Title: **Requirements IPA Fatma Yilmaz**

Subject: **Requirements , Specificatioon**

This document is an extract of the document TsNet V2 Requirements. It contains the requirements which are relevant for the IPA.

Key Words: TsNet V2

Document Storage:	Local
Document Category:	ProjectRecord
Revision:	1
Revision Date:	2016-03-08
Document Status:	Final - without Approval
Author:	Michael Speckien, 5556
Department:	IC BT CPS R&D ZG CS SAP
Responsible:	Michael.Speckien@siemens.com
Company:	Siemens Schweiz AG, Building Technologies Division Control Products & Systems
Based on Template:	Workbook_Small; 4; 2014-11-05; Donat Hutter, 3531

Revision History

Rev	Date	Author	Remarks
1	08-Mrz-2016	Michael Speckien, 5556	Status = **Final - without Approval**
0.1	08-Mrz-2016	Michael Speckien, 5556	Status = **Working** - Extracted from document TsNet V2 Requirements

Table of Contents

1. Introduction

1.1 Purpose of the document

This document serves as requirement based specification for the IPA. It is an extract from the overall document TsNet V2 Requirements [1], which is still in status “working”

1.2 Scope, Field of application

The task for this IPA is a dedicated part of the development TsNet V2.

1.3 Document References

[1] TsNet V2 Requirements, Michael Speckien 2016

1.4 Definitions, Acronyms and Abbreviations, Conventions

1.4.1 Glossary

Term	Description

1.4.2 Conventions in this Document

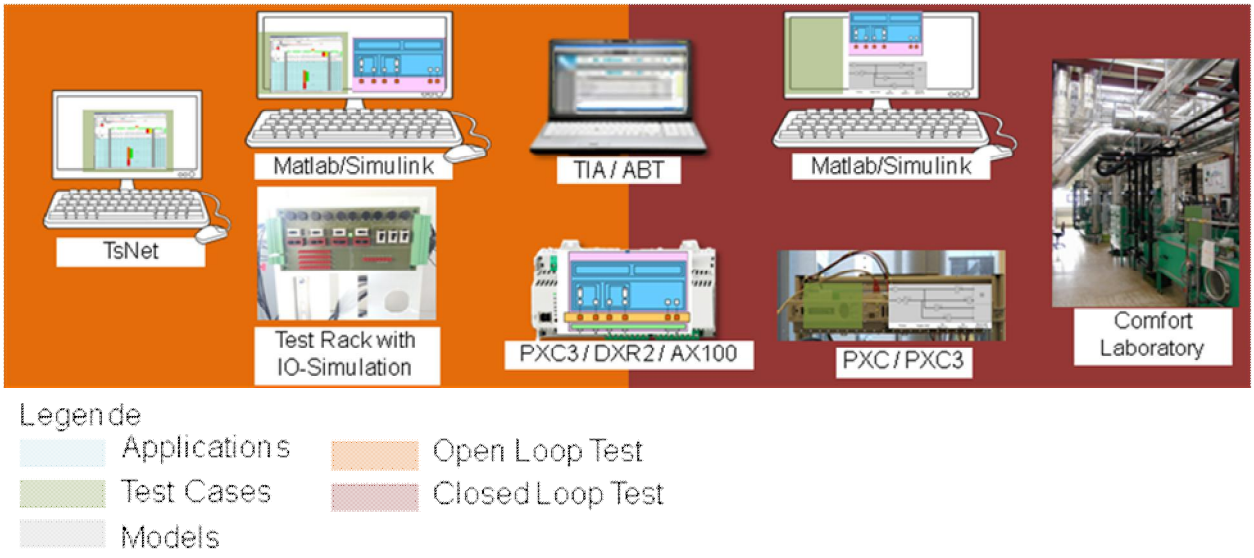
Issues which are interesting for understanding the complete TsNet V2 project, but which are not relevant for the IPA are labeled with “Not relevant for IPA”

1.5 Open Issues in this Document Version

No open issues, this document is valid for the IPA.

2. Actual situation

2.1 Overview TsNet



TsNet is a pc running software package consisting of a test specification part (Excel based template) and a runtime part (test operation and BACnet stack)
TsNet allows executing application tests with hardware-in-the-loop by triggering variables via BACnet, reading the controller's reaction via BACnet and comparing it with the expected results.
TsNet in application testing is usully combined with a test rack, allowing additional triggering via physical inputs and easy monitoring of physical output signals.
Actually TsNet is mainly used for open-loop-tests, however it can be combined with a room simulation on a controller.

3. Requirements

The actual version of TsNet should get some improvements to simplify the dayly use in application testing.
This chapter is an extract of [1]. It shows only requirements which are relevant for the IPA.

TsNetV2-0020	**Functional Test Specification (independent of the test environment)**
DESCR:	The test specification actually is partly depending on the ABT engineering data, especially datapoint names. So when changing ABT engineering data, it is required to rework the test specification. This increases the work required for regression tests. → It should be possible to change datapoint names and controller names without reworking the test specification
PRIO:	0
REF:	Chapter 4.5, 5.1.2
TEST:	Covered by the specification in 5.1. No specific test required

TsNetV2-0050	**Workflow orientation**
DESCR:	TsNet specification template does not support a defined workflow. So it is the user's task to maintain consistent data. → TsNet sheets shall include a change control and a status information. It shall indicate possibly inconsistent data and show the user how to get the data consistent.
PRIO:	0
REF:	Chapter 4.3, 5.1.2,
TEST:	Follow the described workflow in 5.1.2 and verify the status of the sheet and the GUI according to the description of the functions "Modify", "Check List", "Check Devices". Verify the data in the fileds "Status", "Info" and "Status Device"

TsNetV2-0060	**Programming guideline**
DESCR:	The template has been developed over a long period with different developers resulting in different programming styles and different user interfaces, for example limitations in copy/paste. → The template should be redesigned following programming guidelines.
PRIO:	0
REF:	Chapter 4.2, 4.4
TEST:	Non-functional requirement, not to be tested

TsNetV2-0120	**Support of MS/TP**
DESCR:	The template shall support MSTP via IP/MSTP router
PRIO:	0

REF:	Chapter 5.1.1, 5.1.5
TEST:	Covered by the specification in 5.1. No specific test required

4. Design proposal for TsNet Test Specification Template

Fulfilling the requirements, a complete redesign of the TsNet specification template is required.

4.1 Location and file structure

The tool TsNet runtime is delivered with an installation routine.
The test specification template can be copied to any location on the disk or a server and shall be renamed to a project specific name like Test_AF_CenOpMod11.xlsm.
The complete data for running the tests is stored in the different sheets of the excel file.
*: Worksheets marked with * are not relevant for the IPA.

Structure of the Excel file:

Excel file	Worksheets	Description
Excel_file.xlsm	Config*	Basic configuration of the test
	Overview*	Main sheet for navigating, generating, executing and documenting tests
	Workflow*	Documentation of the TsNet V2 workflow
	Spec-Devices	**List of the BACnet devices used in the test This sheet and its functions shall be created in the IPA**
	Spec-Objects*	List of all BACnet objects used in the test
	For each test step	
	Step_<name>_<src>*	Specification of the test step source, can be in the form of a text (_TXT), or a vertical table (_VT)
	Step_<name>_VT*	Generated from <src> if <src> is TXT format. Required before executing the test.
	Step_<name>_Trend*	Optional generated from VT: Trend view of the specification for documentation
	Step_<name>_Script*	Generated from VT. Required before executing the test Data for TsNet Runtime
	Step_<name>_Result*	Generated after test execution from TsNet Runtime
	Imported data from external tools	
	EDE	EDE list from engineering tool
	Data for external tools	
	QCdata*	Export data for QC (HP ALM)
	POOSdata*	Specification file for data logging tool POOS
	Templates used for creating empty sheets for test steps	
	Template Step_TXT*	Empty template for test spec as text
	Template Step_VT*	Empty template for test spec as vertical table
	Template Step_Trend*	Empty template for trend view
	Template Step_Script*	Template for test script
	Template Step_Result*	Template for result
	Templates used for generating external data	
	Template QC*	Template for QC
	Template POOS*	Template for POOS
	Internal sheets	
	Enum*	BACnet enumerations for objects, properties etc
	Help*	Sheet with all help references
	Choices	Non-BACnet enumerations for the user interface

		SDU*		Shortname and description from “All_Translatable_HQ_Texts_ger_20151201.xls”

4.2 User Interface Rules

General improvements to the User Interface

- Follow the standard convention for input fields:
  white = user input
  grey = read-only

QC Testname	011 AF CenOpMod11 - MT
QC Login	speckiem

- Offer navigation features on each sheet
  Navigation to overview sheet and to related sheets

Step No	Step name
1	Init
2	WoBoost
3	Boost

→ link to sheet with Boost

- Each sheet shall be split up into a functional area (grey) and an input area (white)

c:\Programme (x86)\Siemens\TsNet		TsNet Test-PC	192.168.0.233
Alias Controller	**Controllername**	**Comment**	**IP-Addr**
Supervisory	Sprv100	Supervisory functions	192.168.0.100
Central	Cen050	Central functions	192.168.0.50

- Functional area:
  Show information required
  Buttons / links for navigation
  Buttons for calling functions
  Read-only
  Adding cells, columns, lines not possible

- Input area:
  Lists with user data
  Usually horizontal orientation
  Standard Excel function for add lines, filter, sort, fill
  Some lists also allow adding columns

- Some functions allow operation on one or multiple lines or columns.
  Functions for select:
  Selected items are marked with a ✓
  Selecting an item is possible by selecting the line / column with the item to select or by selecting a cell in the line / column of the item to select.
  - Select a line / column and click the ✓ button.
    The selected item is added to the selection.
  - Select a line / column and click the x button.
    The selected item is removed from the selection

- Select multiple lines / columns and click the ✓ button.
  All selected items are added to the selection.
- Select multiple lines / columns and click the x button.
  All selected items are removed from the selection
- Select one or multiple lines / columns and call a function for selected items like “Delete test step”
  If the function called allows multiple selections(like “Run selected test steps”), the previous selection is replaced by all of the selected items.
  If the function allows only 1-n selection like “Insert test step after selected”, the previous selection is replaced by the first of the selected items.

  Proposal:

Select ✓ x	Alias Controller
✓	Supervisory
	Central
	Room

- Help
  Use pdf with link to a specific page.
  Advantages:
  - use graphics, screenshots etc
  - create help from word document automatically
  - Help function once on each sheet

  For easier maintenance, a Help sheet is used with the pdf file and the page for each sheet.

4.3 Workflow orientation (Req 50)

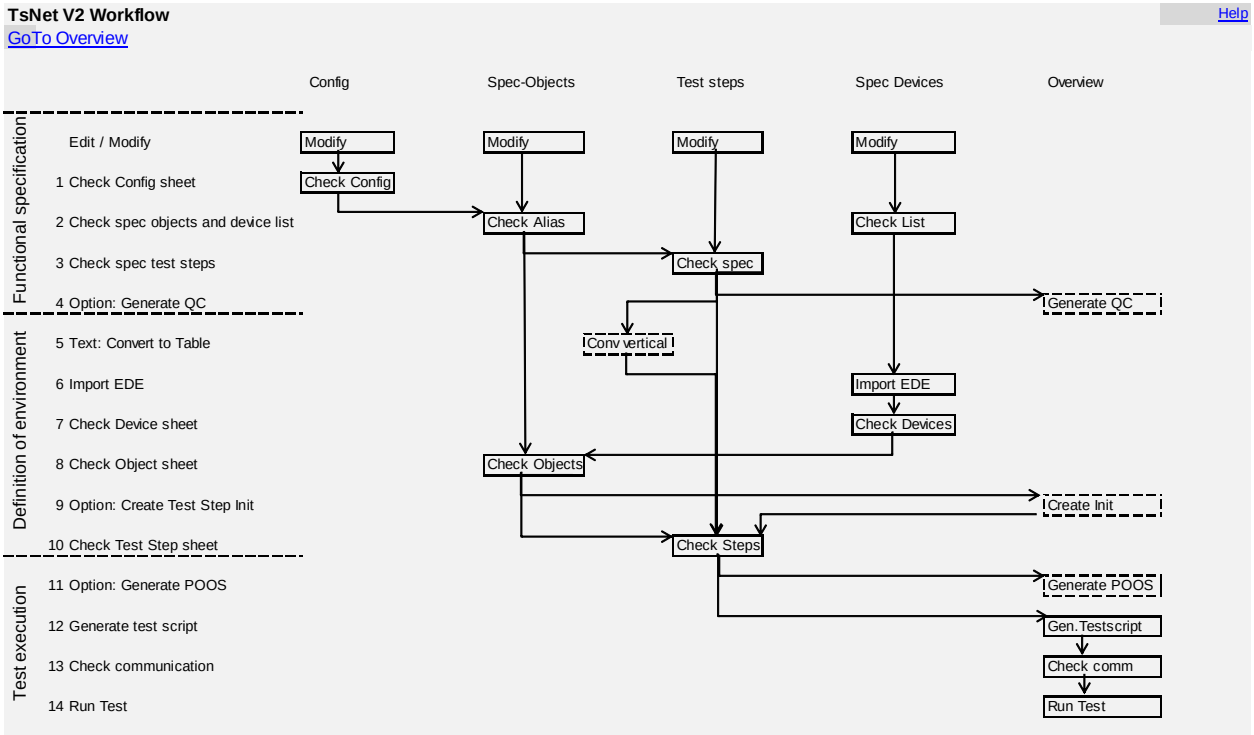
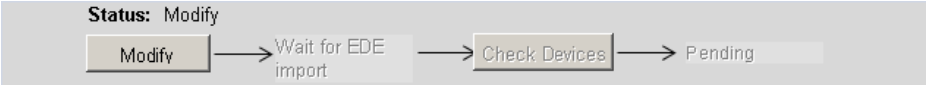


Figure 4–1: Main workflow

The workflow shown is the “straight forward” workflow.

To allow later changes without getting inconsistent, it is required to track changes.

Proposed solution:
Data input into a sheet is followed by a check. After the check has been successfully performed, the sheet is set to the “Checked” state and to read-only. Modification by the user is possible by setting the sheet to the “Modify” state. After a change, the check has to be repeated to get back to the “Checked” state. Each sheet shall show the actual state of the sheet and which steps the user has to execute to follow the workflow including dependencies from other sheets.



!!! The template does not force the user to follow the workflow !!!
It must be possible to ignore the checks and to proceed without having the previous steps done, for example continue running a test immediately after a change in a test step has been made. In this case it is the user’s responsibility to ensure the consistency of the data.

4.4 Programming Guidelines (Req 60)

Please follow the SBT guidelines in
https://workspace.sbt.siemens.com/content/00001136/pal/doc_process/windream/PAL-GL-0012_DE_CodierungsrichtlinieVisualBasic.doc

Here are some additional recommendations to follow.

4.4.1 Avoid formulas

A major concern in the actual TsNet version is the use of formulas within an excel sheet. Formulas are difficult to maintain. For the user, formulas within a sheet make it impossible to use standard excel functions like insert line, delete column, copy/paste etc.
Proposed solution:
To verify user’s input or to calculate values, buttons with VBA code are used within the sheet. Check-buttons also support the workflow orientation.

~~4.4.2 Context menus~~

~~https://msdn.microsoft.com/en-us/library/office/gg469862(v=office.14).aspx~~

~~The use of context menus might be helpful for the overview sheet. To be checked.~~

➔ Context menus are not used, as the user does not try the right mouse button.

4.4.3 Names for Cells and Ranges

Named ranges for the complete Excel file shall only be used for
- dropdown menus (when the list is on an other sheet, names must be used)

All other named ranges shall be local to the worksheet, for example
- definition of writeable areas and read-only areas on a sheet

Restrictions:
Most of the sheets have lists of undefined length, like the list of devices..
No Named ranges must be used for cells or ranges in these areas, as this creates problems when the user copypastes lines or columns.

For lists, the ranges shall be identified by a named cell and empty lines / columns.

The beginning of a list is identified by a named cell above / left of the list. This should be in the read-only area of the worksheet. The cell name shall be also visible in the cell's value.
The first empty line / column in a list is considered as the end of the list.
Example: Named Cell: StartList.
The list starts 1 line below StartList and ends with the last non-empty line.
The range is marked yellow

Select StartList	Alias Controller	Device-Name
✓	Central	Cen050
✓	Room	Room053
✓	Sprv	Sprv100
□	Segment	Segm054
□	Router	ROUTER1

4.4.4 Addressing of Cells within a range

Goal:
- Avoid use of too many named ranges
- Give flexibility for inserting lines and columns
- Changing the position of lines and columns
- Maintain performance by avoiding "search"

Rules
1. The start of a range shall be defined by a named cell
2. Columns and lines within the range shall be defined as constants

Example:
```
Dim FirstLine as integer ' first line of objectlist
Const ColAliasCtrl = 2          'Column with "Alias controller"

' get the first line of the objectlist
        FirstLine = Worksheets("Spec-Device").range("StartList").row + 1
' get the value of the first Alias Controller
        Wert = Worksheets("Spec-Device").range(cells(FirstLine, ColAliasCtrl)).value
```

4.5 Functional Test Specification (Req 20)

This requires splitting up up the test specification into a purely functional part (workflow steps 1..4), an environment specific part (workflow steps 5..10), and the test execution (workflow steps 11..14).
Use cases:
1. Create a functional test specification (for example for QC or for IMSES)
2. Extend a functional specification with test environment data (for testing with TsNet)
3. Remove test environment data from an existing TsNet test

(Replacing a test environment is executing use case 3 and use case 2.)

4.5.1 Alias Names

Instead of BACnet objectnames, the test specification uses "Alias" names for BACnet objects. With using Alias names, the specification is independent of the engineering environment.

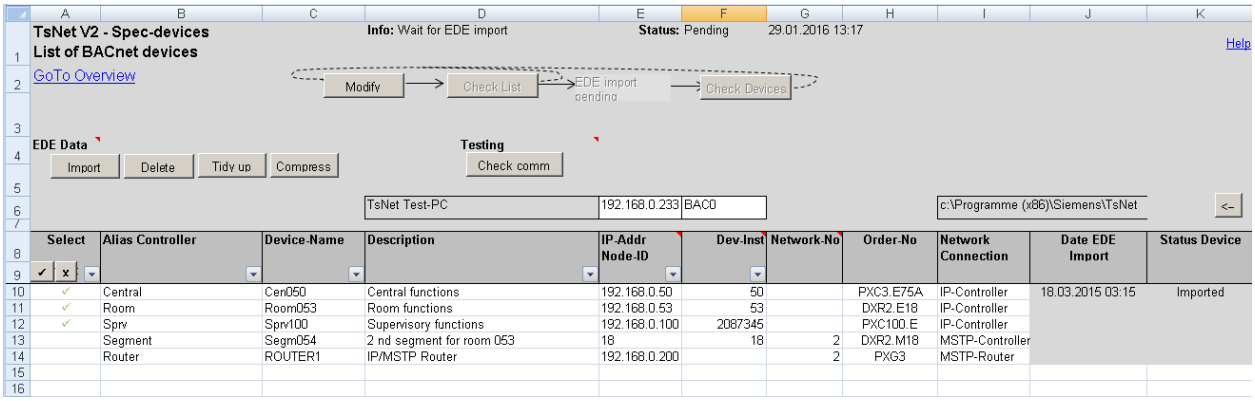
5. Description of Functions and Data

5.1 Sheet Spec-Devices

Required sheet with fix name.
Used for the definition of the devices including the TsNet device itsself.
Goal:

1. Define a network topology with an IP-Network
2. Define a network topology for MSTP or LON network, connected to the IP network via Router
3. Import and handling of BACnet EDE data (Engineering Data Exchange in a "*.csv"-file, see 5.1.6.1).
4. Handling of the workflow including the checks

5.1.1 Example



TsNet V2 - Spec-devices
List of BACnet devices
Info: Wait for EDE import
Status: Pending
29.01.2016 13:17
Help
GoTo Overview
Modify → Check List → EDE import pending → Check Devices
EDE Data
Import | Delete | Tidy up | Compress
Testing
Check comm
TsNet Test-PC | 192.168.0.233 | BAC0
c:\Programme (x86)\Siemens\TsNet <--

Select	Alias Controller	Device-Name	Description	IP-Addr Node-ID	Dev-Inst	Network-No	Order-No	Network Connection	Date EDE Import	Status Device
✓	Central	Cen050	Central functions	192.168.0.50	50		PXC3.E75A	IP-Controller	18.03.2015 03:15	Imported
✓	Room	Room053	Room functions	192.168.0.53	53		DXR2.E18	IP-Controller		
✓	Sprv	Sprv100	Supervisory functions	192.168.0.100	2087345		PXC100.E	IP-Controller		
	Segment	Segm054	2 nd segment for room 053	18	18	2	DXR2.M18	MSTP-Controller		
	Router	ROUTER1	IP/MSTP Router	192.168.0.200		2	PXG3	MSTP-Router		

Remark: only example, no specification of layout.

5.1.2 Workflow

Functional specification

1. Empty sheet (no entries in devicelist) or modification of existing sheet
2. [Modify] ->
   Status: Working
   Define Column Alias Controller, 1 line for each controller used
3. Optional: If already clarified, the following information can be added:
   Device-Name, Description, IP-Addr or Node-ID, Dev-Inst, Network-No, Order-No, Network Connection
4. [Check List] ->
   Depending on the result
   a. Check result is wrong: Info: shows reason of first failure. Status: Error List.
      Continue with correct errors and re-check Check List (→ Step 2)
   b. Check result is OK: Info: Check list OK. Status: OK-List.
      i. Leave worksheet and define objects or test steps
      ii. Continue with defining more details or more devices (→ Step 2)
      iii. Continue with the definition of the environment (→ Step 5)

Definition of the test environment

5. For each controller: Import EDE data
   Select the controller, EDE data [Import]
   Depending on the results:

   a. Import correctly done: Status Device : Imported, Date EDE Import: actual date, Info: Import done. Status: Working. Continue with Import next controller (→ Step 5) or with defining more details or more controllers (→ Step 2)
   b. Import failed: Status Device: Import failed, Date EDE Import: empty, Info: shows reason. Status: Error-import.Continue with import next controller (→ Step 5) or with correcting errors (→ Step 2) or with importing an other EDE file
   c. Import stopped by user: Status Device: Import stopped, Date EDE Import: empty, Info: empty. Status: Working. Continue with import next controller (→ Step 5) or with correcting errors (→ Step 2) or with importing an other EDE file
6. Optional: Handling of EDE data
   [Delete], [Tidy up] or [compress] EDE data
7. [Check Devices]
   depending on the results:
   a. Checks OK: All Status devices: OK. Status: OK-Devices.. Info: OK Devices. Continue with add more devices (→ Step 2) or leave the worksheet and define objects or test steps or start testing
   b. Checks failed: All Status devices: OK or Reason why check failed. Info: first failure when checking. Status: Error-Devices . Continue with Correct errors (→ Step 2) or leave the worksheet and define objects or test steps or start testing.

5.1.3 Functions

Function	Call	Affected data	Description
Modify	Command button	Complete sheet	Switches the sheet from read-only to read-write. Enables all input fields and all functions. Sets status to “Working”
Check List	Command button	Complete sheet	Checks the data and the consistence of input fields. See details 5.1.5.
Check Devices	Command button	Complete sheet	Checks the data and the consistence of all input fields and EDE. See details 5.1.5
Actualise TsNet (<-)	Command button	I6	Reads the TsNetExecutablePath from C:\ProgramData\Siemens\tsNET\tsNET Settings.xml and writes it into TsNetExecutablePath
Import EDE	Comand button	Sheet EDE	Opens dialog to select an EDE file. Opens the EDE file, checks the consistency and adds the data to the worksheet EDE. Details see 5.1.6.
Delete EDE	Command button	Sheet EDE	Removes all EDE data from the selected devices. Selection of multiple lines is possible. For all selected lines: All lines in EDE where EDE.device-obj.-instance = selected TsNet device.Dev-Inst are deleted from EDE. A dialog opens for the user to confirm before deleting.
Tidy up	Command button	Sheet EDE	Removes all EDE data from controllers which are not in the devicelist. Worksheet EDE is searched for lines where object-type = 8. For all lines found: If found line. device-obj.-instance **cannot** be found in Spec-devices.Dev-Inst Then delete all

			lines from EDE with the same device-obj.-instance. A dialog opens for the user to confirm before deleting.
Compress	Command button	Sheet EDE	NOT part of IPA!! Removes all objects of an objecttype which is not included in the BACnet_Objecttype area of the sheet Enum (after confirmation)
Check Comm	Command button	-	NOT part of IPA!! Generates a TsNet script which checks the communication to selected devices. Runs the script and evaluates the result file. Does not influence Status.
Select Button (✓)	Command button	Column A	Selects line in devicelist and marks it with a tick (existing VBA code)
Deselect Button (x)	Command button	Column A	Deselects line in devicelist and removes the tick

5.1.4 Fields

The upper part of the sheet contains general data for all devices

Name	Type	Location in example	Description
Info	R	D1	Last detailled status or error information from any functions called from this sheet
TsNet IP*	W	E6	Outgoing IP-Address for TsNet on the test PC
TsNet Port*	W	F6	Outgoing port for TsNet on test PC. List of BAC0…BAC9
Status*	R	F1	Status of the complete sheet Working: in work, not all entries / checks done yet Error: Check Devices or EDE import or Check List failed OK-List: Check List correct OK-EDE: all EDE imports done and correct OK-ready : ready for testing, all checks done (except Check Comm Devices)
	R	G1	Date of the last status change
StartList*	R	A9	One line above the start of the device list
TsNet Executable Path*	R	I6	Filename and path for executing TsNet (runtime). Used for Check comm and Overview|Gen script

* named cells

The lower part of the sheet from one line below “StartList” downwards contains the devicelist. The devicelist ends with the first empty line.

Name	Type	Location in example	Description
Select	R	A10…Axx	Shows, if a device is selected for further functions (delete…). Handling via select / deselect buttons
Alias Controller	W	B10…Bxx	Alias controllername for functional specification
Device-Name	W	C10…Cxx	BACnet devicename for definition of the actual test environment
Description	W	D10…Dxx	Description of the controller, comment (for documentation)

IP-Addr Node-ID	W	E10…Exx	IP-Address for communication with the controller, Node-ID for MSTP / LON
Dev-Inst	W	F10…Fxx	BACnet device instance
Network-No	W	G10…Gxx	Network-Number for communication via IP/MSTP or IP/LON-Router
Order Number	W	H10…Hxx	Order-No of device (for documentation)
Network Connection	W	I10…Ixx	Selection of device-Type (IP-Controller, MSTP-Controller, LON-Controller, MSTP-Router, LON-Router)
Date EDE Import	R	J10…Jxx	Date/time of the last successful EDE import. Handling via Import EDE
Status Device	R	K10…Kxx	Status of the device, last error / status message: -Not unique : from function Check List / Devices -EDE OK : EDE imported successfully from Import EDE -EDE error: Error in EDE from Import EDE or from Check Devices -Comm OK: Communication test successfully (Check Comm Devices) NOT part of IPA -Comm error: Communication not possible (Check Comm Devices) NOT part of IPA -No entry: OK

5.1.5 Checks

Check List:
All of the following conditions must be fulfilled for Result = OK-List

- Alias Controller: Unique and not empty
- Device-Name: unique, empty is allowed.
- IP-Addr, Node-ID: unique, empty is allowed
- If network connection is IP-Controller or MSTP-Router or LON-Router:
  - IP-Addr, Node-ID: IP-Format (nnn.nnn.nnn.nnn) with nnn <= 255
- If network connection is MSTP-Controler or LON-Controller:
  - IP-Addr, Node-ID: nnn with nnn <= 255
- Dev-Inst: Unique, empty is allowed
- If network connection is MSTP-Controler or LON-Controller: Dev-Inst = IP-Addr Node-ID
- Network-No: empty is allowed
- If network connection is MSTP-Controller
  - If not empty, then Network-No must be identical to the network-No of an MSTP-Router
- If network connection is LON-Controller
  - If not empty, then Network-No must be identical to the network-No of a LON-Router

Check Devices

- All of the following conditions must be fulfilled for Result = OK-Device
- Alias Controller: Unique and not empty
- Device-Name: Unique and not empty
- IP-Addr, Node-ID: Unique and not empty
- If network connection is IP-Controller or MSTP-Router or LON-Router:
  - IP-Addr, Node-ID: IP-Format (nnn.nnn.nnn.nnn) with nnn <= 255
- If network connection is MSTP-Controler or LON-Controller:
  - IP-Addr, Node-ID: nnn with nnn <= 255

- Dev-Inst: Unique and not empty
- If network connection is MSTP-Controler or LON-Controller: Dev-Inst = IP-Addr Node-ID
- Network-No: nnn with nnn <= 255
- If network connection is MSTP-Router: Network-No not empty
- If network-Connection is LON-Router: Network-No not empty
- If network connection is MSTP-Controller: not empty
    - Network-No must be identical to the network-No of an MSTP-Router
- If network connection is LON-Controller: Not empty
    - Network-No must be identical to the network-No of a LON-Router
- Network connection: not empty
- If Network Connection is IP-Controller, MSTP-Controller or LON-Controller:
    - In Worksheet EDE there must be exactly one row which matches:
      EDE!Object-type = 8
      EDE!Device-object-instance = Spec-Devices!Dev-Inst
      EDE!Object-name = Spec-Devices!Dev-Name

5.1.6 Import EDE

5.1.6.1 EDE format

EDE = BACnet engineering data exchange is used to transfer data from one BACnet engineering system to another. The content of the file is standardized.
The commonly used format is csv with ";" as a separator.
An EDE file is characterized by the "#" as the first character in the first line.
The next lines contain some header information.
After the header information, the next line column 1...2 contain "#mandatory", the following line contains "#keyname", "device-obj.-instance".
The next line is the startline for importing the EDE file.

#				
Various number of lines, depending on the version				
#mandatory	mandatory	mandatory	mandatory	mandatory
# keyname	device obj.-instance	object-name	object-type	object-instance
Start of data				

A file which does not have the entries marked red in the example, is not an EDE file.

Within an EDE file the controller data is defined by a line where object-type = 8.

# keyname	device obj.-instance	object-name	object-type
Building03Room15Temperature	10	RoomTemperature	0
Building03Room15DamperPosition	10	Damper	1
Building03Room15Damper	10	Damper	2
Building03Room15	10	ASB03R15	8
Building17Room01WindowStatus	12	Window	3

5.1.6.2 EDE-Import

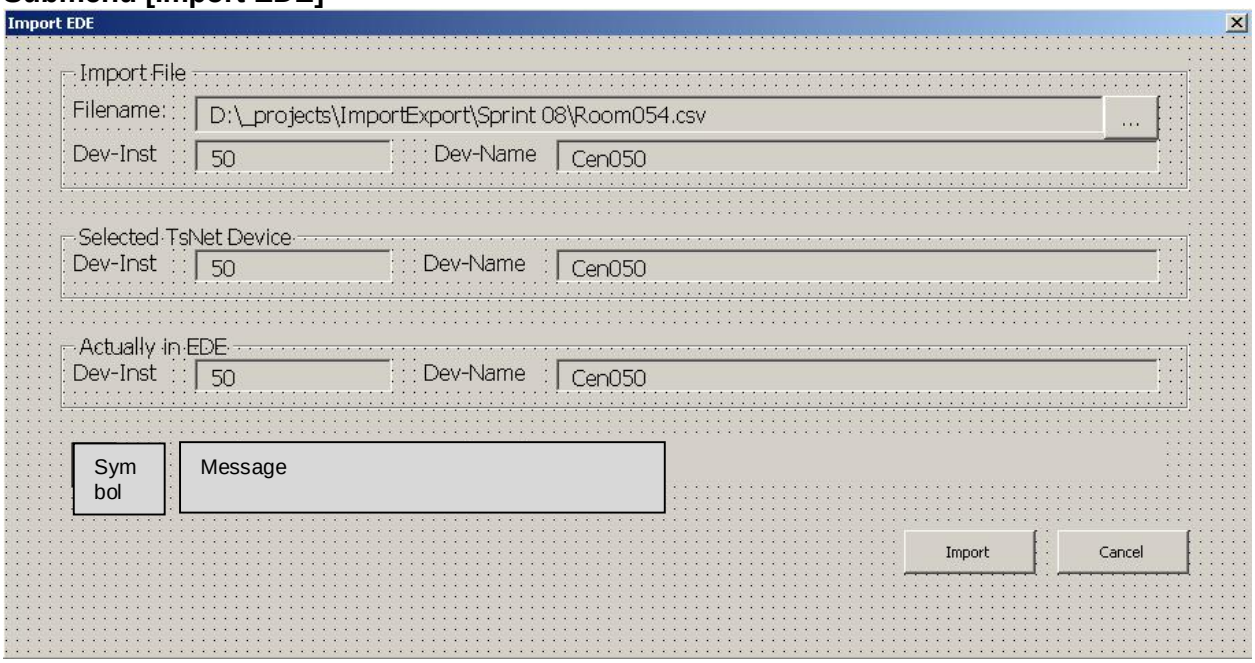
EDE import is started by selecting a device (selected TsNet Device) and clicking the [Import] button.
EDE import is a single selection function which works only with one controller.
A controller is selected
- When a green tick is visible in the Select column or

- When a line is selected or
- When a field in the select column is selected.

The selection is invalid, if
- No device is selected
- The selected device is not of type IP-Controller, MSTP-Controller or LON-Controller

If multiple devices are selected, the first valid controller shall be used.
If a valid controller is found, the green tick in the select column is set to the valid controller and the other devices are deselected.
If no valid device is used, the function shall be aborted with a message to the user.
Otherwise submenu is opened:

Submenu [Import EDE]



Remark: only example, no specification of layout.

Area [Import file]:
Button […]: The EDE-file (*.csv) can be selected. After opening and converting the csv-file, the line with (object-type = 8) and (device-instance = selected TsNet-device Dev-Inst.) is searched.
Field [Dev-Inst] : The device-obj.-instance from the csv file is shown.
Field [Dev-Name]: The Object-name from the csv-file is shown.
If no matching line is found in the csv-file, both fields are left empty.

Area [Selected TsNet device]: Dev-Inst and Device-Name of the selected TsNet device is shown.

Area [Actually in EDE]: in worksheet EDE the line with object-type 8 and (device-instance = selected TsNet-device Dev-Inst.) is searched.
Field [Dev-Inst] : The device-obj.-instance from sheet EDE is shown.
Field [Dev-Name]: The Object-name from the sheet EDE is shown.
If no matching line is found in the sheet EDE, both fields are left empty.

Depending on the situation, the following [symbol] and [message] is shown

Situation	[Symbol	Message
Import file.Dev-Inst <> Selected TsNet device.Dev-Inst or	✗	Selected device not equal to Import device, no import possible. [Import

Import file.Dev-Name <> Selected TsNet device.Dev-Name.		EDE] button is disabled.
Selected TsNet device.Dev-Inst is empty or Selected TsNet device.Dev-Name is empty	✗	Selected device not correctly defined, no import possible. Define devicename and device-instance. . [Import EDE] button is disabled.
Import file is no EDE file (see EDE Format)	✗	No EDE file, no import possible. . . [Import EDE] button is disabled.
Import file.Dev-Name = Selected TsNet device.Dev-Name and Import file.Dev-Inst = Selected TsNet device.Dev-Inst and Actually in EDE. Dev-Inst = Import file.Dev-Inst	✓	Data consistent. Import possible. Import replaces actual EDE for device. [Import EDE] button is enabled.
Import Dev-Name = selected Dev-Name and Import file Dev-Inst = selected Dev-Inst and actual Dev-Inst is empty	✓	Data consistent. Import possible. Imported data will be appended to worksheet EDE. [Import EDE] button is enabled.

Appending EDE data
When the checks described above are successfully executed, and when the user clicks the [Import EDE] button of the [Import EDE] submenu, EDE data is appended to the existing list in worksheet EDE.

1. All lines in the import, where import.device-obj.-instance = selected TsNet device.Dev-Inst are appended to the list.
2. After successful import, [Date EDE import] for the selected device is set to actual date/time, and [Status device] is set to “imported”.

Replacing EDE data
Instead of appending EDE data, tReplacing EDE data is carried out, if already data for the same device-obj.-instance as the selected TsNet device exists in worksheet EDE.

1. All lines in EDE where EDE.device-obj.-instance = selected TsNet device.Dev-Inst are deleted from EDE.
2. All lines in the import, where import.device-obj.-instance = selected TsNet device.Dev-Inst are appended to the list.
3. After successful import, [Date EDE import] for the selected device is set to actual date/time, and [Status device] is set to “imported”.