Title: **TsNet use for System One Application Testing**

Subject: **Engineering Manual**

Project: **103267, DESIGO V5**

The document describes the use of the TsNet toolset for DESIGO application testing.

Key Words:

Related Product(s):	TsNet; SystemOne; PXC3
Document Storage:	ProjectShare
Document Category:	ProjectRecord
Document Number:	PD103267-T-0104_EN
Revision:	0.3
Revision Date:	2013-10-30
Document Status:	Final - without Approval
Author:	Michael Speckien
Department:	Department
Responsible:	Michael.Speckien@siemens.com
Company:	Siemens Schweiz AG, Industry, Building Technologies Division Control Products & Systems
Classification:	Restricted
Based on Template:	HaBuA4len; 2; 2010-01-24; Arnold Broekmeijer

Revision History

Rev	Date	Author	Remarks
0.3	30-Okt-2013	Michael Speckien, 5556	Status = **Final - without Approval** - Updated to new version
0.2	29-Aug-2012	Michael Speckien, 5556	Status = **Final - without Approval** - Update for new tool version

Table of Contents

1. About this document

1.1.1 Copyright

1.2 Document validity

The document is valid for the following tool versions:

TsNet template :	V0.6
TsGen:	V2.3.9322
TsNet:	V2.3.9322
POOS:	V0.04

1.3 Target readers

This document is intended to be used only by application engineering personnel in HQ. Adequate skills are required.

1.4 Terms

The tools have different definitions for the test procedures.

Quality Center	Text	TestTemplate	TsNet
Test Set	Sequence of tests for a delivery. For example module test for a new version of a library.	n.a.	n.a.
Test	Sequence of test steps for a part of a delivery, for example product test for a function block	Test	"Main" Testscript
Test Step	Sequence of input values and reference values for a situation within a test. For example manual operation for an application	Test Case, Test Sheet	Test, Testscript
n.a.	One set of input values, and the reference values expected, after the input values have been applied. For example output shall be true within 10 seconds after manually switched on	Action	n.a.
n.a.	Individual command for writing or reading a value. For example Write True to binary output 2, present value prio 8.	n.a.	Step

2. Tool Overview

All tools described here, are BACnet tools.
So, all variables which are not already BACnet objects, shall be mapped to additional BACnet objects, before the test can start.

Remark: The tools have more functionality than described here. This manual only covers aspects which are relevant for application testing.

2.1 TsNet

Test Sequencer Runtime Tool

Usage: Writing and verifying BACnet data based on a pre-defined test sequences
Concept: Test sequence is defined with an Excel sheet. The test sequence is executed step by step. It contains mainly
- setting of BACnet properties
- Reading of BACnet properties, and verifying the value against a predefined reference value or range
- Starting time delays

TsNet supports all objects and properties supported by the DESIGO system, also structured data.
Restrictions: TsNet allows no list or array datatypes except priority array.
TsNet cannot execute tests with a reaction time < 2 sec, and works only with constant reference values.

2.2 TestScriptGen

Test Script Generator for Application test.

Usage: Converting Input / output tables with BACnet values from the test template into TsNet test sequences..

2.3 TestTemplate

Test template for application tests

Usage: Definition of test conditions and results in an Excel sheet.
Concept: Tests are defined by a sequence of Input conditions and output reference values. The BACnet objects and devices for the tests are defined by names.
An EDE list and a device list allows the conversion of the names into BACnet object IDs.
The data entered can be forwarded to POOS and Quality center.

2.4 POOS

Datalogger for tracing BACnet properties.

Usage: Tracing and trending BACnet properties cyclically.
Concept: The properties to be traced are defined in an Excel file. They are read cyclically and saved in a textfile.

3. Installation

3.1 Download

3.1.1 TsNet tools

Get the TsNet, the TestScriptGen, the TestTemplate and POOS from the System One Sharepoint

https://workspace.sbt.siemens.com/content/10003162/team_appl/Documents/Forms/compact.aspx?RootFolder=%2Fcontent%2F10003162%2Fteam%5Fappl%2FDocuments%2F70%5FAppl%5FTesting%2F03%5FTestenvironment%2F03%5FTestenvironment%5FDevelopment%2FTsNet&View=%7B084FC38D%2D45C2%2D47C1%2DA4C5%2D74B89693D5D7%7D

Load the TsNetxxx.zip file to a temporary directory and extract.
Start the installer and follow the instructions.
Extract the TestTemplate to your project folder.

Extract POOS to a project files folder. This tool needs no installation.

3.1.2 Additional tools

3.1.2.1 Wireshark

It is recommended to install the wireshark network logger and analyser.
This is freeware and available on the internet:

http://www.wireshark.org/download.html

Get the latest stabile release, copy it to a temporary location and follow the user guide to install it. Installing Wireshark also installs WinPcap (This ist the logger). The use of wireshark is often restricted by the IT. Please ask your IT department.

3.1.2.2 GetEDE

This is a small tool from BT Germany, which scans for BACnet devices and their objects online.
It creates an EDE list, which can be used for TsNet.
It can be downloaded from the System One Sharepoint, location see above.

Download it to a temporary location and extract it to the program files folder.
This tool needs no installation.

3.1.2.3 Quality Center Excel Addin

This is used to export Application Test Specifications to HP Quality center.
You find it when opening Quality Center, then go to
“Help“ → “Add-In-Page“ →
“More HP Quality Center Add-ins“ → “Microsoft Excel Add-in“
Follow the instructions and install the add-in.

3.2 Parameterisation

3.2.1 Remarks on BACnet

TsNet, POOS, GetEDE, Cimetrics BACstack, DESIGO INSIGHT and many more are BACnet clients. Cimetrics Explorer and WAGO Explorer require that Cimetrics BACstack is running

There can only be one BACnet client with the same IP address and port on the network.

So when you want to run above applications simultaneously, you need individual IP addresses for each application. When you have only one IP address, you must close all other BACnet clients or Stopp the Cimetrics BACstack before starting an other BACnet client.

Example for IP addresses on a test PC connected to a private network:
192.168.251.5 : Cimetrics BACstack (Cimetrics Explorer, WAGO Explorer)
192.168.251.100 : TsNet
192.168.251.101 : VTS (TestShell for sending BACnet requests)
192.168.251.102 : DESIGO INSIGHT
192.168.251.103 : GetEDE
192.168.251.104 : SmartTalk ((TestShell for sending BACnet requests)
192.168.251.105 : POOS

3.2.2 Parameterisation TsNet

Go to the program folder, where TsNet was installed ..(x86)\Siemens\TsNET Toolset\TsNet.

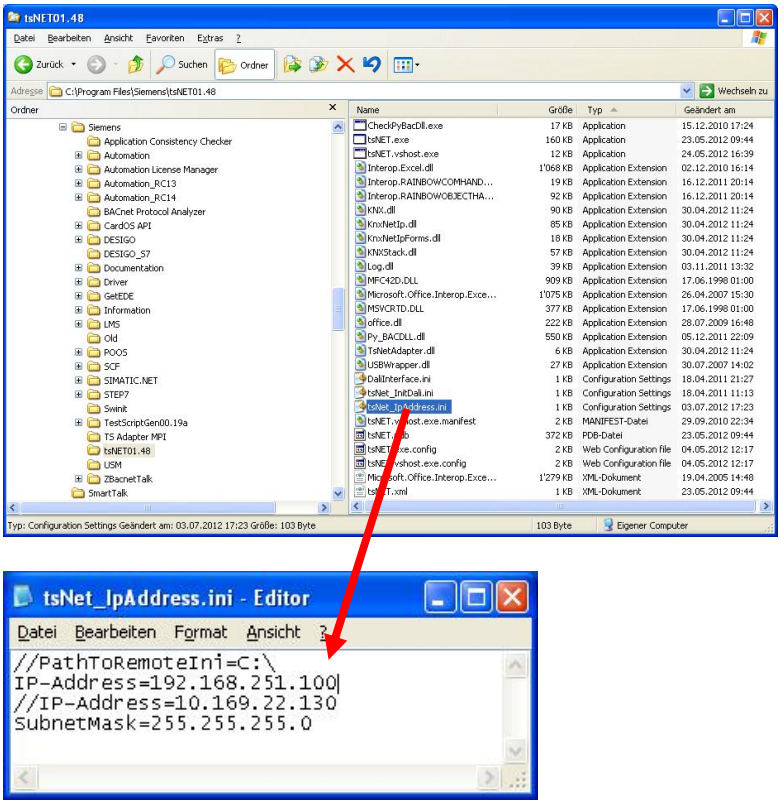


Figure 3–1: Parameterisation TsNet

Remark: On a CAT client, you need administrator rights. Instead of editing the file, you have to copy it to a temporary location, edit it there, delete it from the original location, and move it back to the original location.

3.2.3 Parameterisation POOS

Go to the folder, where POOS was extracted, and enter your IP address in the file tsNet_IpAddress.ini.

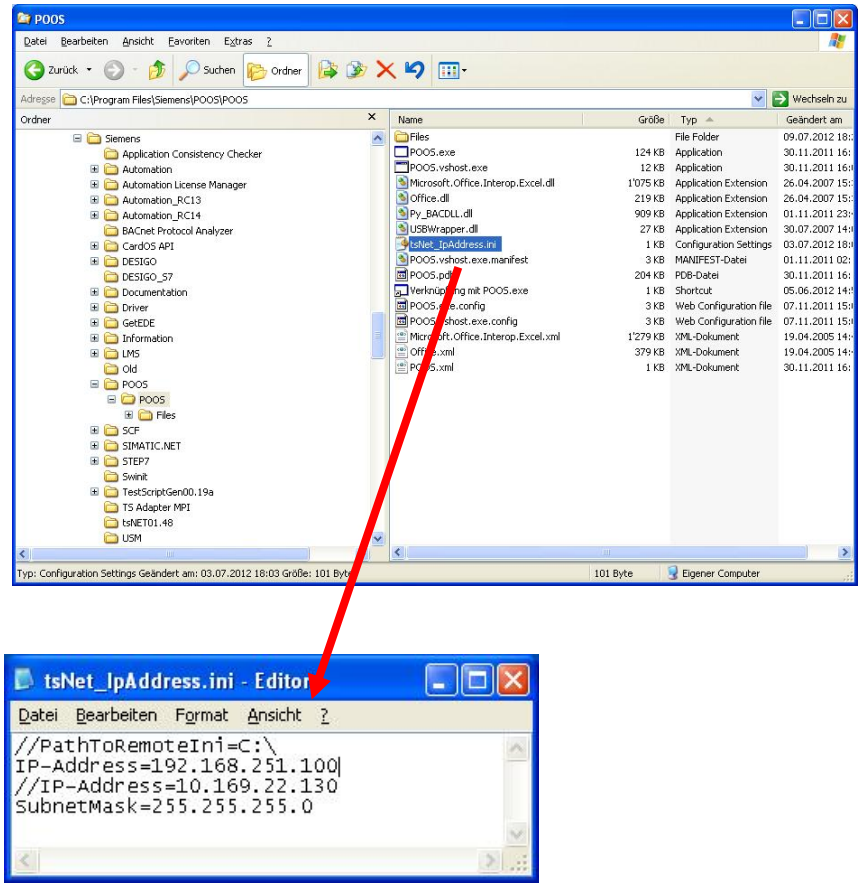


Figure 3–2: Parameterisation POOS

It is recommended to create a shortcut to POOS.exe for easier starting POOS.

3.2.4 Parameterisation TestScriptGen

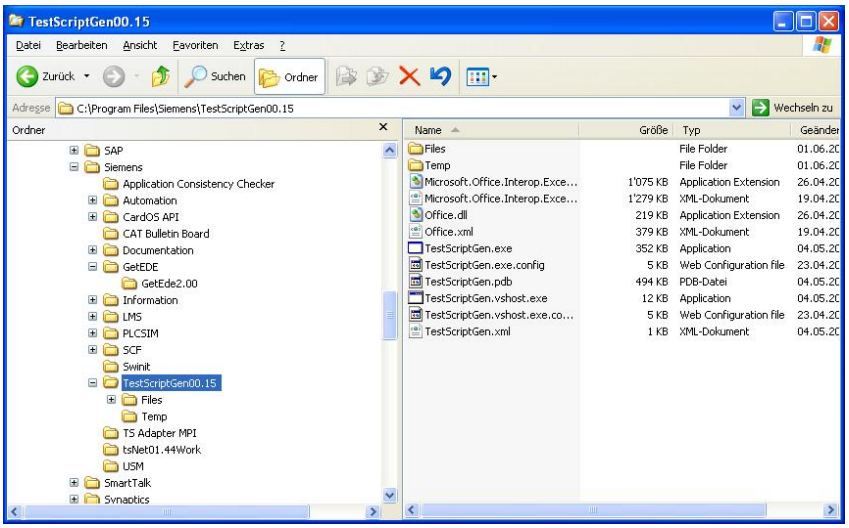


Figure 3–3: Shortcut to TestScriptGen

3.3 TestTemplate

The test template is used once per test.
It is recommended, to parameterize the test template according to the environment of your test set, and then copy and rename it for each test required.

4. Test Specification

Open the template TestTemplate*.xls and save it to your project with a new name, for example as "Test_Docu.xls".
General
- Input fields are marked in grey or turquoise. Required input fields have a red margin.
- Please do not delete or rename any worksheets, do not delete, insert or rename cells, columns or lines unless explicitly allowed
- Please do not write into the "TemplateXXX" sheets even when marked in grey or turquoise

Sequence
- Fill in the config sheet : Test and test environment data
- Fill EDE sheet : BACnet object data
- Go to Overview sheet and insert a test step "StepName"
- Go to the sheet TestCaseStepname and enter objects, conditions and results

4.1 Config Sheet, Configuration

	A	B	C	D	E	F	G	O	P
1	**Application Test**	**Config Sheet**				Test Details	Product Test SubAF ShdPscOp11	Comment	
2									
3	**Test Properties**						Subordinate application function Shading presence operation 11		
4	Test Name		**SubAF ShdPscOp11**				Manual action to be taken during test execution with TsNet using BACnet		
5	Test last modified		30.10.2013 <-				Comments into QC		
6	Author Test		speckiem <-						
7	Test Version		1.00 ++ +						
8	Tester		Michael Speckie <-						
9	Type of Automation		Manual Action	Manual action to be taken during test executi					
10	CC Location of test definition		Enter location for archiving file in Clear case						
12	Type of test		PT	Product Test					
13	Source of Test Object		Pre-VVS	Preliminary valid version set					
14									
15	**TestObject**		Name	Version	Description				
16	SubAF		**ShdPscOp11**	3.0005	Shading presence operation 11				
17									
18	Source Library		TRA Dev	32.1					
19									
20	**Test Environment**								
21	**Quality Center**								
22	QC Path		Define complete QC path here						
23	QC Testname		Define QC testname						
24									
25	**Automation Device**								
26	BACnet-Port		BAC0						
27	Device-No	Name	IP-Addr	Type		Comment			
28	50	Room	192.168.0.50	PXC3-75A					
29									
30	**Test-PC**								
31	Working Dir Testscripts		D:_projects\SysOne\CenFctV1.2\Test\Scripts						
32	Working Dir Results		D:_projects\SysOne\CenFctV1.2\Test\Results						
33	S7-Project		S7Projectnamexxx						
34	S7-Project Location		D:\						
35									
36	Log with Wireshark								
37	Location TShark		c:\program files\wireshark\tshark.exe						
38	Tshark Interface		2						
39	Tshark Filter		-f "udp port 47808"						
40									
41	Location tsnet		C:\Program Files (x86)\Siemens\tsNET Toolset\tsNET						
42									
43	Version Test Template		V0.6	M.Speckien					
44	Version Test Script Gen		2.3.9493.0	W.Müller					
45	Version Tsnet		2.3.9493.0	W.Müller					

Figure 4–1: Config Sheet

Field "Comment" is just for information, is not copied to QC

Test Properties
Fill in test properties here...

Test Name and test Description are shown after the test Object has been entered (Info)
Test Last Modified : <-- enters today's date (info)
Author Test : <-- enters actual username. Name must match to QC login for QC import (QC Test)

Test Version is the version of the test specification : ++ and + increment the version
Tester is the actual tester, when not used with QC (Info)
Type of automation: describes if any maniual intervention during test execution is required.
CC file is the location, where the this test specification is archived in Clear Quest (Info)
Test Details describe the test of the test object and are used for QC import
Type of test: describes what type of test is done
Source of test object : describes, where the test object is taken from

Test Object
Enter the main test object first.
Select whether an AF, a chart or something else is tested. Enter name, version and description of the main test object. Example : "Function" "BoostHeat" "0.5" "Overall Boost Heat Function" or "AF" "CenOpMod02" "0.0011" "Central Operation Mode 2"

Insert additional lines for each additional component required for the test, for example other AFs, test rooms or simulation programs

Use delete line and insert line command buttons to add / delete more test objects.

Test Environment
Describe the test environment.
- Enter Quality center path and testname.
  This is used for QC import and must match exactly with the QC test plan structure.
- Enter the list of all automation devices which are used for the test (see details below)
- Enter the test PC's data locations and program file locations (see details below)

Automation Devices
Enter data of all automation devices used in the test
- enter BACnet port for all devices
- for each device enter the following data (start with the device where the test object is running)
    - Device Instance Number
    - Device Name
    - Device IP Address
    - Device Type (for info)

make sure that all data is correct, as the data is used for online communication

Use delete line and insert line command buttons to add / delete more devices

Test PC
The Test PC shall have a working area to store temporarily the generated test scripts and test results
- Enter the working directrories for the testscripts and the results
- Enter the S7 project name and location for info

Log With Wireshark
Tick here, if you want to log the Ethernet traffic with wireshark
If so, enter the following data
- location of tshark.exe on your test PC (tshark is the background version of wireshark)
- tshark physical ethernet interface number (if you do not know, run "ipconfig" and count for your interface)
- filter for capturing ethernet traffic. The default filter captures only BACnet traffic on BAC0 (refer to wireshark help for more options)

Data is logged in the Results directrory

Location TsNet
Location of tsnet.exe on your test PC.
This is used for creating a "Main" test script running all test cases.

--> when finished please continue with EDE sheet

4.2 EDE Sheet, BACnet objects

EDE data is available directly from the ABT tool and from the XWP tool. (Export to EDE)
If you do not have an EDE list, run the tool GetEDE to obtain one from your test configuration online. The actual version of GetEDE only supports standard BACnet objects.

How to enter EDE DATA
- Go to the EDE sheet.
- Never delete lines 1 to 4, they are required
- If there is already old data in the list, delete all content from line 5 onwards
- now open an actual EDE list of your test configuration with excel, and copy all data from line 5 onwards
- paste this data into the EDE sheet from line 5 onwards
- If you have more devices involved, repeat this for each device
- optional for easier handling : delete all lines with objects that have nothing to do with your tests (file objects, notification class,...)
- look for device objects and verify that device instance and names match to the entries in the config sheet

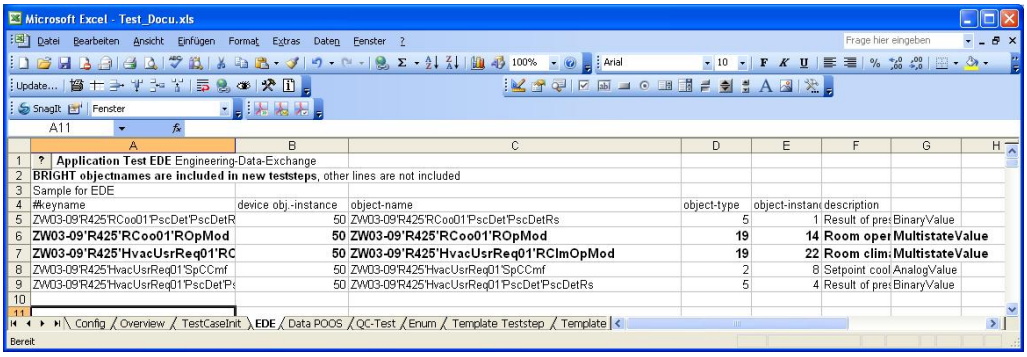


Figure 4–2: EDE Data Sheet

--> continue with sheet Overview

4.3 Overview Sheet, Create Test Steps

Go to sheet “Overview“.
Each test step gives an own test sheet.

- Always use the buttons New / Rename / Delete Test Sheets, never delete or insert a line.
- Step 1 is a predefined test step. It includes the call for all following test steps.
- The template already includes an empty test step (TestCaseInit) as an example, select it and click "Delete", before starting with your own test steps.

Test cases are defined

- New, without entries
  Then select Step 1 or another existing test step, and append a new test sheet by clicking "Insert empty test"
  Enter a name, for example "InitialiseInputs", then enter the number of inputs and outputs required (the number can later be changed)
  → a new empty test step with the name given above is created.

- New, containing BACnet objects from the EDE list
  Go to EDE, select the BACnet objects which are required for this test case, and format them **BRIGHT**.
  Go to Overview
  Then select Step 1 or another existing test step, and append a new test sheet by clicking "Insert from EDE"
  Enter a name, for example "InitialiseInputs"
  → a new test case with the name given above is created. This test case already includes the BACnet objects, which have been formatted **BRIGHT** in the EDE list

- Copied from an existing test case in this TsNet file
  Select an existing test case, and click "Copy and Paste"
  Enter an new name
  → the selected test sheet is copied with the new name

- Copied from an existing test case from another TsNet file
  Then select Step 1 or another existing test step, and append a new test sheet by clicking "From TsNet file”. You are prompted to enter a neme, then to open a file, and to enter the name of the existing test case.
  → The defined test case from the other TsNet file is copied behind the selected test sheet. If required, the sheet is converted into the new format.

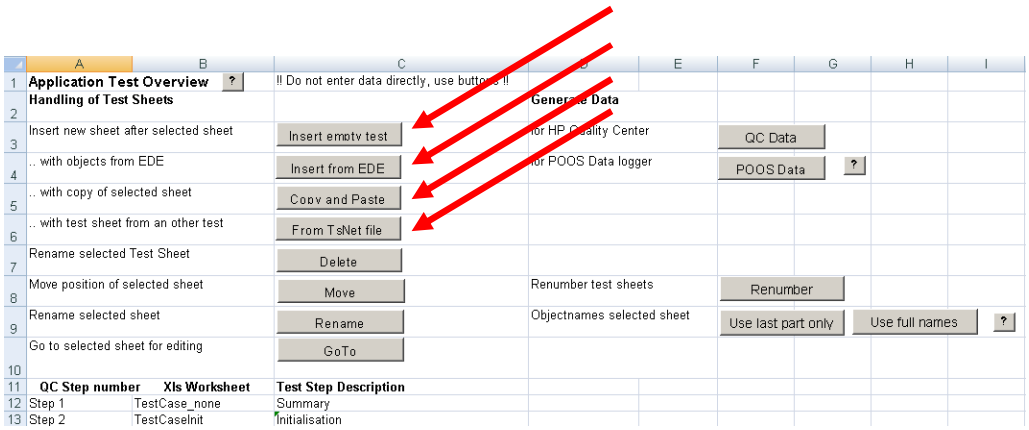


Figure 4–3: Overview Sheet

- Continue with creating further Test Steps.
- For deleting a test step, click "Delete Selected Test Sheet".
- You can also move a test step to another position or rename a test step
- Please renumber the test steps before generating Quality center data.

--> Next Step : Define your tests by writing them into the test step worksheets

4.4 TestCase sheet, Define Tests

Test Steps are defined in the sheet TestCaseXXX (with the name entered before). After creation of the TestCase sheet, it is either empty, or it is filled with BACnet objects from the EDE list, or it is filled with data from an existing test step.

In : The test tool gives input to the device, i.e writes an output object
Out : the test tool reads values from the device, i.e reads from an value object

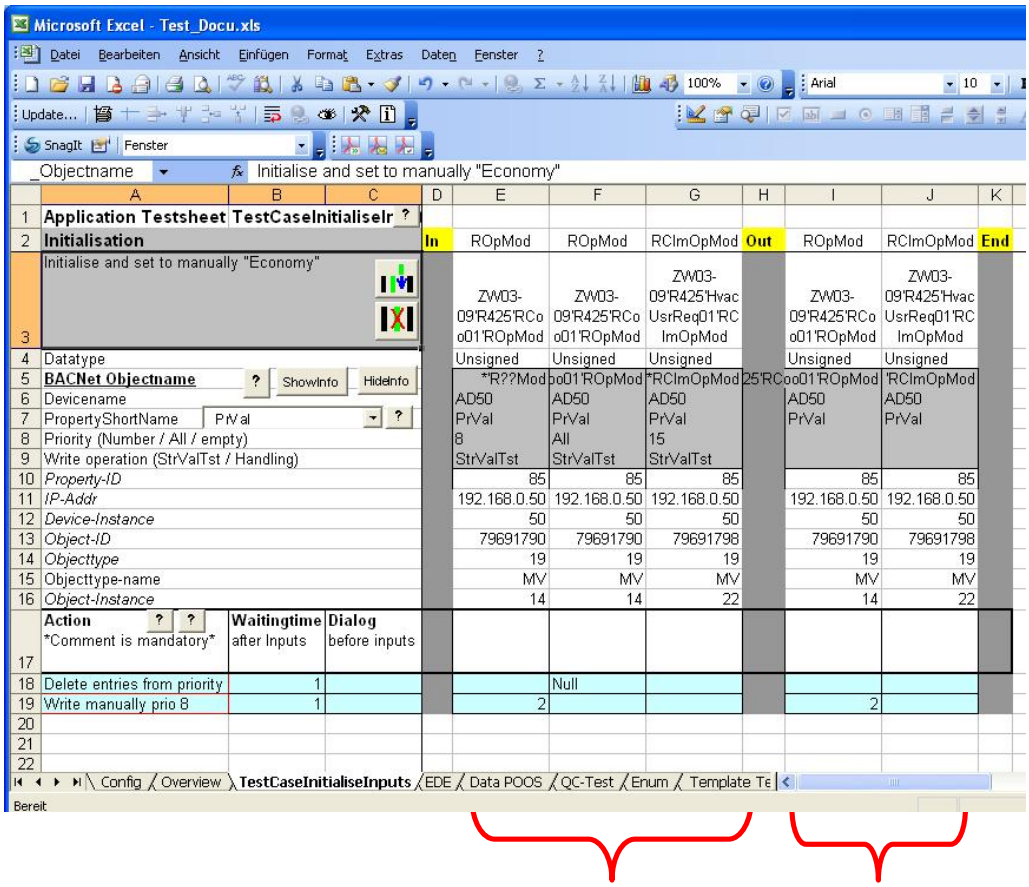


Figure 4–4: Definition of Test Steps

Please enter Summary (Line 2) and Description (Line 3) for the complete Test Step. The data is used for Quality Center.

4.4.1 Define BACnet Objects and Properties

The BACnet objects and properties are entered in columns.

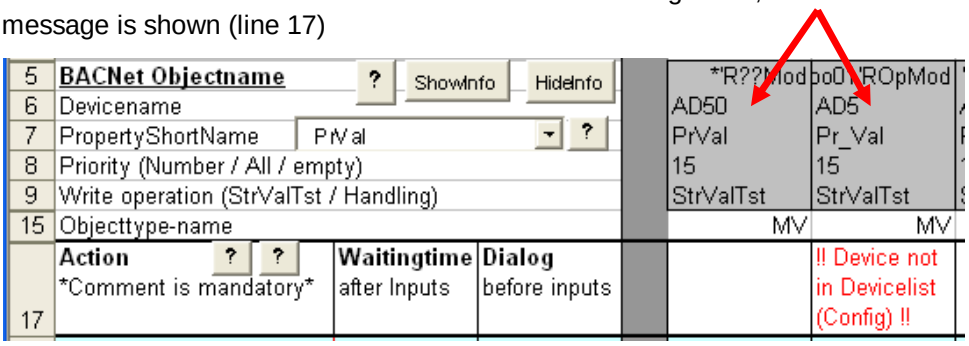
Objects are selected by marking one of the lines 5..9
Then an object can be deleted by clicking the symbol 
or an object can be copied and inserted by clicking the symbol  .

When writing to / reading from different properties or different priorities of the same object, then copy / paste the object, so that you have one column for each property and each priority.

Objects can be used for reading and / or writing

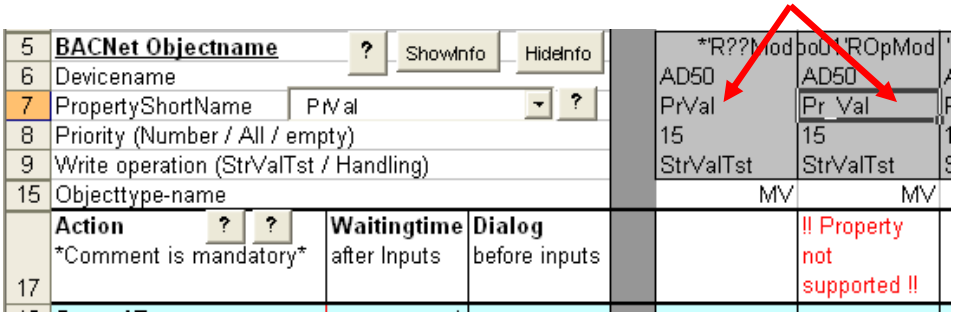
An object is defined by

- entering the device name (line 6)
  The device must match to one of the devices in the config sheet, otherwise an error message is shown (line 17)

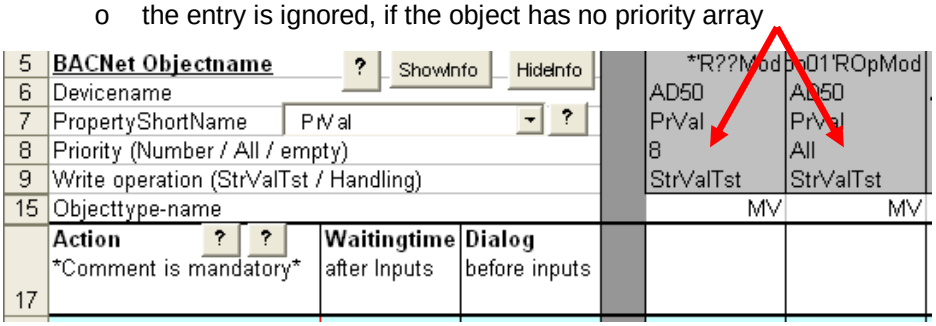


- entering the property short name (line 7)
  - o enter property short name (mostly PrVal) or
  - o select a name from the dropdown list and copy / paste it.

  If the short name is incorrect or the property is not supported, then an error message is shown



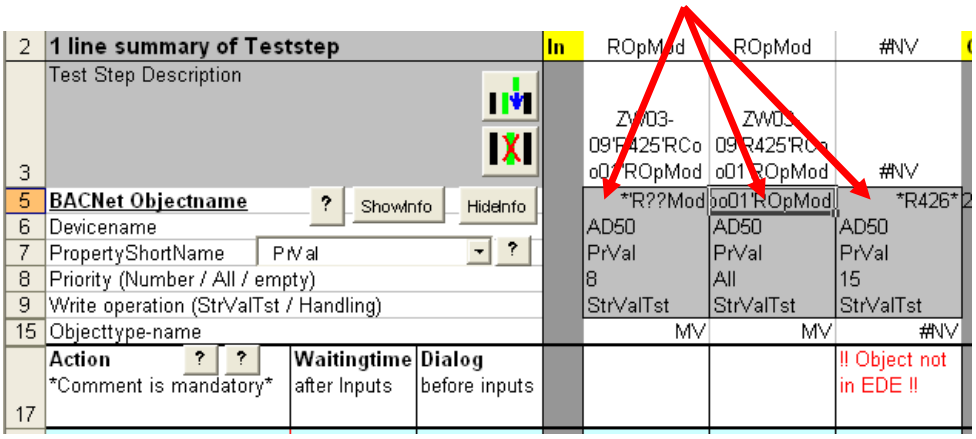
- entering the priority to be written to (line 8)
  - o as a number, if the object has a priority array or
  - o as the keyword "All", if all priorities shall be written with the same value
  - o the entry is ignored, if the object has no priority array



- Entering the write operation
  - o StrValTst : the BACnet acknowledge from the device is tested, if an error occurs (This is the normal case)
  - o Handling : the BACnet acknowledge is not tested (This is only used, when testing the device's error handling)

- entering the objectname (line 6)
  The objectname can be given as
    - o full name (for example ZW03-09'R425'RCoo01'ROpMod) or
    - o part of the name with wildcard (for example *'ROpMod)

  If the objectname matches exactly to an entry in the EDE list, the full name is displayed in line 3, the last part of the name is displayed in line 2 and the objecttype is displayed in line 15. Otherwise an error message is shown in line 17.



Clicking to "ShowInfo" displays additional lines showing additional BACnet object data and communication data. Clicking "HideInfo" hides this information again.

When the environment is changed, the devicelist and the EDE list can be updated. The defined BACnet objects remain correct, as long as the objectname, objecttype and the devicename are not changed.

When using objectnames with wildcards (*'ROpMod) instead of full names (ZW03-09'R425'RCoo01'ROpMod), it is possible to re-use tests, even if the first part of the objectname has changed (for example to ZW11'R803'RCoo01'ROpMod)

4.4.2 Define actions

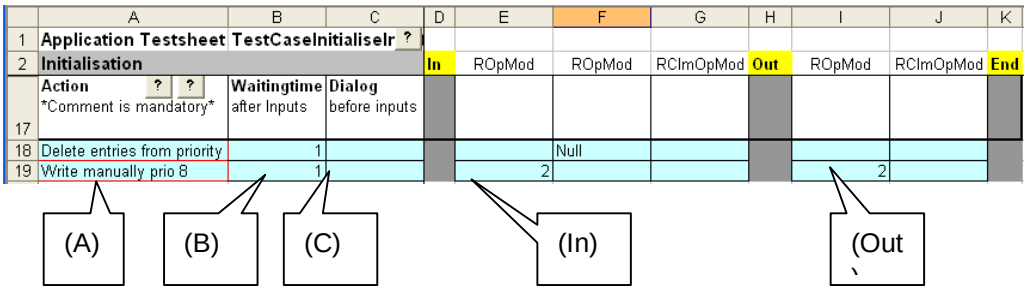


Figure 4–5: Actions

An action is a part of a test step. An action consists of

- a comment (A)
- Optional : A waiting time to wait, after the In values have been written (B)
- Optional : a dialogbox with the results “OK” or “false” (C)
- A set of values for input columns to be written to the device simultaneously (In)
- A set of values for output columns to be read and to be compared with reference values (Out)

Lines for actions can be added or deleted with Excel standard functions insert / delete line

Actions are defined with entries in the blue fields.

Comment: for documentation and for QC (required)
WaitingTime : Time in sec the test execution stops after all inputs (In) are done. Give the controller at least 1 sec to process the input data.
Dialog : The test stops, and a dialog field with the text pops up during the runtime of the tests before the inputs are processed. The dialog (for example “Verify that the blinds are moving up”) can be answered with “OK” or “Failed”.
Values for input columns: values which are written to device, object and property via BACnet.

Boolean: 0 or 1
Enumeration: 15
Real : 12.34
String : "String"
Bitstring : “0010”
Relinquish a priority : Null
Structures: Elements are separated with "Alt Return"
for example blinds command (Command, Height, Angle, P-Pos)
Type : Enum, Real, Real, Unsigned
7
30.0
50.0
0

Values for output columns: value the property is compared with, after reading from the device via BACnet.

Same format as above, additionally a range can be defined
Enumeration range : 3 .. 10
Real range : 12.0 .. 13.6
Range in structure, for example blinds present_value (height, Angle)
50.0
0 .. 100
Comparisons for real values are automatically done with a tolerance

Values left blank are not processed

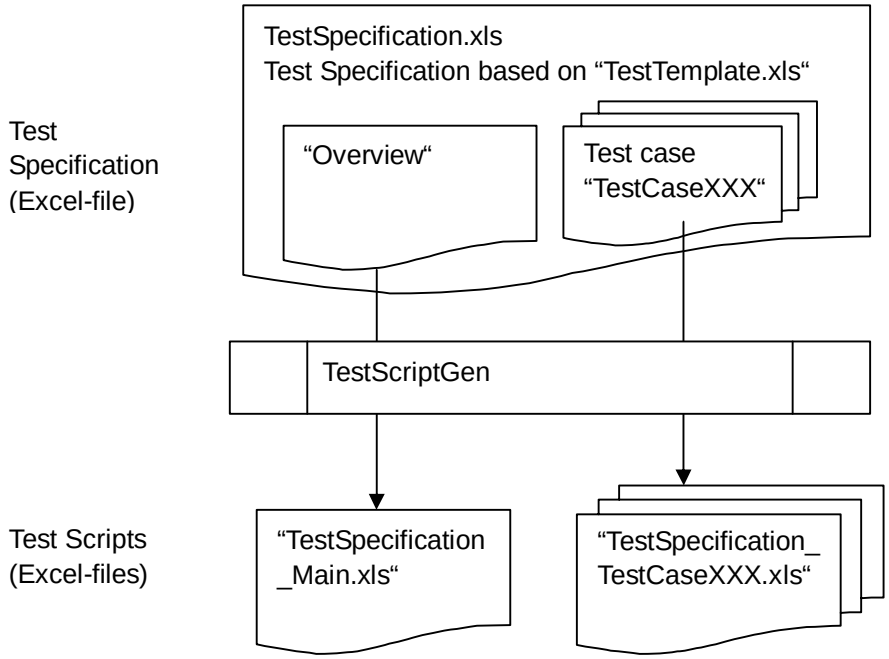
There must be at least one entry with a test result in each action.
--> otherwise the test step's result is undefined

5. Creating Test Scripts with TestScriptGen

5.1 Overview

Test Steps as defined in chapter 4.4 TestCase sheet, Define Tests cannot be interpreted directly by TsNet or Quality center. The tool TestScriptGen converts the Test Steps into Test Scripts for TsNet. Test Scripts for TsNet are Excel sheets, which mainly contain a line for each BACnet write or read command, that means one line per value of an action. One Test Case generates one Test Script.

Additionally a “Main“ test script is generated, running all Test Cases one after another.



5.2 File Structure on the test PC

Project Area : Location for project data, can also be on the project share
Test Specification
Filename : any name, for example TestSpecification.xls

Remark : for HQ application tests, the test specification shall be stored in clear case, and the location in clear case shall be entered in the config sheet

Working Area: Temporary Space for generated data, located on the test PC

TestScripts : Location for test script files generated from TestScriptGen
The location is defined in the “Config“ sheet of the test specification as Working Dir Testscripts.

Filenames :
Test Specification filename + “_Main“,
for example TestSpecification_Main.xls
Test Specification filename + Test Case name

For example TestSpecification_TestCaseStep1.xls

Results : Location for logfiles from TsNet or results of generated data
The location is defined in the “Config“ sheet of the test specification as Working Dir Results.

Filenames:

Test Specification filename + Test Case name + "_Result.txt"
For the result file of a TsNet run (Summary)
Test Specification filename + Test Case name + "_Report.txt"
For the report file of a TsNet run (Details for each action)
Test Specification filename + "_Config.txt"
For a textfile with the content of the config sheet
Test Specification filename + "_QCScript.txt"
For the VB script for QC

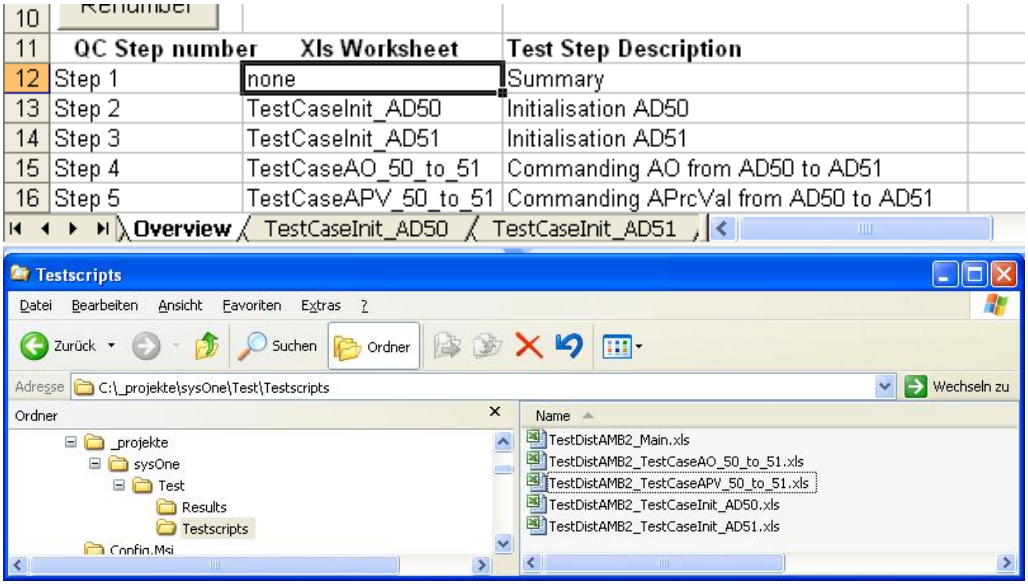


Figure 5–1: Example File Structure

5.3 Running TestScriptGen

Start TestScriptGen.exe.
Select TestScriptAPPL (Test scripts for application)

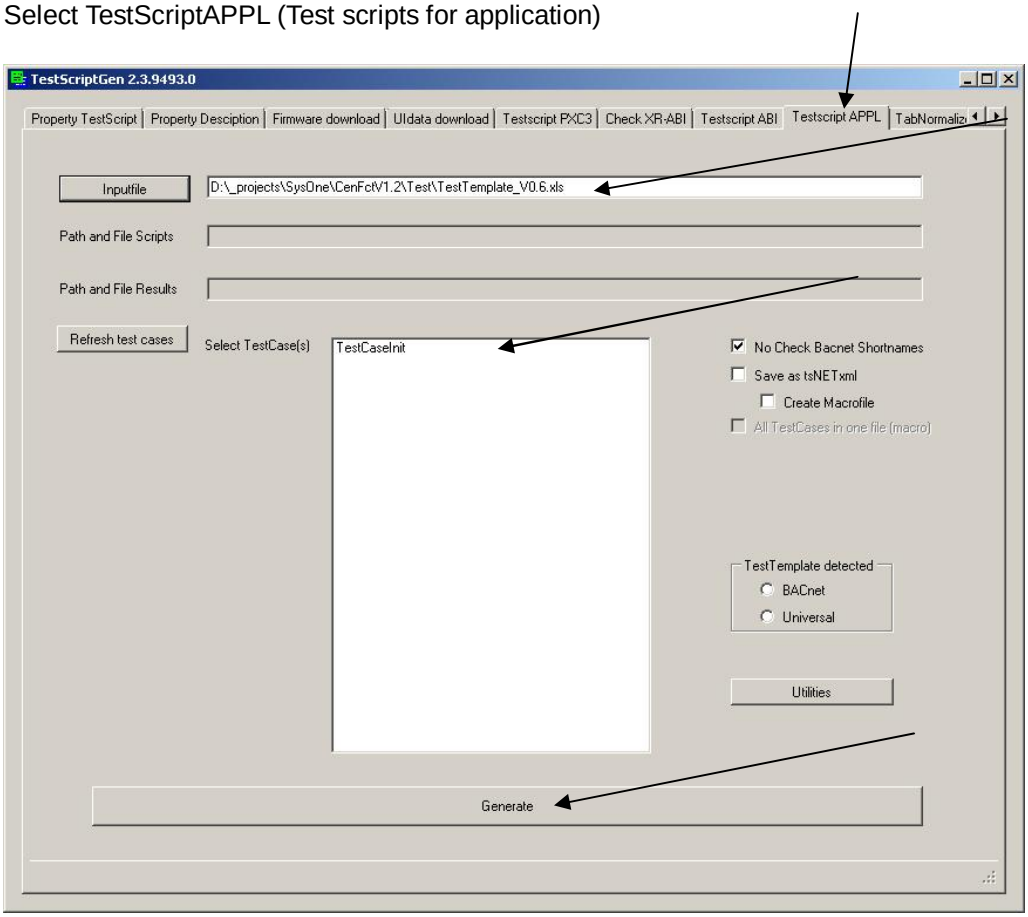


Figure 5–2: TestScriptGen

Then enter the filename of the test specification. The path for scripts and results are taken from the “Config“ sheet of the test specification, and are filled in automatically. Then select the Test Cases (usually all), and press “Generate“.

In case of a possible problem, a list of warnings is displayed. Usually you can ignore these warnings and continue..
The generated Test Scripts are located in the path displayed above.

6. Manual Run Test Cases with TsNet

6.1 Starting TsNet

It is recommended to start with manual testing of each individual Test Case after creating the test specification. So you can easily detect problems.

Start TsNet with TsNet.exe.

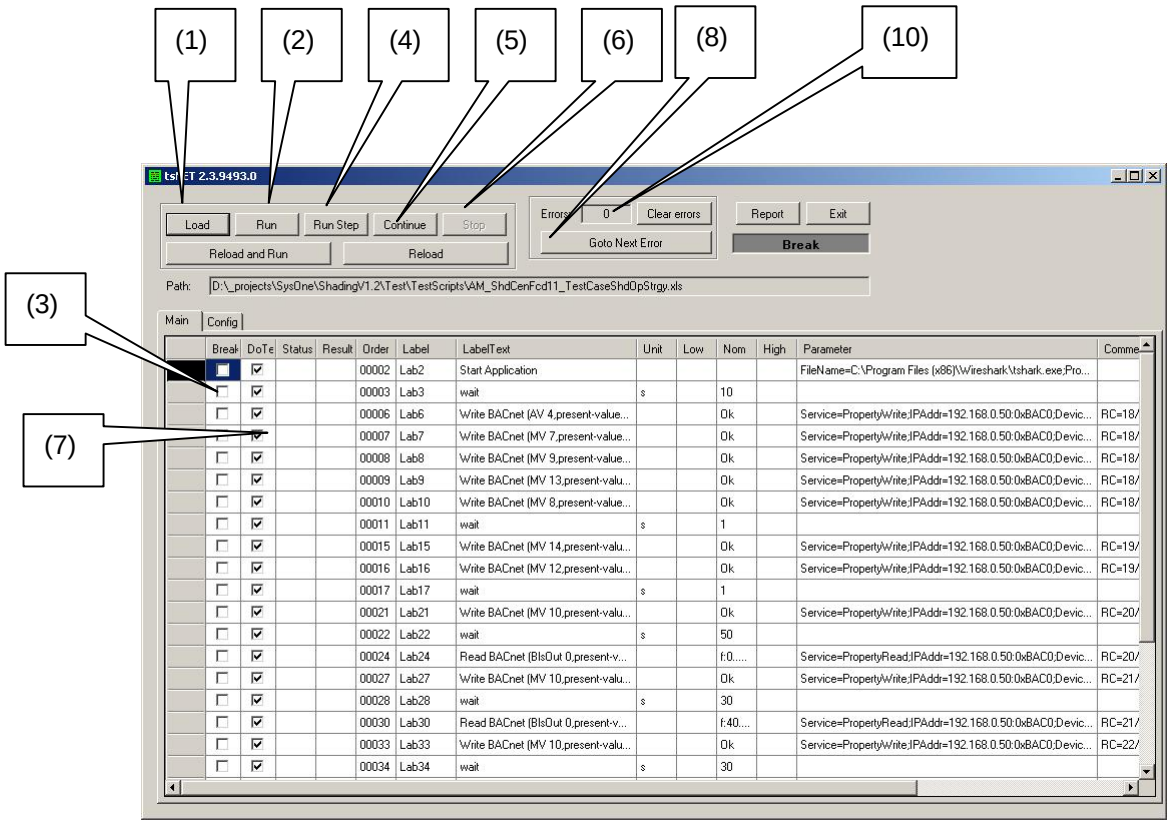


Figure 6–1: TsNet

- Load the testscript to run (1).
- The Test Script is loaded and displayed.

Then you can:

- Run the complete Test Case (2)
- Set a breakpoint (3)
- Run the selected line (4)
- Continue after a breakpoint or a stop (5)
- Stop the execution (6) completely or
- Skip a line (undo tick in 7)

6.2 Tracking Errors during Test Run

Errors are marked with Status = FAILED, and they are counted and displayed (10). The Result column shows the result, this means the BACnet value read, or the BACnet error information or other error information.

You find the next error with "Goto Next Error" (8)

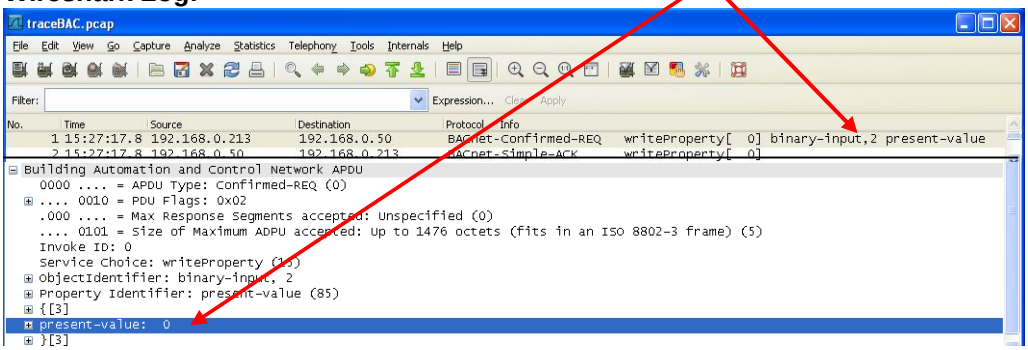
6.2.1 Tracking BACnet issues

Result = BACnet error or other error, but result is not a value
You can either get information from the result error information, or you can get information from the wireshark log.
Look up the BACnet object in the line, column LabelText, and look for the object in the wireshark trace. You find details about the value written / read via BACnet, and probably you find error information, for example „Object not available“.

Test Run TsNet:

Break	DoTe	Status	Result	Order	Label	LabelText	Unit	Low	Nom
☐	☑	Done	*	00002	Lab2	Start Application			
☐	☑	Done	*	00003	Lab3	wait	s		10
☐	☑	PAS...	Ok	00006	Lab6	Write BACnet (BI 2,present-value=...			Ok

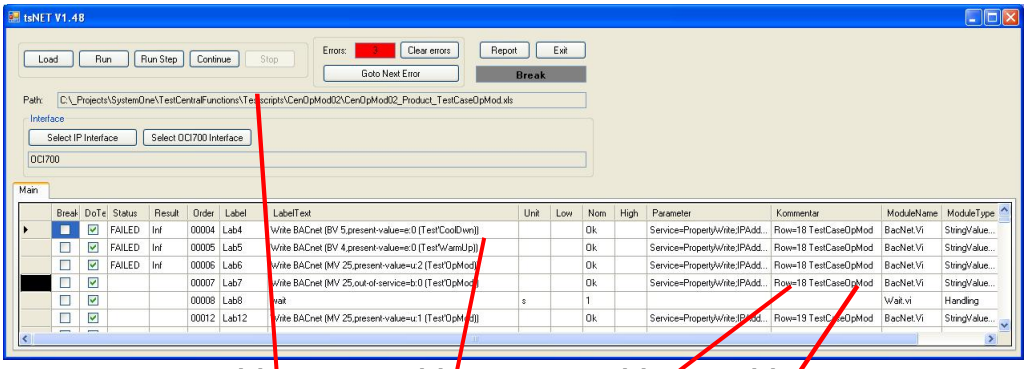
Wireshark Log:



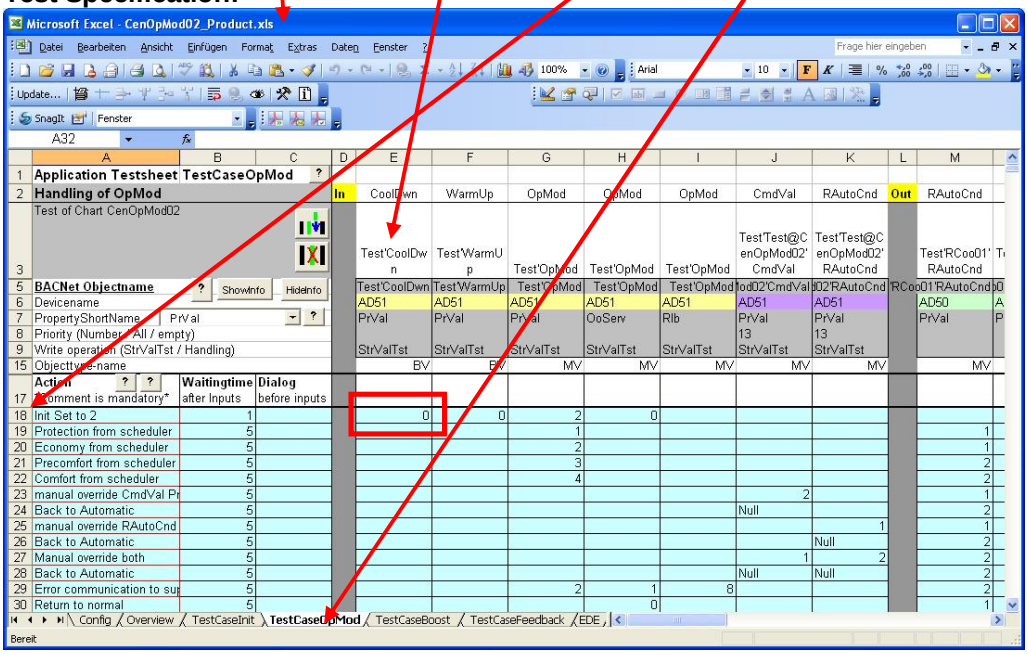
6.2.2 TsNet for tracking value problems

Result is a value, but the value is incorrect
A common failure case is, that the value read (TsNet column Result) does not match to the value expected (TsNet column Nom) or to the range expected (TsNet column Low / High).
See Path for the file with the test specification (1). Open the test specification.
See Kommentar (2) for the test case, and select the test case in the test specification.
See the object in LabelText (3), and find the same object in line 3 of the test sheet. This defines the column in the test sheet.
See Kommentar (4) for the line, and find the line in the test sheet.

TsNet Run



Test Specification:



In the TsNet dialog you can also make temporary changes of your test lines for analysing problems.

TsNet can show a detailled report, which can be saved and stored in your project.

7. Manual Run Main Tests with TsNet

This is reasonable for regression tests, or for tests where you do not expect failures.
The main test contains one line per test step.
When the main test is run, you run all test cases one after the other.

Close any TsNet instances before running main tests.
Then open TsNet.exe and load the Main Test Script "_Main.xls", then run it.

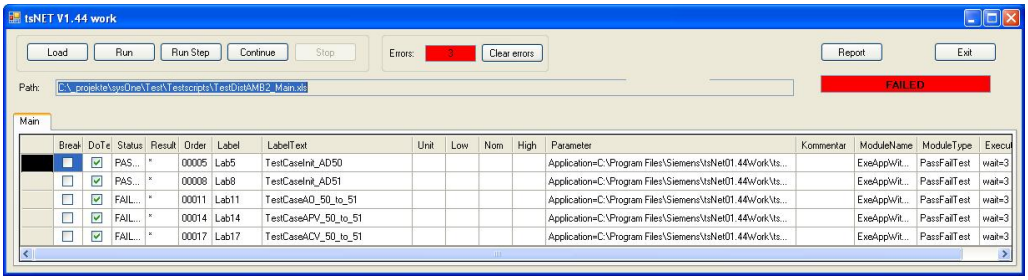


Figure 7–1: Running the Main Test Script

While running main tests, TsNet opens another instance of TsNet running one test case after an other.

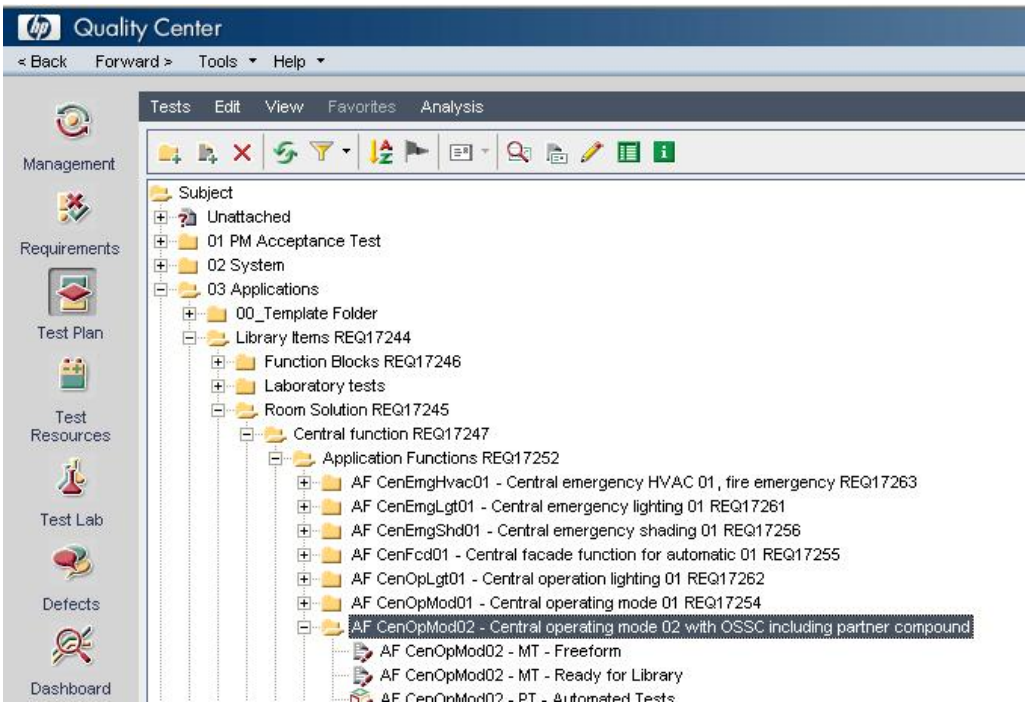
It does not require any user attention during running, however, you only see the summary, and you cannot analyse failures in detail. The test results and the wireshark trace is temporarily logged to the result directrory on your test PC.

Then analyse the results in the result directrory in the "_Report.txt" file.

8. Import Test Specifications to Quality Center

8.1 Prepare Quality Center and Config Sheet

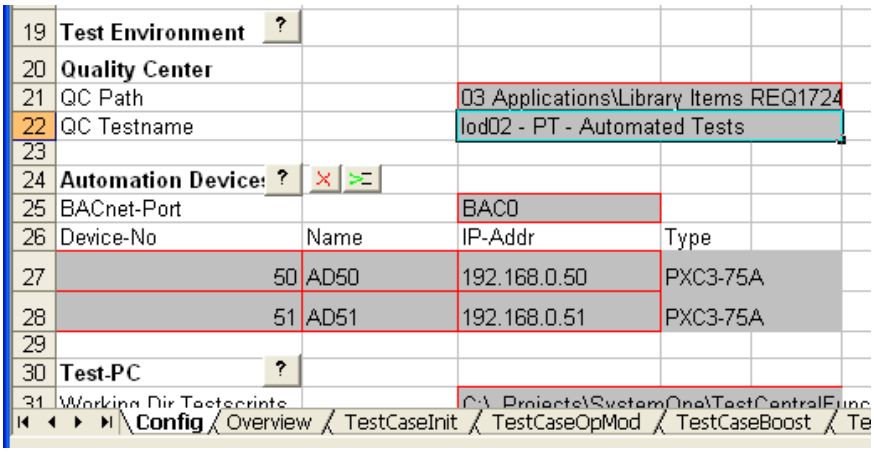
Open Quality Center, select Test Plan.
Build or verify the structure, where the test shall be located.



Follow the complete hierarchy, and copy the full path for the folder, where the test shall be located (Example "03 Applications\Library Items REQ17244\Room Solution REQ17245\Central function REQ17247\Application Functions REQ17252\AF CenOpMod02 - Central operating mode 02 with OSSC including partner compound"

Then enter or verify the path in the test specification, config sheet, Quality center/QC Path.

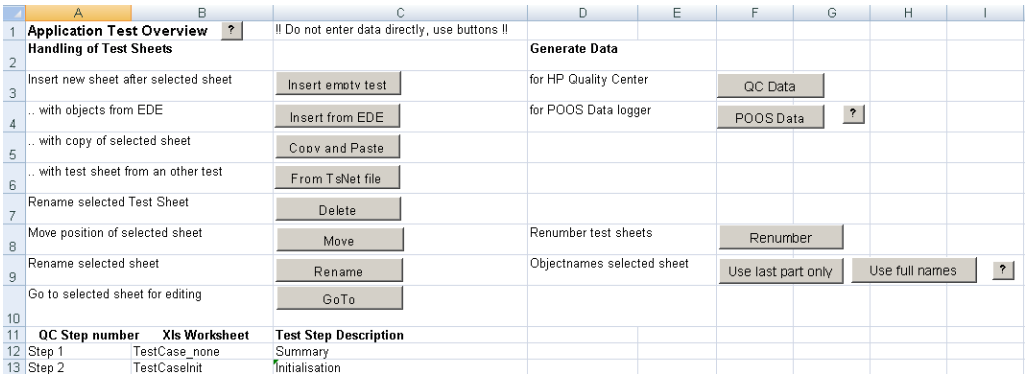
Then enter the test name in QualityCenter\QC Testname. Example "AF CenOpMod02 - PT - Automated Tests".



Verify that the name Author Test in the Config sheet matches to the user name in Quality Center.

8.2 Generate Data

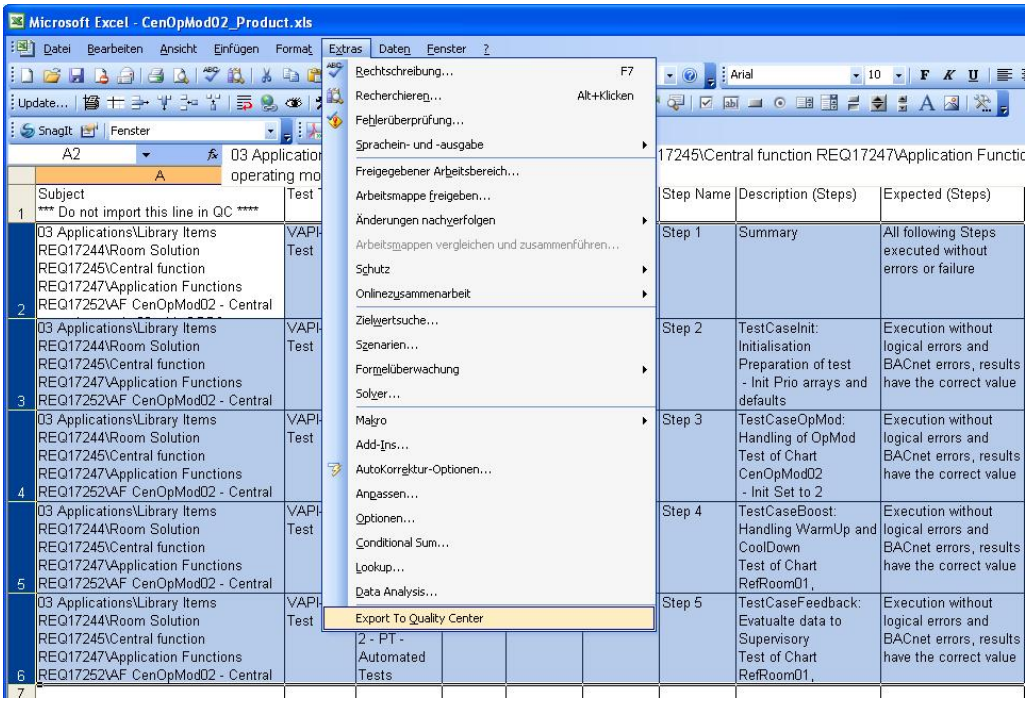
Go to Overview sheet in test specification, and generate the Quality center data by clicking "Generate QC Data"



A new (or updated) sheet "QC-Test" is created.

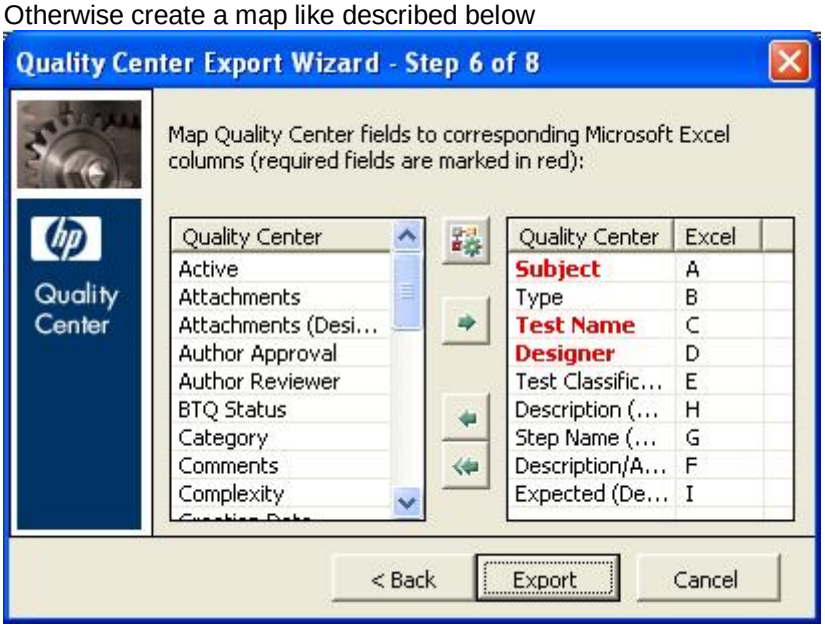
8.3 Export to Quality Center

Go to sheet "QC-Test", and select all lines except line 1
Select Extras → Export To Quality Center



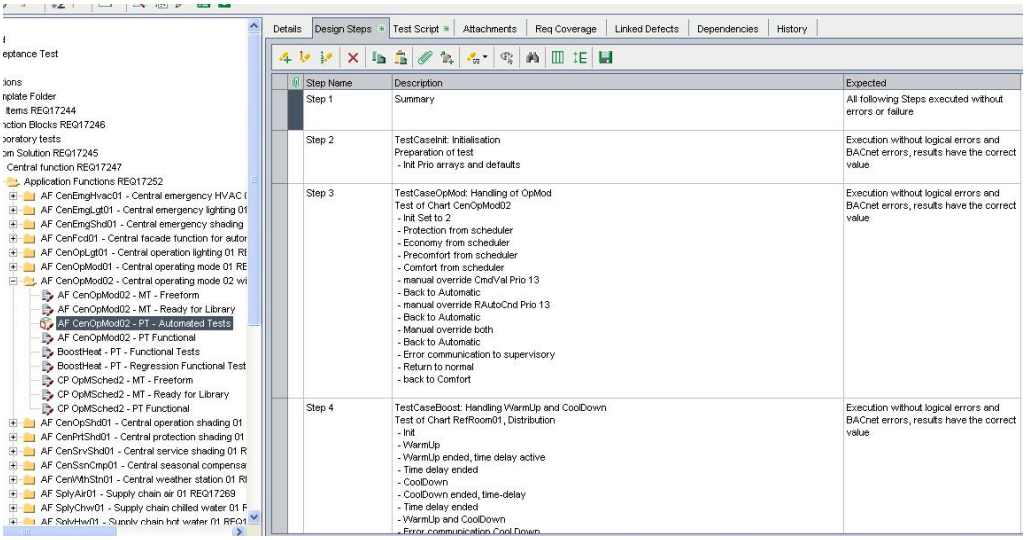
Follow the wizard, enter User Name and User Password, then enter Domain Name and Project Name.
Select type of data you want to export = Tests

Select the map "Tests", if available.
Otherwise create a map like described below



Then export the data.

Verify the result in Quality Center:
A new (or updated) test including test steps has been created, with all comments from the test specification.

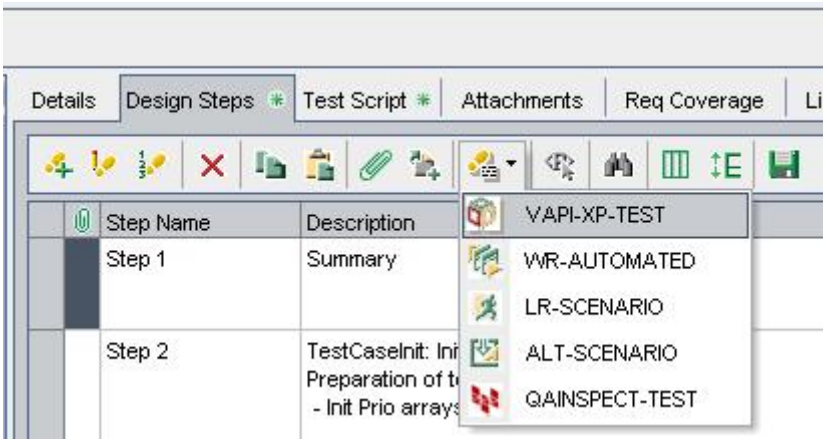


When the test shall only be run manually, no further work has to be done on Quality Center.
Test run and the entry of the test results are done manually by the tester
See next chapter for automatic tests.

9. Running Tests automatically with Quality Center

9.1 Import the VAPI XP test script

Go to Quality Center Test Steps, and add a VAPI-XP-TEST.



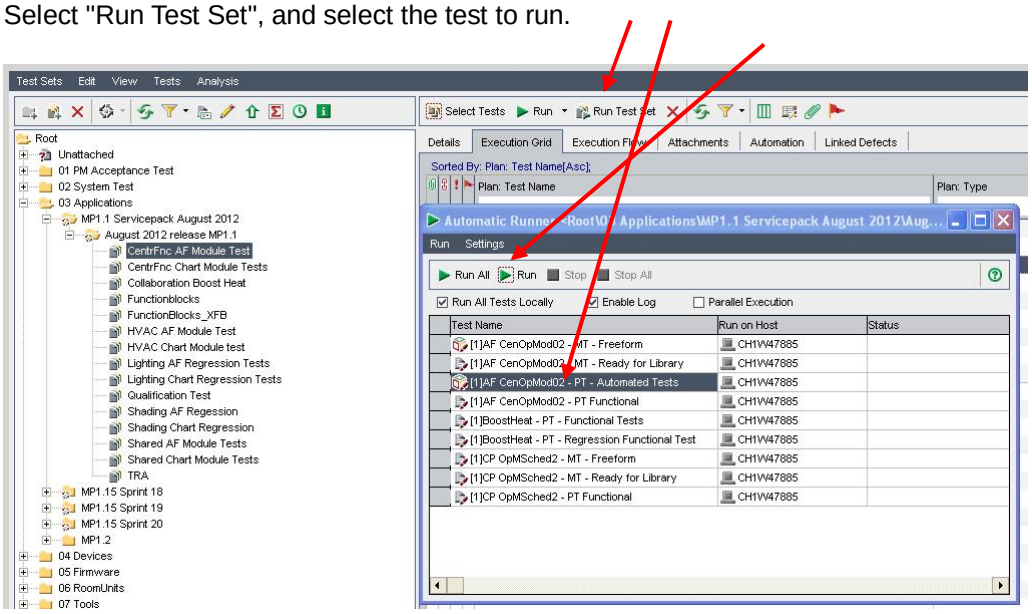
Follow the wizzard using default values (Script Language VBScript, Com/DCom Server Test).
This will create a default test script.
Open the Test Script, mark everything, and delete it.

With the Windows Explorer, go to the result folder (location see test specification, Config sheet, Test-PC, Working Dir Results.
Open the file "_QCScript.txt" with a text editor, mark everything and copy the text to the clipboard (Ctrl-C).

Now go back to Quality Center, Test Script.
The text should now be empty.
Paste the text from the clipboard (Ctrl-V).
Save the script.

9.2 Running Automated Tests

Go to Quality Center Test Lab, and go to the test set you want to run.
Select "Run Test Set", and select the test to run.



Select "Run All Tests Locally" in case you are running Quality center on your test PC, otherwise enter the host name of your test PC. Remark : Often the IT department does not allow DCOM to run scripts on an other PC, so that an other host probably does not work.

Run the test.

Quality Center then downloads the test script to your test PC, and starts the Main Test in TsNet.
Then TsNet starts a new TsNet session for each Test Case. When a test case is finished, the results are entered in the Quality Center test run automatically.
In case of an error, the Main logfile and the main result file are saved as a Quality center attachment.

10. Tracing or Trending values with POOS

10.1 Introduction

Tracing values is useful for dynamic tests, for example closed loop tests

Tracing or trending values during testing can be done with a Building Management Station or by using the POOS tool.
POOS has the following advantages:

- no installation and engineering for the building management station required
- no trend engineering on the controller
- tracing of structured values (for example light or blinds) possible
- tracing other BACnet properties than PrVal possible
- fast polling rate (1 sec) possible
- simple analysis with Excel

POOS is just a data logger for BACnet values, and has no visualization. Visualization is done with Excel after logging is finished.

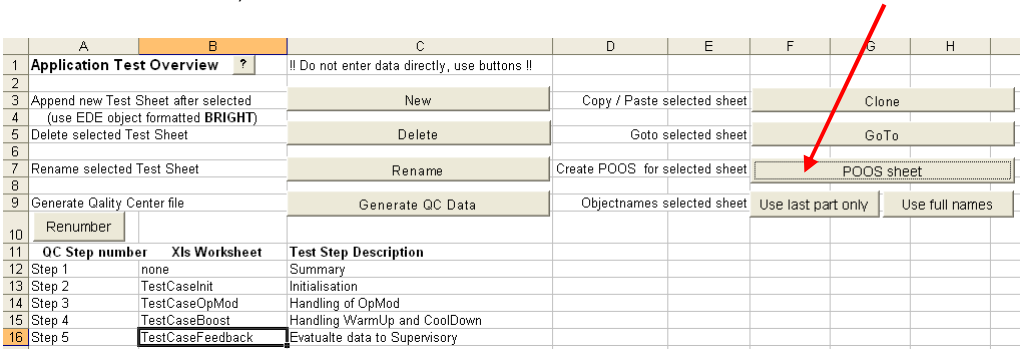
POOS works with a predefined folder for data, located in the tool folder.
It is responsibility of the user to move the log files to the project folder.
POOS has an Excel file for the configuration of the values to be logged.

POOS has no direct connection with TsNet, but they can run both in parallel. The test specification allows creating POOS configuration data from the test cases.

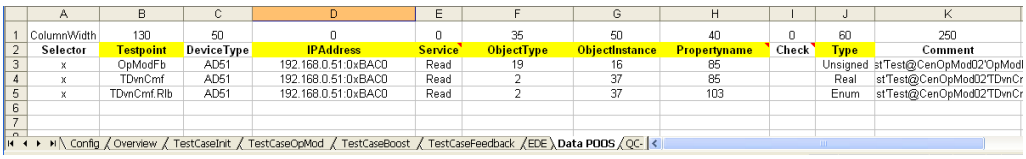
10.2 Creating POOS Configuration data

Within a test specification, the output data of a test case can be converted into POOS configuration data, so that no additional configuration for POOS is required.

Open the test specification, and go to the Overview sheet.
Select the test case, and then click POOS sheet.



A new sheet "Data POOS" is created with the POOS configuration data. Each output variable from the selected Test Case is converted into a line in the POOS configuration.

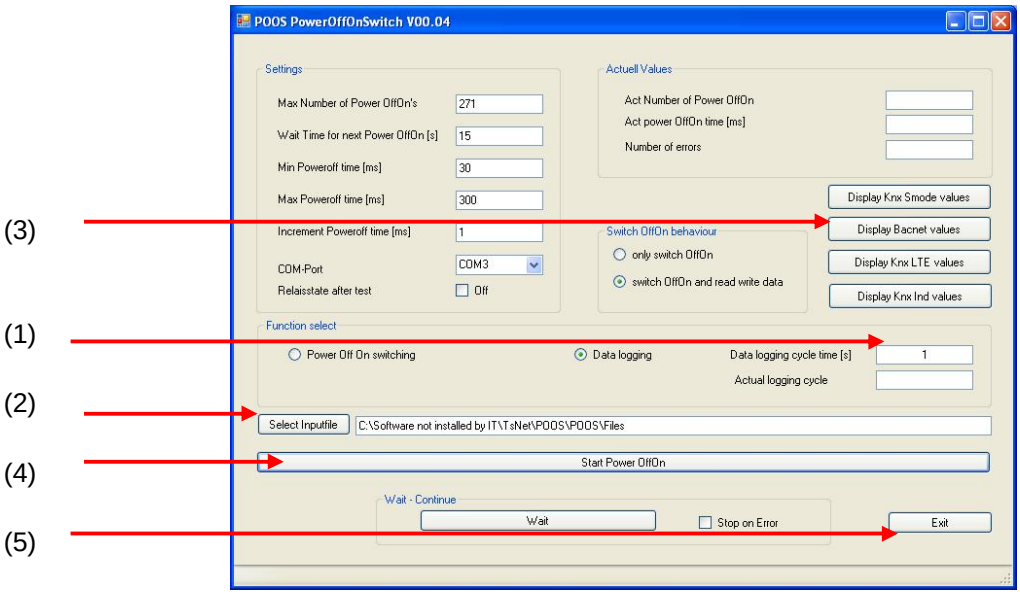


Now, go to the POOS program directory, and open the POOS configuration file "\Files\Test.xls".

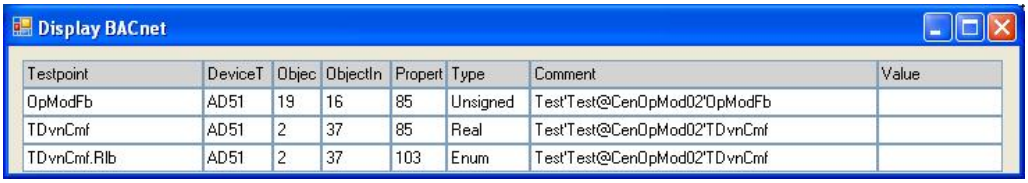
Save it into your project directrory with a new name, for example CenOpMod02_Product_TestCaseFeedback_POOSConfig.xls. Then open the sheet "BACnet", select and delete all data in the sheet.
Go back to the test specification, open the sheet "Data POOS", select all, and copy to clipboard (Ctrl-C).
Go back to the POOS configuration, sheet BACnet, and paste the clipboard data (Ctrl-V).
Save and exit.

10.3 Run POOS for data logging

Start POOS as POOS.exe.
Enter the logging cycle time (1).



Select the configuration file (2) , for example CenOpMod02_Product_TestCaseFeedback_POOSConfig.xls.
Click on Display BACnet Values (3), if you want to watch the variables on the screen during logging.



Testpoint	DeviceT	Objec	ObjectIn	Propert	Type	Comment	Value
OpModFb	AD51	19	16	85	Unsigned	Test'Test@CenOpMod02'OpModFb	
TDvnCmf	AD51	2	37	85	Real	Test'Test@CenOpMod02'TDvnCmf	
TDvnCmf.Rlb	AD51	2	37	103	Enum	Test'Test@CenOpMod02'TDvnCmf	

Then click Start Power OffOn (4) to start logging.

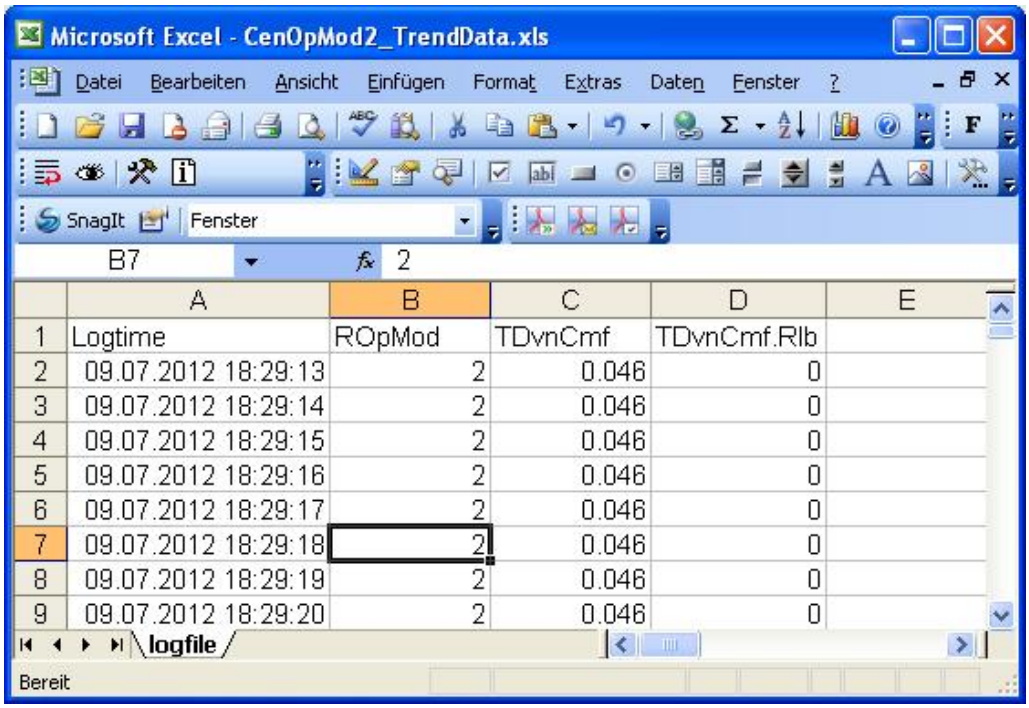
The logged data is always stored in the in the POOS program directrory with the filename "\Files\logfile.txt" as textfile with <Tab> as a separator.

After logging is finished, click Stop Power OffOn (4).

Then click Exit (5), and wait until the POOS shell is closed. This may take some time, as the logging cycle must be finished and all values must be written to the file.

!! The filename and file location is fix. A new start of POOS overwrites already logged data without warning. It iy your responsibility to save the logged data into the project directrory !!

Open the logfile "\Files\logfile.txt" with Excel and save as Excel file to your project directrory, for example as CenOpMod2_TrendData.xls.



11. Get the EDE list with GetEDE

GetEDE shall be used only, if no EDE-Export from the engineering tool is available (for example 3rd party device).

GetEDE is possible, if you

- have online access via BACnet
- have only standard BACnet objects (other objects are ignored)

Start GetEDE.exe and select "Scan Network“
This scans the network for BACnet devices.

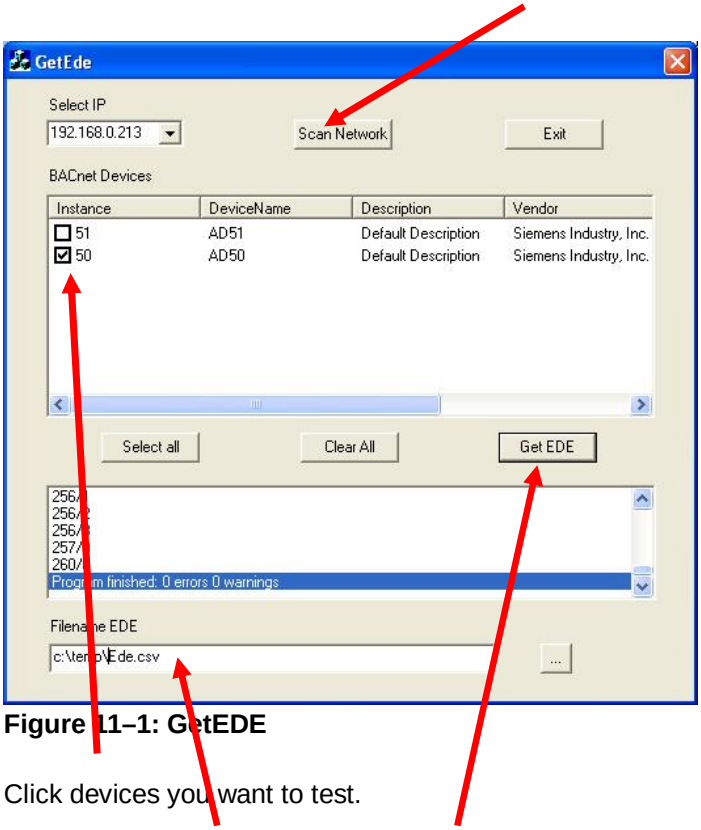


Figure 11–1: GetEDE

Click devices you want to test.

Enter path and filename and select GetEDE.

The EDE list is generated and saved in the csv file.