SAP Plant Connectivity – Example for Usage of Key-Value Pair Buffer

Purpose

The purpose of this page is to illustrate an example how the standard EMP implementation Key-Value Pair Buffer can be used to calculate the difference between measurement readings and then send this delta value to a database. In a tag-based scenario, SAP can only forward the last available measurement reading to PCo, so a configuration using the key-value pair buffer is needed to allow to access the second to last measurement reading and use it in the calculation. In this specific case, the data source is an electricity meter that measures power consumption at regular intervals and sends the readings to PCo. Before PCo writes the current measurement reading to the key-value pair buffer, the second to last measurement reading is read out and used in the calculation. The resulting delta value is then sent to SAP MII or to a database.

Overview

- Prerequisites
  - Create an Agent Instance without Source System
  - Create the second source system, which emulates the real sensor
  - Create an OPC UA Destination System for filling the buffer
  - Create a destination system to access the latest value from the persistent memory
  - Create a simulation destination for sending the calculated delta value
  - Create a "Multiple Call Destination System"
  - Fill the initial value to buffer
  - Combine the above created Multiple Call Destination System with the real data source

Prerequisites

- The screenshots are taken from PCo-Version 15.1.4;
- An electricity meter (data source) is emulated by using the KEPServerEX 6.1;
- A Multiple Call Destination System is used;
- The important building block is the Standard Enhanced Method Processing Implementation Key-Value Pair Buffer;

Create an Agent Instance without Source System

Create an Agent Instance without Source System – this agent instance represents the buffer that can store the last reading.

Create a New Agent Instance Without Source System

[Image: Create a New Agent Instance Without Source System]

Make the Agent Instance an OPC UA Server
Select the Assembly KvpBuffer for the Agent Instance

Apply Assembly (Key Value Pair .dll) and the available (coded) methods. The selected assembly and the methods are offered by the implementation. This agent instance now acts as a server within PCo.
An OPC UA Source System that points to this Agent Instance still needs to be created:

The Link Between the KVP Buffer Agent Instance and the Source System is the Endpoint
The KVP buffer agent instance needs to be running so to perform the connection test.

Create the second source system, which emulates the real sensor.

Creating a Source System for the Data Source.

Subscribe to a tag. The tag represents the measurement value.
Create an OPC UA Destination System for filling the buffer

Creating an OPC UA Destination System

This OPC UA Destination System points to the KVP-Buffer OPC UA Source System
The Method `SetKeyValue` is assigned to this Destination System. For Browsing the Agent (see step “Connection between OPC UA Source System and KVP-Buffer Agent Instance is established”) needs to run.

Select the method `SetKeyValue`. In order to browse for the available methods, the Agent needs to be running.

Create a destination system to access the latest value from the persistent memory

Create a destination system to access the latest value from the persistent memory. This destination system is needed to call up the method for reading a buffer value.

This OPC UA Destination System Points to the KVP-Buffer OPC UA Source System (similarly as illustrated earlier)
Assign the Method _GetValueByKey to this Destination System. This step is different to the step “The Method _SetKeyValue is assigned to this Destination System”, because here the method _GetValueByKey is selected.

Create a simulation destination for sending the calculated delta value

Create a simulation destination for sending the calculated delta value. In real life, this destination system would be a database.
Create a “Multiple Call Destination System”

Create a “Multiple Call Destination System”. This destination system represents the combination of the sequence of the following steps which are carried out by PCo:

- PCoreads the current measurement from the machine (here from the simulator)
- PColookst into the KVP buffer for the latest value
- PCocalculates the difference and sends the delta value to a local file

Add Sequence of Destination System Calls. The following sequence steps are added:

- In the first step (10) you enter the OPC UA destination system for getting the value from the buffer.
- In the second step (20) you enter the OPC UA destination system for writing the latest value to the buffer. This means the buffer always only has one value stored.
- The third step (30) is for sending the data to the simulation destination system which is a placeholder for writing to a real database.

Assigning the Input and Output variables and the internal calculation is the key step of all the logic – here, the Input for “GetValue” is a constant string with the value “LATESTVALUE”
The result from "GetValue" is temporarily stored in two temporary variables (this means they are only used within this MultiCallDestination System)

Input for the Key is the same string (value "LATESTVALUE")
Add Input Variable. Enter the input variable to fill the Key-Value-Pair-Buffer. This is actually the tag/data-point from the data source (sensor) – it will be assigned in the final step.

Set the `inPersistent` variable (destination system variable) to true. This ensures, that the last value is still available after the agent instance has been shut down or stopped.
Enter the expression to calculate the difference between the two values

Fill the initial value to buffer

Before the agent instance started, the initial value needs to be filled into the buffer, otherwise the operation on the preceding step ("Enter the expression to calculate the difference between the two values") will result in an error.

Create another destination system of Type **OPC UA Destination** where the _SetKeyValue is used. This destination system can be run and tested standalone by using the _Test Method button.
Setting the Initial Value (zero) to the Buffer with key „LATESTVALUE“

Optional, if you want to check the content of the buffer, you could create another OPC UA destination system for the Method _GetAllKeyValuePair:

Optional Destination for _GetAllKeyValues in order to check the current content of the Buffer
Combine the above created Multiple Call Destination System with the real data source

Combine the above created Multiple Call Destination System with the real data source. This means, whenever the data source sends a changed value, the steps of the Multiple Call Destination system are executed.

Create a Notification for the Data Source. Simply create a dummy OPC UA data source and a Notification that you add using Drag&Drop.

The Destination expects one input value (in real life, e.g. the current reading)
The running agent instance feeds the destination system with the current value, but only the delta value is sent to the simulation destination system (in this example files are written).

**Related Documents**

Multiple Call Destination System

Standard Enhanced Method Processing Implementation