SIEMENS

SCE Training Curriculum

Siemens Automation Cooperates with Education (SCE) | 09/2015

PA Module P03-02

SIMATIC PCS 7 –
Vertical Integration with OPC

Cooperates with Education
Automation
SIEMENS

Matching SCE Trainer Packages for these curriculum

- **SIMATIC PCS 7 Software block of 3 packages**
  Order No. 6ES7650-0XX18-0YS5
- **SIMATIC PCS 7 Software block of 6 packages**
  Order No. 6ES7650-0XX18-2YS5
- **SIMATIC PCS 7 Software Upgrade block of 3 packages**
  Order No. 6ES7650-0XX18-0YE5 (V8.0 → V8.1) or 6ES7650-0XX08-0YE5 (V7.1 → V8.0)
- **SIMATIC PCS 7 Hardware Set including RTX Box**
  Order No. 6ES7654-0UE13-0XS0

Please note that these trainer packages may be replaced with subsequent packages.
An overview of the available SCE packages is provided at: siemens.com/sce/tp

Continuing education

For regional Siemens SCE continuing education, contact your regional SCE contact partner.
siemens.com/sce/contact

Additional information relating to SIMATIC PCS 7 and SIMIT

In particular, Getting Started, videos, tutorials, manuals and programming guide.
siemens.com/sce/pcs7

Additional information relating to SCE

siemens.com/sce

Note on Usage

The training curriculum for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)“ specifically for training purposes at public educational and R&D facilities. Siemens AG is not liable for the contents.

This document may only be used for initial training on Siemens products/systems. This means it may be copied entirely or partially and handed to trainees for use within the scope of their training. Passing on or copying this document and communicating its contents is permitted within public training and continuing education facilities for training purposes.

Exceptions require written permission by Siemens AG. Contact person: Roland Scheuerer roland.scheuerer@siemens.com.

Violators are subject to damages. All rights including translation rights are reserved, particularly in the event a patent is granted or a utility model or design is registered.

Usage for industrial customer courses is explicitly not permitted. We do not agree to the commercial utilization of these documents.

We would like to thank the Technical University Dresden, particularly Prof. Dr. Leon Urbas and Annett Krause, MS, as well as the Michael Dziallas Engineering Corporation and those who provided support in preparing this SCE training document.

VERTICAL INTEGRATION WITH OPC

TRAINING OBJECTIVE

In this chapter, the students learn to integrate automation systems from diverse manufacturers into higher level programs of the plant control level. The required fundamentals for setting up OPCs and the method of functioning as well as the integration possibilities by means of PCS 7 are explained.

THEORY IN BRIEF

OLE for Process Control (OPC) provides a standard mechanism for communication with a variety of data sources. It makes no difference in this case whether these sources are machines in your factory, or a database in your control center. OPC is based on the OLE/COM technology by Microsoft.

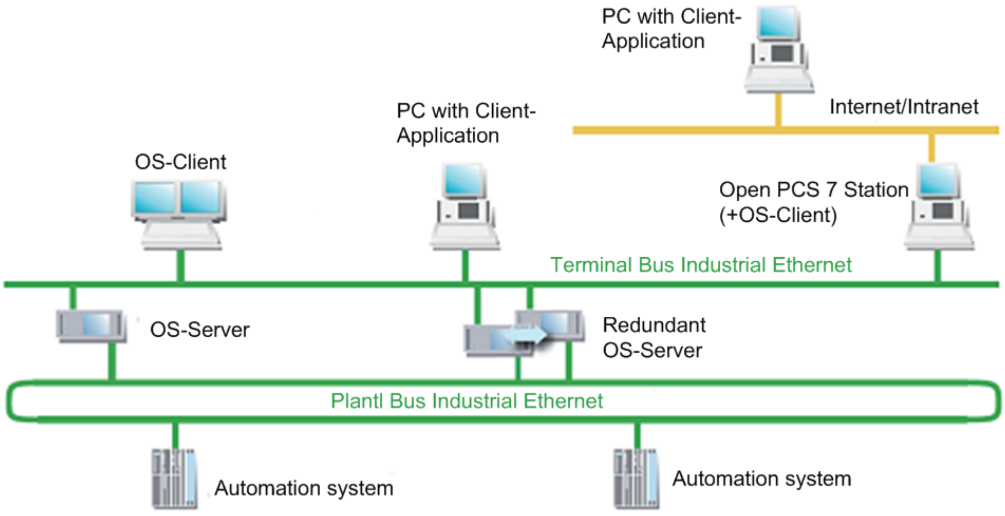

PC with Client-Application
Internet/Intranet
PC with Client-Application
OS-Client
Open PCS 7 Station (+OS-Client)
Terminal Bus Industrial Ethernet
OS-Server
Redundant OS-Server
Plantl Bus Industrial Ethernet
Automation system
Automation system

Figure 1: Integration of the IT world into automation systems [3]

By means of an OPC server, data can be exchanged with external systems without knowledge of the topology and a PCS 7 OS installation being required.

From the view of the user programs generated in higher level programming languages such as C++ and Visual Basic, OPC is a bridge to the process data and the device data of the automation systems. For the device manufacturers, the development of an OPC server is necessary instead of special drivers. For software developers it is of advantage: applications can be written independent of the device. Users in turn have more freedom regarding the selection of devices and software products [1].

THEORY

INTRODUCTION

OPC provides a standardized, open and manufacturer-independent software interface for integrated data communication between automation systems and MS Windows applications with OPC capability. OPC has developed into a de-facto standard for interfacing automation systems by different manufacturers with higher level programs of the plant control level for

– Process visualization (monitoring of individual product lines with cross data traffic)

– Integrated management (ordering, quality control, maintenance, material management, production planning)

The software interface OPC is based on the Windows technologies COM (Component Object Model) and DCOM (Distributed Component Object Model). OPC XML on the other hand is based on the internet standards XML, SOAP and http. COM is the Microsoft protocol standard for communication between objects that are located on one computer but in different programs. With DCOM, COM was expanded with the capability to access objects beyond computer boundaries. This basis makes possible standardized data exchange between applications from industry, office and manufacturing. Communication via DCOM is limited to local networks. Data exchange by means of XML uses the protocol SOAP (Simple Object Access Protocol). SOAP is an XML based protocol that is independent of the platform. With SOAP, applications on the Internet or in heterogeneous computer networks can communicate with each other by means of HTTP (HyperText Transfer Protocol).

The OPC Foundation defined the standard for the OPC interface. Leading corporations in industry automation comprise the OPC Foundation. The OPC servers of the OS system support the following specifications:

– OPC Data Access 1.0, 2.05a and 3.0

– OPC XML Data Access 1.01

– OPC Historical Data Access 1.20

– OPC Alarm & Events 1.10

CLIENT-SERVER PRINCIPLE

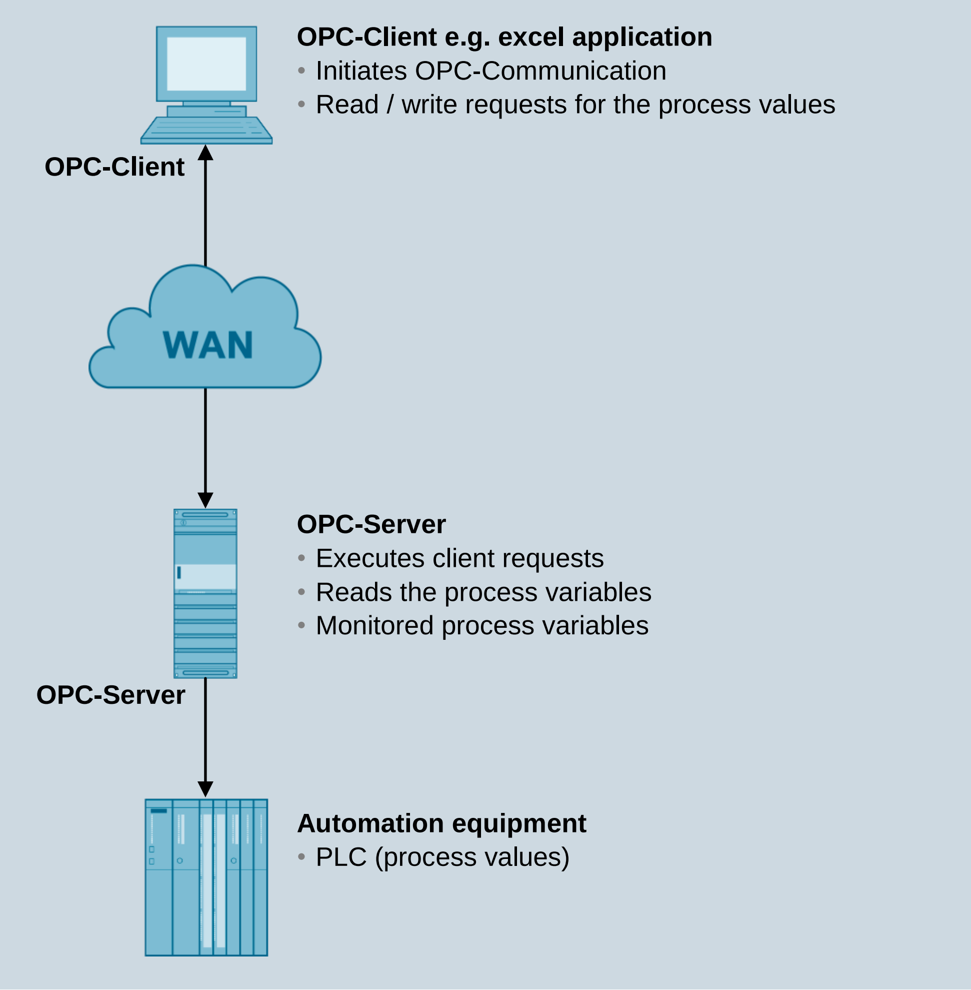
OPC communication is based on the client server principle (refer to

Figure 2). The client takes the initiative and makes a request to the server (service provider). The server replies, executes or supplies. The connection structure needed for this always emanates from the OPC client. The advantage of such a communication scheme is that only the clients have to "know" the server.

The OPC server can access the process data of the automation system.

OPC Server

The OPC software component that provides data when an OPC client requests it is called ***OPC server***. The server has to be installed on the PC since OPC specific entries are required in the Windows registry. “Upwards“, the OPC server supports the interface specification Data Access; “downwards” it is connected through a lower level communication network with the connected automation system as the actual data source.



OPC-Client e.g. excel application

- Initiates OPC-Communication
- Read / write requests for the process values

OPC-Server

- Executes client requests
- Reads the process variables
- Monitored process variables

Automation equipment

- PLC (process values)

Figure 2: Client-server model [1]

OPC Client

OPC components that use an OPC server as data source are called ***OPC clients***. An OPC client is, as a rule, part of the user program that has to be configured. To this end, two OPC interfaces are available:

– The ***custom interface*** (customer specific interface) for programming languages that address interfaces with the function pointer principle, such as C/C++

– The ***automation interface*** for programming languages that address interfaces with object names, such as Visual Basic

OPC SPECIFICATION

OPC Data Access (OPC DA)

Data Access is an OPC specification used to access process data by means of variables. OPC servers for Data Access manage the process variables and the different access capabilities to these variables. This allows the OPC server:

– To read the value of one or several process variables

– To change the value of one or several process variables by writing a new value

– To monitor the value of one or several process variables

– To signal value changes

Process variables are place holders for values that currently have to be determined. The OPC specification defines the interface between client and server programs for managing the process data. Data Access servers make the transparent access to the most varied data sources (for example, temperature sensor) and data sinks (for example, control systems) possible for one or several Data Access clients. These data sources and data sinks can be located on IO cards inserted directly into the PC, but they can also be on any device such as controllers, input/output modules, etc., that are connected by means of serial connections or field busses. A Data Access client can also access several Data Access servers at the same time.

Data Access clients can be very simple Excel sheets or extensive programs (for example, Visual Basic). Data Access clients in turn can be part of larger programs.

OPC Data Access Class Model

When the client accesses data, the hierarchical class model of Data Access assists in adapting the time effort and the result of the content to the current requirements of the application. Data Access differentiates three classes:

- OPC server
- OPC group
- OPC item

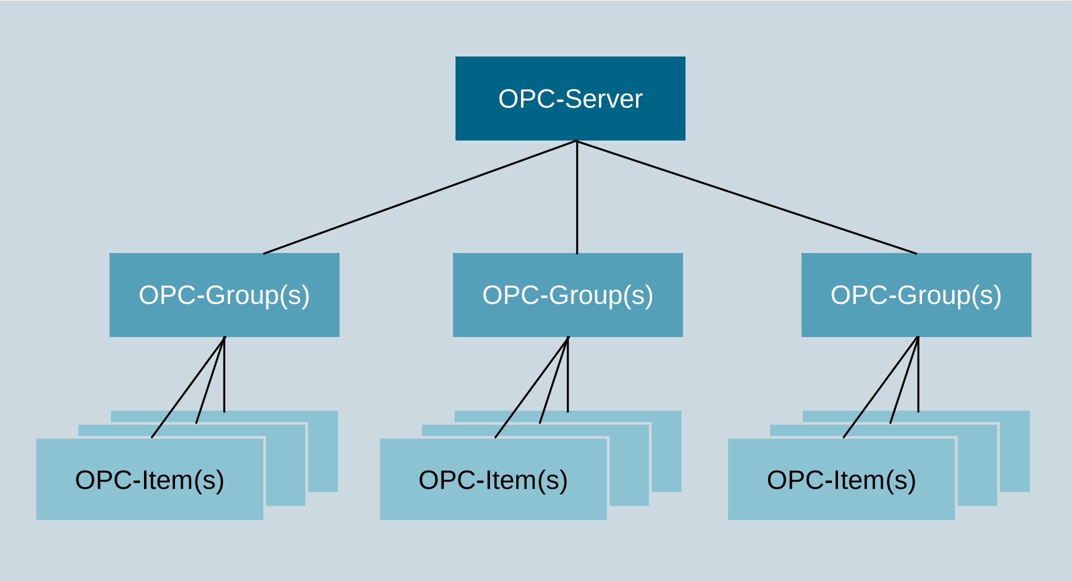


Figure 3: Class model of the Data Access Interface [2]

The client application uses COM calls of the operating system only to generate an object of the class OPC server. The other objects are generated by means of corresponding OPC methods of the class OPC server or lower level classes.

The highest class is the ***OPC server***. Every OPC server belongs to this class. This class represents the access for all other services of the Data Access server. With the aid of class specific attributes, information about the status, the version and (optional) the name space of the available process variables can be obtained. Objects of the class OPC server manage the instances of the lower level class OPC Group.

The class ***OPC group*** is directly subordinated to the class OPC server and structures the process variables utilized by the OPC server. The OPC client is able to use several objects of this class simultaneously. With the aid of the objects of OPC group, the client can generate useful units of process variables and perform operations with them. For example, all process variables of a screen page of the operator interface can be combined into one group. The class OPC group defines methods that can be used to read and write the values of the process variables.

The class ***OPC item*** represents the actual process variable and allows for polling individual data. Each variable is an element (item) in the OPC server's name space and is identified with an item ID. The item ID is specified by the manufacturer of the server and has to be unique within the server's name space. Each value has the following properties:

- ***Value***
  The last recorded value of the variable
- ***Quality***
  How meaningful is the value. If the quality is good, it was possible to determine the value with surety
- ***Timestamp***
  Point in time when the current value of the variable was determined. With each value change indicated to the client, the time stamp is updated. If the value of the variable does not change, the timestamp remains the same also.

Variables have to be specified when the OPC interface is called to obtain process values. By indicating variables, the client is able to request the needed values from the server. The client has to sign up every requested variable at the server to specify the variables that are to be read. Variables can be read and written synchronously as well as asynchronously.

The client can transfer the monitoring of variables to the server. If the value of a variable changes, the server sends a corresponding message to the client. The variables provided by the OPC server can be subdivided into:

- ***Process variables***
  represent measured and control variables of input/output devices
- ***Control variables***
  Using these variables triggers certain supplementary services, such as the transmission of pass words
- ***Information variables***
  These variables are made available by the communication system and by the OPC server and provide information about the state of connections, devices, etc..

Below a few examples of the variables of an OPC Data Access server:

- Control variables of a PLC
- Data of a measurement data acquisition system
- Status variables of the communication system

OPC eXtensible Markup Language DA (OPC XML DA)

OPC XML is a standard that makes possible communication with a platform-independent protocol over the Internet. The client is no longer tied to a Windows environment (DCOM). With the HTTP protocol and the SOAP interface, other operating system -for example LINUX- can monitor and exchange OPC data via the Internet.

SOAP provides a simple and transparent mechanism for exchanging structured and typified information between computers in a decentralized, distributed environment. SOAP is a basis for XML based information exchange.

Data access by means of OPC XML has a functional scope based on OPC Data Access. Change controlled feedback regarding data changes as in the case of DCOM interfaces is not provided for OPC XML because of the lose internet connection.

Accessing methods directly from the Internet represents a considerable security risk. For that reason, SOAP uses exclusively the Internet http channel (HTTP = Hyper Text Transfer Protocol) that can be managed easily with a firewall.

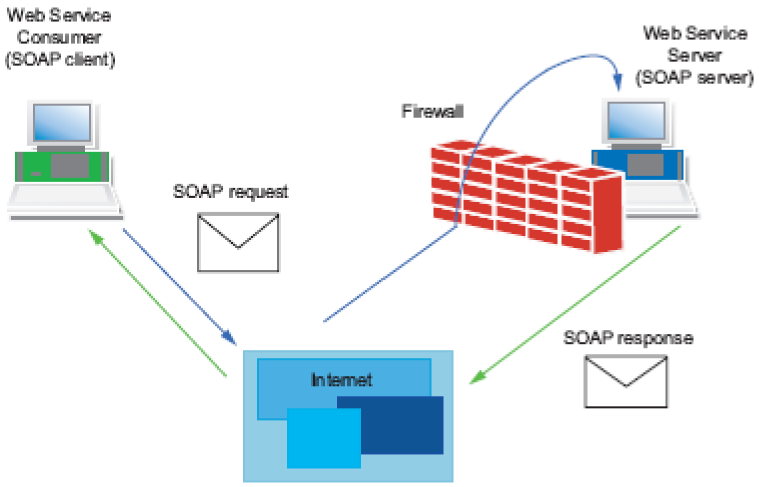

Web Service Consumer (SOAP client)
Web Service Server (SOAP server)
Firewall
SOAP request
SOAP response
Internet

Figure 4: Data transfer with HTTP [2]

OPC Alarms & Events (OPC A&E)

OPC Alarm & Events is an additional specification for transmitting process alarms and events. Events are special states in the process that have to be indicated to the recipient. The OPC specification defines three types of events for this:

– Condition related events

  They indicate the state transitions defined in the OPC state model and are tied to defined conditions

– Tracking events

  They indicate process changes; for example, when a user changes the setpoint of a controller

– Simple events

  They indicate all other stateless events; for example, the failure of a system component

Alarms & Events servers are used for the following, for example:

– Detecting events – for example, reactor level reached

– Determining the state of an event - product tank full

– Confirming an event – reactor level reached detected,

– Monitoring the confirmation – the confirmation is monitored by the reactor alarm indicator; the alarm was detected; the warning signal can be switched off

New events can be indicated without confirmation also. By means of filter criteria, the OPC client sets the events that are indicated to the OPC client. All events corresponding to the set filter criteria have to be forwarded by the originator of the event up to the user. This differentiates Alarms & Events from Data Access. For variable monitoring, only the value changes in the specified time frame are indicated.

The OPC specification defines the syntax of the interface for message receipt. The manufacturer of the OPC server specifies the event types that a server provides.

OPC SERVER OF SIMATIC NET

The open OPC interface is the central interface of the products on the PG/PC of SIMATIC NET. The OPC server of SIMATIC NET supports all communication protocols and services that the communication modules make available.

The OPC server of SIMATIC NET supports for all protocols the interface specification OPC Data Access. For protocols that have mechanisms for transmitting events (S7 communication) OPC Alarms & Events is supported also.

The OPC server of SIMATIC NET makes possible the access to the industrial communication networks PROFIBUS and Industrial Ethernet of SIMATIC NET. It provides OPC clients with the values of process variables or indicates events from the peer device. To this end -with the aid of the protocol software and the communication processor of SIMATIC NET- it accesses the peer devices via the communication network (refer to).

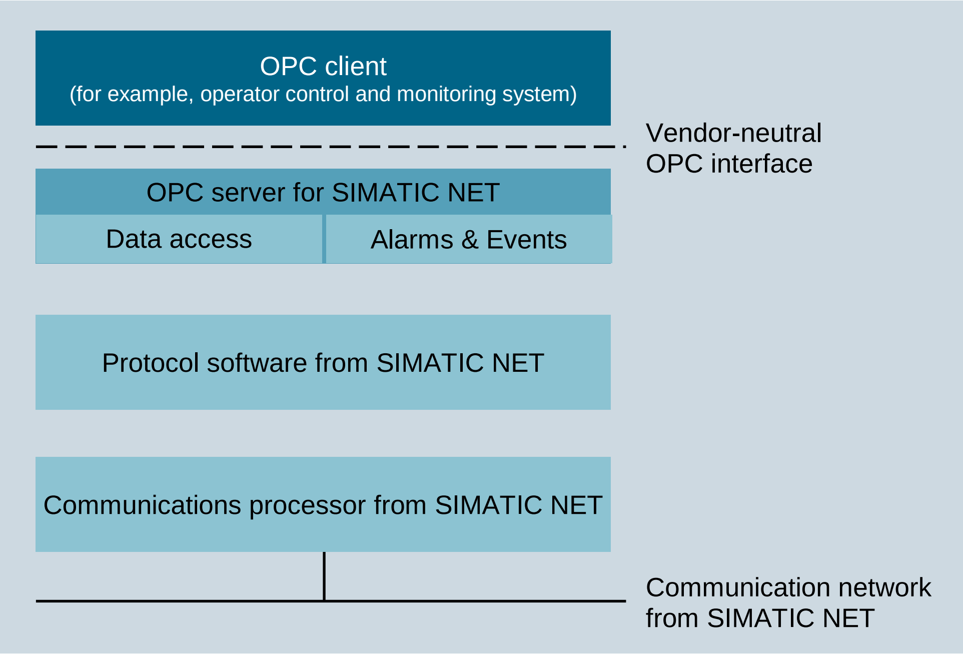

OPC client
(for example, operator control and monitoring system)
Vendor-neutral
OPC interface
OPC server for SIMATIC NET
Data access
Alarms & Events
Protocol software from SIMATIC NET
Communications processor from SIMATIC NET
Communication network
from SIMATIC NET

Figure 5: OPC Server for SIMATIC NET with OPC Client [2]

OPEN PCS 7

OpenPCS 7 was developed for use on multi-user systems. By means of a separate PC station with OpenPCS 7 (OpenPCS 7 station), data can be exchanged with external systems without knowledge of the topology and without a PCS 7 OS installation being necessary. This server collects from existing OS server systems the data via OPC and makes it available in a concentrated form. For that reason, OpenPCS 7 can be used exclusively on multi-user systems.

OpenPCS 7 can be used for data exchange with the following levels:

- Automation level
- Plant control and production level
- MES level (Manufacturing Execution Systems)
- ERP level (Enterprise Resource Planning)

LITERATURE

[1] Wellenreuther, G.: Zastrow, D. (2009) Automatisieren mit SPS (4. Auflage). Vieweg + Teubner. (Automating with PLC)

[2] SIEMENS (2010): SIMATIC NET: Industrial Communication with PG/PC Volume 1 - Basics System Manual. C79000-G8976-C172-09. (http://support.automation.siemens.com/WW/view/en/42783968)

[3] SIEMENS (2015): Process Control System PCS7: Engineering System (V8.1). A5E32712311-AC. (http://support.automation.siemens.com/WW/view/en/90663380)

Step by Step Instructions

Task

The example below shows how to access variables in the OS from Microsoft Excel through OPC. In these step by step instructions, select the level of Reactor R001 to access them.

The settings necessary for this are made in an Excel file. The required macros already exist in the Excel file and are started automatically with the correct settings.

Detailed knowledge regarding macro-programming is not assumed in this training module nor is it provided. Refer to the manuals and technical literature on Microsoft Excel.

To diagnose the availability of the OPC server of WinCC, the software OPC Scout V10 of SIMATIC NET is used.

Training Objective

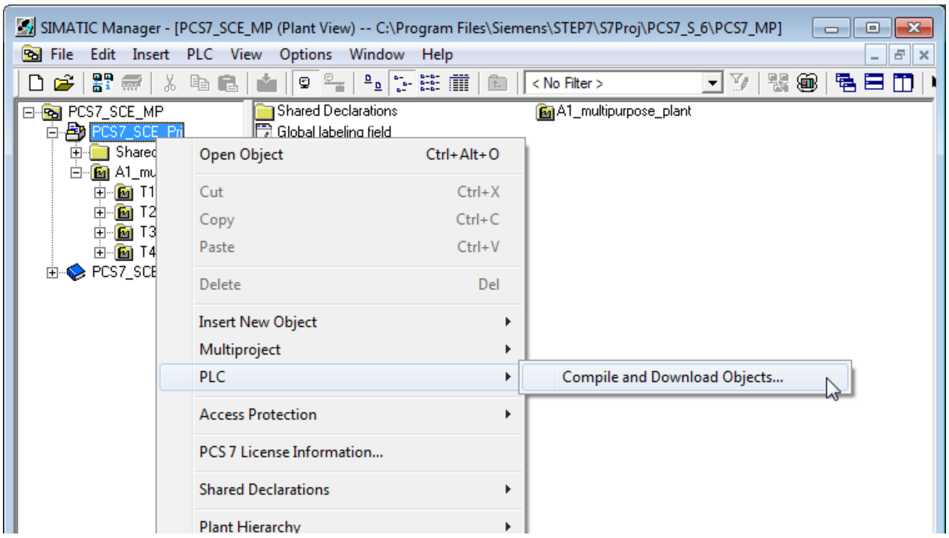
In this chapter, the student learns:

- Diagnosis of the local OPC servers with the OPC Scout V10
- Display of variables available via OPC in OPC Scout V10
- Settings in Excel for OPC access to the variables provided in WinCC
- Test of a Microsoft Excel application with access to variables of WinCC via OPC

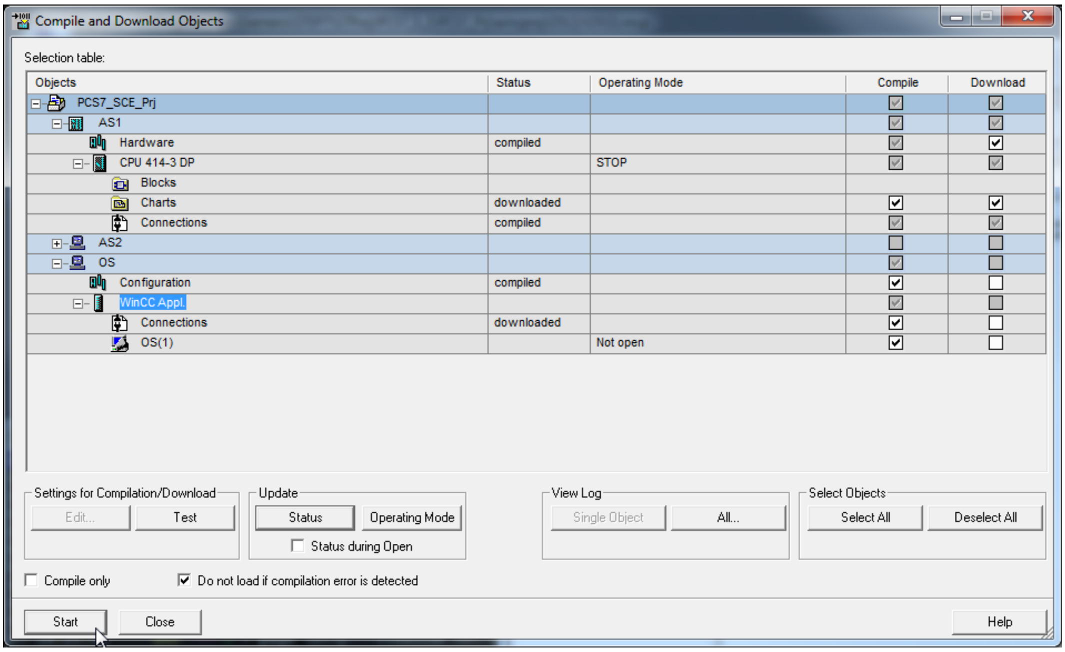
These instructions are based on the project ‘PCS7_SCE_0301_Ueb_R1504_en.zip’ and the Microsoft Excel file ‘PCS7_SCE_0302_OPC_Template_R1305_en.xls‘.

Programming

1. After opening the exercise project from chapter “Advanced Layouts for UIs P03-01“, start PLCSIM and in the ‘Plant View’ Compile and Download Objects of the project is opened. (→ SCE_PCS7_Prj → PLC → Compile and Download Objects)



2. Before you 'Start' the compilation, make sure the S7-PLCSIM is in STOP mode. Everything is compiled and downloaded regarding the charts. Regarding the OS, compile without ‘Memory reset’. (→ Start)


Compile and Download Objects

Selection table:

Objects	Status	Operating Mode	Compile	Download
PCS7_SCE_Prj			☑	☑
AS1			☑	☑
Hardware	compiled		☑	☑
CPU 414-3 DP		STOP	☑	☑
Blocks				
Charts	downloaded		☑	☑
Connections	compiled		☑	☑
AS2			□	□
OS			☑	□
Configuration	compiled		☑	□
WinCC Appl.			☑	□
Connections	downloaded		☑	□
OS(1)		Not open	☑	□

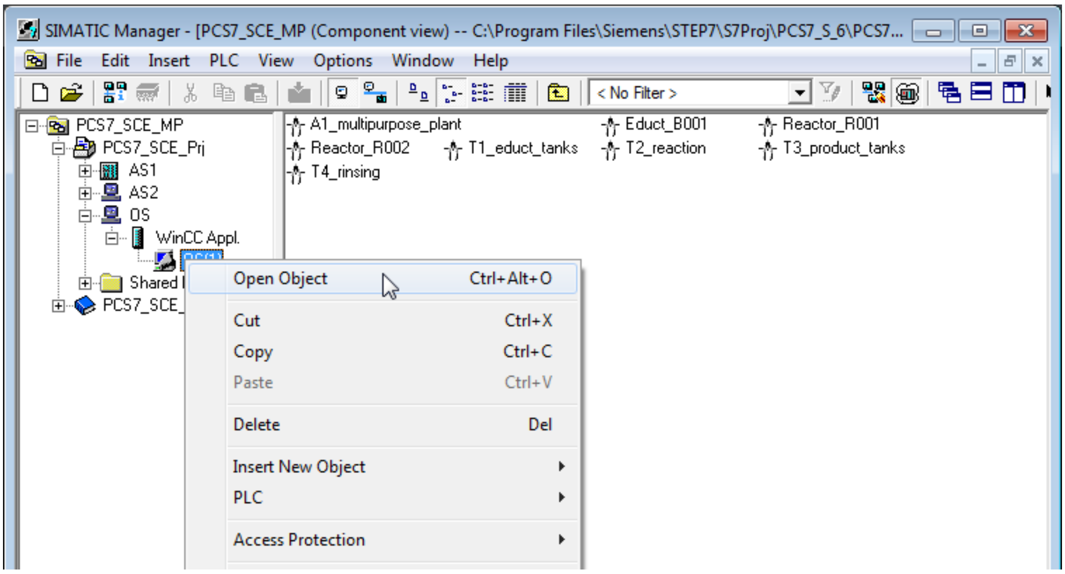
□ Status during Open

□ Compile only

☑ Do not load if compilation error is detected

3. Next, set S7 PLCSIM to the ‘RUN’ mode and open WinCC.

(→ SIMATIC PC Station(1) → WinCC Appl → OS(1) → Open Object)

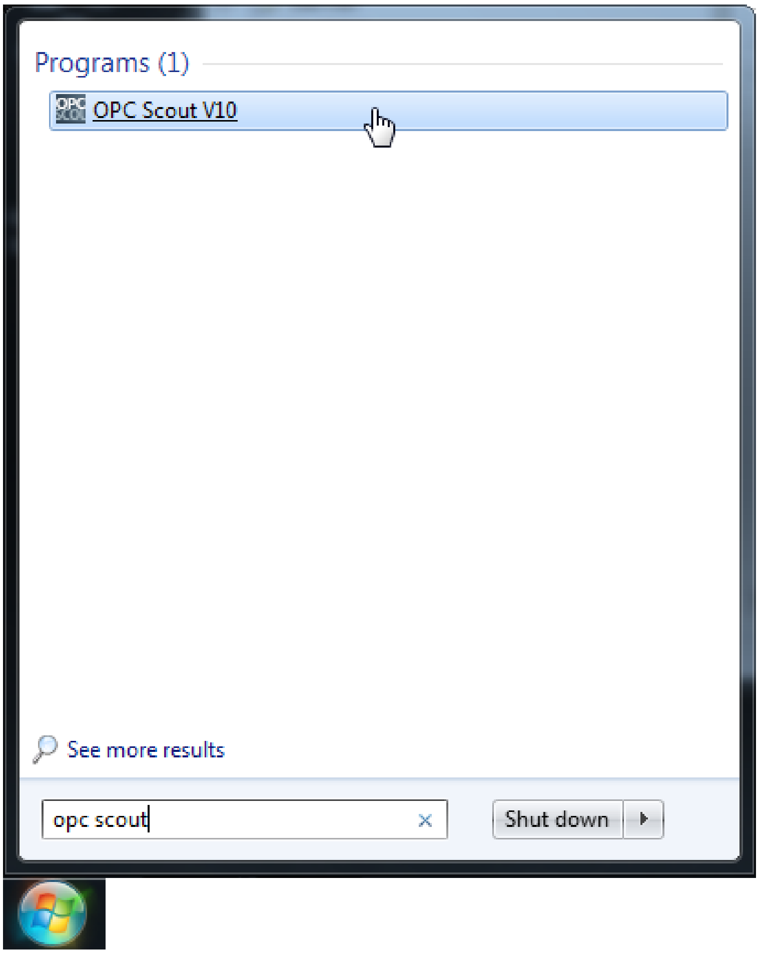


4. Then, in WinCC, activate Runtime.

(→ ▶ Activate)

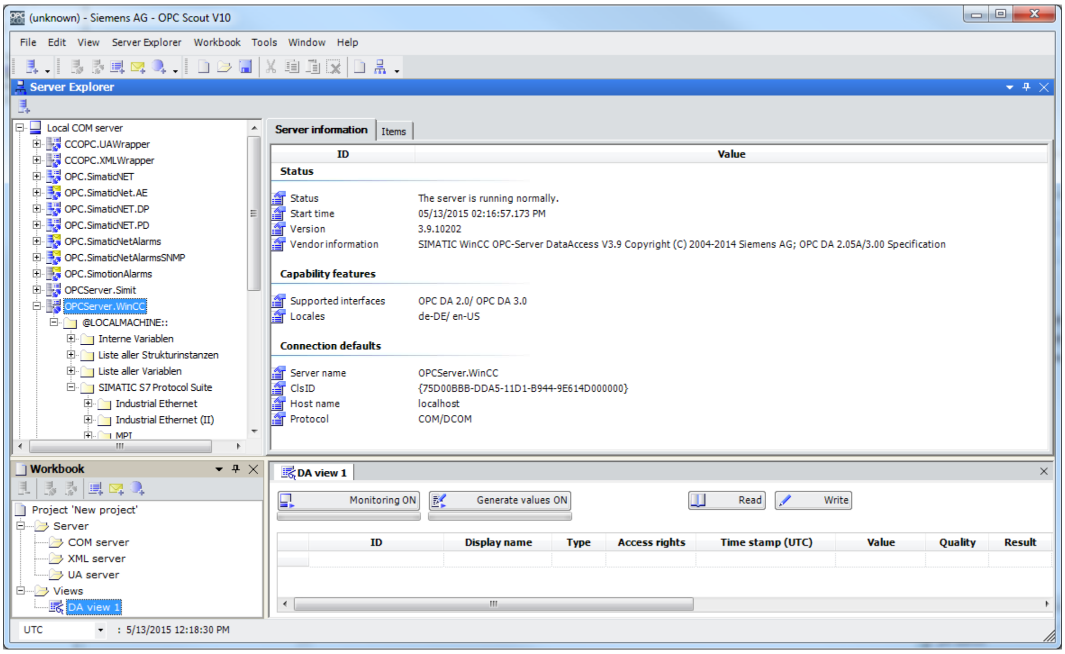
5. After Runtime is completely powered up, start ‘OPC Scout V10’.

   (→ Start → Siemens Automation → SIMATIC → SIMATIC NET → OPC Scout V10)



opc scout

6. In the OPC Scout V10, diagnostic data for the different OPC servers of the local PC is displayed. If it is not possible to set up the connection to a certain server, this is indicated as an error.

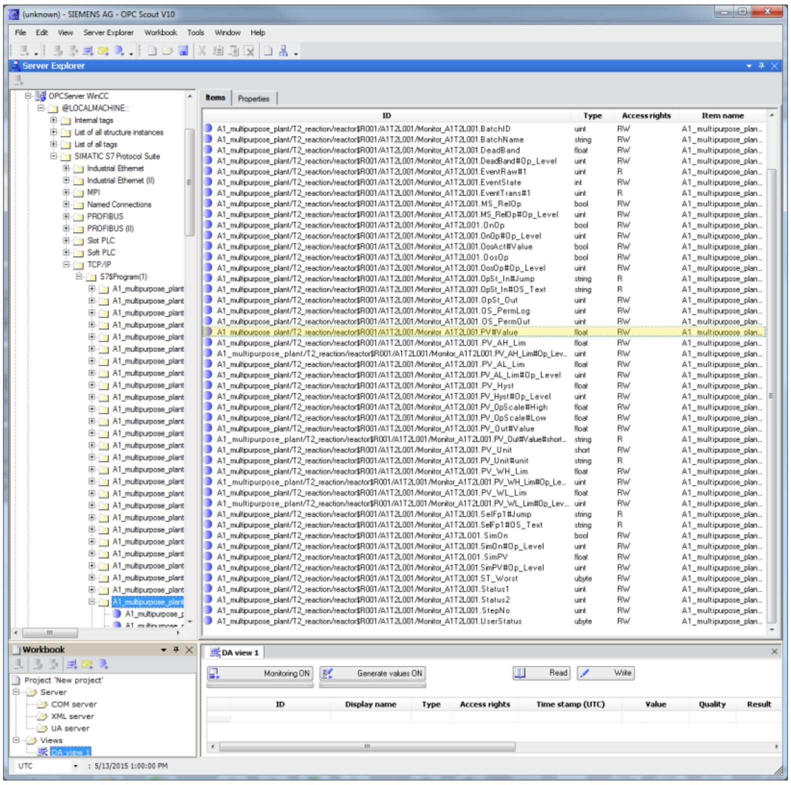


ID	Value
Status	
Status	The server is running normally.
Start time	05/13/2015 02:16:57.173 PM
Version	3.9.10202
Vendor information	SIMATIC WinCC OPC-Server DataAccess V3.9 Copyright (C) 2004-2014 Siemens AG; OPC DA 2.05A/3.00 Specification
Capability features	
Supported interfaces	OPC DA 2.0/ OPC DA 3.0
Locales	de-DE/ en-US
Connection defaults	
Server name	OPCServer.WinCC
ClsID	{75D00BBB-DDA5-11D1-B944-9E614D000000}
Host name	localhost
Protocol	COM/DCOM

ID	Display name	Type	Access rights	Time stamp (UTC)	Value	Quality	Result

: 5/13/2015 12:18:30 PM

7. The WinCC server is called ‘OPCServer.WinCC’. Below this server, a file structure can be opened down to the CFC blocks that allow monitoring. In this example, we selected the block ‘Monitor_A1T2L001’ to display the level of Reactor R001. (→ OPCServer.WinCC → @LOCALMACHINE → SIMATIC S7 Protocol Suite → TCP/IP → S7 Program →A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/ Monitor_A1T2L001)

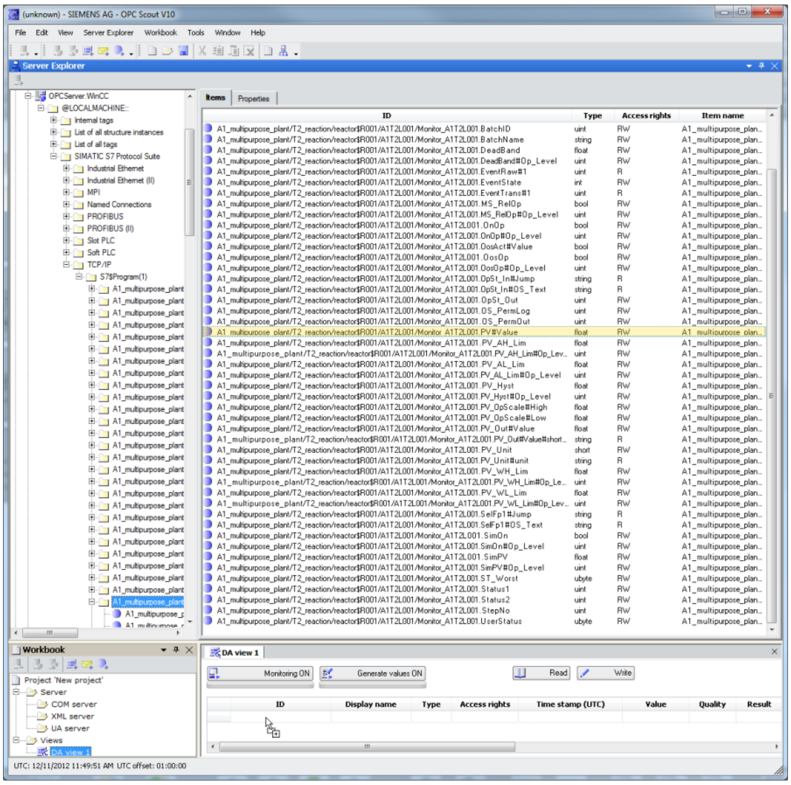


ID	Type	Access rights	Item name
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.BatchID	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.BatchName	string	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.DeadBand	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.DeadBand#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventRaw#1	uint	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventState	int	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventTrans#1	uint	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.MS_RelOp	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.MS_RelOp#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OnOp	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OnOp#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosAct#Value	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosOp	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosOp#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_In#Jump	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_In#OS_Text	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_Out	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OS_PermLog	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OS_PermOut	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim#Op_Lev...	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_Lim	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_Lim#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Hyst	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Hyst#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_OpScale#High	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_OpScale#Low	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Out#Value	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Out#Value#short...	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Unit	short	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Unit#unit	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_WH_Lim	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_WH_Lim#Op_Le...	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_WL_Lim	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_WL_Lim#Op_Lev...	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SelFp1#Jump	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SelFp1#OS_Text	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SimOn	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SimOn#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SimPV	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SimPV#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.ST_Worst	ubyte	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.Status1	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.Status2	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.StepNo	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.UserStatus	ubyte	RW	A1_multipurpose_plan...

ID	Display name	Type	Access rights	Time stamp (UTC)	Value	Quality	Result

8. From the data of this monitoring block, we want to monitor the input ‘PV#Value’. For this reason, drag it to the lower area of ‘DA view 1’.

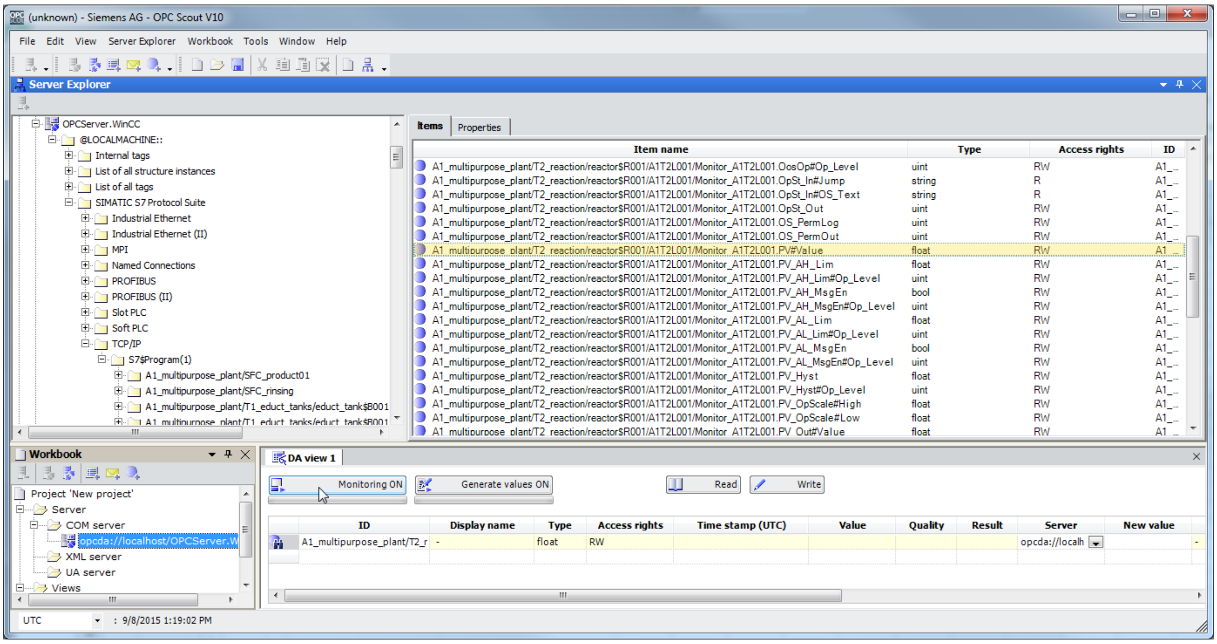
(→A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001 PV#Value)



ID	Type	Access rights	Item name
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.BatchID	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.BatchName	string	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.DeadBand	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.DeadBand#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventRaw#1	uint	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventState	int	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventTrans#1	uint	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.MS_RelOp	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.MS_RelOp#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OnOp	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OnOp#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosAct#Value	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosOp	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosOp#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_In#Jump	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_In#OS_Text	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_Out	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OS_PermLog	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OS_PermOut	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim#Op_Lev...	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_Lim	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_Lim#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Hyst	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Hyst#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_OpScale#High	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_OpScale#Low	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Out#Value	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Out#Value#short...	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Unit	short	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Unit#unit	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_WH_Lim	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_WH_Lim#Op_Le...	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_WL_Lim	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_WL_Lim#Op_Lev...	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SelFp1#Jump	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SelFp1#OS_Text	string	R	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SimOn	bool	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SimOn#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SimPV	float	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.SimPV#Op_Level	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.ST_Worst	ubyte	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.Status1	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.Status2	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.StepNo	uint	RW	A1_multipurpose_plan...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.UserStatus	ubyte	RW	A1_multipurpose_plan...

ID	Display name	Type	Access rights	Time stamp (UTC)	Value	Quality	Result

9. Next, click on ‘Monitoring ON’. Now, in addition to other information, we can monitor the value, time stamp and quality. This shows you whether the ‘OPCServer.WinCC’ works faultlessly. (→ Monitoring ON)



(unknown) - Siemens AG - OPC Scout V10

File Edit View Server Explorer Workbook Tools Window Help

Server Explorer

- OPCServer.WinCC
  - @LOCALMACHINE::
    - Internal tags
    - List of all structure instances
    - List of all tags
    - SIMATIC S7 Protocol Suite
      - Industrial Ethernet
      - Industrial Ethernet (II)
      - MPI
      - Named Connections
      - PROFIBUS
      - PROFIBUS (II)
      - Slot PLC
      - Soft PLC
      - TCP/IP
        - S7$Program(1)
          - A1_multipurpose_plant/SFC_product01
          - A1_multipurpose_plant/SFC_rinsing
          - A1_multipurpose_plant/T1_educt_tanks/educt_tank$B001
          - A1_multipurpose_plant/T1_educt_tanks/educt_tank$B001

Items | Properties

Item name	Type	Access rights	ID
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosOp#Op_Level	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_In#Jump	string	R	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_In#OS_Text	string	R	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_Out	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OS_PermLog	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OS_PermOut	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value	float	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim	float	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim#Op_Level	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_MsgEn	bool	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_MsgEn#Op_Level	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_Lim	float	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_Lim#Op_Level	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_MsgEn	bool	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_MsgEn#Op_Level	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Hyst	float	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Hyst#Op_Level	uint	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_OpScale#High	float	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_OpScale#Low	float	RW	A1_...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Out#Value	float	RW	A1_...

Workbook

- Project 'New project'
  - Server
    - COM server
      - opcda://localhost/OPCServer.W
    - XML server
    - UA server
  - Views

DA view 1

Monitoring ON | Generate values ON | Read | Write

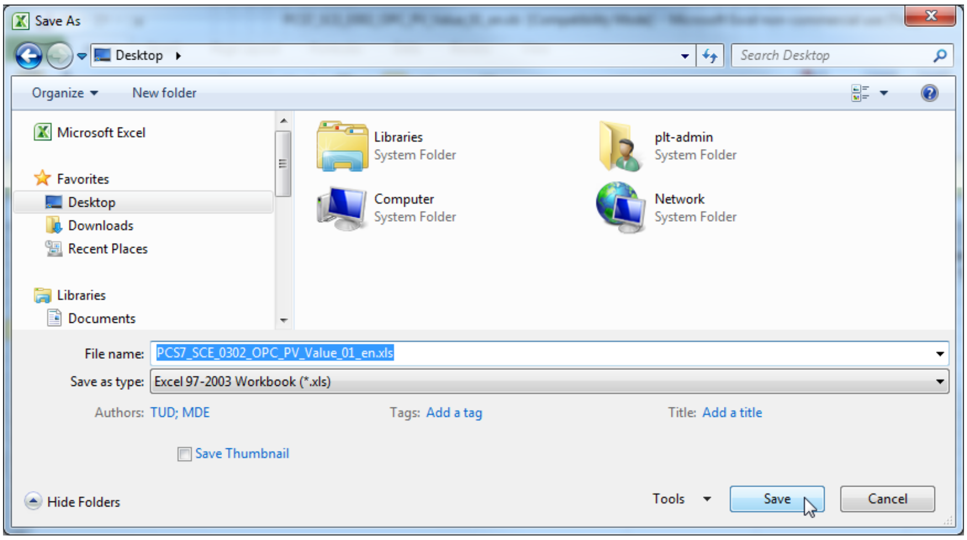
ID	Display name	Type	Access rights	Time stamp (UTC)	Value	Quality	Result	Server	New value
A1_multipurpose_plant/T2_r	-	float	RW					opcda://localh	

UTC : 9/8/2015 1:19:02 PM

10. If this is the case, open the Microsoft Excel file included in the module with a double click. (→ PCS7_SCE_0302_OPC_Template_R1305_en.xls)

11. Then, save it in Microsoft Excel under a new name.

(→ File → Save As → PCS7_SCE_0302_OPC.xls)



Save As

Desktop ▸ | Search Desktop

Organize ▾ | New folder

- Microsoft Excel
- Favorites
  - Desktop
  - Downloads
  - Recent Places
- Libraries
  - Documents

Libraries System Folder	plt-admin System Folder
Computer System Folder	Network System Folder

File name: PCS7_SCE_0302_OPC_PV_Value_01_en.xls

Save as type: Excel 97-2003 Workbook (*.xls)

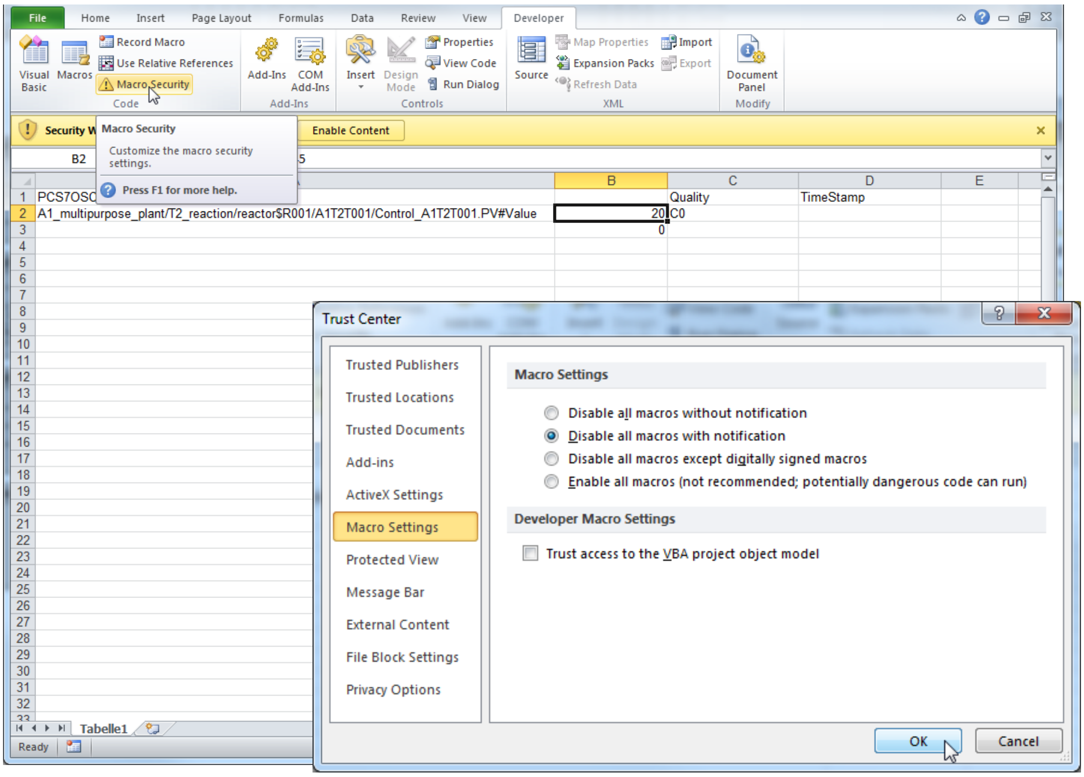
Authors: TUD; MDE | Tags: Add a tag | Title: Add a title

□ Save Thumbnail

Hide Folders | Tools ▾ | Save | Cancel

12. In Microsoft Excel, we now have to set the security settings correctly for the macros to be started.

    (→ Developer → Macro Security → Macro Settings → Disable all macros with notification → OK)



Macro Security

Customize the macro security settings.

Press F1 for more help.

	A	B	C	D	E
1	PCS7OSC		Quality	TimeStamp	
2	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2T001/Control_A1T2T001.PV#Value	20	C0		
3		0			

Trust Center

Macro Settings

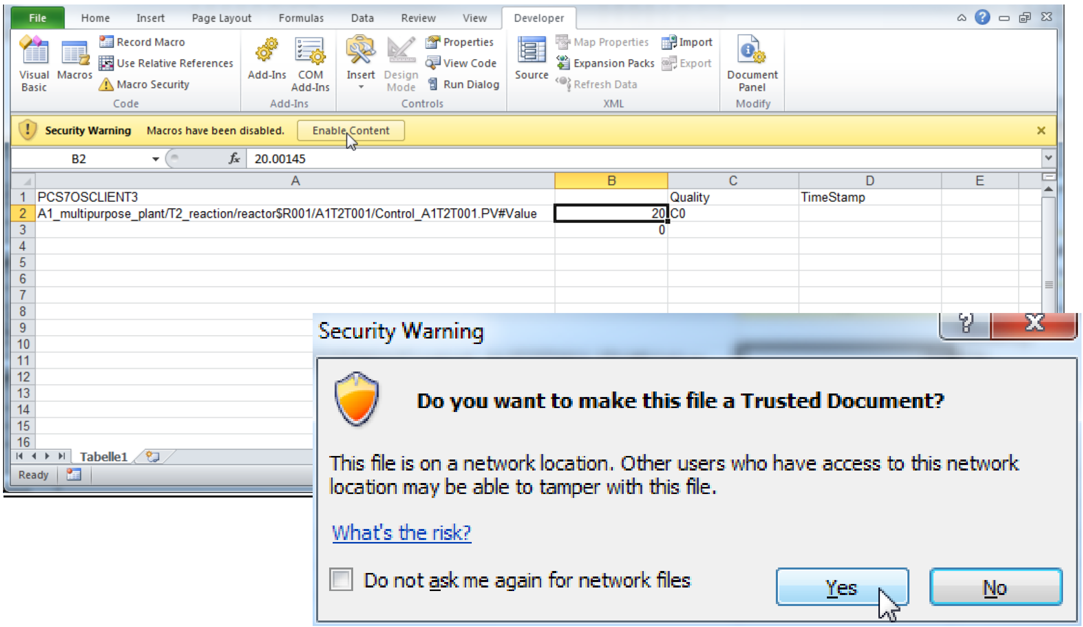
○ Disable all macros without notification
● Disable all macros with notification
○ Disable all macros except digitally signed macros
○ Enable all macros (not recommended; potentially dangerous code can run)

Developer Macro Settings

□ Trust access to the VBA project object model

13. Then, in ‘Options‘, activate the macros for this session.

    (→ Options → Enable Content → OK)



Security Warning Macros have been disabled.

B2 fx 20.00145

	A	B	C	D	E
1	PCS7OSCLIENT3		Quality	TimeStamp	
2	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2T001/Control_A1T2T001.PV#Value	20	C0		
3		0			

Security Warning

Do you want to make this file a Trusted Document?

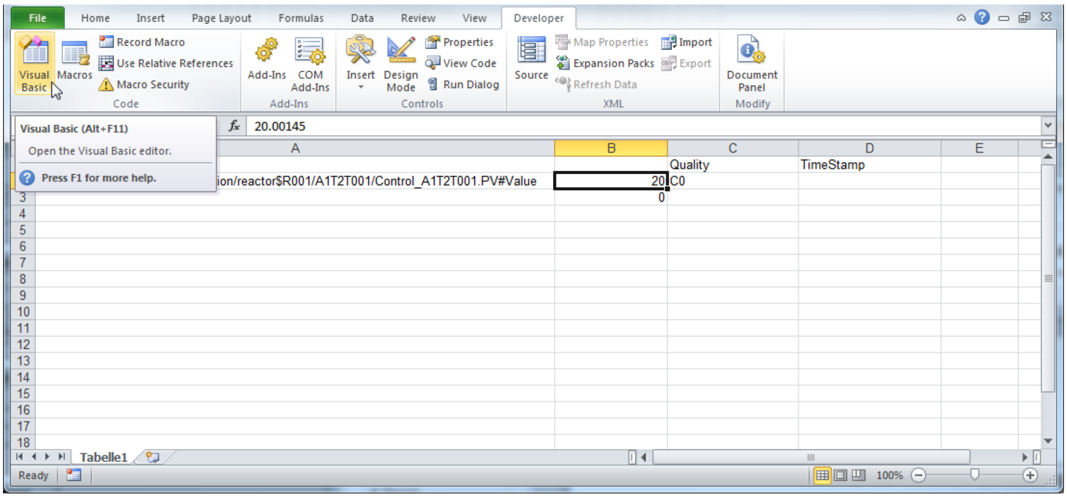
This file is on a network location. Other users who have access to this network location may be able to tamper with this file.

What's the risk?

□ Do not ask me again for network files

14. . For the additional settings, open ‘Visual Basic’ in Microsoft Excel.

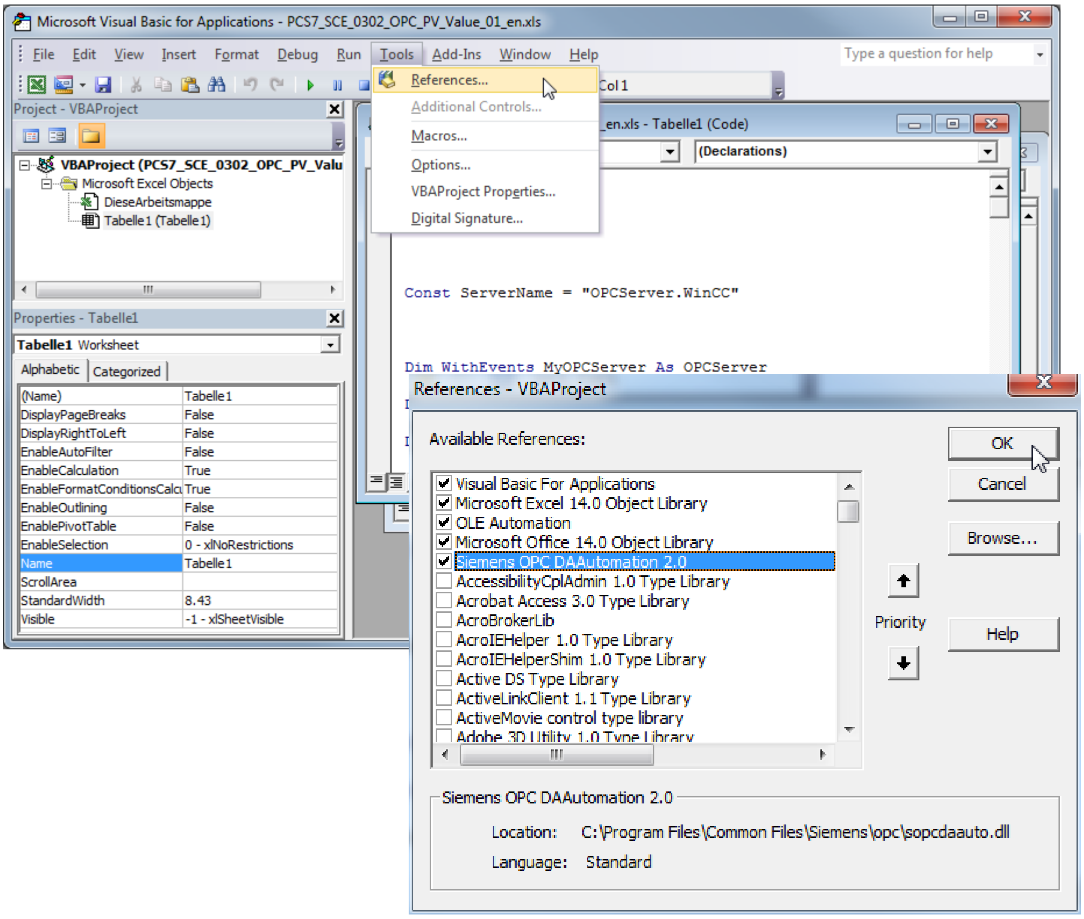
    (→ Developer → Visual Basic)



15. There, under ‘Tools’ open the option ‘References’. (→ Tools → References)

16. For the variables to be accessed in OPCServer 'WinCC’, the reference to the dll ‘Siemens OPC DA Automation 2.0’ has to be activated; if this reference is not available, it has to be entered here with ‘Browse’. The path is then ‘C:\Program Files\Common Files\Siemens\opc\sopcdaauto.dll’.

    (→ Siemens OPC DA Automation 2.0 → Browse → C:\Program Files\Common Files\ Siemens\ opc\sopcdaauto.dll → OK)



(Name)	Tabelle1
DisplayPageBreaks	False
DisplayRightToLeft	False
EnableAutoFilter	False
EnableCalculation	True
EnableFormatConditionsCalcu	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestrictions
Name	Tabelle1
ScrollArea	
StandardWidth	8.43
Visible	-1 - xlSheetVisible

```
Const ServerName = "OPCServer.WinCC"


Dim WithEvents MyOPCServer As OPCServer
```

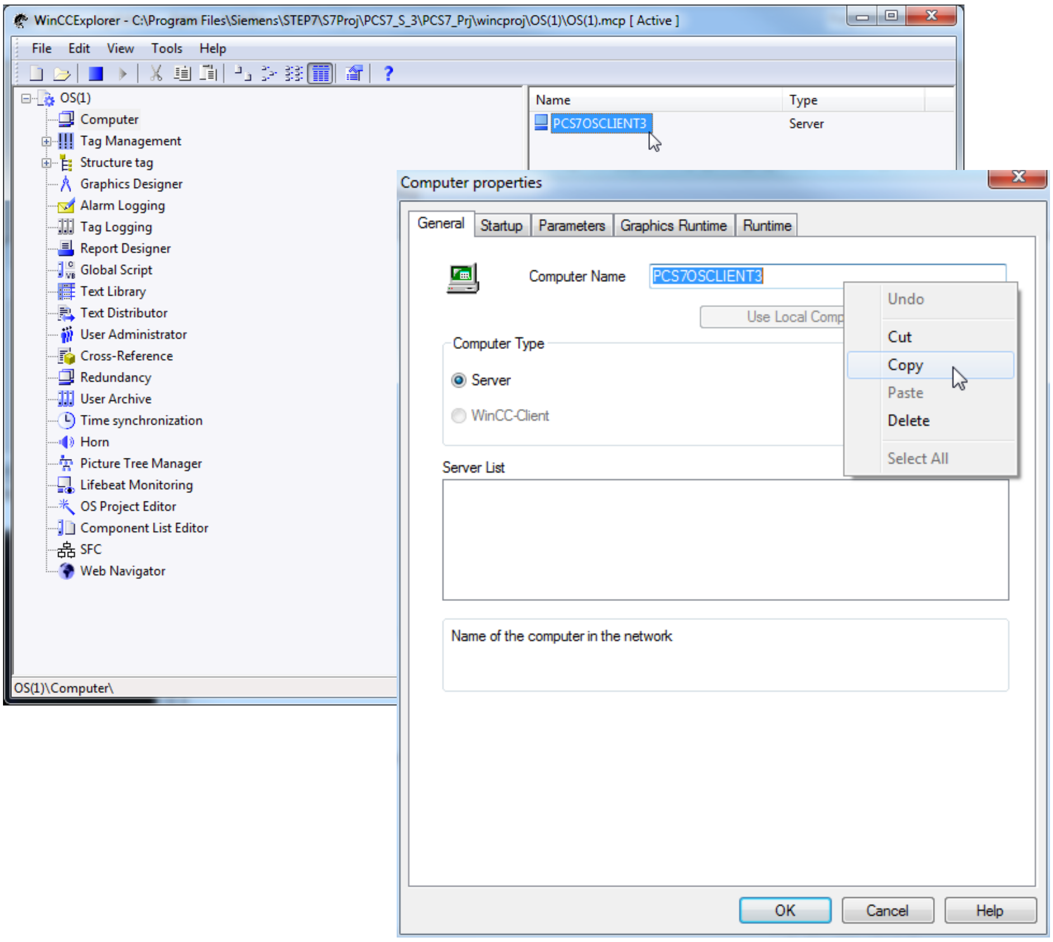
☑ Visual Basic For Applications
☑ Microsoft Excel 14.0 Object Library
☑ OLE Automation
☑ Microsoft Office 14.0 Object Library
☑ Siemens OPC DAAutomation 2.0
□ AccessibilityCplAdmin 1.0 Type Library
□ Acrobat Access 3.0 Type Library
□ AcroBrokerLib
□ AcroIEHelper 1.0 Type Library
□ AcroIEHelperShim 1.0 Type Library
□ Active DS Type Library
□ ActiveLinkClient 1.1 Type Library
□ ActiveMovie control type library
□ Adobe 3D Utility 1.0 Type Library

Siemens OPC DAAutomation 2.0

Location: C:\Program Files\Common Files\Siemens\opc\sopcdaauto.dll

Language: Standard

17. Now, the computer name has to be entered in field A1. We can copy it in WinCC under Computer properties. In field A2, the complete variable name has to be entered; it can be copied from the OPC Scout V10. (→ A1 → PCS7OSLIENT3 → A2 → A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value)



WinCCExplorer - C:\Program Files\Siemens\STEP7\S7Proj\PCS7_S_3\PCS7_Prj\wincproj\OS(1)\OS(1).mcp [Active]

Name	Type
PCS7OSCLIENT3	Server

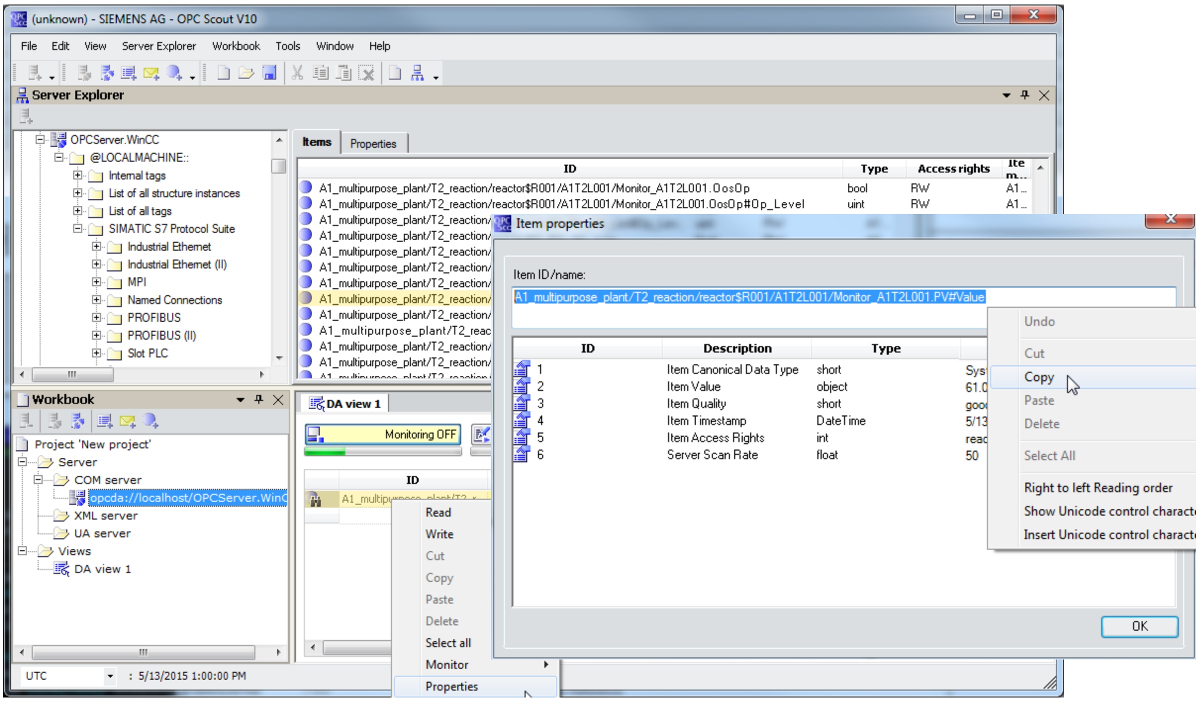
Computer properties

Computer Name: PCS7OSCLIENT3

Computer Type: ● Server ○ WinCC-Client

Name of the computer in the network

OS(1)\Computer\



(unknown) - SIEMENS AG - OPC Scout V10

ID	Type	Access rights	Ite m...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosOp	bool	RW	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OosOp#Op_Level	uint	RW	A1...

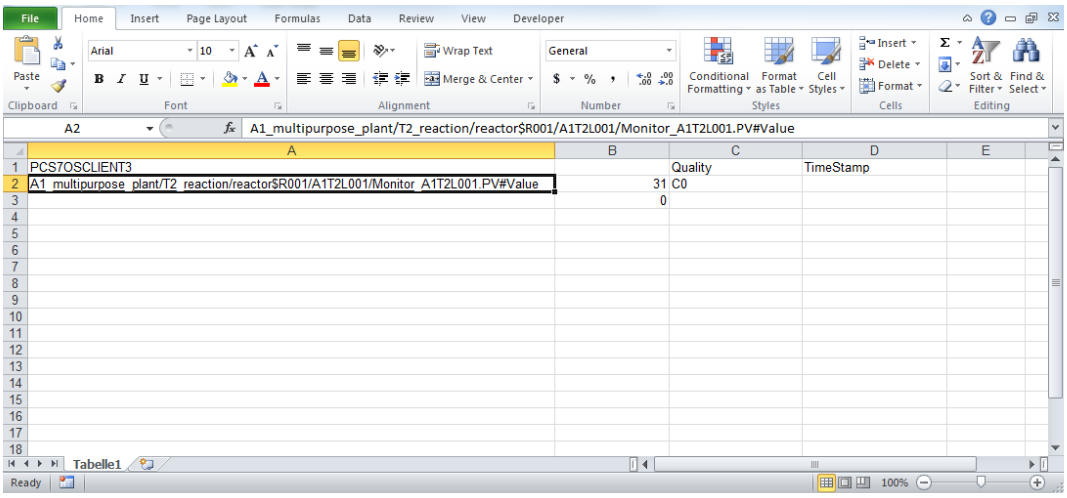
Item properties

Item ID/name: A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value

ID	Description	Type	
1	Item Canonical Data Type	short	Sys
2	Item Value	object	61.0
3	Item Quality	short	goo
4	Item Timestamp	DateTime	5/13
5	Item Access Rights	int	read
6	Server Scan Rate	float	50

Project 'New project'
Server
COM server
opcda://localhost/OPCServer.WinC
XML server
UA server
Views
DA view 1

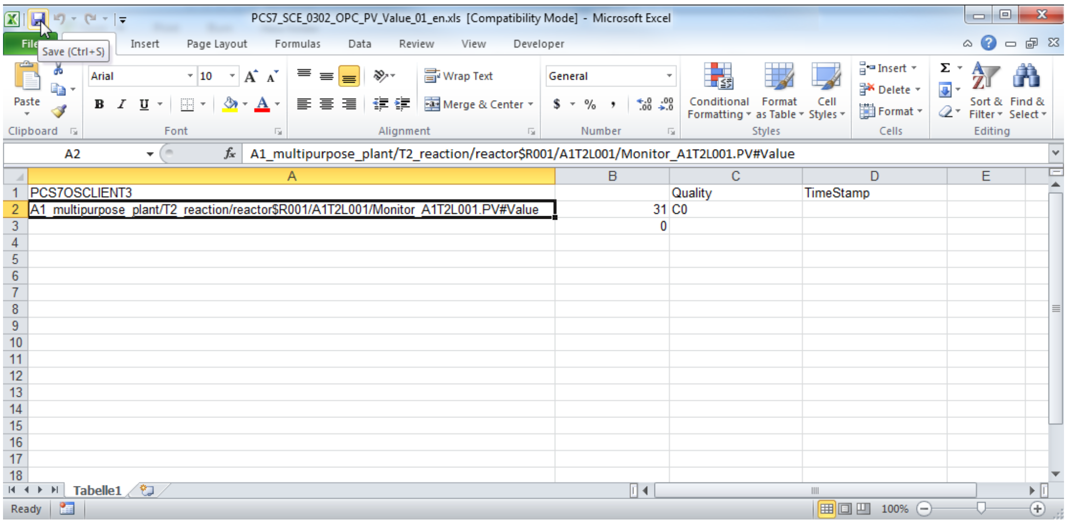
UTC : 5/13/2015 1:00:00 PM



A2 = A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value

	A	B	C	D	E
1	PCS7OSCLIENT3		Quality	TimeStamp	
2	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value	31	C0		
3		0			

18. Next, save and close the Microsoft Excel file prior to reopening it with a double click. (→ Save → ☒ → PCS7_SCE_0302_OPC.xls)

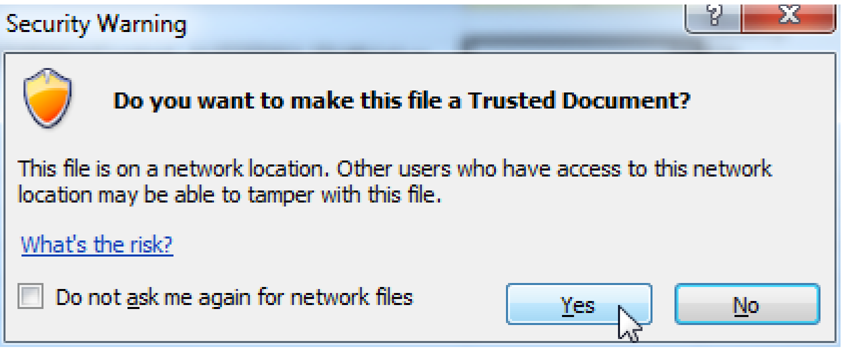


PCS7_SCE_0302_OPC_PV_Value_01_en.xls [Compatibility Mode] - Microsoft Excel

A2 = A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value

	A	B	C	D	E
1	PCS7OSCLIENT3		Quality	TimeStamp	
2	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value	31	C0		
3		0			

19. During opening, we take note of the security warning and here also, we enable the macros for the following session. (→ Options → Enable macros)



Security Warning

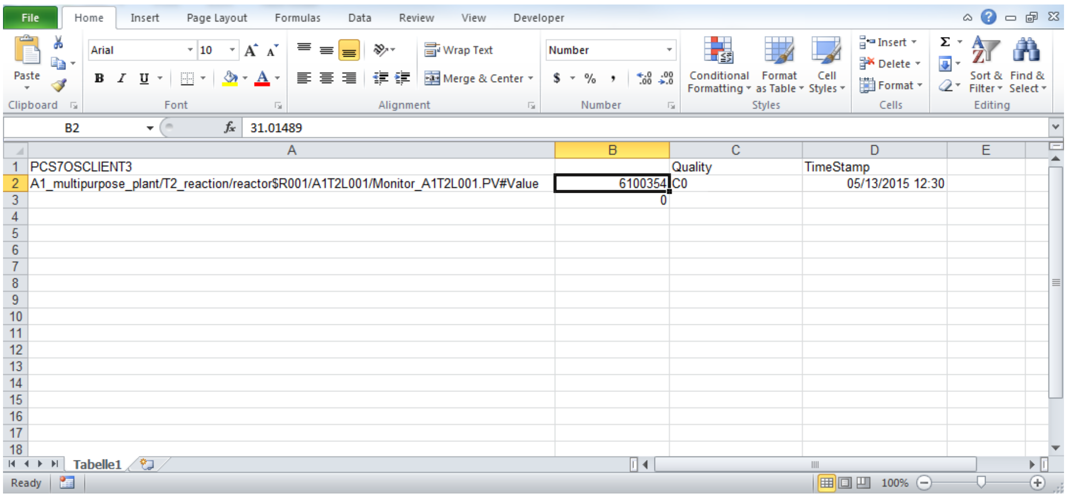
Do you want to make this file a Trusted Document?

This file is on a network location. Other users who have access to this network location may be able to tamper with this file.

What's the risk?

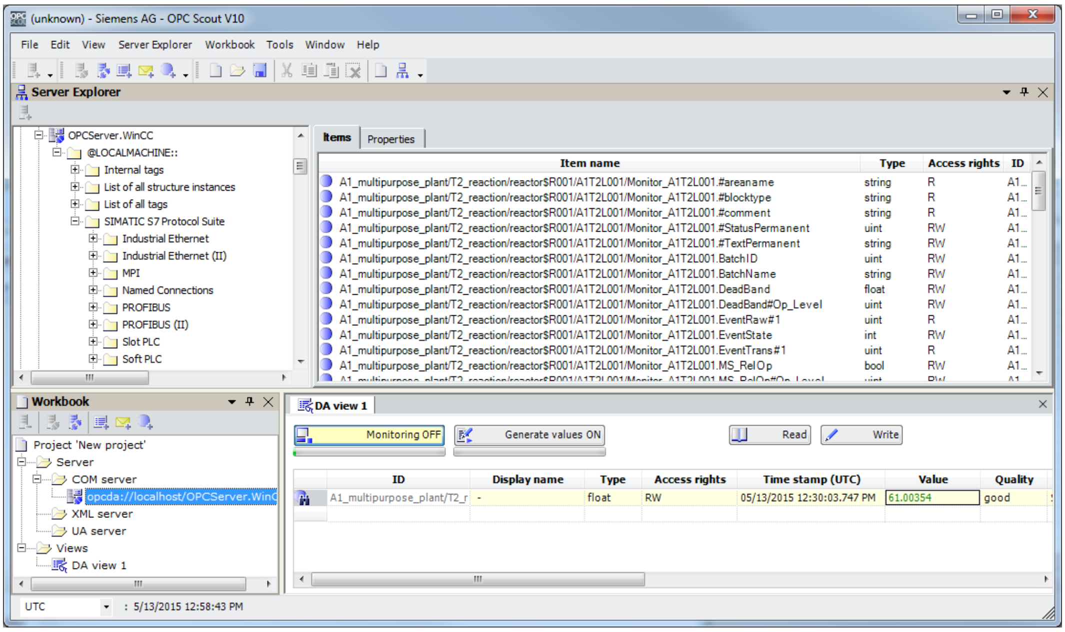
□ Do not ask me again for network files

20. Now, in field B2, the variable can be monitored.



B2 | 31.01489

	A	B	C	D	E
1	PCS7OSCLIENT3		Quality	TimeStamp	
2	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value	6100354	C0	05/13/2015 12:30	
3		0			

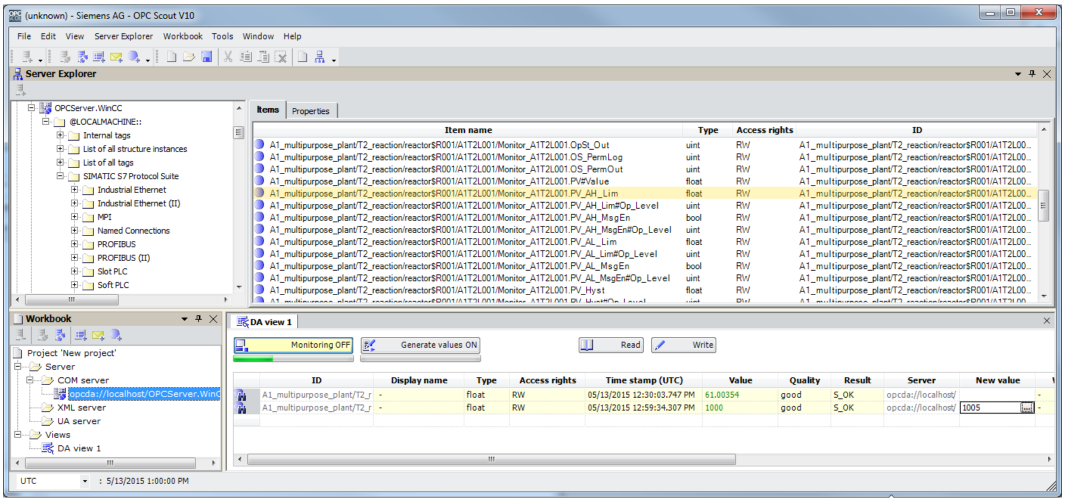


(unknown) - Siemens AG - OPC Scout V10

Item name	Type	Access rights	ID
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.#areaname	string	R	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.#blocktype	string	R	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.#comment	string	R	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.#StatusPermanent	uint	RW	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.#TextPermanent	string	RW	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.BatchID	uint	RW	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.BatchName	string	RW	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.DeadBand	float	RW	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.DeadBand#Op_Level	uint	RW	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventRaw#1	uint	R	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventState	int	RW	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.EventTrans#1	uint	R	A1...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.MS_RelOp	bool	RW	A1...

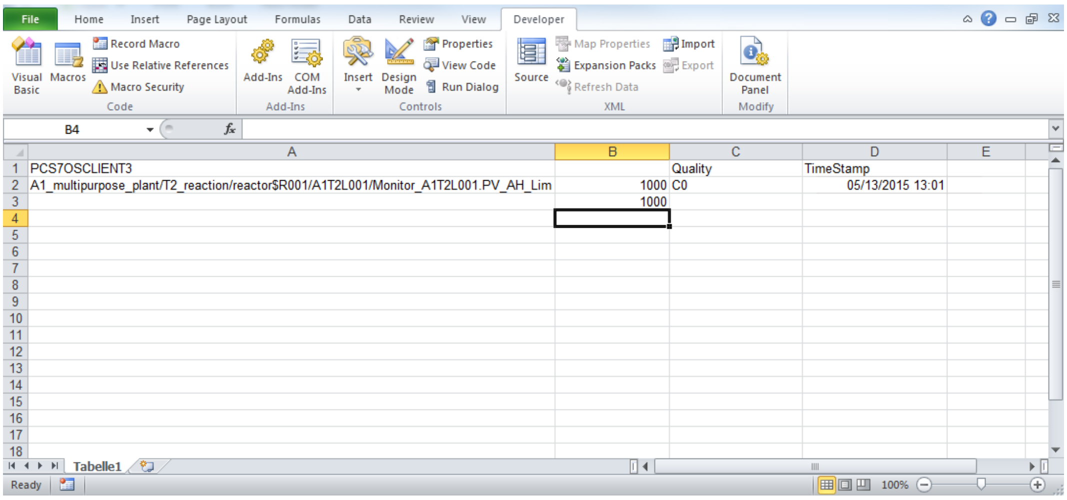
ID	Display name	Type	Access rights	Time stamp (UTC)	Value	Quality
A1_multipurpose_plant/T2_r	-	float	RW	05/13/2015 12:30:03.747 PM	61.00354	good

21. In field B3, a new value can be written for the variable. We cannot use the process value as above because it would overwrite itself immediately. It is best to use a fixed parameter, for example, A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim to test this functionality. With the OPC Scout, change the value to 1005, and then with the Excel file back to 1000.



Item name	Type	Access rights	ID
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OpSt_Out	uint	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OS_PermLog	uint	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.OS_PermOut	uint	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV#Value	float	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim	float	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim#Op_Level	uint	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_MsgEn	bool	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_MsgEn#Op_Level	uint	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_Lim	float	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_Lim#Op_Level	uint	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_MsgEn	bool	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AL_MsgEn#Op_Level	uint	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...
A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_Hyst	float	RW	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L00...

ID	Display name	Type	Access rights	Time stamp (UTC)	Value	Quality	Result	Server	New value
A1_multipurpose_plant/T2_r	-	float	RW	05/13/2015 12:30:03.747 PM	61.00354	good	S_OK	opcda://localhost/	
A1_multipurpose_plant/T2_r	-	float	RW	05/13/2015 12:59:34.307 PM	1000	good	S_OK	opcda://localhost/	1005



	A	B	C	D	E
1	PCS7OSCLIENT3		Quality	TimeStamp	
2	A1_multipurpose_plant/T2_reaction/reactor$R001/A1T2L001/Monitor_A1T2L001.PV_AH_Lim	1000	C0	05/13/2015 13:01	
3		1000			
4					

EXERCISES

In the exercises, we apply what we learned in the Theory section and in the Step by Step Instructions.

The existing multi-project PCS7_SCE_0301_Ueb_R1504_en.zip and the table PCS7_SCE_0302_OPC_Template_R1305_en.xls will be used for this.

The objective of this exercise is: identifying and reading a measured value by means of an OPC item.

TASKS

1. Using PCS 7 and the OPC Scout, identify the corresponding variable name of temperature measurement in Reactor R001.
2. Modify the variable name in the table from the step by step instructions with the new name and read out the value.