

# SAP Plant Connectivity 2.2: Wire to the Outside World for SAP Business Suite-Applications – Sample Query



## Applies to:

SAP Business Suite 7 including Enhancement Packages, SAP Plant Connectivity 2.2. For more information, visit the [Manufacturing homepage](#).

## Summary

Set up your first data exchange queries for your SAP Business Suite applications and OPC Data Access (DA) Servers: Understand how sample report RPCO\_BS\_INT\_TEST works and how easy data point values can be read and change. Implement your first piece of ABAP coding in a user-exit for reading external data.

**Author:** Dr. Uwe Dittes

**Company:** SAP AG

**Created on:** January 31, 2012

## Author Bio



Dr. Uwe Dittes made his PhD in chemistry at the University of Heidelberg, followed by certification as technical operation manager (Chamber of Commerce [IHK]). He worked several years in chemical and pharmaceutical industry (development & production). Dr. Dittes joined the development Organization of SAP in 2001. He has a strong expertise in ERP core manufacturing (SAP PP-PI, SAP PP-SFC) and Business Suite application integration with SAP plant connectivity (PCo). He has also been working as external lecturer at the Baden-Württemberg cooperative State University (State) in Mannheim, teaching fundamentals of SAP ERP, lean production, cloud computing, and mathematics.

## Table of Contents

General Remarks.....	3
Test program RPCO_BS_INT_TEST .....	3
Function Keys on Initial Screen .....	4
Selection Options on Initial Screen .....	4
Parameters for Communication between SAP Business Suite and PCo Agent .....	4
Parameters for Tag Processing .....	5
Parameters for ALV Result Display.....	6
Result List .....	6
Displayed Tag Information .....	6
Function Keys in Result List.....	7
Input Help for Browsing of Namespaces.....	7
Read and Change Data Point Values of OPC DA Servers Using RPCO_BS_INT_TEST.....	8
Read (Retrieve) Data Point Values .....	8
Change (Store) Values of Data Points .....	10
Process Steps for Data Exchange using Queries .....	12
ABAP Sample Implementation for Reading Data Point Values.....	13
Default Values for Order Confirmation .....	13
Scenario.....	13
Implementation .....	13
Sample Coding .....	13
Optimization possibilities .....	16
Related Content.....	17
Copyright.....	18

## General Remarks

We recommend reading first and second part of this set of documents for getting familiar with basic concepts of integrating external data in SAP Business Suite applications by SAP Plant Connectivity (PCo). First chapters of this document provide information about SAP test report `RPCO_BS_INT_TEST`. With the help of this report you are able to execute PCo queries for reading and writing data to connected OPC Data Access (DA) servers. You don't need to write a single line of ABAP code to get the report run.

For those of you who want to implement their own query calls in your applications we also provide an example for accessing data: Default values for time ticket confirmation of production orders (transaction `CO11N`) are provided by tag values of OPC DA server.

In addition please take following remarks into account:

- This document is based on the implementation guide that describes how to connect external data sources to Business Suite with SAP Plant Connectivity (PCo) Release 2.2.
  - Implementation Guide is part of SAP note 1576651 and could be updated without further notice.
- Please also check for corrections for following application components:
  - BC-SRV-PCO Plant Connectivity (Pco): ABAP part
  - MFG-PCO MII Plant Connectivity (replaces component MDS)
- To connect your application, you need to have a good knowledge in ABAP programming and ABAP Objects programming. In addition, you should know how SAP Plant Connectivity (PCo) works.
- You must refer to the system and development prerequisites specified in the documents.
- If you need additional help for the implementation after reading this document, you should request SAP remote-consulting (see SAP note 83020).
- The complimentary Customer Support of SAP will not support errors that are caused by modifications of the SAP standard system or by customer development (see SAP note 7).

## Test program `RPCO_BS_INT_TEST`

The new test program `RPCO_BS_INT_TEST` provides a basic understanding of the communication between *PCo* and the *Business Suite* applications. This program shows how easy it is to carry out the necessary implementations on the side of the *Business Suite* applications, so that external data sources can be connected to the *Business Suite* via *PCo*.

The focus of the test program is on the data exchange between the *Business Suite* application and external data sources can get read/write access to the *PCo* tags. The test program contains the following default values for the application handle:

- APPLICATION: SAPTESTING
- HANDLE: LOG

You can overwrite the default values of the test program with your own values if necessary.

If you want to carry out the sample implementation for the processing of notifications using the test program, you must pass the value `SAPTESTING` to the application handle for the application (APPLICATION).

The test program contains all functions that are provided in the SAP standard for the integration of external data sources using tag access options:

- Display of the supported agent features of *PCo* agent instances
- Browsing of namespaces for the tag hierarchies
- Reading of tags
- Writing of tag value changes
- Subscription to tags to be used for notifications
- Display of information on subscribed tags
- Extension of subscriptions
- Deletion of subscriptions
- Display of notification templates
- Display of application logs generated for the log object `S_PCO`.

## Function Keys on Initial Screen

On the initial screen of the test program, the following function keys are displayed:

Function key	Meaning
<i>Execute</i>	The test report is executed according to the selection conditions maintained on the initial screen.
<i>Get Variant</i>	Previously created report variants can be loaded. The selection conditions of the initial screen must previously have been saved as a report variant ( <i>Save</i> pushbutton).
<i>Log Display</i>	Branches to the selection of application logs that were generated and saved for the integration of <i>Business Suite</i> applications.
<i>Features/Properties of PCo Agent</i>	Display of properties (reading, writing, subscription, support of native filters, etc.) of the <i>PCo</i> agent instance connected with the <i>Business Suite</i> system via a selected RFC connection.
<i>Properties of RFC Destination</i>	Enables the branching to the detail view of the RFC connection data maintenance for the selected RFC connection (transaction <i>SM59</i> ).

## Selection Options on Initial Screen

### Parameters for Communication between SAP Business Suite and PCo Agent

For the communication between the Business Suite and the PCo agent, make entries for the following parameters:

Parameter	Meaning
RFC connection of the <i>PCo</i> agent instance	The F4 input help on a field lists all the RFC connections of the type TCP/IP that exist in the system.
Checkbox <i>RFC Connection Test</i>	If you select this checkbox, the system carries out an RFC connection test for the <i>PCo</i> agent that is to be accessed via the RFC connection when the object instance is generated. The results of the connection test are displayed in the status line.
Input field <i>Application</i>	The contents of the field are interpreted as the application name of the application handle and must have 10 characters.
Input field <i>Application handle</i>	The contents of the field are interpreted as the “handle content” of the application handle. The field content must contain at least one character. A maximum of 80 characters can be maintained as handle information.
Checkbox <i>Logging active</i>	If the checkmark is set, application logs are generated and saved automatically when functions of the test report are executed.
Checkbox <i>Automatic Load Features</i>	When the object instances are generated, the features of the corresponding <i>PCo</i> agent instances are loaded and buffered if you set this checkmark.
Checkbox <i>Test Mode (without PCo)</i>	If you set this checkmark, the test report runs in test mode and uses hard-coded data for the results display. There is no communication with <i>PCo</i> . The test mode serves primarily to simulate the interface.

## Parameters for Tag Processing

For the processing tags, the following parameters are significant:

- Field Mode Namespace Browsing. This field enables one of three different modes:
  - Mode Preselection by Tag Hierarchy: Here the tag hierarchy of the connected external data source is displayed in a tree structure. If you select individual tags in the hierarchy, then the respective tag values, together with the fully qualified path and further details will be displayed when the test report is executed. If you select a group folder, the system will determine all tags (child nodes) having the group folder as a superior node. When you then execute the program, the system determines further detail information on the tags and displays them in the results list.
  - Mode Preselection by Specifying Qualified Path ID (F4 help): When you select this mode, an additional input field for the fully qualified path is displayed. Proceed as follows:
    - After pressing the F4 pushbutton on this field, the tag hierarchy is displayed in a tree structure.
    - You can now select an entry (tag or tag group). The system returns you to the initial screen, where you can see your selection.
- The system returns you to the initial screen, where you can see your selection with the fully qualified path of the tag or tag group.
- Now you can run the program. The system reads further tag data from the external data source and displays this in the results list. If you select a tag group, the display of the respective detail information is displayed in the results list.
- Mode Preselection via Filter Value for Tags. When you select this mode, two further fields are displayed for entry:
- Input field Masking: Via the F4 help you can select the type of filter to be used for tag searches. Depending on the PCo settings made for the corresponding agent instance, the search may be supported by the selected filter type. The following filter types are available:
- Filter type NATIVE: The filter uses the filter options of the connected data source (this filter access usually has very high performance)
- Filter type REGEX: The filter uses regular expressions
- Filter type LEGACY: The filter uses the filter logic of the former PCo function UDS (Universal Data Sources), which is still in use by SAP MII for the connection of external data sources.
- Input field Filter-String for Tag Alias: Here you can search for the name or alias of a tag using wildcards (\*,\*). If tags are found, they are displayed in the results list when you run the test program.
- For tag determination, there are the following functional restrictions:
- No tags are found for which the search string is only a part of the fully qualified path, for example:
- Searching for `'*Tag_A'` returns the tag `'Root/Temperature/TempTag_A'`, but not the tag `'Root/TagGroup_A/Temp_A'`
- The fully qualified path information cannot be determined for these tags. For the tag subscription however, the fully qualified path of the tags is required. This means that subscriptions are not possible. In this test program, therefore, the function keys for the subscription, extension of subscriptions and deletion of subscriptions is hidden in the ALV results display. In addition values of column Tag Alias (shortened) are not editable.
- Input field Maximal value (MaxRows)
- Using this input parameter, you can limit the number of tags and tag groups found. If a tag group is selected, and the maximal value is set at "6", the system reads the first six sub-groups as well as the first six tags, which are child nodes for the selected tag group. If more tags or tag groups exist than were selected, the system will output the corresponding message.
- Input field Number of Decimals (Floating Point)
- For floating point figures, the decimal display is usually preferred. A value of "0" leads to the display of floating point numbers without limiting the number of decimals. Other values limit the number of decimals and lead to rounded floating point numbers.
- Due to the internal display of the decimal places for a floating point number of the type "f" using dual fractions, there is no exact number corresponding to every number in the decimal system. This means that assignments and interim results of calculations can contain rounding errors, which can only be avoided by using a two-level rounding procedure.

## Parameters for ALV Result Display

### Input field *ALV Display Variant*:

For the display of the results list, you can maintain application-specific ALV display variants. The input help for this field contains the variants maintained.

## Result List

If tags were found for the selection conditions defined, the system subsequently reads additional tag data and displays the results in a table overview (ALV grid).

Tag Alias (Shortened)	Qualified Tag ID	Subsc	Tag Value (Shortened)	Value Chg.	Short Descript	Time Stamp	Date	Time
UD_CONF_QUANT	Channel_0_User_DefinedA_Users tags/UD_CONF_QUANT		0.000000000000000		SQL Data Type DOUBLE	23.120.228.144.554	28.02.2012	15:45:54
UD_CONF_RETURNS	Channel_0_User_DefinedA_Users tags/UD_CONF_RETURNS		0.000000000000000		SQL Data Type DOUBLE	23.120.228.144.554	28.02.2012	15:45:54
UD_CONF_SCRAP	Channel_0_User_DefinedA_Users tags/UD_CONF_SCRAP		0.000000000000000		SQL Data Type DOUBLE	23.120.228.144.554	28.02.2012	15:45:54
UD_SHORT	Channel_0_User_DefinedA_Users tags/UD_SHORT		0		SQL Data Type SMALLINT	23.120.228.144.554	28.02.2012	15:45:54
UD_STATUS	Channel_0_User_DefinedA_Users tags/UD_STATUS		String 100		SQL Data Type LONGVARCHAR	23.120.228.144.554	28.02.2012	15:45:54

Figure 1: Result List (ALV Grid) of Report RPCO\_BS\_INT\_TEST

## Displayed Tag Information

The following data is displayed in the results list:

Information	Meaning
Name or alias of the tag ("Tag alias")	The alias can be changed when subscribing. As long as a subscription exists, the alias name will always be displayed. Otherwise, the system will output the tag name.
Fully qualified path information ("Qualified Tag ID")	Display of the fully qualified path in the internal <i>PCo</i> format (with "/" as a separator): <ul style="list-style-type: none"> <li>For the communication with <i>PCo</i>, fully qualified paths must always be passed to <i>PCo</i> from tags in the <i>PCo</i>-internal format.</li> </ul>
Subscription (icon)	Indicates whether subscriptions exist for a tag or not
Shortened display of the tag value (max. 40 characters)	<ul style="list-style-type: none"> <li>The field is ready for input</li> <li>Invalid tag values are indicated by the string "---".</li> </ul>
Value change	In order to carry out changes in values for the specified tags, you next have to carry out the function to write the value change.
Short description	Short text on the SQL data type of the tag
Timestamp	Timestamp in UTC format
Date and Time fields	Day and time of day

## Function Keys in Result List

The following function keys are available:

Function key	Meaning
<i>Display Agent Features</i>	Dialog box display of all functions available for the PCo agent
<i>Application Logs</i>	Display of application logs that are used for the application log object <b>S_PCO</b>
<i>Display Subscriptions</i>	Dialog box display of information on subscriptions that exist for the specified application handle: <ul style="list-style-type: none"> <li>Name of the subscription</li> <li>Name or alias of the tab</li> <li>Fully qualified path of the tag in native display</li> </ul>
<i>Notification Templates</i>	Dialog box with the name and description of notification templates maintained for the selected PCo agent instance
<i>Subscribe</i>	Selection of tags from the results list for which notifications are to be created if the tag values change. <ul style="list-style-type: none"> <li>After pressing this pushbutton, the system displays the existing notification templates maintained for the PCo agent.</li> <li>After selecting a notification request, the subscription is executed.</li> <li>Tags that were successfully executed will display an icon in the "Subscr." column of the results list.</li> </ul>
<i>Delete Subscriptions</i>	Removal of subscriptions: <ul style="list-style-type: none"> <li>If no entries are marked in the results list, the system deletes all subscriptions that exist for the specified application handle.</li> <li>If individual entries of the results list are selected for which subscriptions exist, the system deletes these tags from the subscriptions.</li> </ul>
<i>Extend Subscription</i>	Extension of the validity period for all subscriptions of the application handle. The validity period is extended by changing the validity start to the value of the current date and time.
<i>Write Values Change</i>	Passing of changed tag values (in column <i>Value Chg.</i> ) to the external data source that is linked to the Business Suite via a PCo agent instance
<i>Update Display</i>	Renewed loading of detail information for the selected tags

## Input Help for Browsing of Namespaces

The SAP Business Suite applications require an easy way to process tags and tag groups of an external data source. The elementary search help **S\_PCO\_ELM\_BROWSE\_TAG** (package **S\_PCO**) serves as a sample template for the selection of a tag or a tag group. The namespace of the tags is displayed in a dialog box. After you select a tag or tag group, the search help returns the fully qualified path of the selected object.

The following search parameters are available:

Search help field	Meaning
MAX_ENTRIES	Entry of the maximum value for the selection of tags and tag groups. The entered value must be > 0.
RFC_DEST	Entry of the RFC connection for the selected PCo agent instance
SIMUL_MODE	Execution of the search help in simulation mode ("X"). Note that there is no communication with the PCo agent instance, but that instead, hard-coded data is accessed.
TAG_ID	Output field for the first 255 characters of the fully qualified path for the selected object, in the PCo-internal format (with "/" as separator)
TAG_DESCR	Output field for the description of the tag: <ul style="list-style-type: none"> <li>This only receives values in simulation mode.</li> </ul>

Search help field	Meaning
	<ul style="list-style-type: none"> <li>In productive mode, the tag description can currently not be determined: This PCo metadata is currently not returned to the calling Business Suite application when a PCo request is executed.</li> </ul>
NODE_TEXT	Output field for the name of the tag or tag group
IS_GROUP	Output field for the indicator specifying whether the selected object is a tag group or a tag

Due to the width of 464 characters, it is not possible to create a personal search input help. The output of the fully qualified path must also be limited to a maximum of 255 characters when the character string is output in search help screens.

If you want to avoid the above restriction, then you can also use the function module **S\_PCO\_CALL\_POPUP\_NAMESP\_BROWS** (package **S\_PCO**, function group **S\_PCO**) instead of the search help.

## Read and Change Data Point Values of OPC DA Servers Using RPCO\_BS\_INT\_TEST

So far you have been creating all required data that allow you accessing external data. Now read and change this data by executing PCo queries.

### Read (Retrieve) Data Point Values

First query provides you with values of data points (tags). Proceed as follows:

- Start management console of PCo 2.2 installation. Start PCo agent instance you created
- Log on to Business Suite test system where you created RFC connection which is linked to started PCo agent instance. Start transaction SE38. Enter program RPCO\_BS\_INT\_TEST and press 'Execute' button (F8 key). Selection screen of report appears after pressing F8.
- Selection screen of report:
  - Choose previously created RFC destination as field value for parameter 'RFC Dest. PCo Agent Instance'
  - Unmark check box for 'Test Mode'
  - Keep default values for other parameters, e. G.:

**PCo: Example Program for Business Suite Connection**

☺ ☰ ☷ ⌚ 🖋

Communication Between Business Suite - PCo Agent

RFC Dest. PCo Agent Instance

RFC connection test

Application

Application Handle

Logging Active

Test Mode (Without PCo)

---

Processing Tags

Mode Namespace Browsing

Maximum Number of Rows

No. Decimals (Floating Point)

**Figure 2: Selection Screen Parameters for Calling PCo DA Agent**

- Press 'Execute' button (F8 key). Because mode for browsing hierarchy of data points (namespace) is set to 'Preselection by Tag Hierarchy', system first displays nodes in an ALV Tree control:

Tag Hierarchy	Namespace Path Information	Description
Namespace (PCo Agent)		
_System	_System	
Channel_0_User_Defined	Channel_0_User_Defined	
Channel_1	Channel_1	
Channel_2	Channel_2	
Channel_3	Channel_3	
Channel_4	Channel_4	
_DataLogger	_DataLogger	

Figure 3: Node Hierarchy (ALV Tree)

- Open folders and mark several tag nodes (by keeping CTRL/STRG key pressed). Press ENTER

Tag Hierarchy	Namespace Path Information	Description
Namespace (PCo Agent)		
_System	_System	
Channel_0_User_Defined	Channel_0_User_Defined	
_System	Channel_0_User_Defined/_System	
A_Device_8_Bit_UD	Channel_0_User_Defined/A_Device_8_Bit_UD	
A_Uwes tags	Channel_0_User_Defined/A_Uwes tags	
_System	Channel_0_User_Defined/A_Uwes tags/_System	
_Hints	Channel_0_User_Defined/A_Uwes tags/_Hints	
Uwes_Gruppe1	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1	
UD_CONF_QUANT	Channel_0_User_Defined/A_Uwes tags/UD_CONF_QUANT	
UD_CONF_REWOR	Channel_0_User_Defined/A_Uwes tags/UD_CONF_REWORK	
UD_CONF_SCRAP	Channel_0_User_Defined/A_Uwes tags/UD_CONF_SCRAP	
UD_SHORT	Channel_0_User_Defined/A_Uwes tags/UD_SHORT	
UD_STATUS	Channel_0_User_Defined/A_Uwes tags/UD_STATUS	
Uwes_Trigger_String	Channel_0_User_Defined/A_Uwes tags/Uwes_Trigger_String	

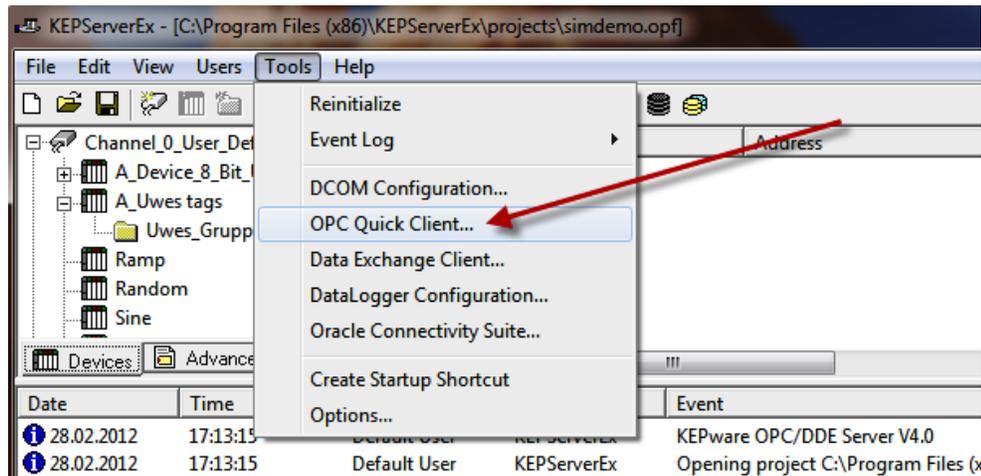
Figure 4: Selected Data Points and Data Point Group Folders

- Result list displays detail data of selected tag nodes:

Tag Alias (Shortened)	Qualified Tag ID	Subst.	Tag Value (Shortened)	Value Chg.	Short-Descript.	Time Stamp	Date	Time
UD_CONF_QUANT	Channel_0_User_Defined/A_Uwes tags/UD_CONF_QUANT		0,0000000000000000		SQL Data Type DOUBLE	20.02.2012 19:00:01	28.02.2012	17:06:01
UD_CONF_REWORK	Channel_0_User_Defined/A_Uwes tags/UD_CONF_REWORK		0,0000000000000000		SQL Data Type DOUBLE	20.02.2012 19:00:01	28.02.2012	17:06:01
UD_SHORT	Channel_0_User_Defined/A_Uwes tags/UD_SHORT		0		SQL Data Type SMALLINT	20.02.2012 19:00:01	28.02.2012	17:06:01
Uwes_Boolean	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Uwes_Boolean				SQL Data Type BOOLEAN	20.02.2012 19:00:01	28.02.2012	17:06:01
Uwes_String	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Uwes_String		String 42		SQL Data Type LONGVARCHAR	20.02.2012 19:00:01	28.02.2012	17:06:01
Uwes_String 2	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Uwes_String 2		String 43		SQL Data Type LONGVARCHAR	20.02.2012 19:00:01	28.02.2012	17:06:01
Uwes_String 2	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Uwes_String 2		String 44		SQL Data Type LONGVARCHAR	20.02.2012 19:00:01	28.02.2012	17:06:01

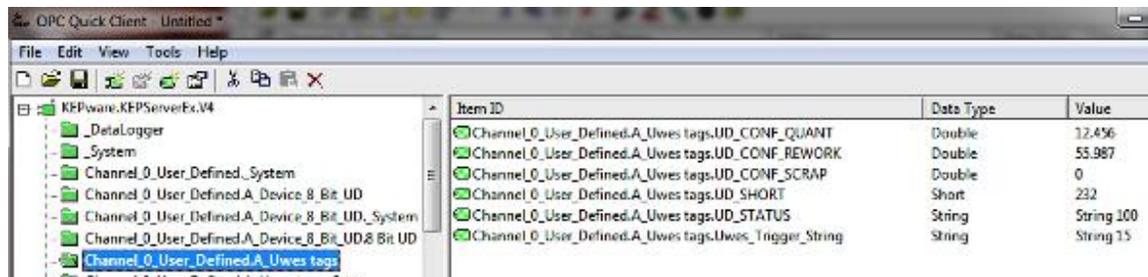
Figure 5: Results of Selection (ALV Grid)

OPC Server suppliers provide their own client applications that allow reading and changing values of data points. As an example Kepware offers such an external OPC client tool named *OPC Quick Client*.



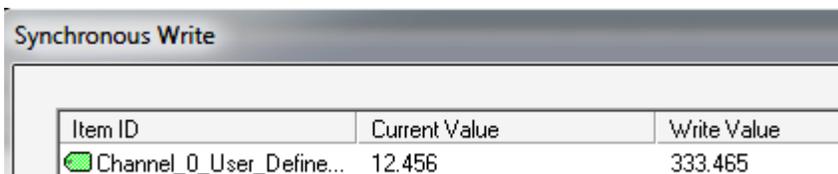
**Figure 6: OPC Quick Client**

You can use OPC Quick Client for double-checking results of data point value retrieval. But it also allows you to change data point values:



**Figure 7: View on Selected Data Points Using OPC Quick Client**

- Change values for your selected data points by using OPC Quick Client.



**Figure 8: Value Change (OPC Quick Client)**

- Then switch to result list of test report RPCO\_BS\_INT\_TEST again.
- Press 'Refresh' button which is located in toolbar of result list (F9 key).
- Result list shall display changed tag values after refresh is executed



**Figure 9: Result List after Refresh**

### Change (Store) Values of Data Points

Next level of difficulty is reached when you try to change values of data points by SAP Business Suite applications. For this purpose we also use our sample report for selecting data points and reading their values. So proceed as described in previous chapter.

- Take care that you select OPC DA data points that are assigned to changeable memory areas of the OPC server (so you can overwrite their values). Also check that values of data points aren't periodically changed by OPC server using functions for value increase by ramping or random value assignment.
- On result list, change values in editable column and press ENTER

- Rows that contain changed values are marked by pencil icon

Overview of Selected Tags			
Tag Alias (Shortened)	Qualified Tag ID	Tag Value (Shortened)	Value Chg.
UD_CONF_QUANT	Channel_0_User_Defined/A_Uwes tags/UD_CONF_QUANT	12,456	
UD_CONF_REWORK	Channel_0_User_Defined/A_Uwes tags/UD_CONF_REWORK	55,98700000000000	
UD_SHORT	Channel_0_User_Defined/A_Uwes tags/UD_SHORT	232	
Mein_Boolean	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Mein_Boolean		
Mein_String	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Mein_String	New value	
Uwes String	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Uwes String	String 43	
Uwes String 2	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Uwes String 2	String 44	

Figure 10: Enter new Values for OPC DA Data Points

- Push button Write Value Change (F2) to update marked data points.



Figure 11: Transfer Changed Values to OPC DA Server

- After storing changed data system returns information about query execution via log messages
  - Logging messages are displayed because checkbox of report parameter 'Logging Active' was ticked

Type	Message Text
	Operation successful

Figure 12: Logging Messages

- After storing changes system automatically refreshes result list:

Overview of Selected Tags			
Tag Alias (Shortened)	Qualified Tag ID	Tag Value (Shortened)	Value Chg.
UD_CONF_QUANT	Channel_0_User_Defined/A_Uwes tags/UD_CONF_QUANT	12,45600000000000	
UD_CONF_REWORK	Channel_0_User_Defined/A_Uwes tags/UD_CONF_REWORK	55,98700000000002	
UD_SHORT	Channel_0_User_Defined/A_Uwes tags/UD_SHORT	232	
Mein_Boolean	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Mein_Boolean		
Mein_String	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Mein_String	New value	
Uwes String	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Uwes String	String 43	
Uwes String 2	Channel_0_User_Defined/A_Uwes tags/Uwes_Gruppe1/Uwes String 2	String 44	

Use external OPC client tools for cross-checking your value changes.

## Process Steps for Data Exchange using Queries

The following steps are necessary to carry out and evaluate *PCo* queries:

Step	System	Activity
1	SAP Business Suite	Creation of an RFC connection (transaction <i>SM59</i> ): <ul style="list-style-type: none"> <li>• Maintain the Gateway options</li> <li>• Provide a name for the registered server program</li> <li>• Optional: Maintain the data for a secure network communication (<i>SNC</i>)</li> </ul>
2	SAP Business Suite	Maintenance of authorizations (transaction <i>PFCG</i> , authorization object <b>S_PCO_INT</b> ) for the authorization profile of the user to be utilized for the RFC data exchange between the Business Suite and <i>PCo</i> . You can use the following <i>PFCG</i> profiles as templates: <ul style="list-style-type: none"> <li>• <i>SAP_BC_SRV_PCO_BS_INT_ADMIN</i> and</li> <li>• <i>SAP_BC_SRV_PCO_BS_INT_USER</i></li> </ul>
3	SAP Plant Connectivity ( <i>PCo</i> )	Generating of a <i>PCo</i> agent instance for the external data source which is to communicate with the <i>Business Suite</i> application. First you must add a source and a destination system (see next steps).
4	SAP Plant Connectivity ( <i>PCo</i> )	Adding of the <i>PCo</i> source system <ul style="list-style-type: none"> <li>• Selection of a source system type (<i>OPC DA agent</i>, <i>OLE DB agent</i>, etc.)</li> <li>• Maintenance of the RFC server connection data</li> </ul>
5	SAP Plant Connectivity ( <i>PCo</i> )	Adding of the <i>PCo</i> destination systems <ul style="list-style-type: none"> <li>• Selection of the <i>PCo</i> destination system type <i>RFC Destination</i></li> <li>• Maintenance of the RFC Client data <ul style="list-style-type: none"> <li>○ Selection of <i>SAP NetWeaver</i> for the <i>PCo</i> Client type</li> <li>○ Maintenance of the application server and additional connection data for the corresponding <i>SAP Business Suite</i> system</li> </ul> </li> </ul>
6	SAP Plant Connectivity ( <i>PCo</i> )	Creation of the <i>PCo</i> agent instance for the above-mentioned <i>PCo</i> source and destination systems <ul style="list-style-type: none"> <li>• Maintenance of the service user name and password for the execution of the <i>PCo</i> agent instance as an <i>MS Windows</i> service</li> <li>• Maintenance of the <i>PCo</i> Query Ports for the source system type <i>SAP NW RFC Server</i></li> <li>• Maintenance of the RFC server settings <ul style="list-style-type: none"> <li>○ Entering the name of the registered server program that was previously specified in the <i>SAP RFC</i> connection data</li> <li>○ Assignment of the <i>PCo</i> destination system providing the information for the Repository structure of the <i>SAP DotNet Connector</i> (<i>SAP NCo</i>)</li> </ul> </li> </ul>
7	SAP Business Suite	Implementation of method calls for the <i>ABAP Wrapper</i> classes of package <i>S_PCO</i> . <ul style="list-style-type: none"> <li>• Assignment of a name for the application handle to be passed to <i>PCo</i></li> <li>• Creation of the method calls in the appropriate coding segments (<i>User Exit</i>, <i>BAdl</i>, etc.) of the <i>Business Suite</i> applications</li> </ul>
8	SAP Plant Connectivity ( <i>PCo</i> )	Starting of the <i>PCo</i> agent instance
9	SAP Business Suite	Execution of the <i>Business Suite</i> application

Method calls of the NetWeaver integration layer must be implemented at a suitable point in the code of the SAP Business Suite application. Embedding these method calls in the Business Suite applications can be time consuming depending on the complexity. Searching and finding appropriate user exits or BAdI calls in the code of the Business Suite application is often the biggest challenge.

## ABAP Sample Implementation for Reading Data Point Values

The following sections contain ABAP sample implementations showing how easily the exchange of information between external data sources and Business Suite applications can be implemented on the ABAP side.

For the communication with the external data source, it is necessary in addition to create and start the corresponding PCo agent instances.

### Default Values for Order Confirmation

#### Scenario

In this scenario, production progress of a discrete manufacturing facility is monitored via confirmations. For this, the user creates a time ticket confirmation in transaction CO11N. After pressing the pushbutton *Propose actual data*, the input fields are prefilled with the corresponding data of the operation. The machine used for production provides the actual data for yield, scrap and manual rework. This data is to be transferred when the actual data is automatically proposed for the corresponding input fields.

#### Implementation

A reading access to the data of the external data source is required. A suitable coding section for the implementation of the corresponding PCo query would be the Include **ZXCOFU06**, which belongs to the user exit **CONFPP01**. The coding of the user exit is processed when the user presses the pushbutton *Propose actual data*.

#### Simplification

For reasons of simplification, the fully qualified paths are specified in hard-coded form for the tags of the data source. The entry of a fully qualified path must always be specified in the PCo-internal format (with "/" as separator). Please note the following:

- Use the test program **RPCO\_BS\_INT\_TEST** or the search help **S\_PCO\_ELM\_BROWSE\_TAG** to determine the fully qualified path.
- If the fully qualified path of the tag is too long, you can use function module **S\_PCO\_CALL\_POPUP\_NAMESP\_BROWS** and evaluate the output parameter **ET\_SEL\_NODES**. You can get the fully qualified path as a string variable from the component **TAG\_ID**.

#### Implementation of a User Exit

1. Start transaction CMOD and create a new project.
2. Assign the enhancement **CONFPP01**.
3. Switch to the component view and position the cursor on the entry **EXIT\_SAPLCORF\_101** of the function module exit.
4. When you double-click, the system switches to the display of the function module **EXIT\_SAPLCORF\_101**.
5. Create the implementation for the user exit by putting the cursor on **ZXCOFU11** and double-clicking it.
6. Insert the following source code.
7. Save and activate your changes and finally, activate the user exit.

#### Sample Coding

The following sections of the corresponding ABAP coding are provided below:

- Generation of an application handle
- Instantiation of a wrapper class for the integration of PCo (**CL\_PCO\_PAC**)
- Creation of a buffer table with information on the tags whose values are to be read

- Calling of the wrapper method for the reading of tag information
- Conversion of query results (data reference) into floating point values and assignment to the corresponding fields of the confirmation structure.
- Error handling or success message

## ABAP Coding

```

*&-----*
*& Include          ZXC0FU11
*&-----*

DATA:
  lt_tag_data      TYPE pco_t_tag_data,
  lt_tag_results  TYPE pco_t_query_result_tag_data.
DATA:
  ls_appl_handle  TYPE pco_s_appl_handle,
  ls_tag_data     TYPE pco_s_tag_data,
  ls_tag_result   TYPE pco_s_query_result_tag_data.
DATA:
  lv_log_handle   TYPE balloghndl,
  lv_msg_txt      TYPE string.
DATA:
  lo_pco_exc      TYPE REF TO cx_pco_bs_int,
  lo_pco_msg      TYPE REF TO cl_pco_query_message,
  lo_pco_pac      TYPE REF TO cl_pco_pac,
  lo_pco_util     TYPE REF TO cl_pco_utility,
  lo_tag_query    TYPE REF TO if_pco_tag_query.

CONSTANTS:
  lc_pco_rfc_dest TYPE rfcdest VALUE 'PCO_BS_INT_UD_Q63',
  lc_tag_alias_quant TYPE string VALUE 'UD_CONF_QUANT',
  lc_tag_alias_rework TYPE string VALUE 'UD_CONF_REWORK',
  lc_tag_alias_scrap TYPE string VALUE 'UD_CONF_SCRAP',
  lc_tag_id_quant TYPE string
    VALUE 'Channel_0_User_Defined/A_Uwes tags/UD_CONF_QUANT',
  lc_tag_id_rework TYPE string
    VALUE 'Channel_0_User_Defined/A_Uwes tags/UD_CONF_REWORK',
  lc_tag_id_scrap TYPE string
    VALUE 'Channel_0_User_Defined/A_Uwes tags/UD_CONF_SCRAP'.

FIELD-SYMBOLS:
  <ld_value> TYPE any.

CLEAR: lt_tag_data, lt_tag_results.

* General part: Map default values for AFRUD_EXP
MOVE-CORRESPONDING afrud_imp TO afrud_exp.
* UoM
afrud_exp-meinh = caufvd_imp-gmein.

* PCo integration part
ls_appl_handle-appl = 'PRODORCONF'.

* Create application handle: Use order number, plant, sequence,
* operation, and ID of confirmation

```

```
CONCATENATE caufvd_imp-aufnr caufvd_imp-werks afrud_imp-aplf1
afrud_imp-vornr afrud_imp-rueck INTO ls_appl_handle-handle.
```

```
TRY.
```

```
CREATE OBJECT lo_pco_pac
```

```
EXPORTING
```

```
  is_appl_handle = ls_appl_handle
  iv_check_dest  = abap_false
  iv_load_feat   = abap_false
  iv_log_active  = abap_true
  iv_rfc_dest    = lc_pco_rfc_dest.
```

```
lo_tag_query = lo_pco_pac->get_tag_query_obj( ).
```

```
* Fill buffer table with tag information to be read for confirmation
```

```
* (produced quantity, rework, scrap)
```

```
ls_tag_data-tag_id   = lc_tag_id_quant.
```

```
ls_tag_data-tag_alias = lc_tag_alias_quant.
```

```
INSERT ls_tag_data INTO TABLE lt_tag_data.
```

```
CLEAR ls_tag_data.
```

```
ls_tag_data-tag_id   = lc_tag_id_rework.
```

```
ls_tag_data-tag_alias = lc_tag_alias_rework.
```

```
INSERT ls_tag_data INTO TABLE lt_tag_data.
```

```
CLEAR ls_tag_data.
```

```
ls_tag_data-tag_id   = lc_tag_id_scrap.
```

```
ls_tag_data-tag_alias = lc_tag_alias_scrap.
```

```
INSERT ls_tag_data INTO TABLE lt_tag_data.
```

```
CLEAR ls_tag_data.
```

```
* Read tag data via PCo agent
```

```
lo_tag_query->read_tag(
```

```
  EXPORTING
```

```
    it_tag_data = lt_tag_data
```

```
  IMPORTING
```

```
    et_tag_result = lt_tag_results ).
```

```
* Convert data references into confirmation values
```

```
LOOP AT lt_tag_results INTO ls_tag_result.
```

```
  IF sy-subrc = 0.
```

```
    CASE ls_tag_result-datatype.
```

```
*      Float point values
```

```
      WHEN cl_pco_utility=>gc_sdt_float OR
```

```
      cl_pco_utility=>gc_sdt_real OR
```

```
      cl_pco_utility=>gc_sdt_double.
```

```
*      Get value from data reference
```

```
      ASSIGN ls_tag_result-value->* TO <ld_value>.
```

```
      IF <ld_value> IS ASSIGNED.
```

```
        CASE ls_tag_result-tag_alias.
```

```
          WHEN lc_tag_alias_quant.
```

```
            afrud_exp-lmnga = <ld_value>.
```

```
          WHEN lc_tag_alias_rework.
```

```
            afrud_exp-rmnga = <ld_value>.
```

```
          WHEN lc_tag_alias_scrap.
```

```
            afrud_exp-xmnga = <ld_value>.
```

```
        ENDCASE.
```

```
      ENDIF.
```

```
      WHEN OTHERS.
```

```
*        Unsupported data format
```

```
      ENDCASE.
```

```

ELSE.
  RAISE EXCEPTION TYPE cx_pco_bs_int
  EXPORTING
    textid      = cx_pco_bs_int=>error_query_exec
    error_cause = text-ert.
ENDIF.
ENDLOOP.
* Send success message
CONCATENATE text-suc ls_appl_handle-appl ls_appl_handle-handle
  INTO lv_msg_txt SEPARATED BY space.
CREATE OBJECT lo_pco_msg
  EXPORTING
    iv_msg_sev = lo_pco_msg->gc_msg_sev_info
    iv_msg_text = lv_msg_txt.

CATCH cx_pco_bs_int INTO lo_pco_exc.
* Error occurred
CREATE OBJECT lo_pco_util.

lo_pco_util->add_exc_obj_to_log(
  EXPORTING
    is_appl_handle = ls_appl_handle
    iv_activity     = lo_pco_util->gc_act_read
    iv_bal_subobj   = lo_pco_util->gc_bal_subobj_tag
    iv_msg_severity = cl_pco_query_message=>gc_msg_sev_error
    iv_req_type     = cl_pco_query=>gc_tag_req_type
    io_exc_obj     = lo_pco_exc
  CHANGING
    cv_log_handle = lv_log_handle ).

lv_msg_txt = lo_pco_exc->get_longtext( ).
MESSAGE lv_msg_txt TYPE cl_pco_utility=>gc_msgty_error.
ENDTRY.

```

### Optimization possibilities

There are the following ways of optimizing the performance of the system:

- Dynamic determination of path information via the work center of the operation to be confirmed:
  - Use of characteristics classification for work centers or equipment, in order to store parts of the path as characteristic values
- Evaluation of PCo user messages:
  - Table **PCO\_MSG\_OBJ** of the exception object **LO\_PCO\_EXC**
  - Evaluation of the exporting parameter **ET\_PCO\_MSG\_OBJ** of method **READ\_TAG**

## Related Content

[How to Create Custom Query Agent Using SAP PCo 2.1 SDK](#)

[Manufacturing Execution For Process Industries](#)

[Production Planning and Control for Discrete Industries](#)

[State of the Art Maintaining Process Instructions with XSteps - Available Functional Enhancements](#)

For more information, visit the [Manufacturing homepage](#).

## Copyright

© Copyright 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.