Plant Connectivity (PCo) – Using a Single Agent to Handle Multiple Notifications

Applies to:

SAP MII 12.x, Plant Connectivity 2.x. For more information, visit the Manufacturing homepage.

Summary

SAP Plant Connectivity (PCo) can be used to actively monitor various plant system events and deliver real-time notifications to SAP applications, such as SAP Manufacturing Integration and Intelligence (MII). A single PCo Agent can be used to handle multiple notifications and pass XML messages to SAP MII transactions. Data from these messages can be extracted for further use within the transaction.

Author: Diana Hoppe

Company: SAP Labs, LLC

Created on: 4 April 2011

Author Bio

Diana Hoppe holds a Bachelors Degree in Computer Science from West Chester University and is currently in the Computer Science Masters Program there. Over the past 6 years she has worked in MII technical support and SAP MII training material development. She is currently a member of the Regional Implementation Group, Applications under the direction of Jeremy Good.

Table of Contents

Prerequisites 3
The Simple Scenario 3
Pre-Steps 3
Configure the Source System 5
Configure the Destination System 5
Configure the Agent Instance 6
Create the SAP MII Transactions 7
Configure the First Notification 7
- Configure the Trigger 7
- Configure the Output 8
- Configure the Destination 8
- View and Copy the Sample Message 9

Configure the Second Notification 9
- Configure the Trigger Expression 9
- Configure the Output 9
- Configure the Destination 10
- View and Copy the Sample Message 10

Expand the First Transaction 10
Expand the Second Transaction 13
Testing the Notification Delivery 14
Start the Agent 15
Review the Output Files 15
Summary 15
References 15
Related Content 16
Copyright 17

Prerequisites

- You have read the Plant Connectivity installation guides available at http://service.sap.com/instguides
- PCo has been successfully installed on your machine
- You have downloaded the MatrikonOPC trial version
- You have access to SAP MII 12.x

The Simple Scenario

On the shop floor a centrifugal pump starts, using its rotating impeller to increase the pressure of a fluid. When the target pressure is reached, a control valve opens allowing the fluid to flow.

We would like PCo notifications to trigger SAP MII transactions. One transaction initiates a BAPI call to tell ECC when the pump has started and when it has stopped. The second transaction initiates a BAPI call to create a Plant Maintenance Notification in ECC if the fluid pressure is high and the fluid flow rate is low, which may indicate a valve problem that requires manual intervention. Each SAP MII transaction will receive an XML message from PCo, containing information that is configured inside of PCo. Each transaction is responsible for fetching any additional data required to build the BAPI Request document.

The purpose of this paper is twofold: to show you how to use a single PCo Agent to handle multiple notifications and to demonstrate how to extract data from the PCo XML message for further processing in an SAP MII transaction.

Note: On the SAP Developer Network (SDN) you can find the SAP MII Manufacturing Integration and Intelligence Templates. These downloadable templates provide content-based examples of how to integrate SAP ERP business processes with shop floor-related processes using SAP MII. You can use these templates as a starting point for developing MII applications that involve any level of integration with SAP ECC, including remote-enabled function modules (RFCs, which include BAPIs). For example, in the scenario stated above, there are transaction templates for creating Production Order Time Tickets and for creating a Plant Maintenance Notification.

The “pieces” used to build this scenario:

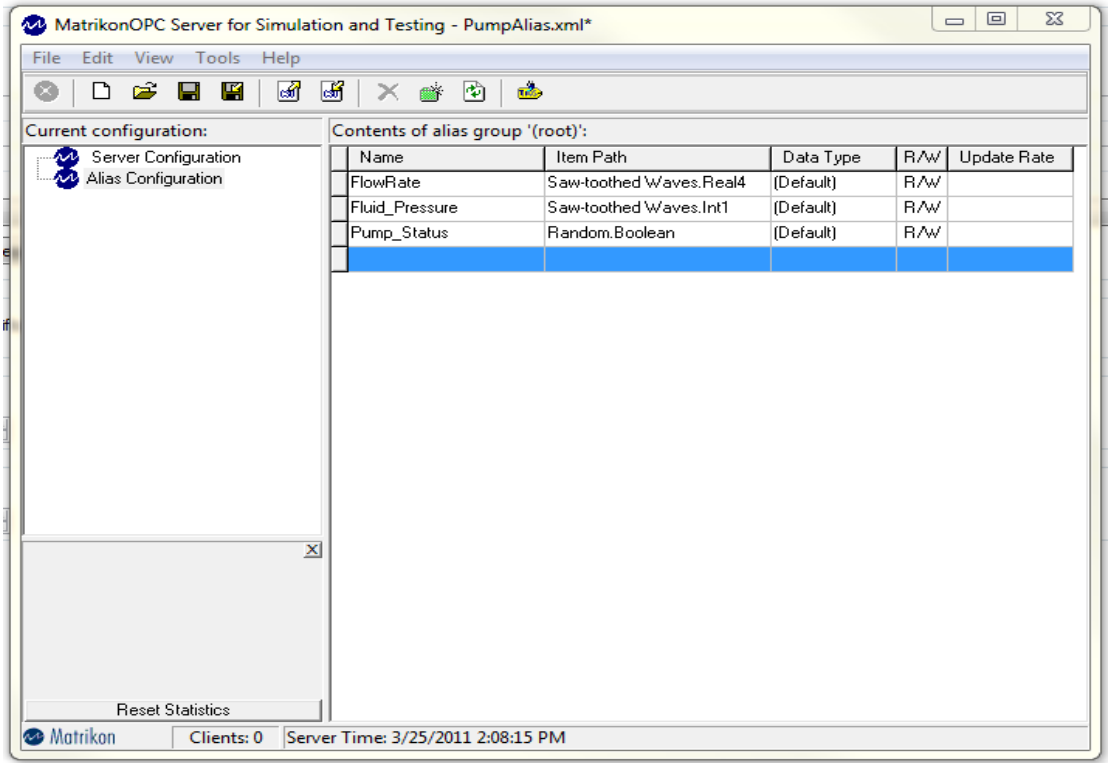
a. The trial version of the MatrikonOPC, currently available at http://www.matrikonopc.com/downloads/178/index.aspx
b. SAP Plant Connectivity 2.1, available on the SAP Service Marketplace http://service.sap.com
c. SAP MII, version 12.1.8

Pre-Steps

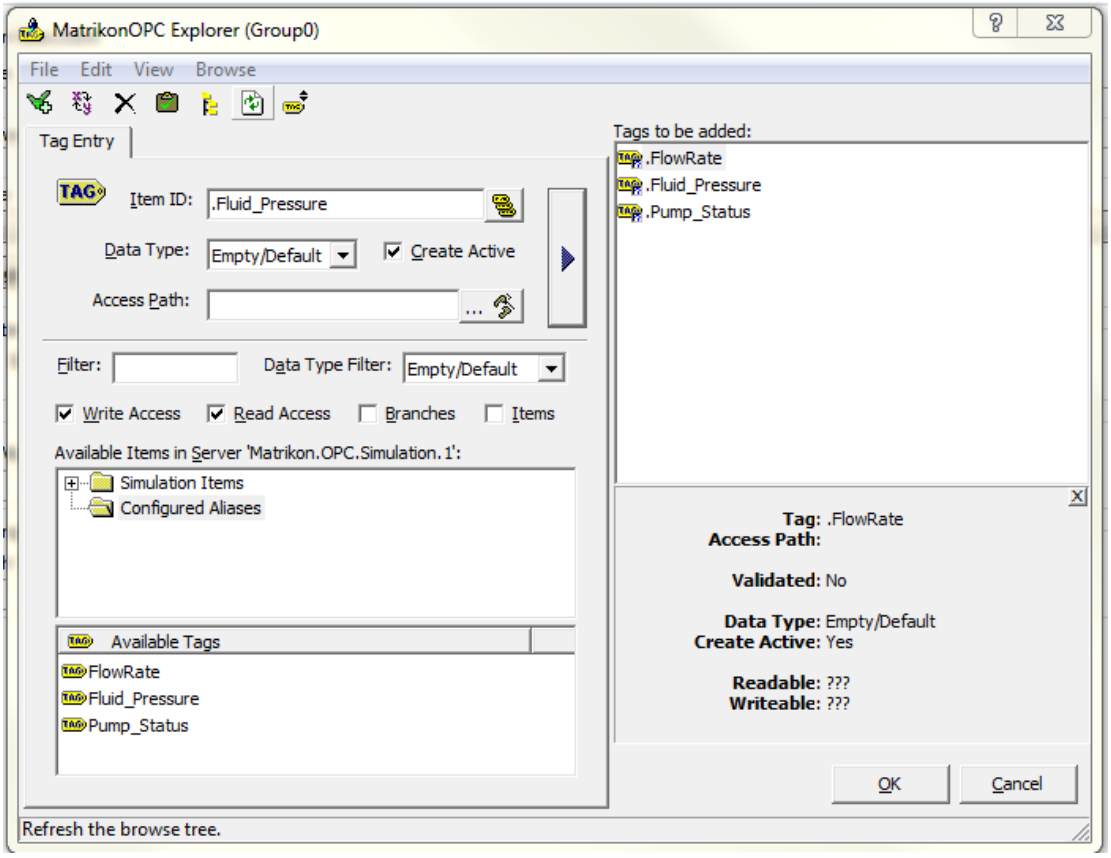
In the *MatrikonOPC Server for Simulation*, an Alias file was configured with the following tag names and mappings:

Fluid_Pressure	Saw-toothed Waves.Real.4
FlowRate	Saw-toothed Waves.Int1
Pump_Status	Random.Boolean

The Alias file configuration in the MatrikonOPC Server:

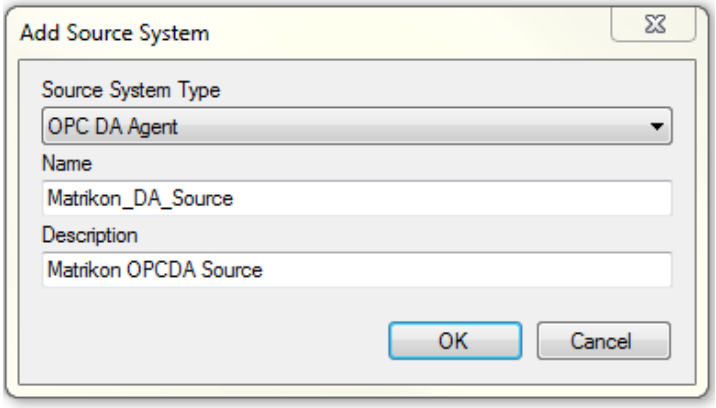


Next, the Alias File was added to the *MatrikonOPC Explorer* – the green checkmark activates the selected tags.



Configure the Source System

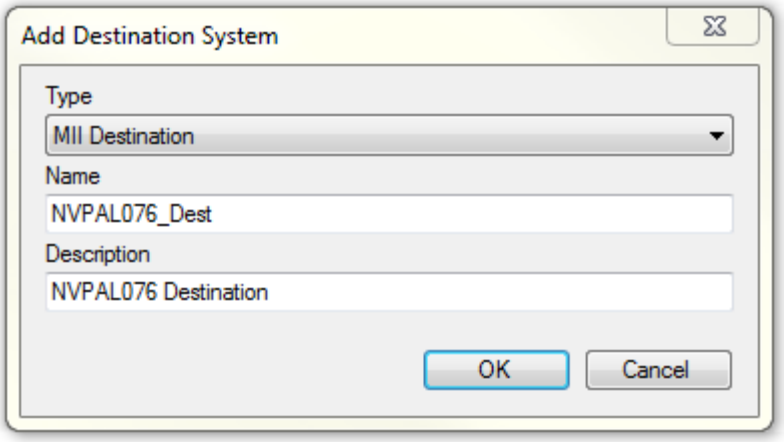
1. Open the *PCo Management Console*.
2. Select *File -> New -> Source System* (or use the add icon located in the *Destination Systems* area).



3. In the *Add Source System* dialog, select *OPC DA Agent* as the *Source System Type*.
4. Give the *Source System* a meaningful *Name* and *Description*, then click the *OK* button to close the dialog.
5. On the *Server* tab, enter the name of the OPC Server or use the *Browse* button to obtain a list of available servers.
6. On the remaining tabs (Settings, Aliases, and Reliability), keep the default settings.
7. Save the newly created *Source System*.

Configure the Destination System

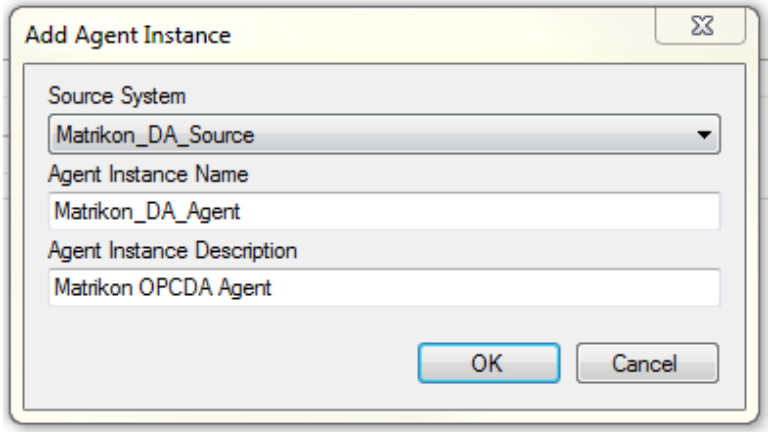
1. Select *File -> New -> Destination System*.



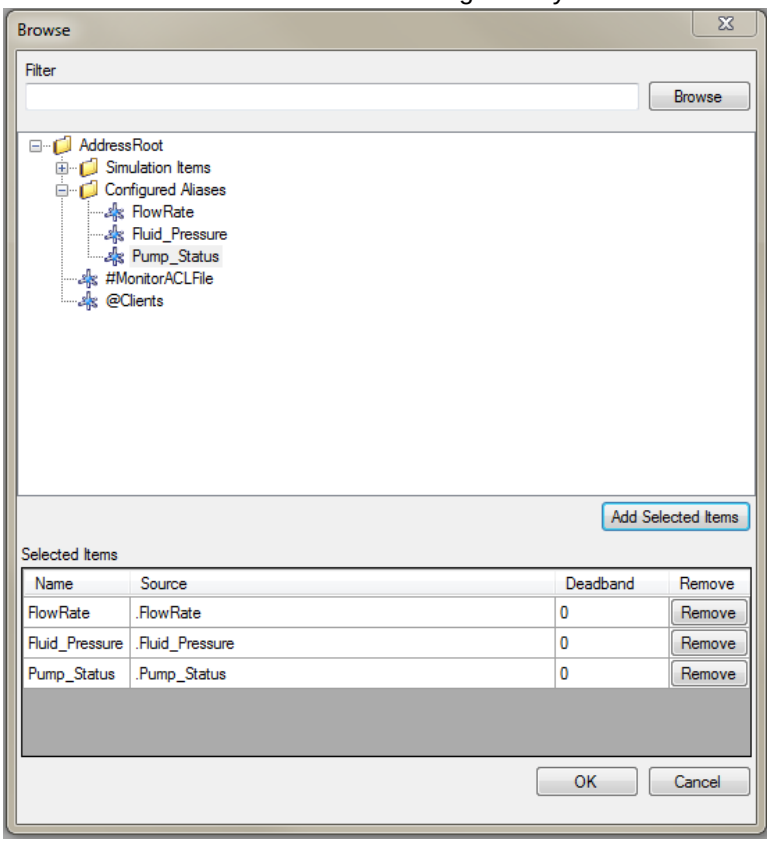
2. In the *Add Destination System* dialog, select *MII Destination*.
3. Give the *Destination System* a meaningful *Name* and *Description*. Click the *OK* button to close the dialog.
4. On the *Server* tab, enter the MII Server details:
    a. *Server Name* and *Port* (the same server and port used to logon to MII)
    b. The MII *Version*
    c. The MII *User Name* and *Password*
5. Keep all other default settings.
6. Test the connection using the *Test Connection* button.
7. If the connection test succeeds, save the newly created Destination System. If the connection test fails, review the MII Server details.
8. Save the newly created *Destination* system.

Configure the Agent Instance

1. *Select File -> New -> Agent Instance*.
2. Choose the *Source System* you created earlier from the dropdown list.



3. Give the *Agent* a meaningful *Name* and *Description*.
4. On the *Host* tab, use the default settings.
5. The tag or tags to monitor are identified on the *Subscription Items* tab.
    a. Click the *Browse* button to search for the desired tag or tags.
    b. In the *Browse* dialog, you may enter a filter, for example, *Pump**, to narrow the tag list
    c. Click on the *Browse* button of the *Browse* dialog and expand the Address Root.
    d. Locate the desired tag or tags, and after each selection, click the *Add Selected Items* button (even though it says *Add Selected Items* you can only add them one at a time).
    e. Click OK to close the Browse dialog when you are finished.



6. Leave the default settings on all remaining tabs. Save the Agent by clicking on the *Save* icon in the *Agent Instances* area.

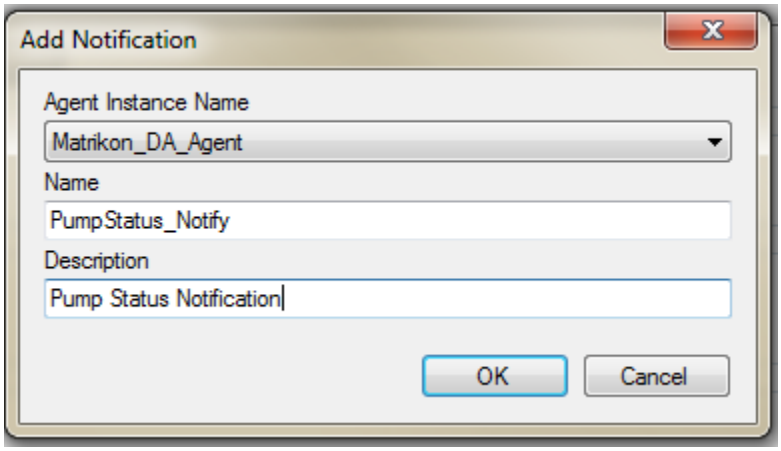
Create the SAP MII Transactions

Before the PCo notifications are created, the transactions that will be triggered by them must be created.

1. Open the Workbench of the MII Destination system you configured earlier.
2. From the *File* menu, select *New -> Transaction*.
3. Add a Transaction Property with the following parameters:
    a. Name: notifyIn
    b. Description: PCo XML notification message
    c. Data Type: XML
    d. Value: leave empty for now
4. Save the transaction as *TimeTicketCreate* in the folder of your choice.
5. Resave this transaction as *PMNotificationCreate*, so that you now have two transactions ready for further configuration.

Configure the First Notification

1. Click on the add notification icon  . You can also go to *File -> New -> Notification*.
2. In the *Add Notification* dialog, select the appropriate *Agent Instance Name* from the dropdown list.
3. Give the Notification a meaningful *Name* and *Description*.
4. Click the OK button when finished.



Configure the Trigger

1. Make sure the *Enabled* box is checked.
2. In this example, we will configure the Trigger Type as “Always”. Select *Always* from the *Trigger Type* dropdown list.

Note: A trigger type of “Always” means that when the value of the subscription item changes, the notification is delivered. Refer to the PCo Documentation for an explanation of the different trigger types.

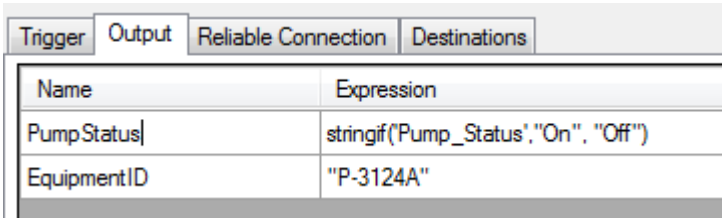
3. No trigger expression is needed when you use a Trigger Type of *Always*. Instead, the tag will be added to the output message – this will tell PCo what tag is to be monitored “Always”. You will do this in the next section.

Configure the Output

On the Output tab, you configure information to be included as part of the notification message payload that will be sent to the SAP MII transaction. The content of the notification message for this example will be the pump status (Off/On) and the equipment identifier.

1. From the Output tab, click the *Add Expression* button.
2. From the *Conditional Expression* dropdown list, select *stringif(condition, onTrue, onFalse)*.
   a. Highlight the word “condition” in the stringif expression.
   b. From the *Subscription Item* dropdown, select Pump_Status to overwrite it.
   c. Highlight the OnTrue and type “On”.
   d. Highlight the OnFalse and type “Off”.
   e. The expression should read: stringif('Pump_Status', "On", "Off").
   f. Use the Validate button to check the expression for errors.
   g. Click the OK button when finished.
3. Change the *Name* (which currently reads *Expression1*) to PumpStatus.
4. Build another expression, the hard-coded equipment identification string, “P-3124A” by clicking on the *Add Expression* button. Change the Name of this expression to read *EquipmentID*.

Note: Use single quotes for the subscription item values, double quotes for string literals (the actual characters between the double quotes).

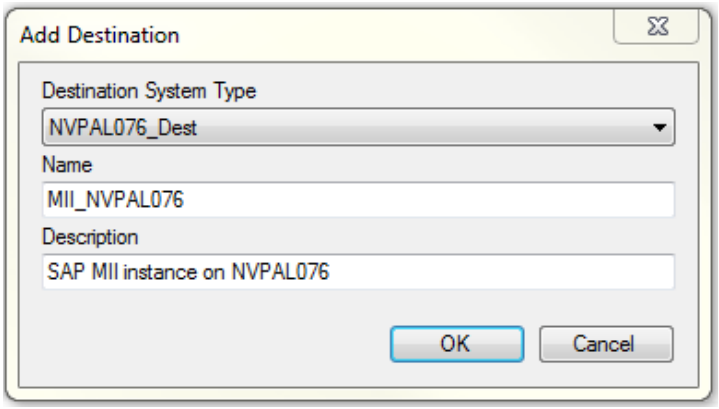


Configure the Destination

On the Destinations tab, the destination of the notification is specified.

Note: In PCo 2.0 and PCo 2.1, SAP MII systems are the only destinations that can be specified.

1. Switch to the *Destinations* tab. Locate and click the *Add Destination System* icon,  .
2. Select the *Destination System* created earlier from *Destination System Type* dropdown list.
3. Give the *Destination System Type* a meaningful *Name* and *Description*. Click *OK* when you are finished.



4. Expand the newly created Destination and then expand the server node.
5. Find the Project folder that contains the *TimeTicketCreate* transaction.
6. Select the transaction. The *Transaction Name* box is populated.
7. From the Input Parameter Name dropdown list, select the Transaction Input property (notifyIn).
8. Save the Notification by clicking on the *Save* icon in the *Agent Instances* area.

View and Copy the Sample Message

In this step a sample message is generated. This XML message is copied and entered as the value of the notifyIn Transaction Property of the *TimeTicketCreate* transaction. Doing so allows you to see its XML structure in MII.

1. Switch back to the Output tab of the notification.
2. Click on the *Test Notification Delivery* button at the bottom right.
3. The *Notification Test Dialog* opens. Click on the *View Sample* button.
4. The sample message opens in a text editor. Copy the message, and paste into the *Value* field of the *notifyIn* transaction property of *TimeTicketCreate*.
5. Save the transaction.

Configure the Second Notification

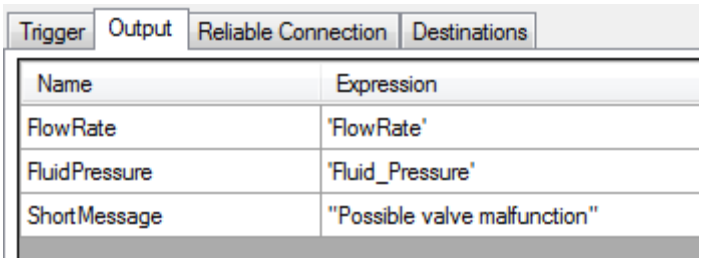
1. Open the Plant Connectivity Management Console and click on your Agent Instance (in this example, Matrikon_DA_Agent).
2. Click on the add notification icon  .
3. *Name* this Notification LowFlow_Notify.
4. Add a *Description*, such as Low Flow Notification.
5. Click *OK* to close the Add Notification Dialog.

Configure the Trigger Expression

1. Trigger Type: OnTrue
   With Trigger Type = On True, a PCo Notification will be generated whenever the expression evaluates to true. However, the expression must evaluate to false before it evaluates to true again for another notification to be sent.
2. Trigger Expression: ‘FlowRate’ < 5.8.

Configure the Output

1. The flow rate
2. The fluid pressure
3. A short message: “Possible valve malfunction”



Configure the Destination

1. Add the Destination (on the Destinations tab) as before.
2. Select the *NotificationCreate* transaction from the file tree.
3. Select *notifyIn* from the *Input Parameter Name* dropdown list.
4. Save the notification by clicking the *Save* icon in the *Agent Instances* area.

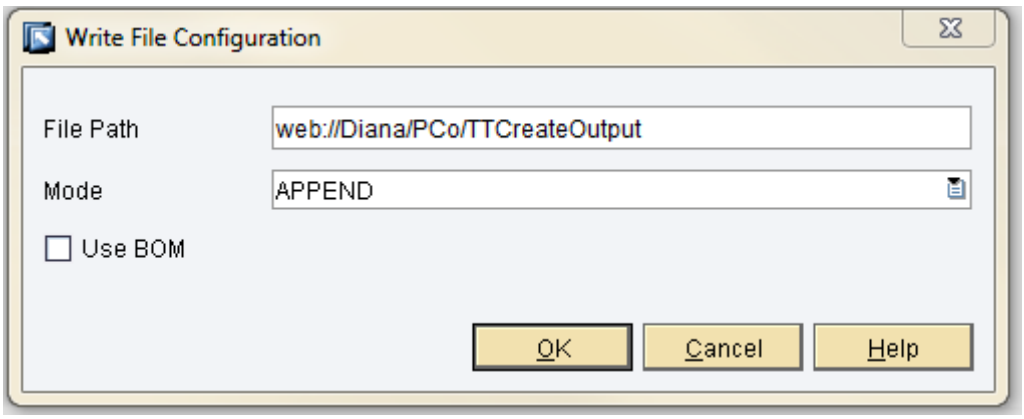
View and Copy the Sample Message

View and copy the sample message as you did for the first notification to the *NoticationCreate* transaction's *notifyIn* transaction property. Be sure to *Save* the transaction when you finish.

Expand the First Transaction

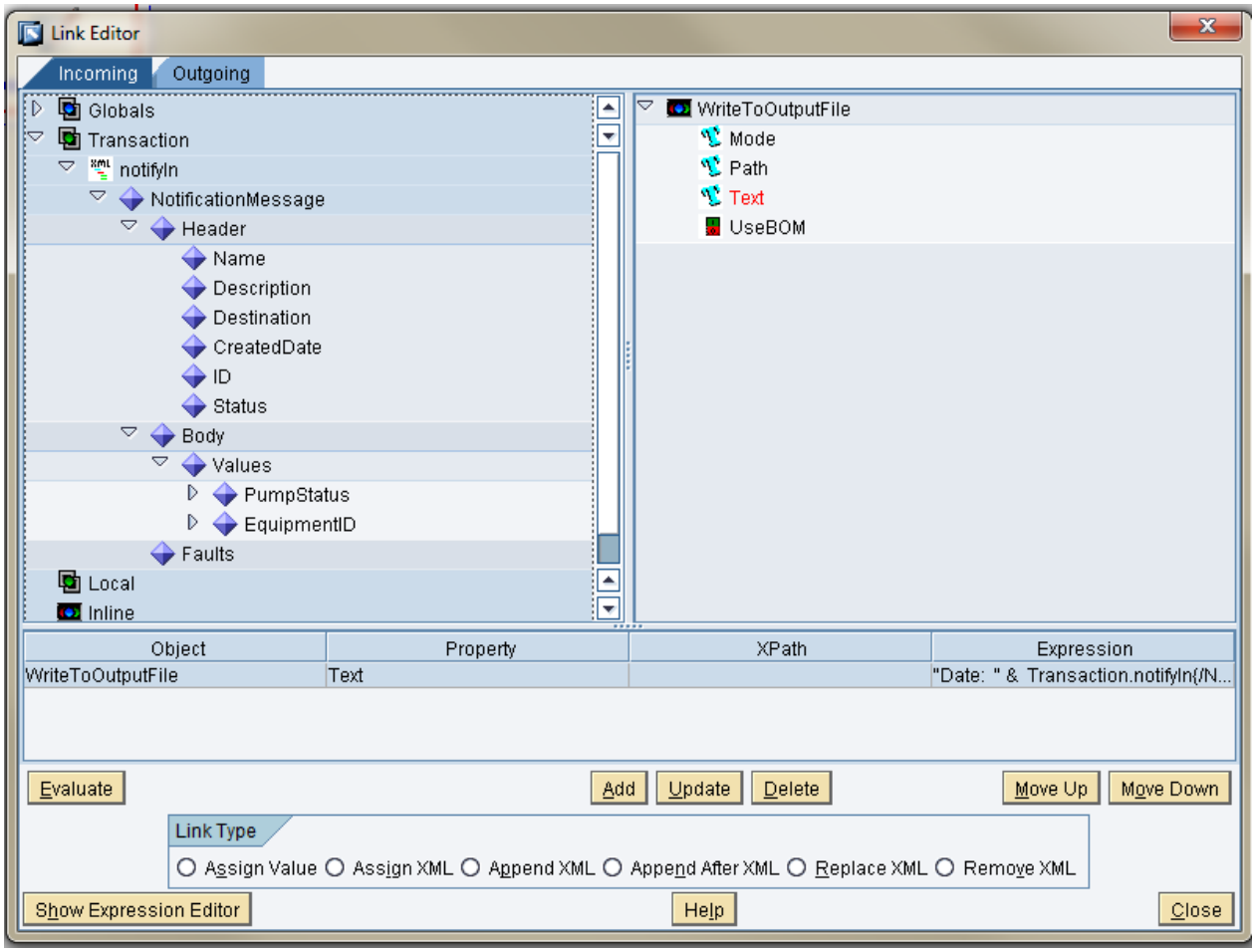
These next steps will illustrate how to retrieve the data from the PCo XML message received by SAP MII. The data will be accessed and written to a file, so that you can actually see PCo at work. Once you understand how to access this data, you can use it in the building of or gathering of additional parameters for a BAPI Request document.

1. To the first sequence of the transaction, add a *Write File* action, located in the *File I/O* action category. Using the Sequence and Actions Property menus, name the sequence *Initial_Seq* and the action *WriteToOutputFile* (It is an SAP MII Best Practice to name you sequences and actions to help make the transaction self-documenting).
2. Open the Configure dialog of the action.
    a. Configure the file path to point to an existing folder and the text file name of your choice on the Web tab (the text file will be created for you). Select *Append* as the *Mode*. Your configuration should look similar to the picture below.



    b. Click *OK* to close the *Write File Configuration* dialog.

3. Open the Link Editor for the *WriteToOutputFile* action.
    a. Expand the Transaction property, followed by the notifyIn property on the source (left) side of the Input tab, as seen in the picture below.
    b. Expand the *WriteToOutputFile* object on the destination (right side).
    c. Select the *Text* property of the WriteToOutputFile.



d. Click the Show Expression Editor button. Using a combination of XPath, string literals, the concatenation operator (&) and the newline operator (crlf), you build the expression that will be written to the output file.

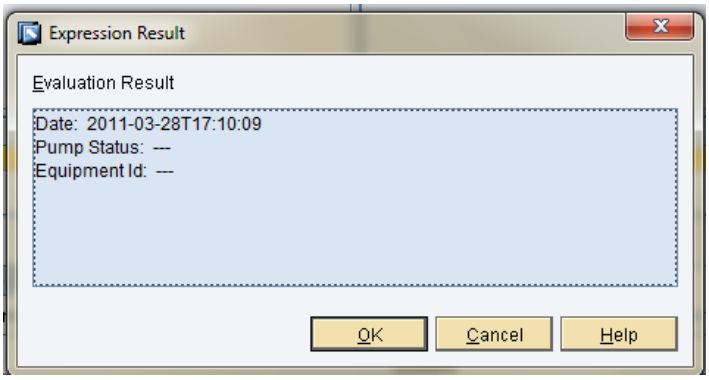
"Date: " & Transaction.notifyIn{/NotificationMessage/Header/CreatedDate} & crlf & "Pump Status: " & Transaction.notifyIn{/NotificationMessage/Body/Values/PumpStatus} & crlf & "Equipment Id: " & Transaction.notifyIn{/NotificationMessage/Body/Values/EquipmentID} & crlf & crlf

Let's dissect the first line of this expression.

"Date: "	String literal, needs quotes, you type in
& (ampersand)	Concatenation operator, type in or use the button to the left of the Expression box
Transaction.notifyIn{/NotificationMessage/Header/CreatedDate}	In the source links, follow the path Transaction -> notifyIn -> NotificationMessage -> Header and click on CreatedDate. Drop and drag this into the correct spot in the Expression box
& (ampersand)	Concatenation operator
crlf (creates a new line)	Type in or select from the Functions dropdown list

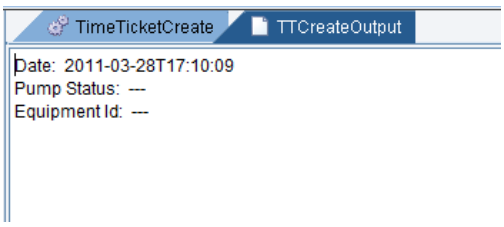
e. Continue building the expression a piece at a time.
f. When finished, test the expression using the *Evaluate* button, your result should look as below.

Note: When the transaction is initiated by a PCo Notification, you will see the actual values.



g. Click the *Add* button to add the expression to the Link Table.
h. Close the Link Editor.

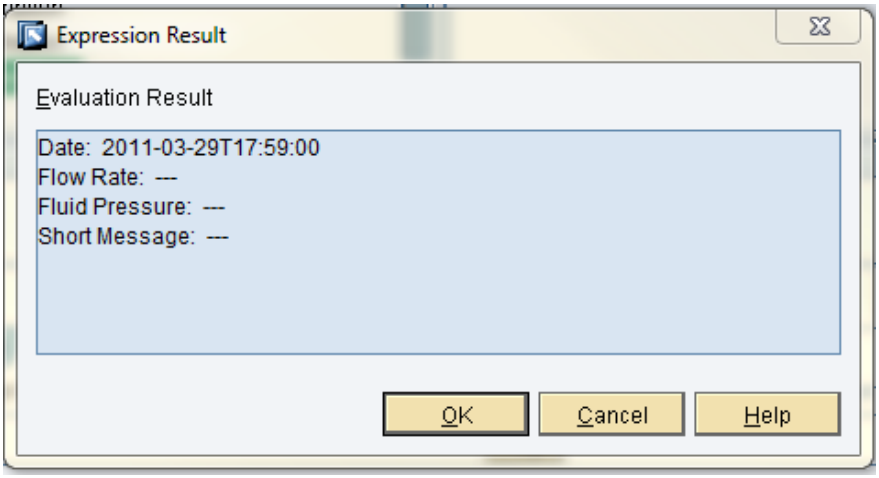
4. Save and test the transaction
   a. After saving, test the transaction. Correct any errors if necessary.
   b. After the transaction runs successfully, switch to the Web tab. Refresh the Web directory of the project in which you saved your text file. The file should appear in the file tree after the refresh.
   c. Open the .txt file by double-clicking on it. The result should look similar to the picture below:



Expand the Second Transaction

As you did in the TimeTicketCreate transaction, you will retrieve the data from the PCo XML message received by SAP MII and write it to a file. Here is a synopsis of the steps:

1. Open the Notification Create transaction.
2. Configure the *WriteToOutputFile* action's *File Path* and *Mode.*
3. Open the Link Editor of the action and Expand the *Transaction -> notifyIn -> NotificationMessage -> Header* and *-> Body* -> Values nodes on the source side (left).
4. Expand the *WriteToOutputFile* on the destination side (right), and highlight the *Text* property.
5. Build an expression that shows the CreatedDate, FlowRate, FluidPressure, and Short Message.
6. Evaluate your expression, it should look something like this:

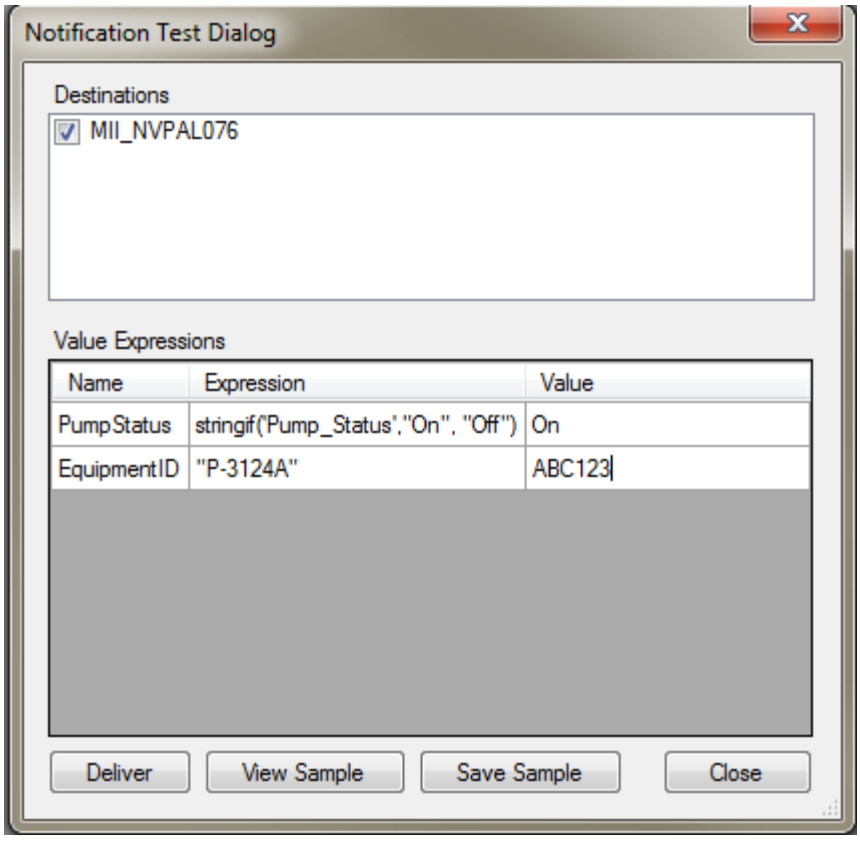


7. Click the *Add* button to add the expression to the Link Table.
8. Close the Link Editor and Save the transaction.
9. Test the transaction.
10. Refresh the Web directory of the project in which you saved your text file. The file should appear in the file tree after the refresh.

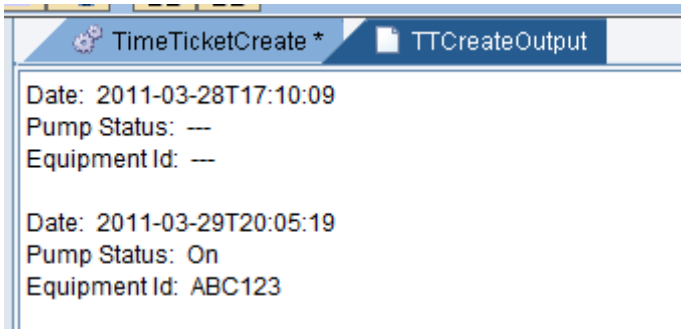
Testing the Notification Delivery

We can test to make sure the notification is delivered to the MII transaction. From the Plant Connectivity Management Console, click on the Agent you created and open the PumpStatus_Notify notification. Follow these steps:

1. Switch to the Output tab.
2. Click on the *Test Notification Delivery* button.
3. In the *Value Expressions* area, assign values to each expression.



4. Click the *Deliver* button. You should receive a "Message sent to destinations successfully" dialog.
5. From the MII Workbench, refresh the Web directory of the project in which you saved your text file. You should see an additional entry that contains the values you entered.



6. Repeat the steps above for the PumpStatus_Notify notification.

Start the Agent

Before starting your Agent, make sure the data source is up and running. In this example, via the *Matrikon Explorer*, the aliased tags were activated.

1. In the Agent Instance area, locate and highlight your Agent. Click on the green arrow button  . The green arrow button turns into a red square button  .
2. If the Agent starts successfully, the red square next to it becomes a green arrow.
3. Let the Agent run for several minutes, and then turn it off by clicking on the red square button.

Review the Output Files

Go back into the Workbench, switch to the Web tab, locate and open the NotificationCreateOutput and TTCreateOutput files. If these files were already open, close them and refresh the Web folder in which they reside. You should see the PCo message data appended in each file.

Summary

Hopefully after reading this paper and following the steps to implement this scenario, you now understand how a single PCo Agent can monitor multiple notifications. Although for the sake of simplicity only two notifications were configured, obviously you can configure more. You should also understand how to configure the "payload" of the XML message that is sent to an MII transaction, and how to retrieve information from the message once inside of the transaction. You should be able to apply these same principles to other supported data sources. For a list of the available PCo Agent types, review the Plant Connectivity Help documentation.

References

Specials thanks to my SAP colleagues Christopher Carney, Eoin Donnelly, Jeremy Good, Igor Becker, Kevin Yurasits, and Michael Appleby, and to Ajay Malempati of Seal Consulting for allowing me to obtain information and ideas for this paper.

Related Content

http://help.sap.com

http://service.sap.com/rkt

http://service.sap.com/instguides

For more information, visit the Manufacturing homepage

Copyright