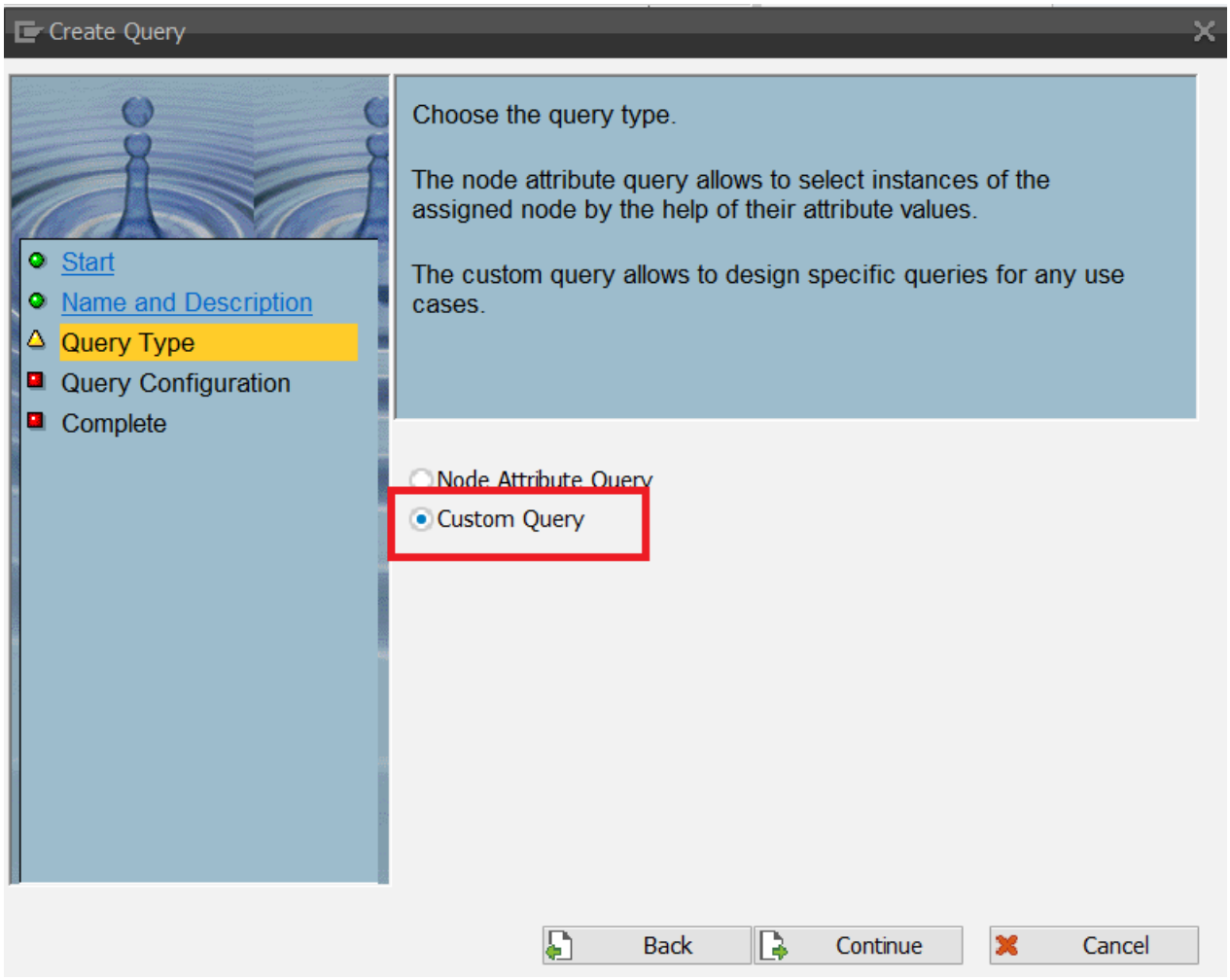

Fig. 11: select the query type –Custom Query

Continue with the next step.

Define the Implementing Class

At runtime, a custom query is represented by an ABAP class that implements the query interface. In this step you are able to define the name of that class. In addition, you have to define the data type for the input parameters. If you do not specify the result type, the framework expects the node structure to be returned.

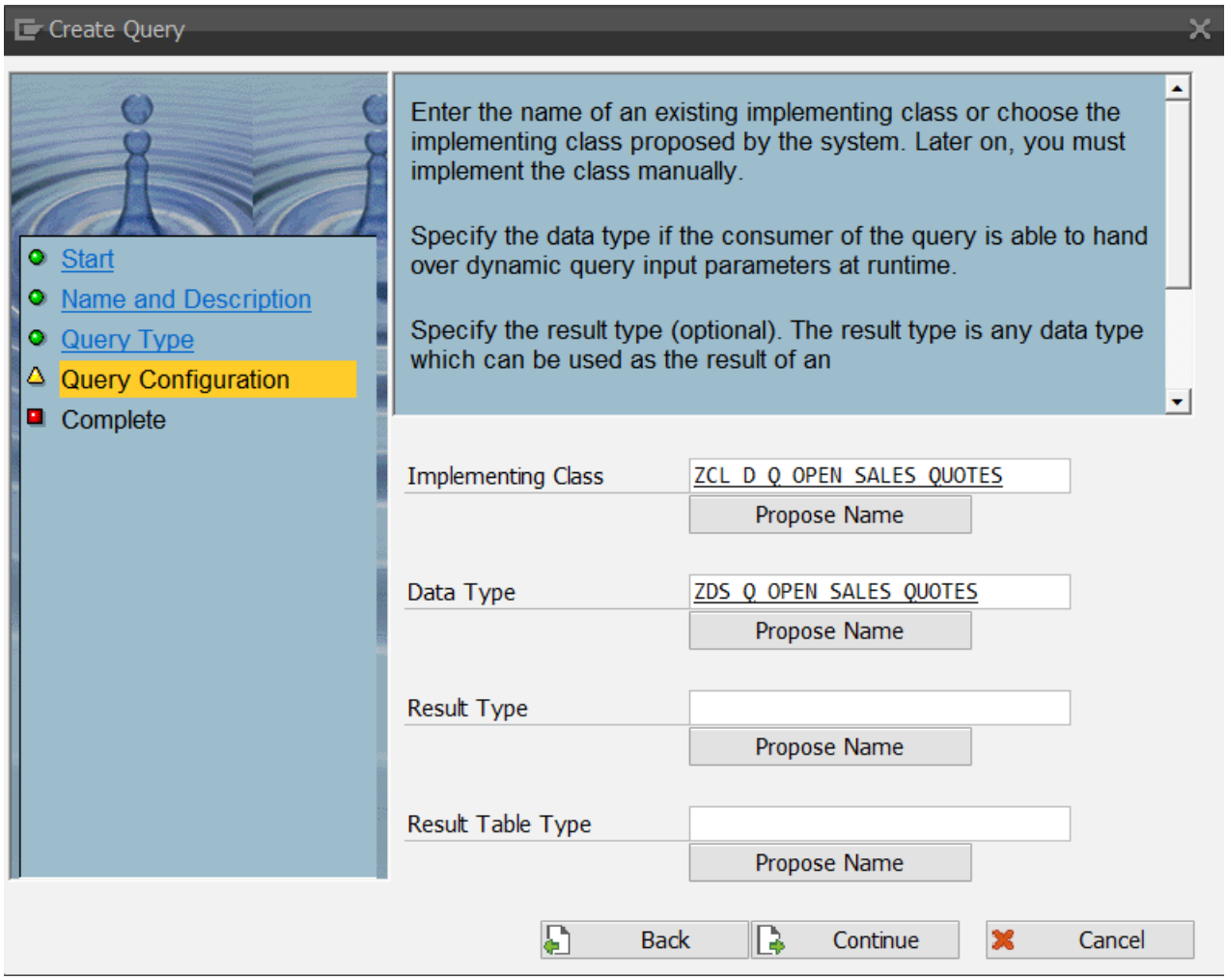

Fig. 12: Enter the name of the implementing class and the data structure

You can use the combined data type of the root node as data type. Normally you do not want to offer every attribute as a selection criterion, only some individual ones. Thus we have to create a new data type in the dictionary. Accept the proposed name and navigate to the DDIC through double-click. Enter the attributes you want to offer as selection parameters. In our case we need the key fields and the QUOTE_ID. We do not offer the QUOTE_STATUS as we will set a constraint for this attribute in the implementation.

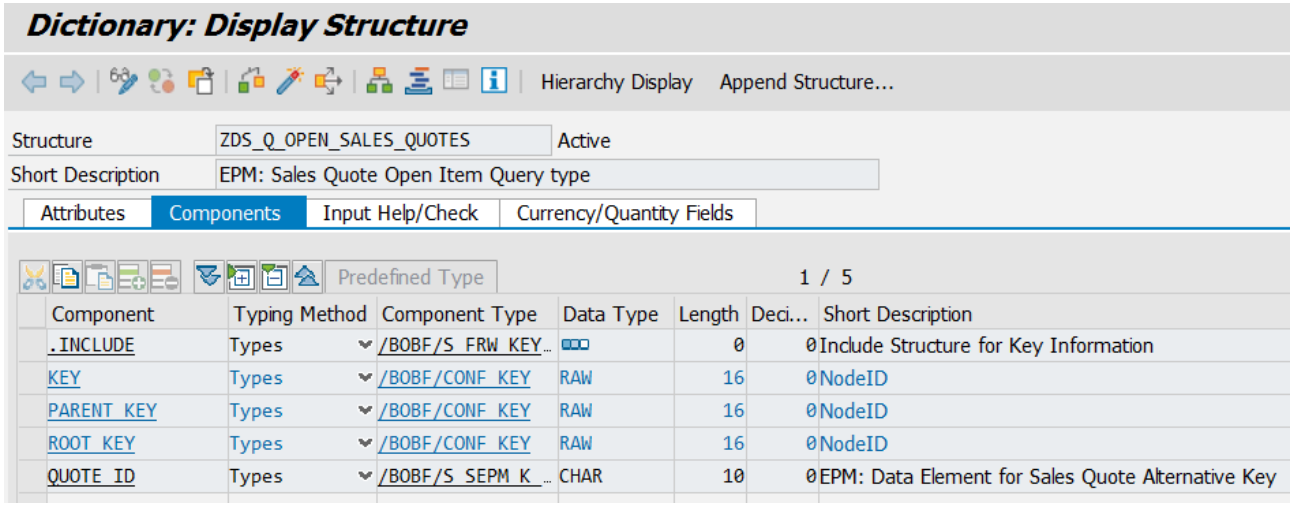

Fig. 13: Input data structure

Activate your definition and return to the wizard. Continue with the next step.

Finish the Wizard

Finish the query creation by choosing **Complete**. In the background, the system updates the BO configuration and generates the implementing class of the query.

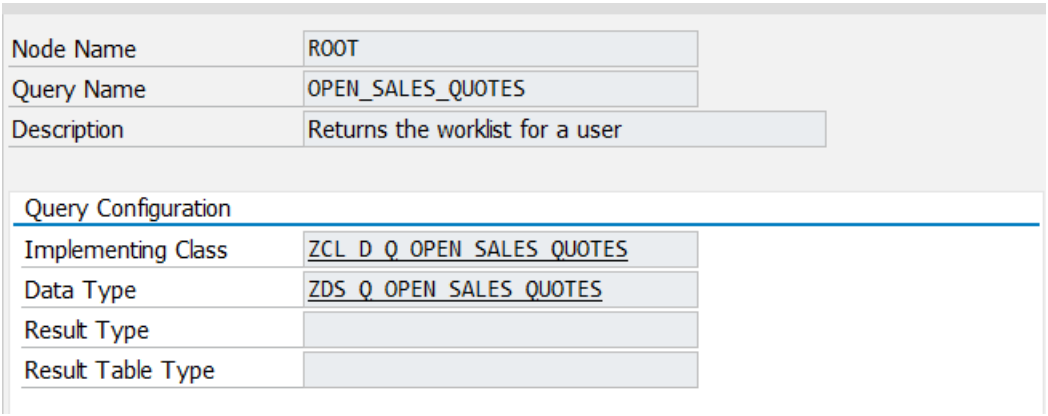

Fig. 14: Result of the Wizard

Implement the Query

The new query is now visible in the Entity Browser pane of the BO configuration view. On the right, the settings of the selected query are shown. Double-click the name of the implementing class to navigate to the class builder. Open the empty implementation of method */BOBF/IF_FRW_QUERY~QUERY* and provide the following source code.

```
  METHOD /bobf/if_frw_query~query.
    DATA: lt_selection_parameters TYPE /bobf/t_frw_query_selparam,
          ls_selection_parameters TYPE /bobf/s_frw_query_selparam.

    CLEAR: eo_message,
           et_key,
           es_query_info,
           et_data.

*   Predefined selection parameters
    ls_selection_parameters-attribute_name =
              zif_d_sales_quote_c=>sc_node_attribute-root-quote_status.
    ls_selection_parameters-sign   = 'I'.
    ls_selection_parameters-option = 'NE'.
    ls_selection_parameters-low    = 'P'.
    APPEND ls_selection_parameters TO lt_selection_parameters.
*   Query utility
    io_query->query(
      EXPORTING
        is_ctx                  = is_ctx
        it_filter_key           = it_filter_key
        it_selection_parameters = lt_selection_parameters
        io_query_authorities    = io_query_authorities
        is_query_options        = is_query_options
        io_query                = io_query
        io_read                 = io_read
        io_modify               = io_modify
        iv_fill_data            = iv_fill_data
        it_requested_attributes = it_requested_attributes
      IMPORTING
        eo_message    = eo_message
        et_key        = et_key
```

```
        es_query_info = es_query_info
        et_data       = et_data ).


  ENDMETHOD.
```

Hint: if you copy the source code directly into the ABAP editor, the formatting will be lost. Copying it into WordPad, on the other hand, preserves at least the line breaks. This format can then be copied into the ABAP editor.

In the first step we set the selection parameters. We want to select all open sales quotes – this means we want to exclude the status published. Then we use the query utility to execute the query.

Activate your code and navigate back to the configuration view of the Business Object Builder.

Result

You are finished with the implementation of the query.

TEST THE QUERY

We are going to test the query.

Procedure

Start Business Object Test Environment

In the configuration view of BOB, press the **Test** button in the toolbar. Then open the folder and select By Query. You see the new query OPEN_SALES_QUOTES for the root node.

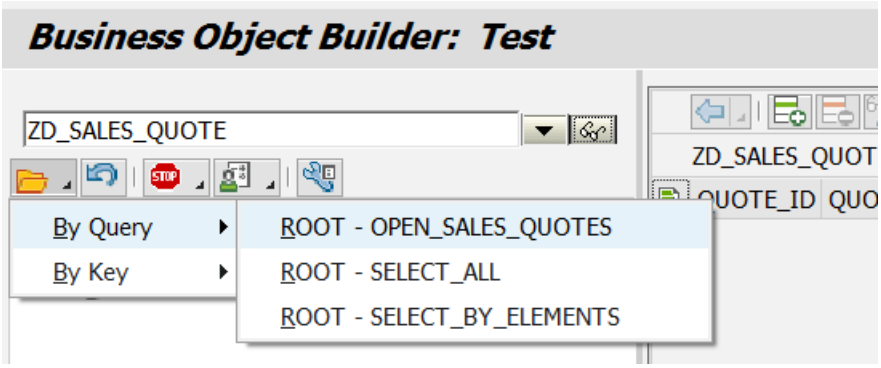

Fig. 15: New Query is visible

Select the Query ROOT – OPEN_SALES_QUOTES.

The next screen offers the selection parameters of the query. The QUOTE_STATUS attribute does not appear since we excluded it in the definition of the data type.

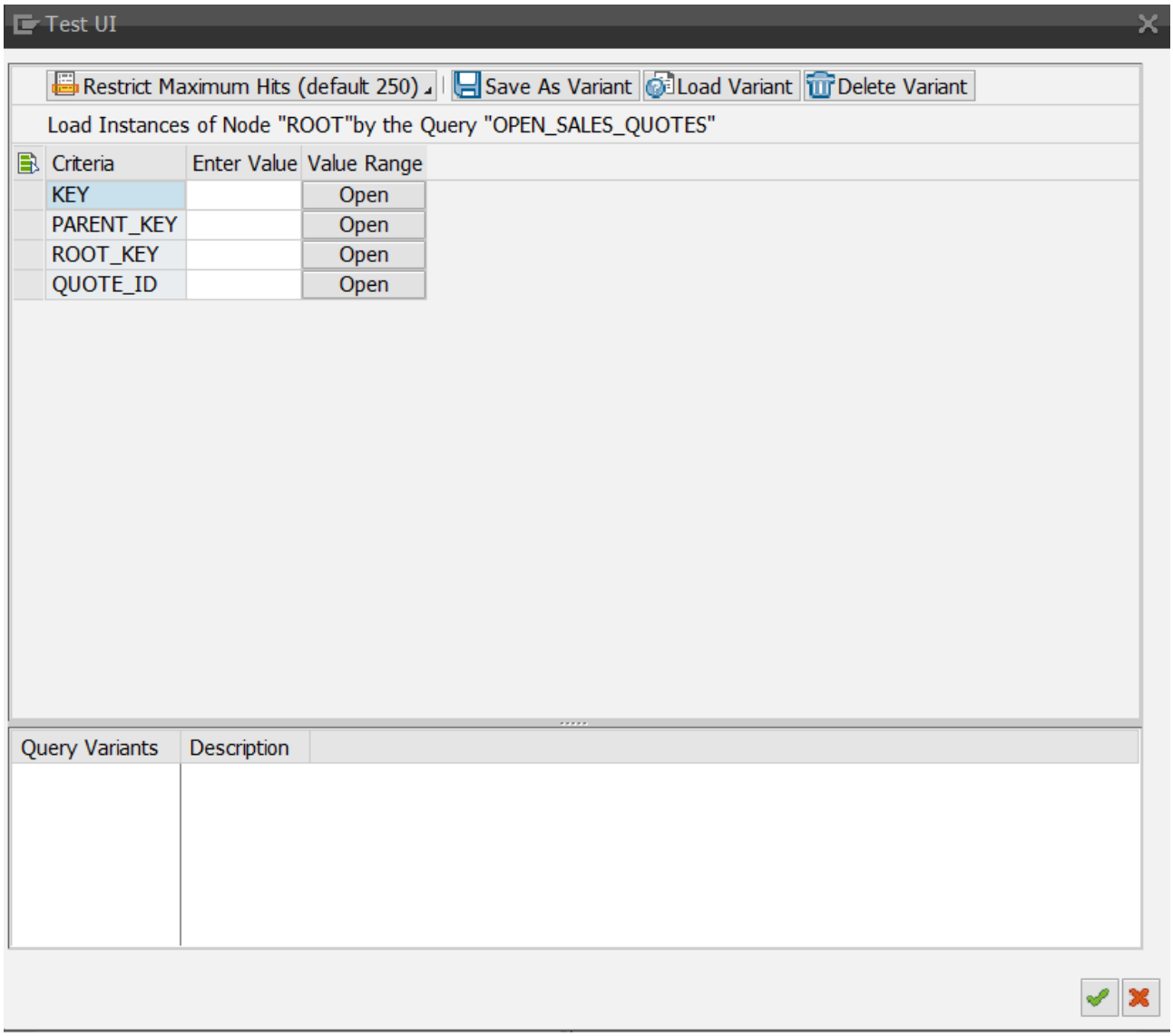

Fig. 16: Selection Screen of the Query – without QUOTE_STATUS

We do not enter further selection parameters but start the query by choosing the Enter button. The query result contains all attributes of the root node since we didn't define a special result type. It does not select the sales quotes in status 'published'.

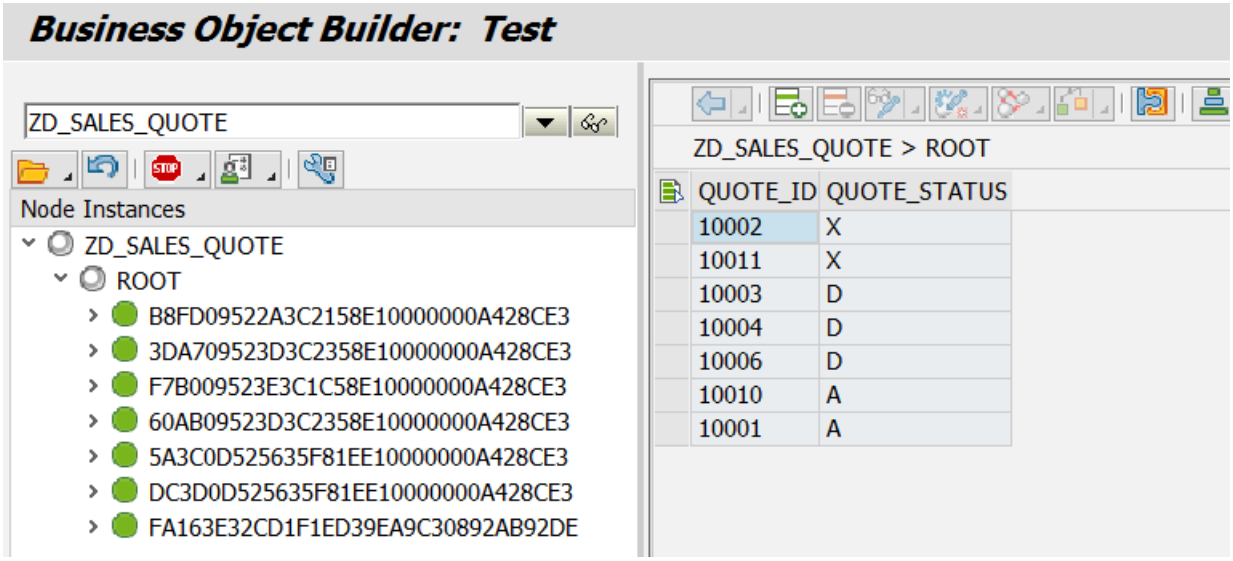

Fig. 17: Query Result

Result

Within a few minutes we have verified that we have implemented the query correctly, without writing any test code. Of course, this does not replace an automated test, but with the help of the Business Object Test Environment you get direct feedback as to whether your Business Object works correctly or not.

We have now gone through the fundamentals of creating and testing queries. I hope you are motivated to try out more things with BOPF, as there is much more that can be discovered. Stay tuned for further articles about our framework.

www.sap.com