

Titel: Individuelle Projekt Arbeit

Thema: Neuerstellung Bedienoberfläche für Testautomationstool TsNet

Dieses Dokument ist nach der „Dokumentations-Empfehlung für die IPA Informatik“ des Kantons Zug, Stand 07.01.2016, aufgebaut und enthält den schriftlichen Teil der „Individuellen Praktischen Arbeit“ von Fatma Yilmaz. Es dient zur Orientierung der Experten und des Fach-Vorgesetzten.

Der erste Teil befasst sich mit der Aufgabenstellung, der Planung und dem Arbeitsjournal. Der zweite Teil beinhaltet die Charakteristik der eigentlichen Arbeit.

Key Words: IPA, TsNet, VBA

Dokument Kategorie:	ProjectRecord
Gültigkeitsdatum:	2016-03-21
Dokument Status:	In Bearbeitung
Autor:	Yilmaz Fatma
Abteilung:	BT CPS R&D ZG CS PT
Verantwortliche Stelle:	fatma.yilmaz@siemens.com
Firma:	Siemens Schweiz AG, Building Technologies Division Control Products & Systems
Basierend auf Vorlage:	Workbook_Standard; 4; 2014-11-05; Donat Hutter,3531

Änderungsgeschichte

Rev	Datum	Autor	Änderungen
0.1	21-Mrz-2016	Yilmaz Fatma	Status = In Bearbeitung <ul style="list-style-type: none">- Journal Tag 1- Entwurf
0.2	22-Mrz-2016	Yilmaz Fatma	Status = In Bearbeitung <ul style="list-style-type: none">- Journal Tag 2- Akzeptanztest- Design
0.3	24-Mrz-2016	Yilmaz Fatma	Status = In Bearbeitung <ul style="list-style-type: none">- Journal Tag 3- Struktogramme erstellen- Implementierung
0.4	29-Mrz-2016	Yilmaz Fatma	Status = In Bearbeitung <ul style="list-style-type: none">- Journal Tag 4- Implementierung

0.5	31-März-2016	Yilmaz Fatma	Status = In Bearbeitung - Journal Tag 5 - Implementierung
0.6	01- April-2016	Yilmaz Fatma	Status = In Bearbeitung - Journal Tag 6 - Implementierung - White Box Test
0.7	04-April-2016	Yilmaz Fatma	Status = In Bearbeitung - Journal Tag 7 - White Box Test - Implementierung
0.8	05- April-2016	Yilmaz Fatma	Status = In Bearbeitung - Journal Tag 8 - White Box Test - Akzeptanz Test
0.9	07- April- 2016	Yilmaz Fatma	Status = In Bearbeitung - Journal Tag 9 - Akzeptanz Test - Schlussbericht
1	08- April- 2016	Yilmaz Fatma	Status = In Bearbeitung - Journal Tag 10 - Diverses

Inhaltsverzeichnis

1. TEIL 1	5
1.1 EINFÜHRUNG	5
1.1.1 IPA	5
1.1.2 Zweck des Dokumentes	5
1.1.3 Ziel Publikum	5
1.2 PROJEKTAUFTRAG GEMÄSS PKORG	6
1.2.1 Ausgangslage	6
1.2.2 Detaillierte Aufgabenstellung	7
1.2.3 Mittel und Methoden	7
1.2.4 Vorkenntnisse	8
1.2.5 Vorarbeiten	8
1.2.6 Neue Lerninhalte	8
1.2.7 Arbeiten in den letzten 6 Monaten	8
1.3 PROJEKTORGANISATION	9
1.3.1 Datensicherung	9
1.3.2 Beteiligte Dienste und Fachabteilungen	9
1.3.3 Verwendete Projektmanagementmethode	10
1.3.4 Ordnerstruktur auf Server	11
1.3.5 Arbeitsplatz	11
1.3.6 Risikobeschreibung	12
1.4 PLANUNG	13
1.4.1 Zeitplan: Soll	13
1.4.2 Zeitplan: Soll-Ist-Vergleich	14
1.4.3 Tätigkeiten	15
1.4.4 Meilensteine	17
1.5 ARBEITSJOURNAL	18
1.5.1 Zweck des Arbeitsjournals	18
1.5.2 Anwendungsbereich, Abgrenzung	18
1.5.3 Aufbau	18
1.5.4 Arbeitsjournale vom 21.03.2016 bis 08.04.2016	19
2. TEIL 2	29
2.1 IPA KURZFASSUNG	29
2.1.1 Ausgangssituation	29
2.1.2 Umsetzung	29
2.1.3 Ergebnis	29
2.2 REALISIERUNG	30
2.2.1 Bedienoberfläche erstellen	30
2.2.2 Informationen zur Bedienoberfläche	32
2.2.3 Struktogramme	34
2.2.4 Implementierung	46
2.3 KONTROLLIERUNG	59
2.3.1 Testkonzept	59
2.3.2 White-Box-Test	61
2.3.3 Akzeptanz-Test	75
2.4 AUSWERTUNG	81
2.4.1 Schlusswort	81
2.5 GLOSSAR	82
2.6 QUELLEN	84
2.7 ANHANG	85
2.7.1 Code	85
2.7.2 Tabelle 19 (Spec-Devices)	85
2.7.3 mdl_SpecDevices	87
2.7.4 frmImportEDE	118

2.7.5 DieserArbeitsmappe	120
--------------------------------	-----

Tabellenverzeichnis:

Tabelle 1 : Projektauftrag	6
Tabelle 2 : Beteiligte Dienste & Fachabteilungen	9
Tabelle 3 : Risikobeschreibung	12
Tabelle 4 : Tätigkeiten	16
Tabelle 5 : Meilensteine	17
Tabelle 6 : Funktionen	32
Tabelle 7 : Felder	32
Tabelle 8 : Unterfunktion-CheckFrmlImportEDE	54
Tabelle 9 : WhiteBox-Testergebnisse	74
Tabelle 10 : WhiteBox-Nachtest	74
Tabelle 11 : Akzeptanztest - Testergebnisse	80
Tabelle 12 : Glossar	83
Tabelle 13 : Quellenverzeichnis	84

Abbildungsverzeichnis:

Abbildung 1-2 : Backup, Siemens Server	9
Abbildung 1-1 : Backup, Festplatte	9
Abbildung 1-3 : IPERKA	10
Abbildung 1-4 : Ordnerstruktur auf Server	11
Abbildung 1-5 : Arbeitsumgebung	11
Abbildung 1-6 : Soll-Zeitplan	13
Abbildung 1-7 : Soll-Ist-Zeitplan	14
Abbildung 2-1 : Bedienoberfläche erstellen	30
Abbildung 2-2 : Strg - Cmd_ImportEDE_Click	35
Abbildung 2-3 : Strg – ActualiseTsNet	35
Abbildung 2-4 : Strg - GetHelp	35
Abbildung 2-5 : Strg – TidyUp	36
Abbildung 2-6 : Strg – DeleteEDE	36
Abbildung 2-7 : Strg - ModifyWBopen	37
Abbildung 2-8 : Strg - ModifyWBS	37
Abbildung 2-9 : Strg - CheckList	38
Abbildung 2-10 : Strg – CheckDuplication	39
Abbildung 2-11 : Strg – ImportEDE	40
Abbildung 2-12 : Strg – CheckFrmlImportEDE	41
Abbildung 2-13 : Strg – appendingEDE	42
Abbildung 2-14 : replacingEDE	43
Abbildung 2-15 : Strg - CheckDevices	44
Abbildung 2-16 : Strg - CheckDevicesInEDE	45
Abbildung 2-17 : Modify	46
Abbildung 2-18 : ListOk Bsp1	47
Abbildung 2-19 : ErrorList Bsp2	48
Abbildung 2-20 : ErrorList Bsp3	48
Abbildung 2-21 : ErrorList Bsp4	49
Abbildung 2-22 : OkDevices Bsp1	50
Abbildung 2-23 : ErrorDevices Bsp2	51
Abbildung 2-24 : ErrorDevices Bsp3	51
Abbildung 2-25 : SubMenu-ImportEDE	52
Abbildung 2-26 : WrongEdeFormat	53
Abbildung 2-27 : TidyUp	56
Abbildung 2-28 : DeleteEDE	57
Abbildung 2-29 : ActualiseTsNET	58
Abbildung 2-30 : Testumgebung	59
Abbildung 2-31 : Information zu Excel	59

1. Teil 1

1.1 Einführung

1.1.1 IPA

IPA steht für **I**ndividuelle **P**raktische **A**rbeit und wird von allen Informatik–Lernenden im letzten Semester der beruflichen Grundbildung durchgeführt. Die Plattform PkOrg ist zuständig für einen organisatorisch einwandfreien Ablauf der IPA.

1.1.2 Zweck des Dokumentes

Die IPA-Dokumentation beinhaltet alle Arbeitsschritte, welche im Rahmen der IPA von Fatma Yilmaz durchgeführt wurden. Der erste Teil befasst sich mit der Aufgabenstellung, der Planung und dem Arbeitsjournal. Der zweite Teil beinhaltet die Beschreibung der eigentlichen Arbeit.

1.1.3 Ziel Publikum

Der Inhalt richtet sich an die Experten und den Fachvorgesetzter der IPA. Damit kann die IPA nacherlebt und eingestuft werden. Des Weiteren können Entwickler von TsNet diese Dokumentation verwenden, um sich zu erkundigen.

1.2 Projektauftrag gemäss PkOrg

Projekttitel	Neuerstellung Bedienoberfläche für Testautomationstool TsNet
Prüfungskandidat	Fatma Yilmaz
Fachvorgesetzter	Michael Speckien
Auftraggeber	Michael Speckien

Tabelle 1 : Projektauftrag [1]

1.2.1 Ausgangslage

Seit geraumer Zeit wird das PC-basierte Test-Tool TsNet V1 beim Testen von Applikationen eingesetzt. Es besteht aus

- einem Definitionsteil (Excel-Template), in dem die Testschritte und die erwarteten Ergebnisse definiert werden sowie
- einem Run-Time-Teil, der mit dem Controller kommuniziert und Testschritte vorgibt und Ergebnisse abfragt.

Das Projekt TsNet V2 beinhaltet die komplette Neuerstellung des Definitionsteils. Im Rahmen dieser Arbeit fällt auch die Neuerstellung der Definition der Gerätetopologie an. Dabei werden die Bedienoberfläche und die zugehörigen Funktionen neu erstellt.

Zusammenfassend handelt es sich um die Erstellung eines Excel- Arbeitsblatts zur Spezifikation der Geräte- und Netzwerktopologie.

Neben dem Eingabedialog sind auch folgende Funktionen gefordert:

- Führen eines Status, der den aktuellen Stand der Eingaben darstellt
- Prüfen der Plausibilität der eingegebenen Daten untereinander und mit anderen Arbeitsblättern
- Import von Daten aus einem .csv- File

Dieser Tätigkeit wird von Fatma Yilmaz in der IPA durchgeführt.

1.2.2 Detaillierte Aufgabenstellung

Als Basis für die Arbeiten gilt eine Requirement-Specification, siehe Aufgabenstellung im Anhang. Neben der Spezifikation werden diverse Beispieldateien zur Verfügung gestellt. Diese dienen zur Erläuterung der Spezifikation und zum Testen der Funktionen.

Aufgabenstellung: Gerätetopologie für TsNet V2 gemäss Spezifikation

- Terminplanung und Projektstatus
- Alle 2 Tage ist ein Statusmeeting mit dem Auftraggeber durchzuführen
- Erstellung der Software
- Erstellung der Softwaredokumentation. Zielgruppe: SW-Entwickler, die das TsNet-Tool warten, pflegen und weiterentwickeln
- Erstellen der Testfälle und Durchführen von Tests
- Test der Eingabedialoge mit korrekten und falschen Werten
- Test der Usability der Eingabedialoge mit Fachperson ausserhalb vom TsNet-V2-Projekt
- Test aller erstellten Funktionen gegenüber der Spezifikation
- Test aller erstellten Funktionen gegen Fehlbedienung und fehlerhafte Daten und Dateien
- Entsprechend den IPA-Regeln ist ein Arbeitsjournal zu führen

Erwartete Lieferungen:

- Terminpläne und Projektstatus alle 2 Tage und am Ende der IPA
- Excel-File mit der Gerätetopologie gemäss Spezifikation
- Kommentierter Source-Code
- Kommentierte Eingabedialoge
- Softwaredokumentation: Es kann jede grafische Darstellung des Workflows, des Programmablaufs und des Datenflusses verwendet werden, sofern sie dazu geeignet ist, einem anderen Entwickler die Weiterarbeit zu ermöglichen. Komplexere Funktionen wie die Plausibilitätstests und der csv-Import sollen graphisch dokumentiert werden (Struktogramm, Flussdiagramm o.ä.). Bei einfachen Funktionen genügt eine Beschreibung in Textform.
- Testspezifikation und Testergebnisse
- Arbeitsjournal

1.2.3 Mittel und Methoden

Entwicklungsumgebung:

- Standard-PC mit Microsoft Windows 7
- Microsoft Excel 2007
 - Programmierung in VBA
- Erstellung der Dokumentation, der Präsentation und weiterer Dokumente mit Microsoft Office 2007
- Firmen-Richtlinie, Codierungsrichtlinie Visual Basic und weitere Vorgaben in der Spezifikation

1.2.4 Vorkenntnisse

Die IPA-Ausführende hat bereits im Vorfeld der IPA an anderen Teilen von TsNet mit der gleichen Arbeitsumgebung gearbeitet. Programmieren mit VBA ist ihr aus dieser Tätigkeit und aus anderen Projekten bekannt.

1.2.5 Vorarbeiten

Im Vorfeld der IPA wurde eine Variantenanalyse zur Gestaltung der Bedienoberfläche durchgeführt. Die Ergebnisse sind in die Spezifikation eingeflossen. Weiterhin wurde die Spezifikation durchgesprochen, so dass die Aufgabenstellung komplett abgeklärt ist.

1.2.6 Neue Lerninhalte

- Selbstständiges Organisieren eines Zeitplans
- Überprüfen eines Projektstatus
- Selbstständiges Software-Design
- Erstellen einer Software-Dokumentation
- Selbstständiges Erstellen eines Testkonzepts

1.2.7 Arbeiten in den letzten 6 Monaten

- Weiterarbeit an einer Bedienoberfläche in Matlab / Simulink (ca. 1 Monat)
- Weiterarbeit an mehreren Excel VBA-Projekten (ca. 3 Monate)
- Programmierarbeiten im Projekt TsNet V2 in Excel VBA (ca. 2 Monate)

1.3 Projektorganisation

1.3.1 Datensicherung

Die gesamte IPA-Ordnerstruktur mit dem betreffenden Code befindet sich auf den Siemens Server, von welchem täglich automatisch ein Backup gemacht wird. Zusätzlich wird vom IPA-Ordner am Ende des Arbeitstages eine Kopie erstellt, welche auf einer lokalen Festplatte gespeichert wird. Dadurch kann man jederzeit auf alle bisherigen Versionen zurückgreifen, falls Daten gelöscht oder zerstört worden sind.



Abbildung 1-2 : Backup, Siemens Server



Abbildung 1-1 : Backup, Festplatte

1.3.2 Beteiligte Dienste und Fachabteilungen

Firma	Siemens Schweiz AG, Building Technologies Division
Abteilung	BT CPS R&D ZG CS PT
Verwendete Software	<ul style="list-style-type: none"> - Microsoft Office 2007 - Microsoft Visio - HUS Struktogrammer
Verwendete Tools	<ul style="list-style-type: none"> - Snipping Tool [ist Bestandteil von Windows 7]

Tabelle 2 : Beteiligte Dienste & Fachabteilungen

1.3.3 Abteilung BT CPS R&D ZG CS PT

Die Abteilung „Applikation“ erstellt Libraries/Funktionsblöcke für freiprogrammierbare Geräte. Diese Geräte werden zur Gebäudeautomation verwendet und steuern Storen, Lichter, Heizung/Lüftung. Mit dem Programm TIA kann man sich die notwendigen Funktionen für ein Gerät zusammensuchen und abändern. Schliesslich kompiliert man und lädt die erzeugte Software auf das Gerät (z.B. DXR- oder PXC-Controller).

Die Geräte setzen Befehle von Sensoren oder Interface (Control-Panel/Web-Browser) um. TIA verwendet eine Art graphische oder diagrammähnliche Programmierung.

1.3.3 Verwendete Projektmanagementmethode

Bei der Arbeit an dem Projekt wird die Projektmanagementmethode IPERKA verwendet

Die Abkürzung IPERKA steht für:

- 1) **I**nformation
- 2) **P**lanen
- 3) **E**ntscheiden
- 4) **R**ealisieren
- 5) **K**ontrollieren
- 6) **A**uswerten

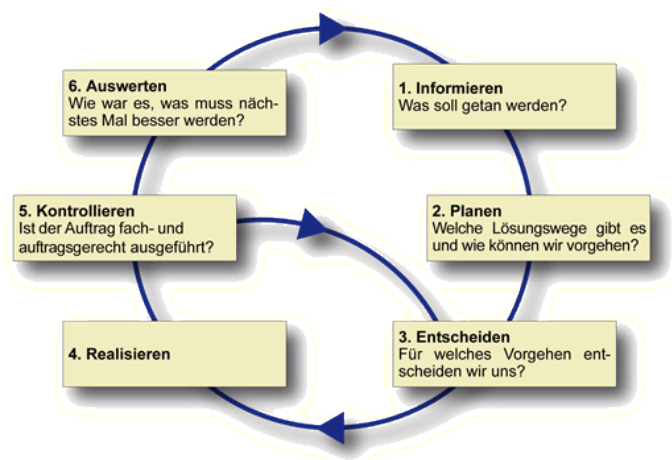


Abbildung 1-3 : IPERKA [2]

Begründung der Wahl:

Die ausgewählte Projektmanagementmethode „IPERKA“ passt gut zu meinem Arbeitsstil, da bei dieser Methode die einzelnen Tätigkeiten, die von mir erstellt wurden, mühelos entsprechenden Schritten zugeordnet werden können. Die Projektmethode wird auch in der Schule gelehrt und hat sich in diversen Projekten in der Schule bereits bewährt.

1.3.4 Ordnerstruktur auf Server

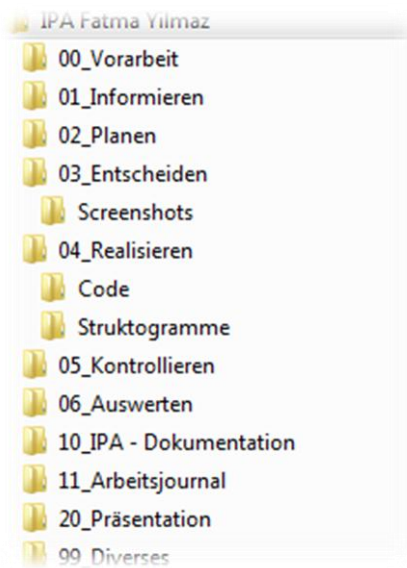


Abbildung 1-4 : Ordnerstruktur auf Server

1.3.5 Arbeitsplatz

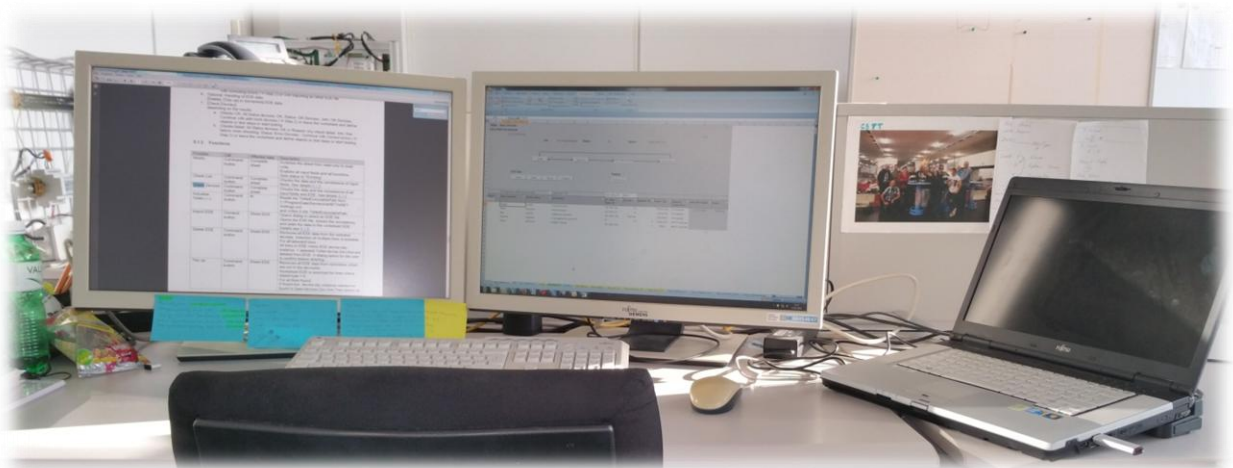


Abbildung 1-5 : Arbeitsumgebung

Der Arbeitsplatz von Fatma Yilmaz befindet sich in Zug im Zählerweg 5, vierter Stock. Ein Fujitsu Laptop mit zwei zusätzlichen Monitoren steht für IPA zur Verfügung.

1.3.6 Risikobeschreibung

In der nachfolgenden Tabelle „Tabelle 1 – 3: Risikobeschreibung“ wird beschrieben, was den Ablauf der IPA in Gefahr bringen könnte.

VBA – Kenntnisse	Falls die IPA- Ausführende bei der Implementierung Probleme haben sollte, könnte es sein, dass der Terminplan nicht eingehalten wird, somit werden die Bestandteile von IPA nicht termingerecht implementiert und getestet. Das würde zu einer Verzögerung im Zeitplan führen.	
	Wahrscheinlichkeit	50%
	Auswirkungsgrad	Kritisch
Know-how im Umgang mit Testdaten	Es werden Testdaten zur Verfügung gestellt, um möglichst viele Testfälle in der Testphase abdecken zu können. Falls für das Verständnis dieser Daten Hilfe benötigt wird, sollte man Hilfe anfordern.	
	Wahrscheinlichkeit	50%
	Auswirkungsgrad	Leicht
Akzeptanztest eines Applikationsentwicklers	Für den Akzeptanztest wird mit der Testperson einen Termin vereinbart. Falls der Mitarbeiter nicht zur Verfügung steht, sollte so schnell wie möglich eine Ersatzperson gefunden werden.	
	Wahrscheinlichkeit	10%
	Auswirkungsgrad	Leicht
Fehlschlag von Akzeptanztest	Wenn der Akzeptanztest fehlschlagen würde, so könnte keine Änderungen oder Korrekturen mehr gemacht werden. Der Zeitplan kann nicht eingehalten werden.	
	Wahrscheinlichkeit	25%
	Auswirkungsgrad	Kritisch
Zeitplan	Im Zeitplan wurde eine Pufferzeit von 0.8 h pro Arbeitstag eingeplant. Damit hat die IPA-Ausführende am Schluss 1 Tag als Reservezeit, um allfälligen Verzögerungen zu korrigieren.	
	Wahrscheinlichkeit	10%
	Auswirkungsgrad	Kritisch

Tabelle 3 : Risikobeschreibung

1.4 Planung

1.4.1 Zeitplan: Soll

IPA Fatma Yilmaz		Abhängigkeit		Aufwand		Status		Geplanter Ablauf																					
ID	Tätigkeiten	Voraussetzung	Nächster Schritt	Soll [h]	Ist [h]	Differenz [h]	Meilenstein [Datum]	Status	21.03.2016		22.03.2016		24.03.2016		Karfreitag Ostermontag	29.03.2016		31.01.1900		01.04.2016		04.04.2016		05.04.2016		07.04.2016		08.04.2016	
									Soll	Ist	Soll	Ist	Soll	Ist		Soll	Ist	Soll	Ist	Soll	Ist	Soll	Ist	Soll	Ist	Soll	Ist		
1 Informieren																													
1.1	Projektauftrag lesen und verstehen			0.5					0.5																				
2 Planen																													
2.1	Tätigkeiten und Meilensteine definieren	1.1	2.2	2.0					2.0																				
2.2	Soll- Zeitplan mit geplantem Ablauf	2.1		2.0			21.03.2016		2.0																				
2.3	Vorlagen und IPA-Bericht aufbauen			1.0					1.0																				
2.4	Testkonzept erstellen			1.0								1.0																	
2.5	Arbeitsumgebung einrichten			0.0					0.0																				
3 Entscheiden																													
3.1	Akzeptanztest-Spezifikation ermitteln			4.0							2.0		2.0																
4 Realisieren																													
4.1	Bedienoberfläche erstellen und benennen			2.0			24.03.2016				2.0																		
4.2	Struktogramme/Flussdiagramme erstellen			3.0									3.0																
4.3	Implementierung von "Modify", "CheckList", "CheckDevices", "Select, Deselect"	4.1	4.4	5.0									2.5			1.5		1.0											
4.4	Implementierung von "Import EDE Data"	4.1	4.5	7.0			31.03.2016									5.0		2.0											
4.5	Implementierung von "Delete EDE Data"	4.1	4.6	4.0														2.5		1.5									
4.6	Implementierung von "Tidy Up EDE Data"	4.6	4.7	4.0																4.0									
4.7	White-Box Testfälle ermitteln	4.3-4.7		4.0																		4.0							
4.8	Usability Improve/ Error Handling	4.1	4.9	2.0																		1.0		1.0					
4.9	Help-Funktion erstellen	4.1		1.0			04.04.2016																	1.0					
5 Kontrollieren																													
5.1	White-Box-Test durchführen	4.7		3.0																				3.0					
5.2	Akzeptanz-Test durchführen	3.1		3.0																				1.0		2.0			
6 Auswerten																													
6.1	Schlussbericht verfassen			2.0			07.04.2016																			2.0			
7 Diverses																													
7.1	Zeitplan IST-Zustand einfügen			1.0					0.1		0.1		0.1			0.1		0.1		0.1		0.1		0.1		0.1		0.1	
7.2	Arbeitsjournal führen			1.0					0.1		0.1		0.1			0.1		0.1		0.1		0.1		0.1		0.1		0.1	
7.3	IPA-Bericht führen			12.0					1.0		1.0		1.0			1.0		1.0		1.0		1.0		1.0		1.0		4.0	
7.4	Statusmeeting mit Auftraggeber			3.0					0.5				0.5					0.5				0.5				1.0			
7.5	Gespräch mit Erstexpert (Kick-Off)			2.0							2.0																		
7.6	Administratives/Organisatorisches			1.0					0.1		0.1		0.1			0.1		0.1		0.1		0.1		0.1		0.1		0.1	
7.7	Abgabe: drucken, binden, Upload			1.5																								1.5	
8 Puffer / Reserve																													
8.1	Pufferzeit			8.0					0.8		0.8		0.8			0.8		0.8		0.8		0.8		0.8		0.8		0.8	
Total:				80.0	0.0	0.0			8.1	0.0	8.1	0.0	11.1	0.0		8.6	0.0	8.1	0.0	7.6	0.0	7.6	0.0	8.1	0.0	7.1	0.0	6.6	0.0
x Soll-Zeit																													
x Ist-Zeit																													
Meilensteine																													

Abbildung 1-6 : Soll-Zeitplan

1.4.2 Zeitplan: Soll-Ist-Vergleich

IPA Fatma Yilmaz		Abhängigkeit				Aufwand		Status		Geplanter Ablauf																			
ID	Tätigkeiten	Voraussetzung	Nächster Schritt	Soll [h]	Ist [h]	Differenz [h]	Meilenstein [Datum]	Status	21.03.2016		22.03.2016		24.03.2016		Karfreitag Ostermontag	29.03.2016		31.03.2016		01.04.2016		04.04.2016		05.04.2016		07.04.2016		08.04.2016	
									Soll	Ist	Soll	Ist	Soll	Ist		Soll	Ist	Soll	Ist	Soll	Ist	Soll	Ist	Soll	Ist	Soll	Ist	Soll	Ist
1 Informieren																													
1.1	Projektauftrag lesen und verstehen			0.5	0.5	0.0		OK	0.5	0.5																			
2 Planen																													
2.1	Tätigkeiten und Meilensteine definieren	1.1	2.2	2.0	2.0	0.0		OK	2.0	2.0																			
2.2	Soll- Zeitplan mit geplantem Ablauf	2.1		2.0	2.0	0.0	21.03.2016	OK	2.0	2.0																			
2.3	Vorlagen und IPA-Bericht aufbauen			1.0	1.0	0.0		OK	1.0	1.0																			
2.4	Testkonzept erstellen			1.0	1.0	0.0		OK					1.0	1.0															
2.5	Arbeitsumgebung einrichten			0.0	0.0	0.0		erfolgt	0.0	0.0																			
3 Entscheiden																													
3.1	Akzeptanztest-Spezifikation ermitteln			4.0	2.0	2.0		OK			2.0	2.0	2.0	0.0															
4 Realisieren																													
4.1	Bedienoberfläche erstellen und benennen			2.0	2.5	-0.5	24.03.2016	OK			2.0	2.5																	
4.2	Struktogramme/Flussdiagramme erstellen			3.0	3.0	0.0		OK					3.0	3.0															
4.3	Implementierung von "Modify", "CheckList", "CheckDevices", "Select, Deselect", "Actualise TsNet"	4.1	4.4	5.0	5.0	0.0		OK					2.5	2.5		1.5	1.5	1.0	1.0										
4.4	Implementierung von "Import EDE Data"	4.1	4.5	7.0	8.0	-1.0	31.03.2016	OK								5.0	5.0	2.0	2.0										
4.5	Implementierung von "Delete EDE Data"	4.1	4.6	4.0	4.5	-0.5		OK									2.5	3.0	1.5	1.5									
4.6	Implementierung von "Tidy Up EDE Data"	4.6	4.7	4.0	3.0	1.0		OK											4.0	3.0									
4.7	White-Box Testfälle ermitteln	4.3-4.7		4.0	4.0	0.0		OK													4.0	4.0							
4.8	Usability Improve/ Error Handling	4.1	4.9	2.0	2.0	-1.0		OK											1.0	2.0	1.0	0.0							
4.9	Help-Funktion erstellen	4.1		1.0	1.0	0.0	04.04.2016	OK													1.0	1.0							
5 Kontrollieren																													
5.1	White-Box-Test durchführen	4.7		3.0	3.0	0.0		OK													3.0	3.0							
5.2	Akzeptanz-Test durchführen	3.1		3.0	3.0	0.0		OK													1.0	1.0	2.0	2.0					
6 Auswerten																													
6.1	Schlussbericht verfassen			2.0	2.0	0.0	07.04.2016	OK																					
7 Diverses																													
7.1	Zeitplan IST-Zustand einfügen			1.0	1.0	0.0			0.1	0.1	0.1	0.1	0.1	0.1		0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	
7.2	Arbeitsjournal führen			1.0	1.2	0.1			0.1	0.1	0.1	0.1	0.1	0.1		0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	
7.3	IPA-Bericht führen			12.0	18.0	0.0			1.0	1.0	1.0	1.0	1.0	1.5		1.0	0.5	1.0	1.0	1.0	3.0	1.0	1.0	1.0	2.0	1.0	3.0	4.0	
7.4	Statusmeeting mit Auftraggeber			3.0	3.5	0.1		OK	0.5	1.0			0.5	0.5				0.5	0.5		0.5	0.5			1.0	1.0			
7.5	Gespräch mit Erstexpert (Kick-Off)			2.0	1.2	0.5		OK			2.0	1.2																	
7.6	Administratives/Organisatorisches			1.0	2.8	-0.7			0.1	0.5	0.1	0.1	0.1	0.3		0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	
7.7	Abgabe: drucken, binden, Upload			1.5	1.5	0.0																					1.5	1.5	
8 Puffer / Reserve																													
8.1	Pufferzeit			8.0	1.3				0.8	0.0	0.8	0.0	0.8	0.0		0.8	0.0	0.8	0.0	0.8	0.2	0.8	0.2	0.8	0.5	0.8	0.0	0.8	
Total:				80.0	80.0				8.1	8.2	8.1	7.0	11.1	9.0		8.6	8.3	8.1	8.0	7.6	8.0	7.6	8.0	8.1	8.0	7.1	8.3	6.6	7.2
x Soll-Zeit																													
x Ist-Zeit																													
Meilensteine																													

Abbildung 1-7 : Soll-Ist-Zeitplan

1.4.3 Tätigkeiten

ID	Tätigkeit	Erklärung
1.1	Projektauftrag lesen und verstehen	Die detaillierte Aufgabenstellung gemäss PkOrg lesen und verstehen
2.1	Tätigkeiten und Meilensteine definieren	Arbeitseinheiten finden und erklären und mit definierten Meilensteinen festlegen, zu welchem Zeitpunkt die verschiedenen Aufgaben oder Tätigkeiten fertig sein sollen.
2.2	Soll- Zeitplan mit geplantem Ablauf	In dem Soll-Zeitplan festgelegt, welche Tätigkeiten wann ausgeführt werden.
2.3	Vorlagen und IPA- Bericht aufbauen	Vorlagen für Arbeitsjournal, White-Box-Test und Akzeptanz-Test erstellen und IPA-Dokumentation gliedern
2.4	Arbeitsumgebung einrichten	Die Arbeitsumgebung wurde bereits vor der IPA ausgestattet
3.1	Akzeptanztest- Spezifikationen ermitteln	Mit Hilfe der Anforderungen wird den Akzeptanztest ermittelt, der später von einem Siemens Mitarbeiter durchgeführt wird
4.1	Bedienoberfläche erstellen und benennen	Die Bedienoberfläche wird im Excel grafisch dargestellt. Für die Oberfläche werden die ActiveX- Steuerelemente verwendet.
4.2	Struktogramme/Flussdiagramme erstellen	Die unübersichtlichen User-Interaktionen auf der Bedienoberfläche werden in Struktogrammen oder Flussdiagrammen dargestellt
4.3	Implementierung von "Modify", "CheckList", "CheckDevices", "Select, Deselect", "Actualise TsNet"	Für die Active X Steuerelemente „Modify“, „CheckList“, „CheckDevices“, „Actualise TsNet“ und „Select/Deselect“ werden Funktionen/Prozeduren übertragen
4.4	Implementierung von "Import EDE Data"	Für die Active X Steuerelement „Import“ wird eine Funktion übergeben und wird gemäss Struktogramm implementiert
4.5	Implementierung von "Delete EDE Data"	Für „Delete“ wird eine Funktion/Prozedur verliehen und gemäss Struktogramm implementiert.
4.6	Implementierung von "Tidy Up EDE Data"	Zu „Tidy Up“ wird eine Funktion/Prozedur ausgezeichnet und gemäss Struktogramm implementiert

4.7	White-Box Testfälle ermitteln	Mit dem Stand nach #4.3, #4.4, #4.5, #4.6 werden die einzelne White-Box- Testfälle ausspäht
4.8	Usability Improve/ Error Handling	Die Bedienoberfläche ist anwenderfreundlich und kann auf die ungewünschten Eingaben reagieren
4.9	Help-Funktion erstellen	Die Help Funktion wird integriert
5.1	White-Box-Test durchführen	Die Testfälle, die im Schritt #4.7 definiert worden sind, durchführen
5.2	Akzeptanz-Test durchführen	Die Testfälle, die im Schritt #3.1 definiert worden sind, durchführen
6.1	Schlussbericht verfassen	In einen Schlussbericht gehört die eigene Meinung der IPA-Ausführende und ihre Erlebnisse, was gut und was nicht so gut war
7.1	Zeitplan IST-Zustand einfügen	Der Zeitplan wird am jeden Tag mit dem IST-Stundenanzahl vervollständigt
7.2	Arbeitsjournal führen	Jeden Abend, bevor die IPA-Ausführende nach Hause geht, wird zuerst das Arbeitsjournal ausgefüllt
7.3	IPA-Bericht führen	Alle Arbeitsschritte und Bestandteile werden im IPA- Bericht festgehalten.
7.4	Statusmeeting mit Auftraggeber	Jeden zweiten Tag wird mit dem Auftraggeber ca. 30 min lang über die Arbeit konferiert
7.5	Gespräch mit Experten	Erster Besuchstag. An diesem Tag werden Informationen betreffend IPA ausgetauscht.
7.6	Administratives/Organisatorisches	Unter Administratives/Organisatorisches gehört Email schreiben, Backup machen und Termine planen
7.7	Abgabe: drucken, binden, Upload	Am 08.04 wird der IPA-Bericht mit Deckblatt 1 ausgedruckt, gebunden und dann dem Auftraggeber zugestellt. Der Zweitexperte erhält ein zweites Exemplar mit Deckblatt 2.
8.1	Pufferzeit	Insgesamt ein Tag Pufferzeit

Tabelle 4 : Tätigkeiten

1.4.4 Meilensteine

ID	Meilensteine	Erledigte Arbeitsschritte	Datum
2.2	Soll- Zeitplan mit geplantem Ablauf	1.1 bis 2.2	21.03.2016
4.1	Bedienoberfläche erstellen und benennen	2.3 bis 4.1	24.03.2016
4.4	Implementierung von "Import EDE Data"	4.2 bis 4.4	31.03.2016
4.9	Help- Funktion erstellen	4.5 bis 4.9	04.04.2016
6.1	Schlussbericht verfassen	5.1 bis 6.1	07.04.2016

Tabelle 5 : Meilensteine

1.5 Arbeitsjournal

1.5.1 Zweck des Arbeitsjournals

Das Arbeitsjournal dient zur Orientierung der Fachvorgesetzten und der Experten über den Verlauf des aktuellen Entwicklungsstands, allfällige Probleme und Hilfestellungen während der IPA.

1.5.2 Anwendungsbereich, Abgrenzung

Das Arbeitsjournal ist nur im Zusammenhang mit der IPA von Wert und ersetzt keine Dokumentation. Es beinhaltet die Tätigkeiten, wichtigen Erkenntnisse und Ergebnisse der Arbeit.

1.5.3 Aufbau

Jeden Morgen werden alle Tätigkeiten, die erledigt werden müssen, eingetragen. Am Abend werden die einzelnen Projekttasks mit der endgültigen Ist- Zeit und Status ergänzt.

- Unter „Hilfestellung“ wird beschrieben, ob die IPA- Ausführende von einer Person Informationen zum Projektauftrag erhalten hat.
- Unter „Projektstatus“ wird festgelegt, in welchem Bearbeitungsstand sich das Projekt befindet.
- Unter „Mailverkehr“ steht, an wen während der IPA E-Mails geschrieben wurden.
- Unter „Notizen“ wird die genaue Beschreibung des Arbeitsverlaufs festgehalten.
- Unter „Ausblick“ werden die Tätigkeiten festgehalten, welche gemäss Zeitplan für den kommenden Arbeitstag anstehen oder vom aktuellen Arbeitstag übernommen wurden.

1.5.4 Arbeitsjournale vom 21.03.2016 bis 08.04.2016

21.03.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016									
ID	Tätigkeit						Status		Soll	Ist	
1.1	Projektauftrag lesen und verstehen						abgeschlossen		0.5	0.5	
2.1	Tätigkeiten und Meilensteine definieren						abgeschlossen		2.0	2.0	
2.2	Soll- Zeitplan mit geplantem Ablauf						abgeschlossen		2.0	2.0	
2.3	Vorlagen und IPA-Bericht aufbauen						abgeschlossen		1.0	1	
2.4	Arbeitsumgebung einrichten						abgeschlossen		0	0	
7	Diverses						abgeschlossen		2.6	2.7	
							Arbeitsdauer		≈8.2		
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		E-Mail mit Excel-Tabelle des Zeitplans an Herr Müller									
Notizen											
<p>Heute habe ich mit meiner IPA gestartet. Zuerst habe ich die Aufgabenstellung gemäss PkOrg nochmals genauer durchgelesen und habe mir Notizen gemacht. Danach habe ich die einzelnen Arbeitseinheiten gefunden und kurz erklärt, was diese bedeuten. Mit Hilfe der IPERKA- Methode konnte ich die einzelnen Tätigkeiten, die von mir erstellt wurden, mühelos den entsprechenden Schritten zuordnen. Um 15:00 Uhr hatte ich das erste Statusmeeting mit meinem Auftraggeber und wir haben über die einzelnen Tätigkeiten diskutiert. Am Ende des Meetings musste ich ein paar kleine Änderungen im Zeitplan vornehmen. Am Anfang war ich mir bei der Erstellung von meinem Zeitplan unsicher, bin aber trotzdem mit meiner heutigen Leistung zufrieden.</p>											
Ausblick											
<p>Morgen werde ich die Akzeptanztest-Spezifikationen ermitteln und die TsNet Bedienoberfläche erstellen inkl. benennen.</p>											

22.03.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016									
ID	Tätigkeit						Status		Soll	Ist	
3.1	Akzeptanztest-Spezifikationen ermitteln						Abgeschlossen		2	2	
4.1	Bedienoberfläche erstellen und benennen						Abgeschlossen		2	2.5	
7	Diverses						Abgeschlossen		4.1	2.5	
							Arbeitsdauer		≈7		
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		E-Mail an Herr Müller wegen Treffpunkt									
Notizen											
<p>Der heutige Tag verlief grundsätzlich gut. Zuerst habe ich mit der Tätigkeit 3.1 angefangen und nachdem ich die abgeschlossen habe, erledigte ich die Tätigkeit 4.1.</p> <p>Bei 3.1, die Testfälle finden und definieren, hatte ich ein paar Schwierigkeiten, weil ich nie richtig in der Schule oder im ÜK gelernt habe, wie man Testfällen ermittelt und was beim Definieren von Testfällen wichtig ist. Ich denke das Ermitteln von den Akzeptanztest-Spezifikationen konnte ich mit Hilfe der Anforderungen von Herr Speckien gut lösen.</p> <p>Ich hatte heute mein erstes Gespräch mit dem Erstexperten, am Anfang war ich sehr nervös und hatte Angst, dass ich etwas Falsches sagen würde oder die Fragen, die Herr Müller gestellt hat, nicht beantworten könnte. Da Herr Müller so nett war, wurde mir klar, dass ich keine Angst vor diesem Gespräch haben soll. Er hat mir ein paar hilfreiche Tipps und Tricks gegeben, die ich während meiner IPA anwenden könnte. Zusätzlich hat er mich auf das Testkonzept aufmerksam gemacht und hat mir gesagt, dass ich noch ein Testkonzept erstellen soll, das bei der Bewertung von meiner IPA eine wichtige Rolle spielen wird.</p>											
Ausblick											
<p>Am 24.März werde ich Struktogramme erstellen und beginnen die Funktionen gemäss dem von mir erstellten Zeitplan zu implementieren. Am Donnerstagmorgen schreibe ich zuerst eine E-Mail an Herr Scheuber, wie er die IPA- Dokumentation gerne hätte, ausgedruckt oder per Email geschickt und am Schluss werde ich noch das gewünschte Testkonzept erstellen.</p>											

24.03.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016		24.03.2016							
Nr.	Tätigkeit						Status		Soll	Ist	
4.1	Struktogramme erstellen						Abgeschlossen		3	3	
4.3	Implementierung von Modify						Abgeschlossen		0,5	0,5	
4.3	Implementierung von Checklist						Abgeschlossen		1,5	1,5	
4.3	Implementierung von CheckDevices						In Bearbeitung		0,5	0,5	
2.4	Testkonzept erstellen						Abgeschlossen		0.5	1.0	
7	Diverses						Abgeschlossen		2,6	2,1	
							Arbeitsdauer			≈9	
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		E-Mail an Herr Scheuber									
Notizen											
<p>Heute habe ich zuerst das gewünschte Testkonzept erstellt, worauf mich Herr Müller beim ersten Besuchstag aufmerksam gemacht hat und meinen Zeitplan neu angepasst. Das erstellte Testkonzept habe ich auch noch mit Herr Speckien an unserem Statusmeeting besprochen und er hat mir gesagt, dass er mit dem von mir erstellten Testkonzept zufrieden wäre. Da ich die Akzeptanztest-Spezifikationen schon am 22.03.2016 abgeschlossen habe, hatte ich am 24.03.2016, 2h zur Verfügung, die ich für Akzeptanztest - Spezifikation eingeplant habe. Diese Zeit verwendete ich für das Erstellen des Testkonzepts. Nachdem ich die Tätigkeit 2.4 abgeschlossen habe, erstellte ich die einzelnen Struktogramme, die ich bei der Implementierung verwenden werde. Bei der Implementierung habe ich «Modify», «CheckList», «CheckDevices» abgeschlossen. Zusätzlich habe ich noch meine IPA- Dokumentation ergänzt.</p>											
Ausblick											
<p>Am Dienstagsmorgen werde ich die «Select», «Deselect», «Actualise TsNet» fertig implementieren und am Dienstagnachmittag fange ich denn mit der Implementierung von Import EDE an. Zusätzlich werde ich auch gleichzeitig dokumentieren</p>											

29.03.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016		24.03.2016							
Nr.	Tätigkeit						Status		Soll	Ist	
4.3	Implementierung von Select						Abgeschlossen		0.25	0.25	
4.3	Implementierung von Deselect						Abgeschlossen		0.25	0.25	
4.3	Implementierung von Actualise TsNet						Abgeschlossen		1	1	
4.4	Implementierung von Import EDE						In Bearbeitung		5	5	
7	Diverses						Abgeschlossen		2.1	1.5	
							Arbeitsdauer			≈8.3	
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		Keine									
Notizen											
<p>Durch die von mir erstellten Struktogramme konnte ich die Tätigkeiten 4.3 abschliessen. Aber während der Implementierung musste ich lange überlegen, ob das, was ich implementiert habe, auch so richtig ist. Bei der Codierung wurde auch das Geschriebene immer wieder kommentiert, damit am Ende jeder nachvollziehen kann, was codiert worden ist. Ausserdem musste ich immer die Zeit im Auge behalten. Natürlich darf nicht die ganze Zeit nur codiert werden, ich sollte auch den IPA Bericht ergänzen. Ich denke, den Teil 1 konnte ich mit wichtigen Informationen ergänzen. Beim Dokumentieren stellte sich immer wieder die Frage, ob das reicht, was ich geschrieben habe oder ob ich noch mehr dokumentieren muss/soll. Heute konnte ich mit 4.4 anfangen und ich hoffe, dass ich am 31.03 mit der Implementierung von Import EDE fertig werde.</p>											
Ausblick											
<p>Am 31.03.2016 werde ich die Implementierung von Import EDE abschliessen und mit der Tätigkeit 4.5 anfangen. Ich werde zusätzlich noch die IPA-Dokumentation führen. Die Dokumentation wird mit der Risikobeschreibung ergänzt, die zeigt was den Ablauf von IPA in Gefahr bringen könnte.</p>											

31.03.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016		24.03.2016		31.03.2016					
Nr.	Tätigkeit						Status		Soll	Ist	
4.3	Implementierung von CheckDevices						Abgeschlossen		1.0	1.0	
4.4	Implementierung von Import EDE						Abgeschlossen		2.0	2.0	
4.5	Implementierung von "Delete EDE Data"						In Bearbeitung		2.5	2.5	
7	Diverses						Abgeschlossen		2.6	2.5	
							Arbeitsdauer		≈8		
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		Keine									
Notizen											
<p>Am Morgen wurde mit der Implementierung von Tätigkeit 4.3 angefangen, da ich sie am 24.03.2016 nicht abschliessen konnte. Weil die Funktionalität von Check Devices abhängig von Import EDE war, musste ich warten bis ich die Import EDE fertig implementiert habe.</p> <p>Ich hatte grosse Mühe bei der Tätigkeit 4.4, weil ich eine .csv Datei mit der „Workbooks.Open“ Methode öffnen wollte. In der Hilfe werden Delimiter und von Parameter Format beschrieben. Als ich für das Format benutzerdefiniertes Trennzeichen und für Delimiter =";" eingab, wurde die .csv Datei trotzdem im Textmodus geöffnet und die durch Semikolon getrennten Werte nicht in die Excel Spalten übernommen. Durch lange Recherchen habe ich ein paar Tipps im Internet gefunden, dass man nur mit zwei Parametern das Problem lösen konnte. Als erstes musste ich den Dateinamen mit Pfad und als zweites den Parameter Local als True setzen, da das Argument Local als True Excel anweist, auf die Windows-Ländereinstellungen zurückzugreifen und dort ist das Semikolon im deutschsprachigen Raum standardmäßig als Listentrennzeichen hinterlegt.</p> <p>Beispiel aus dem Code: <code>Workbooks.Open Filename:=strFileName, local:=True</code></p> <p>Nach der fertigen Implementierung von Import EDE, konnte ich die Check-Devices auch abschliessen und das Zusammenspiel von beiden Funktionen überprüfen.</p> <p>Im heutigen Statusmeeting wurde noch über die einzelnen Tätigkeiten besprochen und Herr Speckien hat mir ein kurzes Feedback gegeben und erwähnt, dass ich auf dem richtigen Weg sei und dass er bis jetzt zufrieden ist mit meiner Arbeit.</p>											
Ausblick											
Morgen werde ich die Implementierung von „Delete EDE“ abschliessen und starte mit der Implementierung von „Tidy Up“											

01.04.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016		24.03.2016		31.03.2016					
Nr.	Tätigkeit					Status			Soll	Ist	
4.5	Implementierung von Delete EDE					Abgeschlossen			1.5	1.5	
4.6	Implementierung von Tidy Up					Abgeschlossen			4	3	
7	Diverses					Abgeschlossen			2.1	3.5	
						Arbeitsdauer			≈8		
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		Keine									
Notizen											
<p>Heute wurden die einzelnen Tätigkeiten abgeschlossen. Bei der Implementierung von Tidy Up brauchte ich weniger Zeit als ich eingeplant habe. Die übriggebliebene Zeit konnte ich beim Dokumentieren verwenden. Da leider das Dokumentieren nicht zu meiner Stärke gehört, muss ich jede freie Zeit für das Dokumentieren nutzen. Das Problem beim Dokumentieren ist, dass ich nicht weiss, wie viel ich dokumentieren muss. Ich meine man kann ja alles dokumentieren, jeden einzelnen Schritt, aber mir stellt sich immer wieder die Frage. Was ist wichtiger Qualität oder Quantität?</p>											
Ausblick											
<p>Am Montag werde ich die einzelnen White-Box Testfälle ermitteln und gleichzeitig meinen Code mit If und else Anweisungen ergänzen, damit der User besser informiert wird.</p>											

04.04.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016		24.03.2016		31.03.2016		04.04.2016			
Nr.	Tätigkeit					Status			Soll	Ist	
4.7	White-Box Testfälle ermitteln					Abgeschlossen			4.0	4.0	
4.8	Usability Improve / Error Handling					Abgeschlossen			1.0	2.0	
7	Diverses					Abgeschlossen			2.6	2.0	
						Arbeitsdauer			≈8		
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		Keine									
Notizen											
<p>Um die von mir erstellten Funktionen genau zu testen und die Werte in den Input Felder zu überprüfen, wurden die einzelnen Testfälle für die White-Box Tests ermittelt. Diese Testfälle werden von mir selbst durchgeführt. Ich stellte mir immer wieder die Frage, was könnte der Benutzer alles falsch machen? Mit dieser Frage im Hinterkopf habe ich ca. 40 Testfälle ermittelt.</p> <p>Das Usability Improve war für mich während der Implementierung schon ein Hauptpunkt, da ich bei der Implementierung schon viele Situationen fand, bei denen ich dachte, dass der User besser informiert werden müsste. Bei der Verwendung der Bedienoberfläche entdeckte ich immer wieder Unschönheiten, welche ich beseitigen musste.</p> <p>Beim Statusmeeting von heute habe ich das Endresultat von der Bedienoberfläche an Herr Speckien vorgestellt und erhielt positives Feedback betreffend Funktionalität von einzelnen Elementen. Mit Herr Speckien wurde noch über die Kommentare im Code geredet und es wurde festgelegt, dass ich den Header mit dem Erstelldatum, mit der Version, mit dem Namen der Funktion und mit meinem Namen ergänzen soll, wie im Beispiel unten.</p> <p>History: 24-03-2016 Fatma Yilmaz V.1 Document Creation for IPA</p> <p>Die kurze Erklärung von einzelnen Funktionen befindet sich oberhalb der erstellten Funktion, wie im Beispiel unten.</p> <p>' ----- '</p> <p>' Description: Opens the sheet in read only mode and disables all functions '</p> <p>' ----- '</p> <p>Sub ModifyWBopen()</p>											
Ausblick											
<p>Morgen werde ich mit der Tätigkeit Help-Funktion starten, danach werde ich selbstständig den White-Box- Test durchführen und der Akzeptanztest wird von einer Applikationsentwicklerin durchgeführt.</p>											

05.04.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016		24.03.2016		31.03.2016		04.04.2016			
Nr.	Tätigkeit					Status			Soll	Ist	
4.9	Help-Funktion erstellen					Abgeschlossen			1.0	1.0	
5.1	White-Box-Test durchführen					Abgeschlossen			3.0	3.0	
5.2	Akzeptanz-Test durchführen					In Bearbeitung			1.0	1.0	
7	Diverses					Abgeschlossen			2.1	3	
						Arbeitsdauer			≈8		
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		Keine									
Notizen											
<p>Heute wurde zuerst mit der Implementierung von Help-Funktion angefangen. Für die Help Funktion wurde eine. Pdf-Datei von Herr Speckien zur Verfügung gestellt. Auf der Bedienoberfläche im Bereich des Header findet man einen blau markierten Help Text. Dieser Help-Text wurde den mit der Pdf-Datei mit Hilfe von einer Funktion verbunden. Durch das Klicken auf den Help-Text wird die Seite 9 geöffnet und der User kann die Informationen über die Test-Spezifikationen finden.</p> <p>Beim Durchführen der White-Box-Tests musste ich ein paar Code Anpassungen machen, die notwendig waren. Die grössten Änderungen musste ich bei der Funktion von TidyUp() machen. Ich bin froh, dass ich viele Testfälle für den White-Box Test erstellt habe. Somit konnte ich die Funktionalität der Bedienoberfläche gut überprüfen. Die fehlgeschlagene White-Box Testfälle wurden in einem Nachtest wiederholt und hatten keine Fehler mehr, das heisst, alle Erwartungen wurden erfüllt.</p> <p>Am Nachmittag hat die Testperson eine kurze Einführung ins Thema von meinem Fachvorgesetzten und mir erhalten. Ich habe die Bedienoberfläche vorgestellt und erklärt, was die einzelnen Elemente auf der Bedienoberfläche eigentlich tun sollten und wo sie die Daten zum Testen finden wird. Zusätzlich bekam sie die Testdaten für den Donnerstag. Eine Testumgebung wurde für sie vorbereitet.</p> <p>In der Dokumentation habe ich das Kapitel Projektorganisation noch ein wenig überarbeitet. Die Kapitel Realisierung und Kontrollierung wurden entworfen.</p>											
Ausblick											
<p>Am Donnerstag wird der Akzeptanztest durchgeführt und ich hoffe, dass der Test nicht fehlschlägt. Ich werde am Donnerstag noch die IPA-Kurzfassung schreiben.</p>											

07.04.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016		24.03.2016		31.03.2016		04.04.2016		07.04.2016	
Nr.	Tätigkeit					Status			Soll	Ist	
5.2	Akzeptanz-Test durchführen					Abgeschlossen			2	2	
6.1	Schlussbericht verfassen					Abgeschlossen			2	2	
7	Diverses					Abgeschlossen				4	
						Arbeitsdauer			≈8		
Hilfestellung		Dokumentation an Berit Klinger zur Überprüfung der Vollständigkeit gesendet									
Projektstatus		Im Zeitplan									
Mailverkehr		Keine									
Notizen											
<p>Heute führte die Siemens Mitarbeiterin den Akzeptanz-Test aus und füllte dann das Testergebnis aus. Das Produkt hat den Test bestanden.</p> <p>Am Nachmittag habe ich die vorhandenen Dokumente zusammengefügt und fehlende Dokumente habe ich mir notiert. Danach habe ich die Notizen abgearbeitet, um die IPA-Dokumentation zu vervollständigen.</p> <p>Ich habe in der Dokumentation die Tabellen, Abbildungen und andere Elemente mit Überschriften versehen.</p> <p>Nachdem ich die IPA - Dokumentation noch mit Glossar, IPA-Kurzfassung und Schlussbericht ergänzt habe, sendete ich Die Dokumentation an Frau Berit Klinger, um die Vollständigkeit überprüfen zu lassen.</p>											
Ausblick											
Morgen werde ich noch weiter an der IPA-Dokumentation arbeiten											

08.03.2016	Tag	1	2	3	4	5	6	7	8	9	10
Statusmeetings		21.03.2016		24.03.2016		31.03.2016		04.04.2016		07.04.2016	
Nr.	Tätigkeit					Status			Soll	Ist	
7.1	Zeitplan IST-Zustand einfügen					Abgeschlossen			0.1	0.1	
7.2	Arbeitsjournal führen					Abgeschlossen			0.1	0.2	
7.3	IPA-Bericht führen					Abgeschlossen			4	4	
7.6	Administratives/Organisatorisches					Abgeschlossen			0.1	1.0	
7.7	Abgabe: drucken, binden, Upload					Abgeschlossen			1.5	1.5	
						Arbeitsdauer			≈7.2		
Hilfestellung		Keine									
Projektstatus		Im Zeitplan									
Mailverkehr		E-Mail an Hauptexperte: Informieren wegen Uploads im Pkorg									
Notizen											
Heute musste ich nur noch Korrekturen machen und meine IPA-Dokumentation verbessern. Ich bin eigentlich pünktlich fertiggeworden, obwohl ich den Aufwand von der Tätigkeit „Administratives/Organisatorisches“ grösser ausfiel als gedacht.											
Ausblick											
IPA – Dokumentation abgeben Nach der IPA : Web Summary und Vorbereitung auf IPA - Präsentation											

2. Teil 2

2.1 IPA Kurzfassung

2.1.1 Ausgangssituation

Das Projekt TsNet V2 beinhaltet die komplette Neuerstellung des Definitionsteils. Gefordert werden die Neuerstellung der Definition der Gerätetopologie und die Erstellung der Bedienoberfläche mit den zugehörigen Funktionen, also einem Excel- Arbeitsblatt zur Spezifikation der Geräte- und Netzwerktopologie. Neben dem Eingabedialog wurden die folgenden Funktionen gefordert: Führung des aktuellen Standes, Prüfung der Plausibilität der eingegebenen Daten untereinander und mit anderen Arbeitsblättern und Import von Daten aus einem .csv- File. Das Ziel dieser IPA ist eine einheitliche Bedienoberfläche und die Neuerstellung der zugehörigen Funktionen.

2.1.2 Umsetzung

Im Rahmen der individuellen praktischen Arbeit werden die Anforderungen, die in der Ausgangssituation genannt wurden, umgesetzt. Da für die Implementierung und Dokumentation der IPA-Ausführenden nur zehn Tage zur Verfügung stehen, wurde ein Zeitplan über zehn Tagen erstellt. Bei der Umsetzung wurde die Projektmanagementmethode IPERKA verwendet und alle zwei Tage ein Statusmeeting mit dem Auftraggeber durchgeführt.

In der Planungsphase wurde zuerst ein Testkonzept erstellt und in der Entscheidungsphase wurden die Akzeptanztest-Spezifikationen mit Hilfe der gewünschten Anforderungen ermittelt.

In der Realisierungsphase wurde die Bedienoberfläche, bei der die ActiveX-Steuerelemente verwendet wurden, im Excel grafisch dargestellt. Anstelle der bisherigen unübersichtlichen User-Interaktionen auf der Bedienoberfläche sind Struktogramme erstellt worden.

Anschliessend wurden in dieser Phase die einzelnen Anforderungen gemäss Requirement-Specification auf der Bedienoberfläche integriert: Im ersten Schritt werden hierfür(?) die Daten und die Konsistenz der Eingabefelder überprüft. Im zweiten Schritt kann der Benutzer, sofern die Überprüfung fehlerfrei ist, die EDE Daten bearbeiten, löschen und importieren. Wenn das Importieren von EDE Daten erfolgreich abgeschlossen ist, werden im dritten Schritt die EDE Daten auf Vollständigkeit und Richtigkeit überprüft.

In der Kontrollphase wurden von der IPA-Ausführenden die White Box Tests, die sie während der Implementierung ermittelt hat, ausgeführt. Der Akzeptanztest wurde am Schluss von einer Siemens Mitarbeiterin durchgeführt.

2.1.3 Ergebnis

Nach dem Abschluss dieser Arbeit sind alle Anforderungen gemäss Requirement-Specification erfüllt. Die Bedienoberfläche verfügt über 10 Buttons, 2 Links, 1 Label, das den Status von Import zeigt und unendlich vielen Eingabefeldern. Für eine gute Orientierung auf der Bedienoberfläche werden die jeweiligen Buttons, die im aktuellen Arbeitsschritt nicht verwendet werden dürfen, deaktiviert.

Der Hauptteil des Programmcodes befindet sich im Excel im Modul „mdl_SpecDevices“. Der Programmcode ist kommentiert und hat einen Kommentar-Header. Die erstellte Bedienoberfläche für die Tabelle „Spec-Devices“ ist intuitiv. Bei allfälligen Unklarheiten können sich die Benutzer mit Hilfe der Help-Funktion informieren.

2.2 Realisierung

2.2.1 Bedienoberfläche erstellen

Da im Vorfeld der IPA bereits eine Variantenanalyse zur Gestaltung der Bedienoberfläche durchgeführt worden ist, wurde in dieser IPA darauf verzichtet. Zudem soll die Bedienoberfläche so geplant werden, dass der User keine Schwierigkeiten beim Bedienen hat. Die Bedienoberfläche wurde benutzerfreundlich und leicht verständlich umgesetzt. Der Workflow ist klar ersichtlich und gibt den Benutzer Informationen über den Fortschritt.

Die Bedienoberfläche ist in drei Gruppen unterteilt. Im ersten Schritt werden die Daten und die Konsistenz der Eingabefelder überprüft. Im zweiten Schritt kann der Benutzer, wenn die Überprüfung fehlerfrei ist, die EDE Daten bearbeiten, löschen und importieren. Wenn das Importieren von EDE Daten erfolgreich abgeschlossen ist, werden im dritten Schritt die EDE Daten auf Vollständigkeit und Richtigkeit überprüft.

The screenshot shows the 'TsNet V2 - Spec Devices' interface. At the top, there's a 'List of BACnet devices' section with a 'Go To Overview' link and a 'Help' link. Below this, a workflow diagram shows the process: 'Modify' → 'Check List' → 'EDE Import pending' → 'Check Devices'. The 'Info' section shows 'In Progress' status and 'Datum: 05.04.2016 11:35:10'. The 'EDE Data' section has buttons for 'Import', 'Delete', 'Tidrup', and 'Compress'. The 'Testing' section has a 'Check comm' button. Below these sections is a table of BACnet devices.

Select	Alias Controller	Device-Name	Description	IP - Addr Node- ID	Dev-Inst	Network- No	Order - No	Network Connection	Date EDE import	Status
✓	Central	Cen120	Central functions	192.168.0.50	50		PXC3 E75A	IP-Controller		
✓	Room	Room053	Room functions	192.168.0.55	51	2	DXR2 E18	IP-Controller		
	Supervisor	Spr100	Supervisory functions	192.168.0.57	54	2	PXC100 E	MSTP-Router		
	Segment	Segm054	2 nd segment for room 053	180	180	2	DXR2 M18	MSTP-Controller		
	Router	ROUTER1	IP/MSTP Router	192.168.0.56	23	2	PXG3	LON-Router		

Abbildung 2-1 : Bedienoberfläche erstellen

Übersicht über die implementierten Workflowschritte:

1. Leeres Blatt: keine Einträge in der Geräteliste
2. {Modify}
 - a. Status: Working
 - b. Info: In Progress
 - c. Definieren von Alias Controller, Device-Name, Description, IP-Addr Node ID, Dev-Inst, Network-No, Order-No, Network Connection
3. {Check List}

Ist abhängig von dem Ergebnis

 - a. Überprüfung falsch:
 - Info : zeigt den Grund der Fehler Status: Error – List
 - Fahren Sie mit dem Fehler zu korrigieren und erneut zu überprüfen {Schritt 2}
 - b. Überprüfung richtig:
 - Info : Check List is OK Status: OK– List
 - Worksheet verlassen und Objekte oder Test Steps definieren
 - Weitere Details oder Geräte definieren{Schritt 2}
 - Weiterfahren mit definieren von Testumgebung{Schritt 4}

4. Für jeder Controller: EDE Daten importieren

Selektieren von Controller aus der Geräteliste, klicken auf {Import}

Ist abhängig von dem Ergebnis

a. Daten wurden richtig importiert:

- Info: Import is done Status: Import - Ok
- Date EDE import: aktuelles Datum Status Device: imported
 - Importieren von weitere Controller{Schritt 4}
 - Weitere Details oder Geräte definieren{Schritt 2}

b. Import fehlgeschlagen

- Info : zeigt den Grund der Fehler Status: Import– Error
- Date EDE Import: aktuelles Datum Status Device: -
 - Importieren von nächsten Controller{Schritt 4}
 - Korrigieren von Fehler {Schritt 2}
 - Importieren von einem anderen File

c. Import ist abgebrochen

- Info : zeigt den Grund der Fehler Status: Working
- Date EDE Import: aktuelles Datum Status Device: Import– Stopped
 - Importieren von nächsten Controller{Schritt 4}
 - Korrigieren von Fehler {Schritt 2}
 - Importieren von einem anderen File

5. EDE Daten bearbeiten

{Delete}, {Tidy Up}

6. {Check Devices}

Ist abhängig von dem Ergebnis

a. Überprüfung ist okay

- Info : the selected device is OK Status: OK-Device
 - Weitere Details oder Geräte definieren{Schritt 2}

b. Überprüfung falsch:

- Info : zeigt den Grund der Fehler Status: Error – Device
Fahren Sie mit dem Fehler zu korrigieren und erneut zu überprüfen{Schritt 2}
 - Korrigieren von Fehler {Schritt 2}

Erklärung: { } symbolisiert ein Button bei der Beschreibung der Workflowschritte.

2.2.2 Informationen zur Bedienoberfläche

Funktionen

Schaltflächenname	Objektname	Funktion	Wo?
Modify	Cmd_Modify	ModifyWBS	Mdl_SpecDevices
Check List	Cmd_CheckList	CheckList	Mdl_SpecDevices
		CheckDuplication	Mdl_SpecDevices
Check Devices	Cmd_CheckDevices	CheckDevices	Mdl_SpecDevices
		CheckDuplication	Mdl_SpecDevices
		CheckDevicesInEDE	Mdl_SpecDevices
Import	Cmd_Import	ImportEDE	Mdl_SpecDevices
		frmImportEDE	Formulare
		CheckFrmImportEDE	Mdl_SpecDevices
		appendingEDE	Mdl_SpecDevices
		replacingEDE	Mdl_SpecDevices
		openFilecsv	Mdl_SpecDevices
Delete	Cmd_Delete	Delete	Mdl_SpecDevices
Tidy Up	Cmd_TidyUp	TidyUp	Mdl_SpecDevices
Help	= 'Spec-Devices'!\$O\$3	GetHelp	Mdl_SpecDevices
[..]	Cmd_ActualiseTsNet	ActualiseTsNet	Mdl_SpecDevices
[✓]	Cmd_Select	cmd_Select_Click	Tabelle 19 (Spec-Devices)
[x]	Cmd_Deselect	cmd_Deselect_Click	Tabelle 19 (Spec-Devices)

Tabelle 6 : Funktionen

Felder

Definierte Bereichsnamen	Zellenbereichen	Beschreibung
cl_Info	= 'Spec-Devices'!\$E\$4	Die letzte detaillierte Status- oder Fehlerinformationen von allen aus diesem Blatt genannten Funktionen werden gezeigt
cl_Status	= 'Spec-Devices'!\$I\$4	Zeigt den Status der zuletzt ausgeführten Funktion
cl_Date	= 'Spec-Devices'!\$L\$4	Zeigt das Datum der zuletzt ausgeführten Funktion
cl_TsNetExecutablePath	= 'Spec-Devices'!\$A\$17	Zeigt den Pfad für die Ausführung von TsNet (RunTime)
cl_TsNetIP	= 'Spec-Devices'!\$J\$17	Zeigt die Ausgangs-IP-Adresse für TsNet auf dem Test-PC
cl_TsNetPort	= 'Spec-Devices'!\$K\$17	Zeigt den Ausgangs-Port für TsNet auf dem Test-PC
GoTo Overview	= 'Spec-Devices'!\$B\$3	Link zum Wechseln auf die Tabelle „Overview“

Tabelle 7 : Felder

Die Entwicklungsumgebung in VBA – Editor

Menü Ansicht/Projektexplorer. Hier wird das geladene Projekt TestTemplate.xlsm für TsNet V2 und dessen Objekte, Module und Formulare angezeigt. Der Teil der IPA sind nur die rot umrahmten Elemente und auf der rechten Seite wird angezeigt, was die IPA-Ausführende während der IPA implementiert hat.

VBAProject (TestTemplate.xlsm)

Microsoft Excel Objekte

- DieseArbeitsmappe** (rot umrahmt)
 - Sheet10 (Template Teststep)
 - Sheet12 (Step_WoBoost_VT)
 - Sheet14 (Step_WoBoost_TXT)
 - Sheet15 (Step_Boost_VT)
 - Sheet17 (Step_TestSheet_VT)
 - Sheet3 (Cover)
 - Sheet32 (Step_WoBoost_Trnd)
 - Sheet4 (Grundlagen)
 - Tabelle1 (Enum)
 - Tabelle10 (Step_WoBoost_Result)
 - Tabelle11 (Choices)
 - Tabelle12 (Spec-Objects)
 - Tabelle13 (Spec-Devices-)
 - Tabelle14 (Workflow)
 - Tabelle15 (Step_WoBoost_Script)
 - Tabelle16 (Spec- Objects V.1)
 - Tabelle17 (Config V.1)
 - Tabelle18 (Overview)
 - Tabelle19 (Spec-Devices)** (rot umrahmt)
 - Tabelle2 (Config)
 - Tabelle20 (Tabelle1)
 - Tabelle22 (InsertedData)
 - Tabelle3 (Template POOS)
 - Tabelle36 (EDE)
 - Tabelle37 (Tabelle18)
 - Tabelle4 (Overview V1)
 - Tabelle5 (Template QC-Test)
 - Tabelle6 (Help)
 - Tabelle7 (QC-Test)
 - Tabelle8 (Data POOS)
 - Tabelle9 (SDU)

Formulare

- frmImportEDE** (rot umrahmt)

Module

- mdl_Config
- mdl_Overview
- mdl_SpecDevices** (rot umrahmt)

workbook_open ()

cmd_Modify_Click ()
 cmd_ActualiseTsNet_Click ()
 cmd_Select_Click ()
 cmd_Deselect_Click ()
 cmd_CheckList_Click ()
 cmd_Import_Click ()
 cmd_Delete_Click ()
 cmd_TidyUp_Click ()
 cmd_CheckDevices_Click ()
 Worksheet_SelectionChange ()

cmd_ImportEDE_Click ()
 cmd_OpenFileName_Click ()
 cmd_CancelEDE_Click ()

ModifyWBopen ()
 ModifyWBS ()
 ActualiseTsNet ()
 CheckDuplication ()
 CheckList ()
 CheckDevices ()
 CheckDevicesInEDE ()
 ImportEDE ()
 CheckFrmImportEDE ()
 appendingEDE ()
 replacingEDE ()
 DeleteEDE ()
 openFilecsv ()
 TidyUp ()
 GetHelp ()

2.2.3 Struktogramme

Struktogramme sind grafische Darstellungen von Programmlogik für die Aufgaben und Lösungen. Um vor dem eigentlichen Programmieren einen Überblick zu bekommen, können Programmierer mit Hilfe von einem Struktogramm erkennen, wie das Programm aufgebaut ist oder aufgebaut sein soll. Ebenso lässt sich herausfinden, wie die Abläufe sein sollen, wo die einzelnen Schleifen und die Bedingungen sind. [3]

Im Umfang der IPA wird nur für die komplexeren Interaktionen ein Struktogramm erstellt:

- Cmd_ImportEDE_Click

Die folgenden Funktionen werden mit einem Struktogramm dargestellt:

- ActualiseTsNet
- CheckDevices
- CheckDevicesInEDE
- CheckDuplication
- CheckList
- ModifyWBOpen
- ModifyWBS
- ImportEDE
- CheckFrmlImportEDE
- appendingEDE
- replacingEDE
- DeleteEDE
- TidyUp
- GetHelp

Erklärung:

< > : umschliessen von Active-X Steuerelemente oder Variablen

:= definiert gleich

- frm = Userform
- cmd = Command Button
- txt = Text box

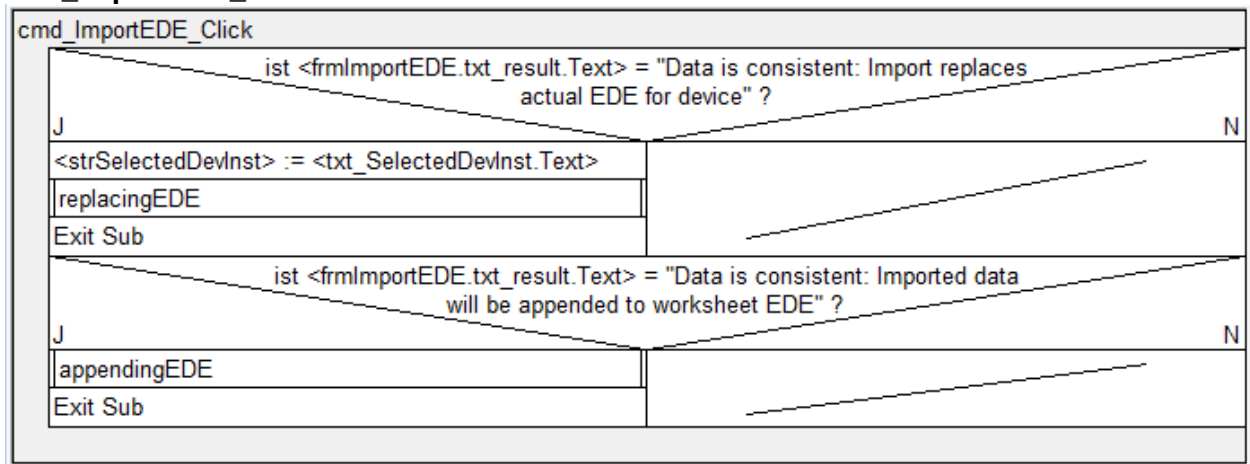
cmd_ImportEDE_Click

Abbildung 2-2 : Strg - Cmd_ImportEDE_Click

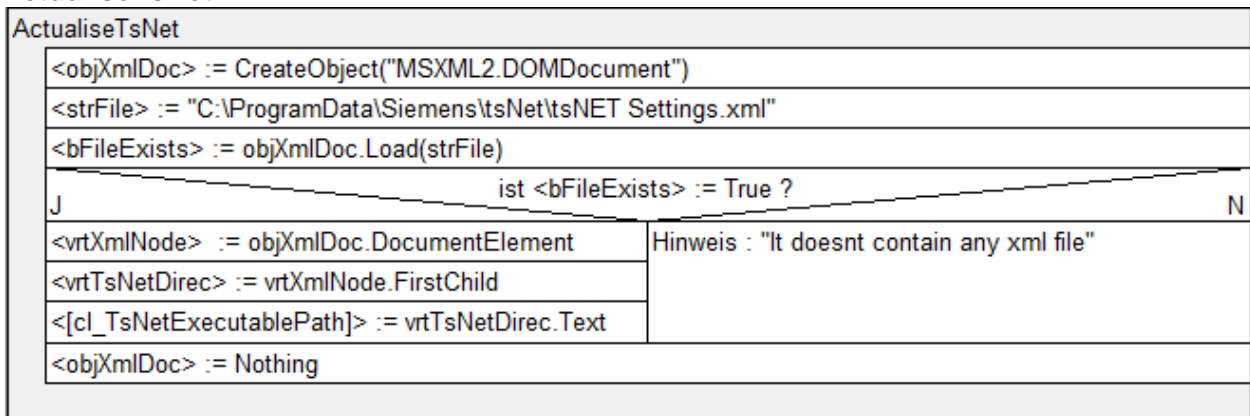
ActualiseTsNet

Abbildung 2-3 : Strg – ActualiseTsNet

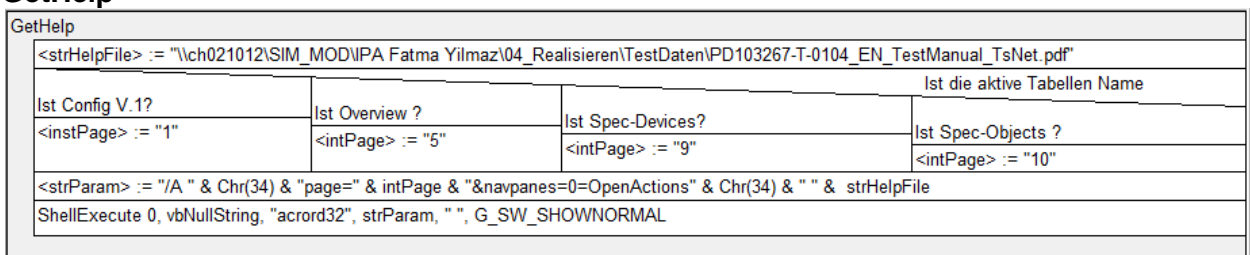
GetHelp

Abbildung 2-4 : Strg - GetHelp

TidyUp

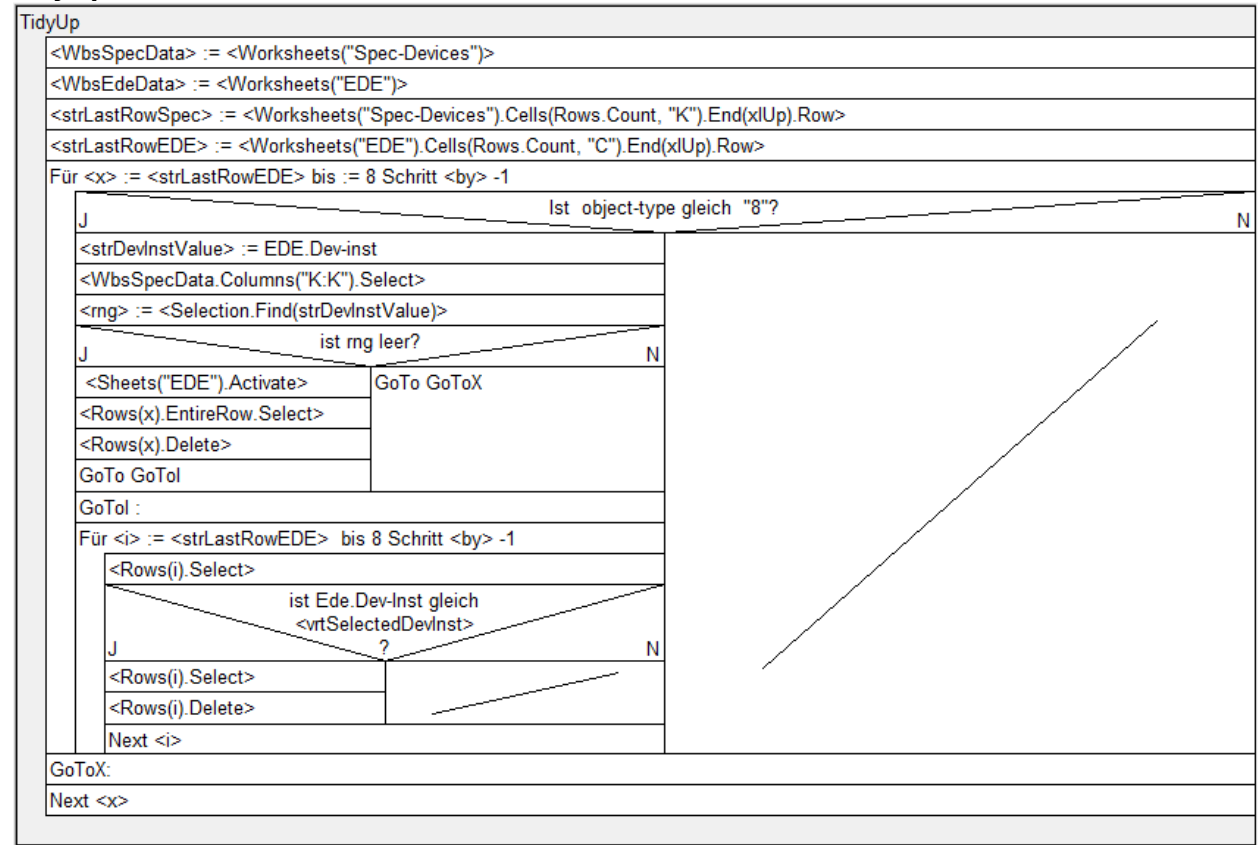


Abbildung 2-5 : Strg – TidyUp

DeleteEDE

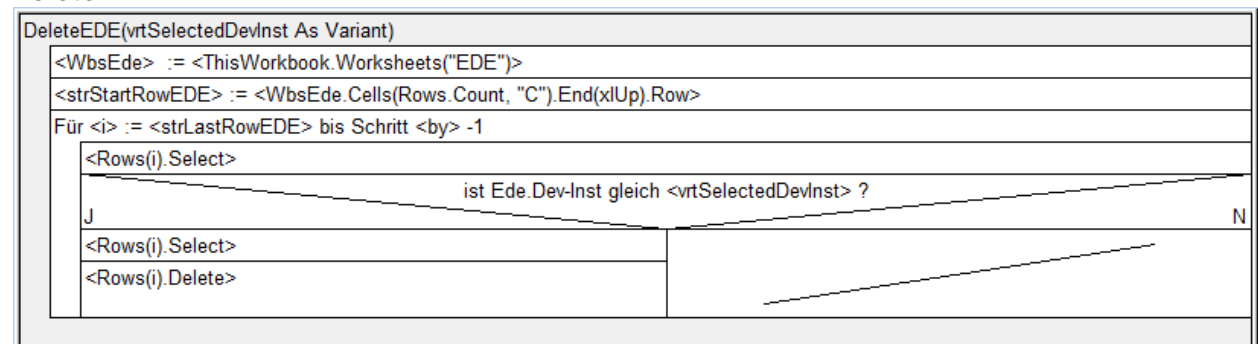


Abbildung 2-6 : Strg – DeleteEDE

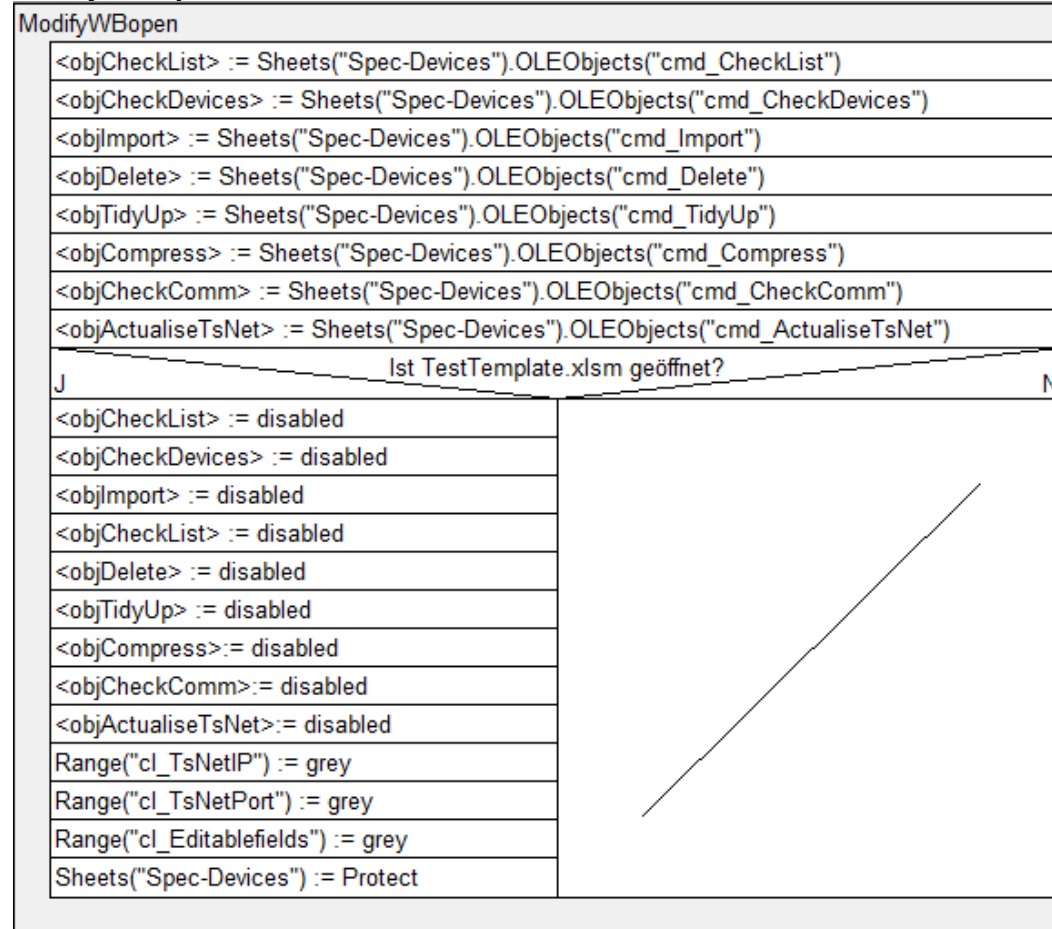
ModifyWBOpen

Abbildung 2-7 : Strg - ModifyWBOpen

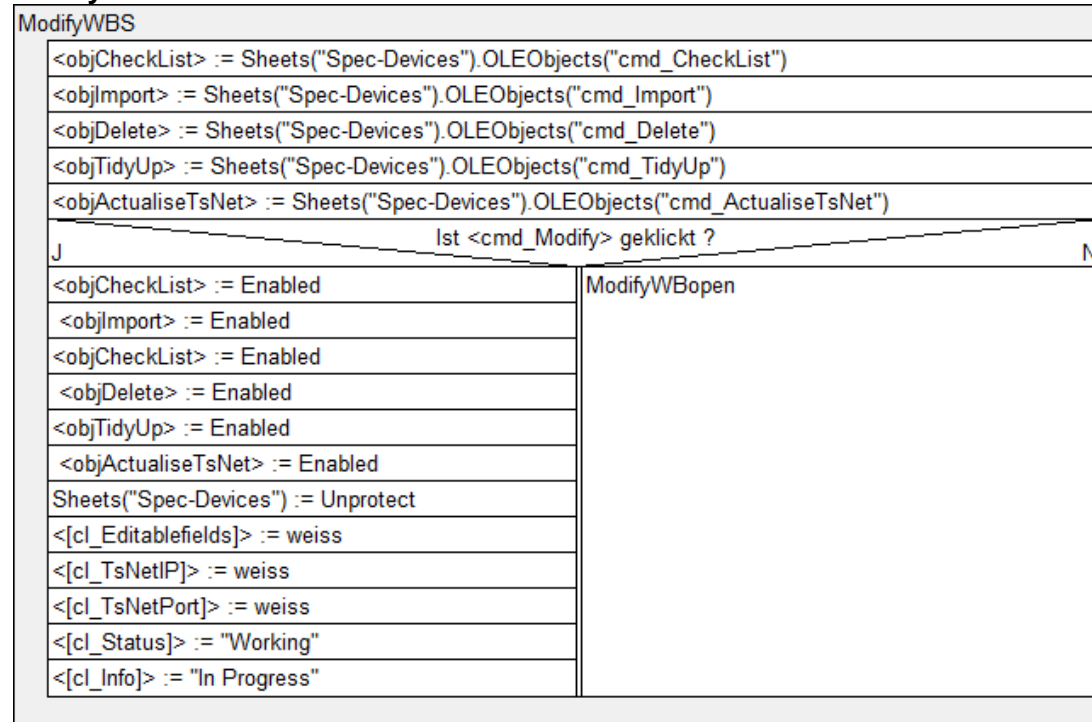
ModifyWBS

Abbildung 2-8 : Strg - ModifyWBS

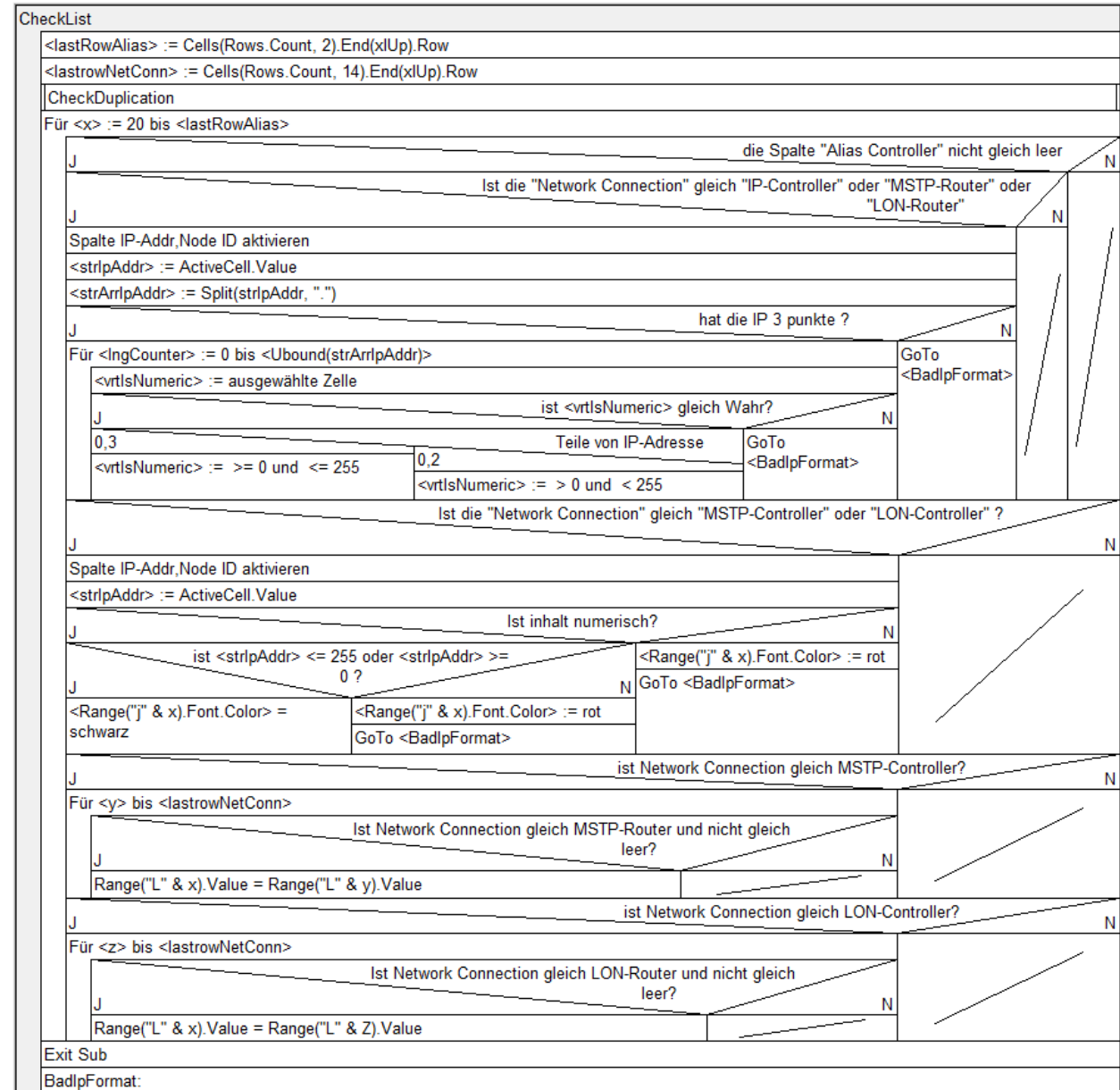
CheckList

Abbildung 2-9 : Strg - CheckList

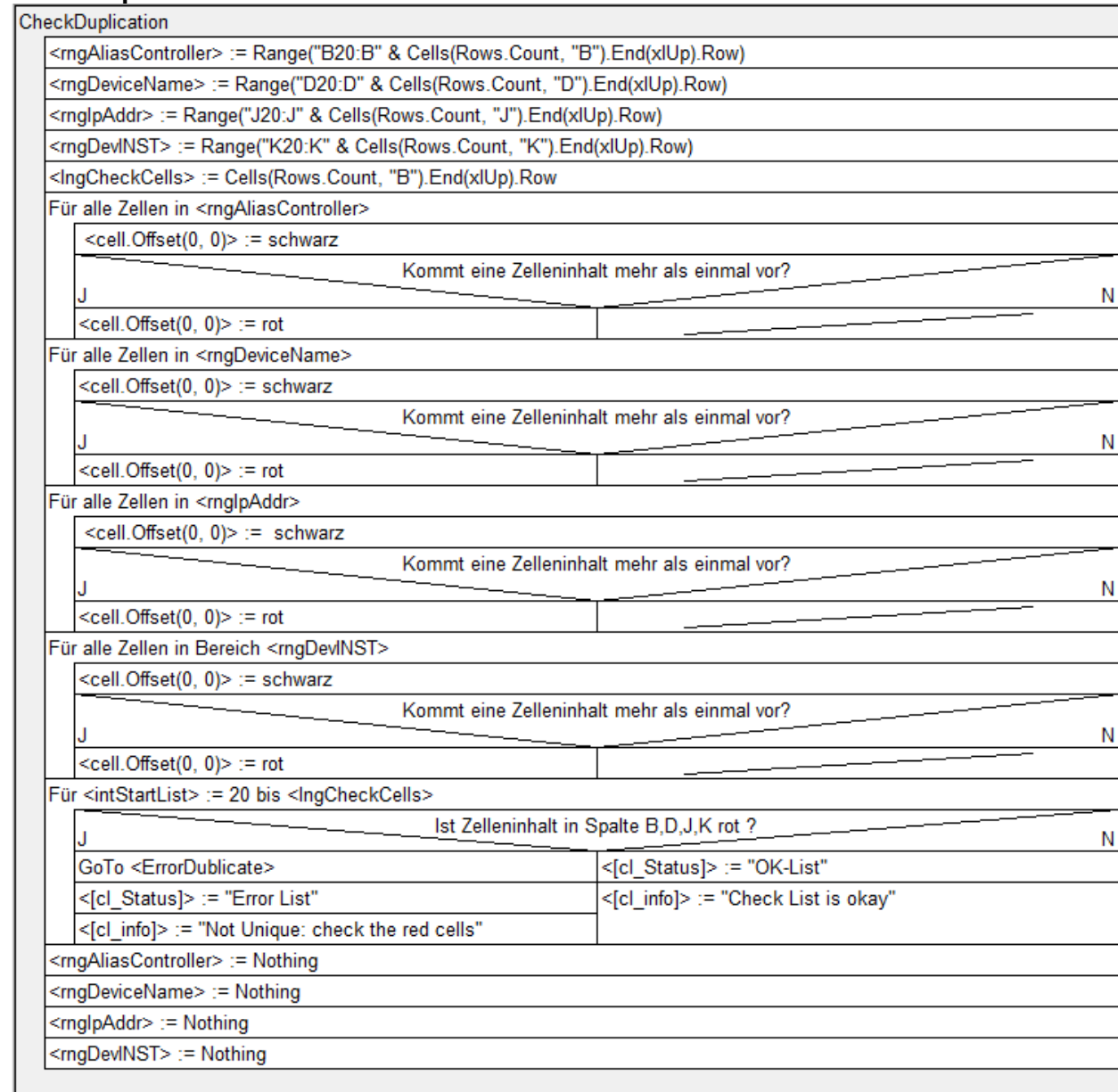
CheckDuplication

Abbildung 2-10 : Strg – CheckDuplication

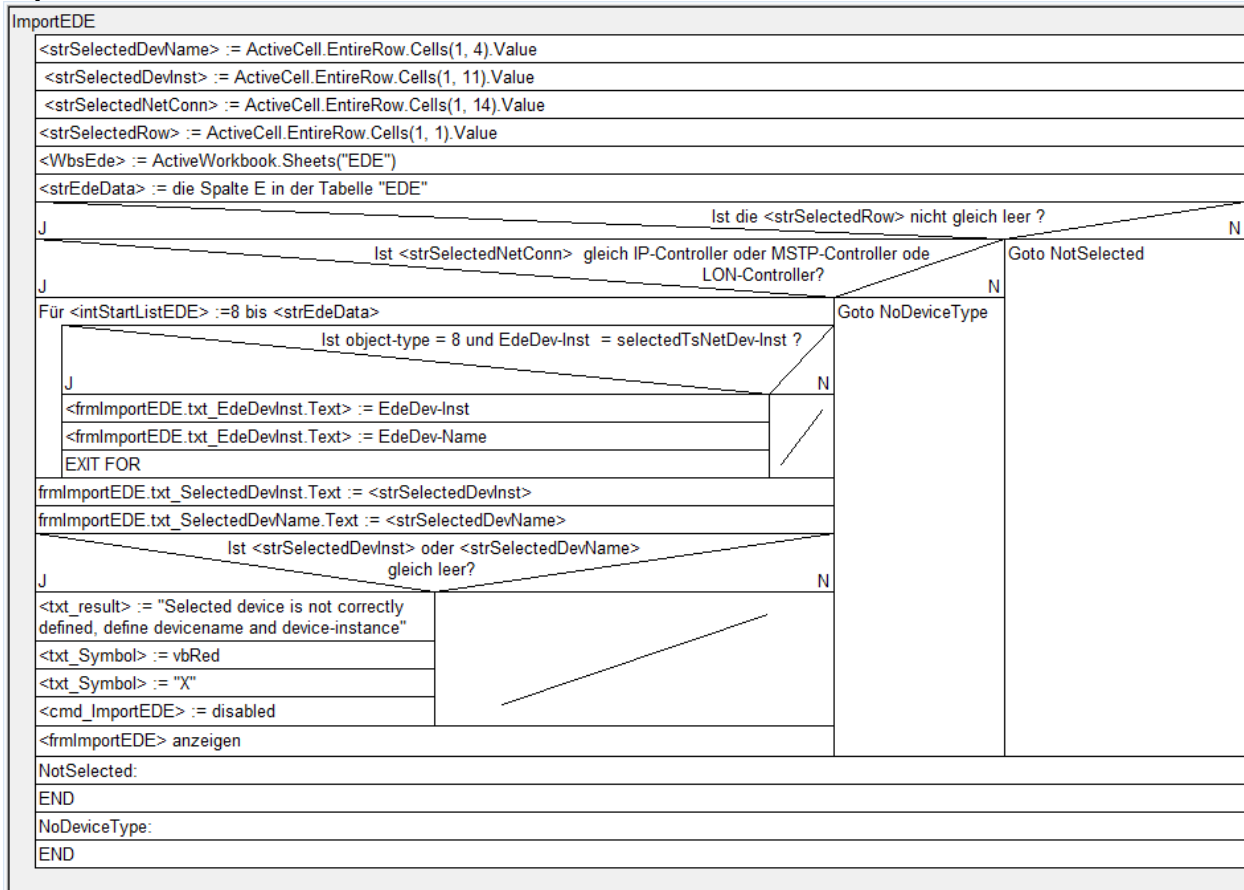
ImportEDE

Abbildung 2-11 : Strg – ImportEDE

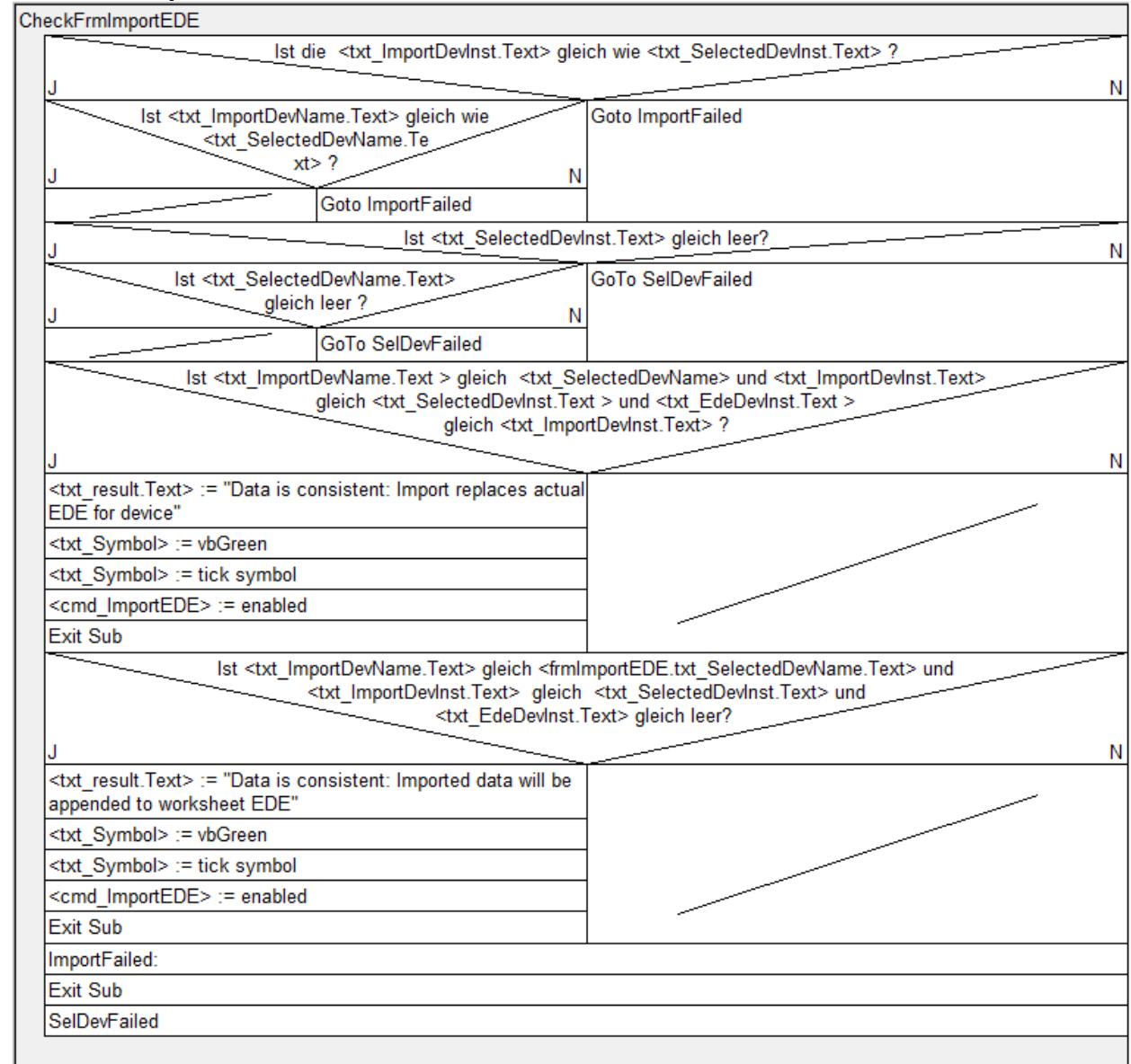
CheckFrmlImportEDE

Abbildung 2-12 : Strg – CheckFrmlImportEDE

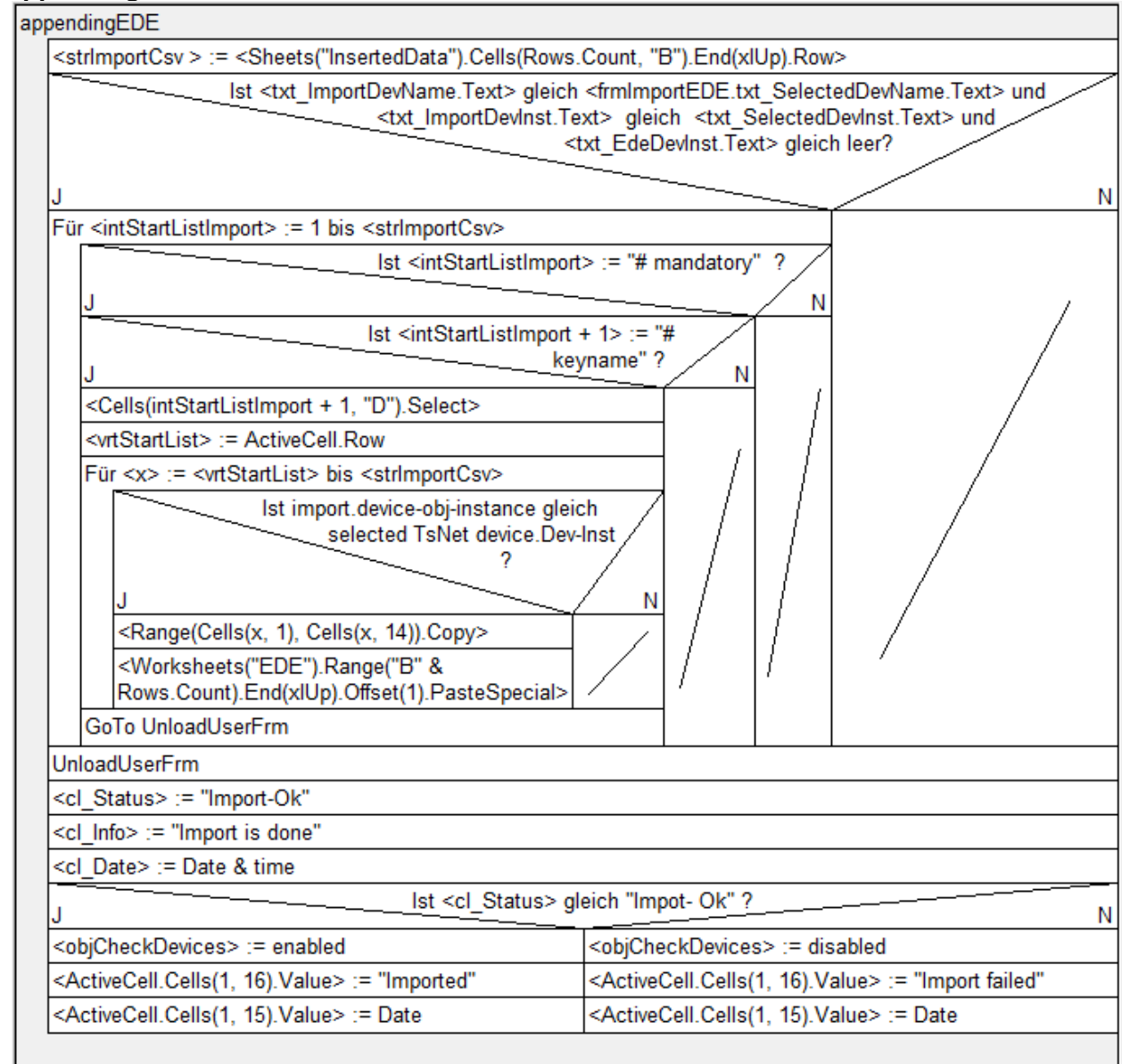
appendingEDE

Abbildung 2-13 : Strg – appendingEDE

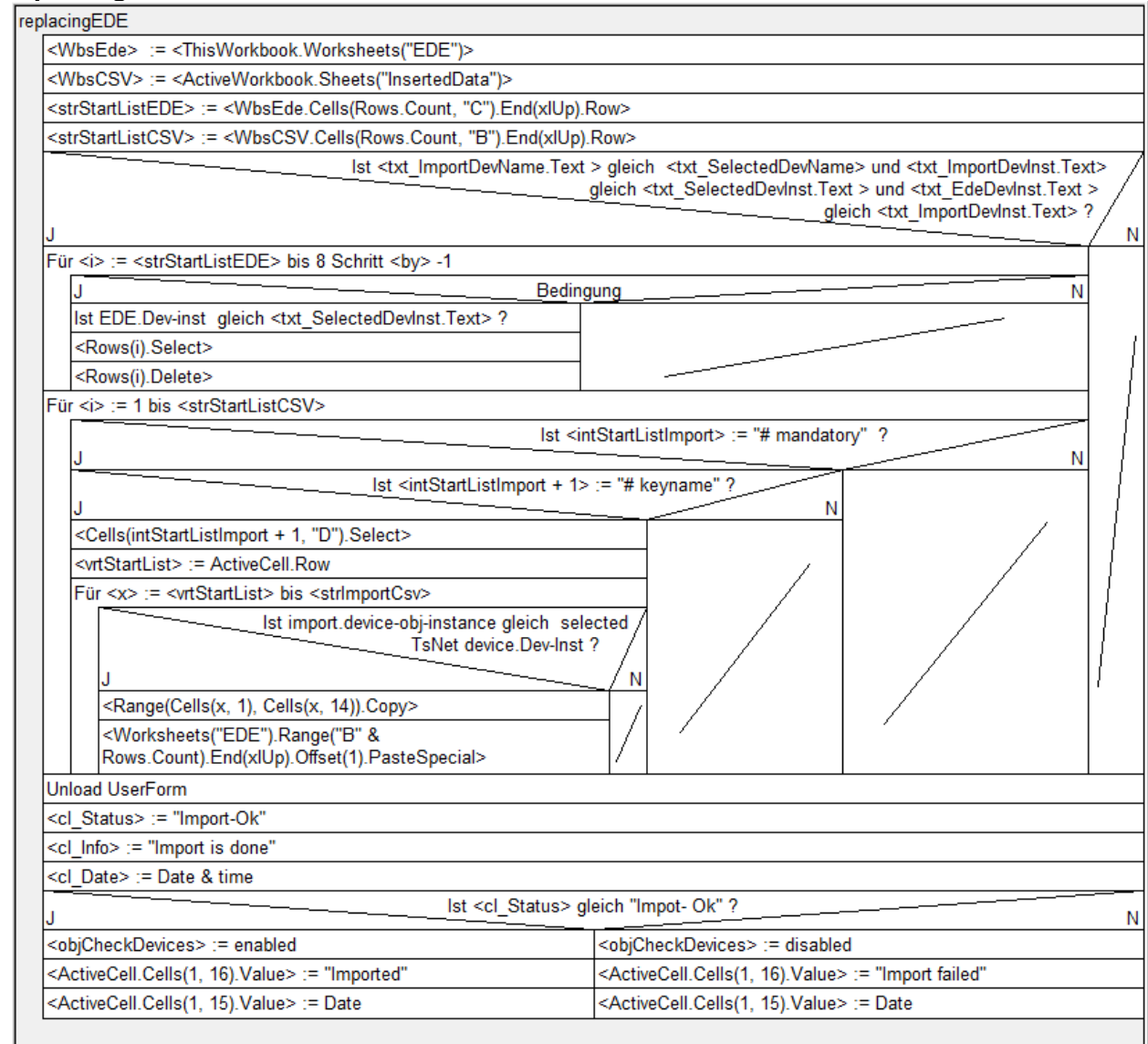
replacingEDE

Abbildung 2-14 : replacingEDE

CheckDevices

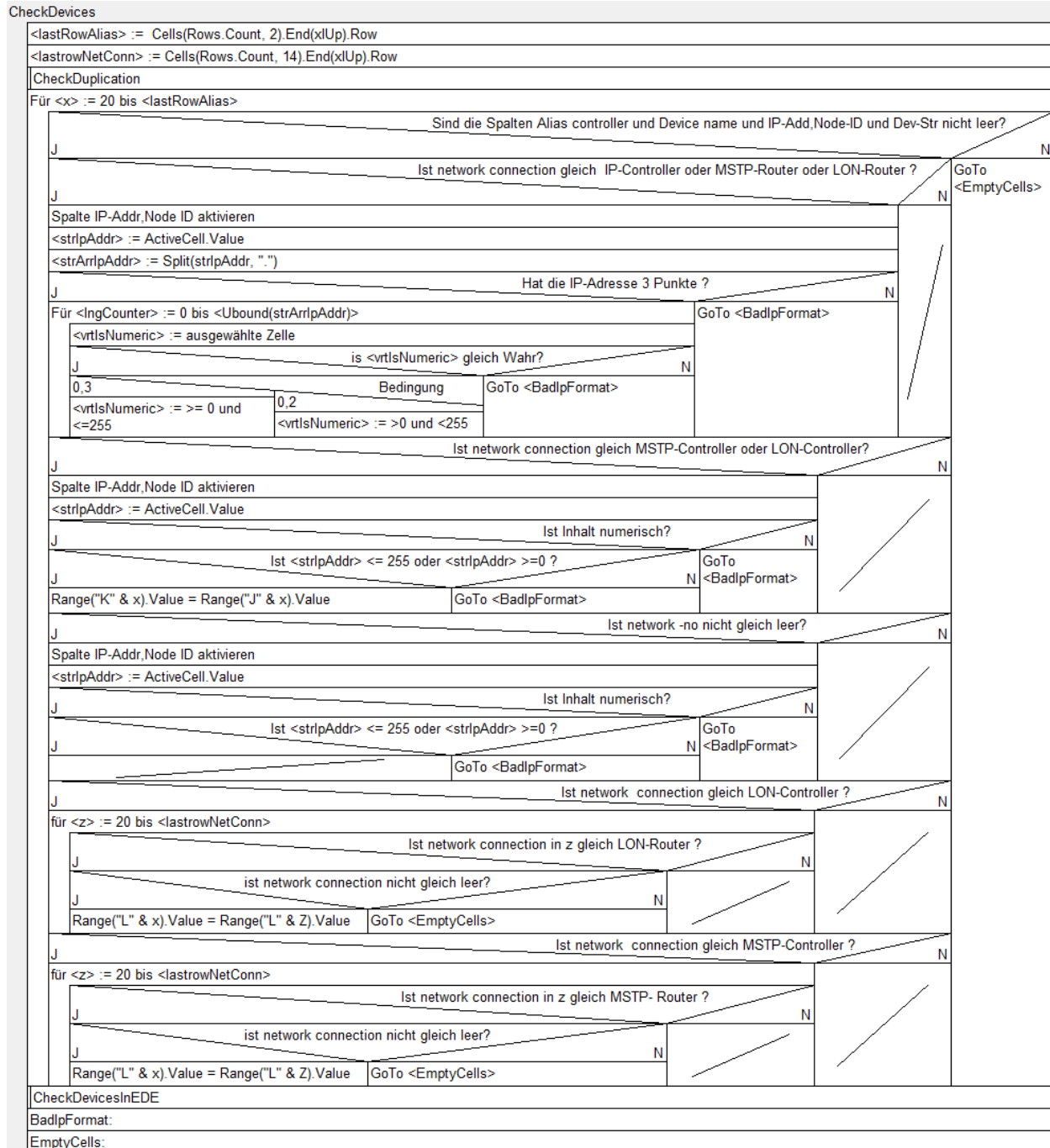


Abbildung 2-15 : Strg - CheckDevices

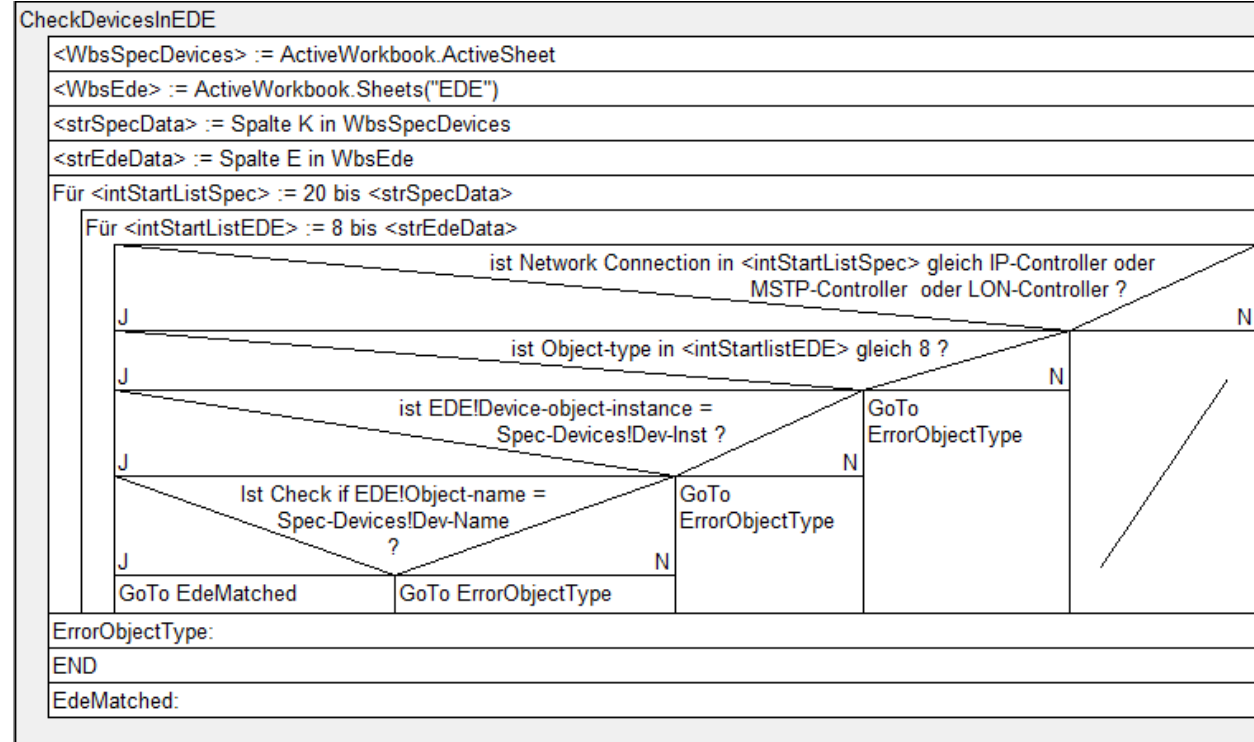
CheckDevicesInEDE

Abbildung 2-16 : Strg - CheckDevicesInEDE

2.2.4 Implementierung

Nachfolgend werden die wichtigsten Funktionen aufgelistet und erläutert.

ModifyWBopen ()

Die Funktion ModifyWBopen () wird durch Klicken auf den Button „Modify“ aufgerufen. Sie wechselt die Tabelle „Spec-Devices“ von einem schreibgeschützten Modus zu einem Lese-Schreib Modus. Alle Eingabefelder und Funktionen werden aktiviert. Die Funktion setzt den Status auf „Working“ und die Info auf „In Progress“

Beispiel:

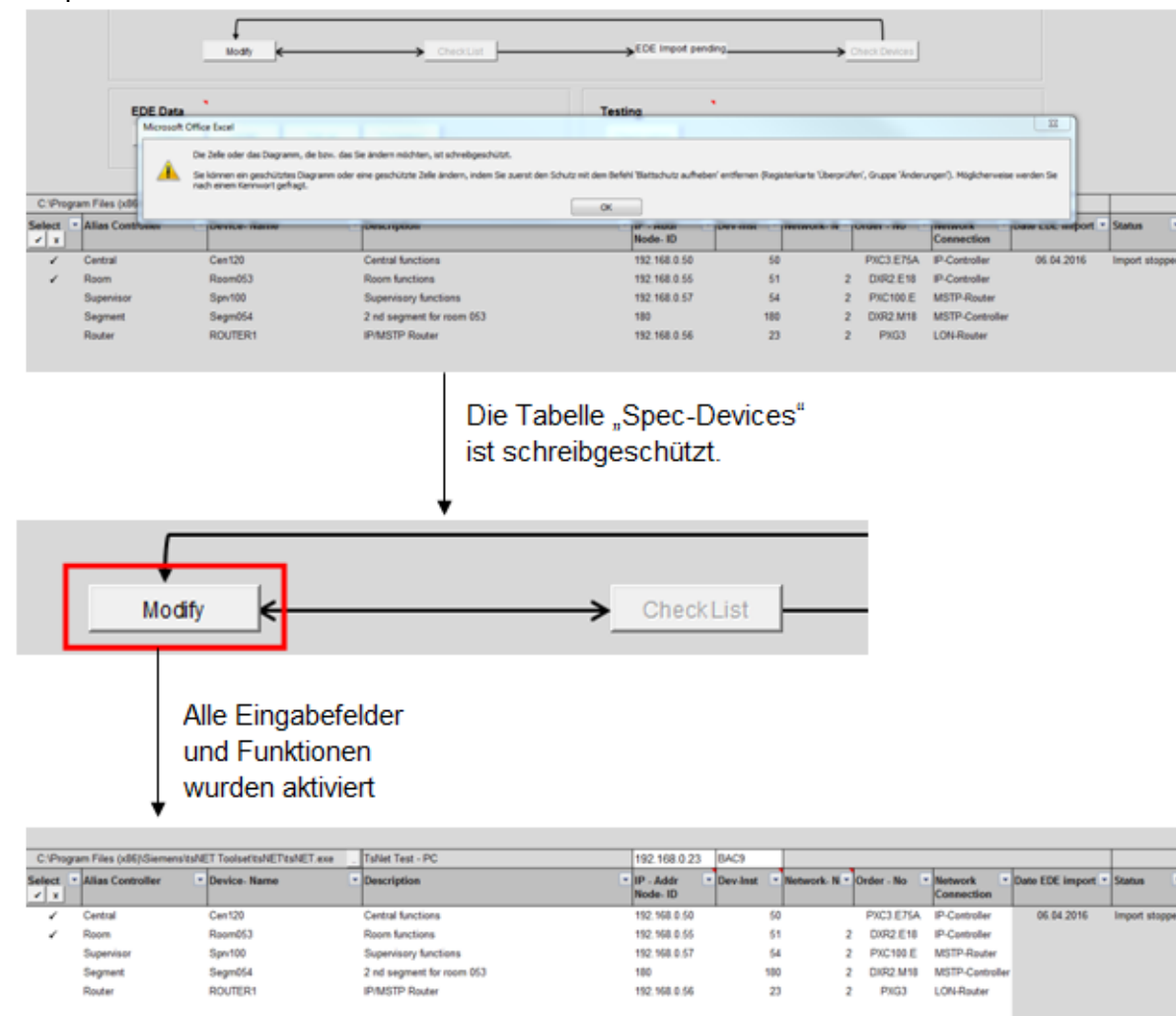


Abbildung 2-17 : Modify

CheckList ()

Die Funktion CheckList () wird durch Klicken auf den Button „Check List“ aufgerufen. Sie überprüft die Daten und die Konsistenz der Eingabefelder. Dabei werden die leeren Felder ignoriert, ausser in der Spalte „Alias Controller“: hier sind leere Zellen nicht erlaubt. Die Daten in den folgenden Spalten müssen einmalig sein:

- „Alias Controller“
- „Ip-Addr
- Node-ID“
- „Device- Name“
- „Dev-Inst“.

Wenn alle Überprüfungen fehlerfrei sind, wechselt die Tabelle „Spec-Devices“ zu einem Lese-Schreib Modus: Die Eingabefelder und Funktionen werden deaktiviert und der Status wird auf „OK-List“ sowie die Info auf „List is okay“ gesetzt. Die Buttons „Import“, „Delete“, „TidyUp“ sind aktiviert.

Falls die Überprüfung Inkonsistenzen ergeben hat, ändert sich der Status auf „Error-List “ und in der Info steht, was der Ursache des Fehlers ist. Die Buttons „Import“, „Delete“, „TidyUp“ sind deaktiviert.

Nachfolgend sind vier Screenshots, die Beispiele für Statusmeldungen zeigen.

Beispiel 1: Überprüfung ok

Result: Info: „List is Okay“, Status Device: „OK-List“

Select	Alias Controller	Device- Name	Description	IP - Addr Node- ID	Dev-Inst	Network- N	Order - No	Network Connection	Date EDE import	Status
✓	Central	Cen120	Central functions	192.168.0.50	50		PXC3.E75A	IP-Controller	06.04.2016	Import stopped
	Room	Room053	Room functions	192.168.0.55	51	2	DXR2.E18	IP-Controller		
	Supervisor	Sprv100	Supervisory functions	192.168.0.57	54	2	PXC100.E	MSTP-Router		
	Segment	Segm054	2 nd segment for room 053	180	180	2	DXR2.M18	MSTP-Controller		
	Router	ROUTER1	IP/MSTP Router	192.168.0.56	23	2	PXG3	LON-Router		

Abbildung 2-18 :ListOk Bsp1

Beispiel 2: Überprüfung failed

Info: „Check the column alias Controller: empty cells“, Status Device: „Error-List“

GoTo Overview Help

Info: Check the alias controller empty column Status Device: Error List Datum: 07.04.2016 09:49:34

Modify ↔ CheckList → EDE Import pending → Check Devices

EDE Data Testing

Import Delete Tidy up Compress Check comm.

Select	Alias Controller	Device-Name	Description	IP - Addr Node- ID	Dev-Inst	Network- N	Order - No	Network Connection	Date EDE import	Status
✓	Central	Cen120	Central functions	192.168.0.50	50		PXC3 E75A	IP-Controller	06.04.2016	Import stopped
	Room	Room053	Room functions	192.168.0.55	51	2	DXR2 E18	IP-Controller		
	Supervisor	Sprv100	Supervisory functions	192.168.0.57	54	2	PXC100 E	MSTP-Router		
	Segment	Segm054	2 nd segment for room 053	180	180	2	DXR2 M18	MSTP-Controller		
	Router	ROUTER1	IP/MSTP Router	192.168.0.56	23	2	PXG3	LON-Router		

Abbildung 2-19 : ErrorList Bsp2

Beispiel 3: Überprüfung failed

Resultat: Info: „Not Unique: Check the red cells“, Status Device: „Error-List“

GoTo Overview Help

Info: Not Unique: check the red cells Status Device: Error List Datum: 07.04.2016 09:49:34

Modify ↔ CheckList → EDE Import pending → Check Devices

EDE Data Testing

Import Delete Tidy up Compress Check comm.

Select	Alias Controller	Device-Name	Description	IP - Addr Node- ID	Dev-Inst	Network- N	Order - No	Network Connection	Date EDE import	Status
✓	Central	Cen120	Central functions	192.168.0.50	50		PXC3 E75A	IP-Controller	06.04.2016	Import stopped
	Room	Room053	Room functions	192.168.0.55	51	2	DXR2 E18	IP-Controller		
	Room	Sprv100	Supervisory functions	192.168.0.55	54	2	PXC100 E	MSTP-Router		
	Segment	Segm054	2 nd segment for room 053	180	180	2	DXR2 M18	MSTP-Controller		
	Router	ROUTER1	IP/MSTP Router	192.168.0.56	23	2	PXG3	LON-Router		

Abbildung 2-20 : ErrorList Bsp3

In diesem Beispiel werden die Verdoppelungen in der Spalte „Alias Controller“ und „Ip-Addr, Node-ID“ rot markiert.

Beispiel 4: Überprüfung failed

Resultat: Info: „Wrong Ip Format in column IP-Addr“, Status Device: „Error-List“

The screenshot shows the 'Info' section with the message 'Wrong Ip Format in Column IP-Addr' and 'Status Device: Error List'. Below this is a flowchart with buttons: Modify, CheckList, EDE Import pending, and Check Devices. The 'EDE Data' section has buttons: Import, Delete, Tidy up, and Compress. The 'Testing' section has a button: Check comm.

Select	Alias Controller	Device- Name	Description	IP - Addr Node- ID	Dev-Inst	Network- N	Order - No	Network Connection	Date EDE import	Status
✓	Central	Cen120	Central functions	192.168.0.50	50	PXC3 E75A		IP-Controller	06.04.2016	Import stopped
	Room	Room053	Room functions	192.168.0.55	51	DXR2 E18	2	IP-Controller		
	Supervisor	Sprv100	Supervisory functions	192.168.0.57	54	PXC100 E	2	MSTP-Controller		
	Segment	Segm054	2 nd segment for room 053	180	180	DXR2 M18		MSTP-Controller		
	Router	ROUTER1	IP/MSTP Router	192.168.0.56	23	PXG3	2	LON-Router		

Abbildung 2-21 : ErrorList Bsp4

Eine der implementierten Bedingungen war die Berücksichtigung von verschiedenen Netzwerkverbindungen:

- Wenn die Network Connection Mstp-Controller oder Lon-Controller ist, dann sollte die Ip-Addr, Node-ID ≤ 255 sein.
- Wenn die Network Connection IP-Controller, MSTP- Router oder LON-Router ist, dann hat die Zeile Ip-Addr, Node-ID das IP – Format 0.0.0.0 mit ≤ 255 .

CheckDevices ()

Die Funktion CheckDevices () wird durch Klicken auf den Button „Check Devices“ aufgerufen. Sie überprüft die Daten und die Konsistenz der Eingabefelder und EDE. Die Daten in den folgenden Spalten müssen einmalig und dürfen nicht leer sein:

- „Alias Controller“
- „Ip-Addr,
- Node-ID“,
- „Device- Name“,
- „Dev-Inst“.

Wenn die Spalte Network Connection „IP-Controller“, „MSTP-Controller“ oder LON-Controller ist, sucht die Funktion in der Tabelle „EDE“ nach folgenden Zeilen:

- EDE.Object-type = 8
- EDE.Device obj.-instance = Spec-Devices.Dev-Inst
- EDE.Object-name = Spec-Devices.Device Name

Wenn die obengenannten Zeilen gefunden und alle Überprüfungen fehlerfrei sind, wechselt die Tabelle „Spec-Devices“ zu einem Lese-Schreib Modus: Die Eingabefelder und Funktionen werden deaktiviert und der Status wird auf „OK-Device“ sowie die Info auf „all devices are ok OK“ gesetzt.

Falls die Überprüfung Inkonsistenzen ergeben hat, ändert sich der Status auf „Error-Device “ und in der Info steht, was der Grund von Fehler ist. Die Buttons „Import“, „Delete“, „TidyUp“ sind deaktiviert.

Nachfolgend sind drei Screenshots, die Beispiele für Statusmeldungen zeigen

Beispiel 1: Überprüfung ok

Result: Info: „all devices are Okay“, Status Device: „OK-Devices“

Select	Alias Controller	Device- Name	Description	IP - Addr Node- ID	Dev-Inst	Network- N	Order - No	Network Connection	Date EDE import	Status
✓	Central	Cen120	Central functions	192.168.0.50	50	PXC3.E75A	2	IP-Controller	07.04.2016	Imported
	Room	Room053	Room functions	192.168.0.55	51	DXR2.E18	2	LON-Router		
	Supervisor	Sprv100	Supervisory functions	192.168.0.57	54	PXC100.E	2	MSTP-Router		
	Segment	Segm054	2 nd segment for room 053	192.168.0.58	180	DXR2.M18	2	LON-Router		
	Router	ROUTER1	IP/MSTP Router	192.168.0.56	23	PXG3	2	MSTP-Router		

Abbildung 2-22 : OkDevices Bsp1

Beispiel 2: Überprüfung failed

Result: Info: „There are empty cells“, Status Device: „Error-Devices“

GoTo Overview Help

Info : There are empty cells Status Device: Error-Devices Datum: 07.04.2016 16:22:59

Modify CheckList EDE Import pending Check Devices

EDE Data

Import Delete Tidy up Compress

Testing

Check comm

Select	Alias Controller	Device-Name	Description	IP - Addr Node- ID	Dev-Inst	Network- N	Order - No	Network Connection	Date EDE import	Status
✓	Central	Cen120	Central functions	192.168.0.50	50	PXC3.E75A	IP-Controller	07.04.2016	Imported	
	Room	Room053	Room functions	192.168.0.55	51	DXR2.E18	LON-Router			
	Supervisor	Sprv100	Supervisory functions	192.168.0.57	54	PXC100.E	MSTP-Router			
	Segment	Segm054	2 nd segment for room 053	192.168.0.58	180	DXR2.M18	LON-Router			
	Router	ROUTER1	IP/MSTP Router	192.168.0.56	23	PXG3	MSTP-Router			

Abbildung 2-23 : ErrorDevices Bsp2

Beispiel 3: Überprüfung failed

Info: „Wrong IP Format in Column IP-Addr“, Status Device: „Error-Devices“

GoTo Overview Help

Info : Wrong Ip Format in Column IP-Addr Status Device: Error-Devices Datum: 07.04.2016 16:26:21

Modify CheckList EDE Import pending Check Devices

EDE Data

Import Delete Tidy up Compress

Testing

Check comm

Select	Alias Controller	Device-Name	Description	IP - Addr Node- ID	Dev-Inst	Network- N	Order - No	Network Connection	Date EDE import	Status
✓	Central	Cen120	Central functions	192.168	50	PXC3.E75A	IP-Controller	07.04.2016	Imported	
	Room	Room053	Room functions	192.168.0.55	51	DXR2.E18	LON-Router			
	Supervisor	Sprv100	Supervisory functions	192.168.0.57	54	PXC100.E	MSTP-Router			
	Segment	Segm054	2 nd segment for room 053	192.168.0.58	180	DXR2.M18	LON-Router			
	Router	ROUTER1	IP/MSTP Router	192.168.0.56	23	PXG3	MSTP-Router			

Abbildung 2-24 : ErrorDevices Bsp3

ImportEDE()

Die Funktion ImportEDE() wird durch Klicken auf den Button „Import“ aufgerufen. EDE Import wird durch die Auswahl einer TsNet Device in der Tabelle „Spec-Devices“ gestartet und funktioniert nur bei Auswahl eines Controllers.

Ein Controller ist ausgewählt:

- Wenn ein Häkchen in der Spalte „A“ sichtbar ist oder
- Wenn eine Zeile aus der Geräteliste ausgewählt ist oder
- Wenn ein Feld in der Auswahl Zeile ausgewählt ist

Die Auswahl eines Controllers ist ungültig:

- Wenn kein Gerät ausgewählt ist
- Wenn der Auswahl kein Typ von IP-Controller, MSTP-Controller oder LON-Controller ist

Wenn die Bedingungen erfüllt sind, wird das Sub Menu „Import EDE“ geöffnet.

The screenshot shows the 'Import EDE' dialog box with the following fields and sections:

- Import File:** Contains 'Filename' (Y:\IPA Fatma Yilmaz\99_Diverses\TestDaten\Cen120.csv), 'Dev-Inst:' (50), and 'Device Name:' (Cen120).
- Selected Device:** Contains 'Dev-Inst:' (50) and 'Device Name:' (Cen120).
- Device in EDE:** Contains 'Dev-Inst:' (50) and 'Device Name:' (Cen120).
- Information:** A checkbox labeled 'Data is consistent: Import replaces actual EDE for device' is checked.
- Buttons:** 'Import' and 'Cancel' buttons at the bottom.

Annotations on the left side of the dialog box:

- Bereich: Selected Device:** Points to the 'Selected Device' section.
- Bereich: Information:** Points to the 'Data is consistent' checkbox.

Annotations on the right side of the dialog box:

- Bereich: Import File:** Points to the 'Import File' section.
- Bereich: Device in EDE:** Points to the 'Device in EDE' section.
- Bereich: Import:** Points to the 'Import' and 'Cancel' buttons.

Abbildung 2-25 : SubMenu-ImportEDE

Im Bereich „Import – File“:

- Die EDE- Datei wird ausgewählt
 - .Csv wird konvertiert
 - EDE-Format überprüft
 - Ein EDE-File:
 - Beginnt mit einer „#“
 - Die folgenden Zeilen enthalten einige Header-Informationen.
 - Nach der Header-Informationen beginnt die nächste Zeile mit "# mandatory" , „mandatory“ , „mandatory“
 - Nach der Zeile „# mandatory“ startet die Zeile mit „#keyname“ , „device obj.instance“ , „object-name“
 - Nach der „# keyname“ sind die EDE- Daten aufgelistet
 - Wenn EDE-Format richtig
 - Die Zeile mit Object-type = 8 und dazugehörige DevObjInstance ausgelesen
 - Die ausgelesene DevObjInstance wird mit der selected Device Dev-Inst verglichen
 - Wenn das stimmt wird im Feld
 - Dev-Inst : device obj.-instance aus der Csv-Datei angezeigt
 - Device Name : object- name aus der Csv-Datei angezeigt
 - Wenn nicht stimmt
 - Dev-Inst = leer
 - Device Name = leer
 - Wenn EDE- Format falsch
 - Wird im Bereich Information
 - Message: „No EDE file, no import possible“ Symbol: „X“
 - Import Button ist deaktiviert

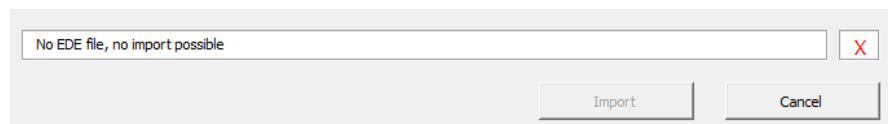


Abbildung 2-26 : WrongEdeFormat

Im Bereich „Selected Device“:

- Die ausgewählte Dev-Inst und Device-Name aus der Tabelle „Spec-Devices“ wird angezeigt.

Im Bereich „Device in EDE“:

- In der Tabelle „EDE“ wird die Zeile mit Object-type = 8 und die dazugehörige DevObjInstance ausgelesen
- Die ausgelesene DevObjInstance wird mit der selected Device Dev-Inst verglichen
 - o Wenn das stimmt, wird im Feld
 - Dev-Inst : device obj.-instance aus der Csv-Datei angezeigt
 - Device Name : object- name aus der Csv-Datei angezeigt
 - o Wenn das nicht stimmt, wird im Feld
 - Dev-Inst = leer
 - Device Name = leer

Unterfunktion: CheckFrmlImportEDE ()

Die Unterfunktion CheckFrmlImportEDE () liefert automatisch je nach Situation die folgenden Symbole und Meldungen:

Situation	Symbol	Message
Import file.Dev-Inst <> Selected TsNet device.Dev-Inst or Import file.Dev-Name <> Selected TsNet device.Dev-Name	X	Selected device is not equal to Import device, no import possible. {Import} Button is disabled
Selected TsNet device.Dev-Inst is empty or Selected TsNet device.Dev-Name is empty	X	Selected device not correctly defined, no import possible. {Import} Button is disabled
Import file is no EDE file	X	No EDE file, no import possible {Import} 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} 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} button is enabled.
		Die Unterfunktion AppendingEDE wird aufgerufen (Erläuterung dazu siehe unten)

Tabelle 8 : Unterfunktion-CheckFrmlImportEDE

Unterfunktion: replacingEDE ()

- in der ausgewählten Import-file wird nach Zeilen Import.Dev-Inst = selectedDev-Inst gesucht. Gefundene Zeilen werden in die Tabelle „EDE“ hinzugefügt
- Nach dem erfolgreichen Import wird in die Zeile [Datum EDE Import] in der Tabelle „Spec-Devices“ das aktuelle Datum hinzugefügt und Status auf „imported“ gesetzt.

Unterfunktion: AppendingEDE ()

- Alle Zeilen in der Tabelle „EDE“, wo EDE.Dev-Inst = selectedDev-Inst sind, werden aus der Tabelle „EDE“ gelöscht.
- in der ausgewählten Import-file wird nach Zeilen Import.Dev-Inst = selectedDev-Inst gesucht. Die Gefundene Zeilen werden in die Tabelle „EDE“ hinzugefügt
- Nach dem erfolgreichen Import wird in die Zeile [Datum EDE Import] in der Tabelle „Spec-Devices“ das aktuelle Datum hinzugefügt und Status auf „imported“ gesetzt.

TidyUp ()

Die Funktion TidyUp wird durch Klicken auf den Button „Tidy Up“ aufgerufen. Die Funktion löscht in der Tabelle „EDE“ die Daten von allen Devices, die nicht in der Tabelle „Spec-Devices“ vorkommen. Um diese zu finden, wird in der Tabelle „EDE“ nach object-type = 8 gesucht und die dazugehörige DevObjInstance herausgelesen. Wenn diese nicht in der Tabelle Spec-Devices.Dev-Inst vorkommt, dann werden alle Zeilen in der Tabelle „EDE“ mit der gleichen device-obj.-instance gelöscht.

Beispiel:

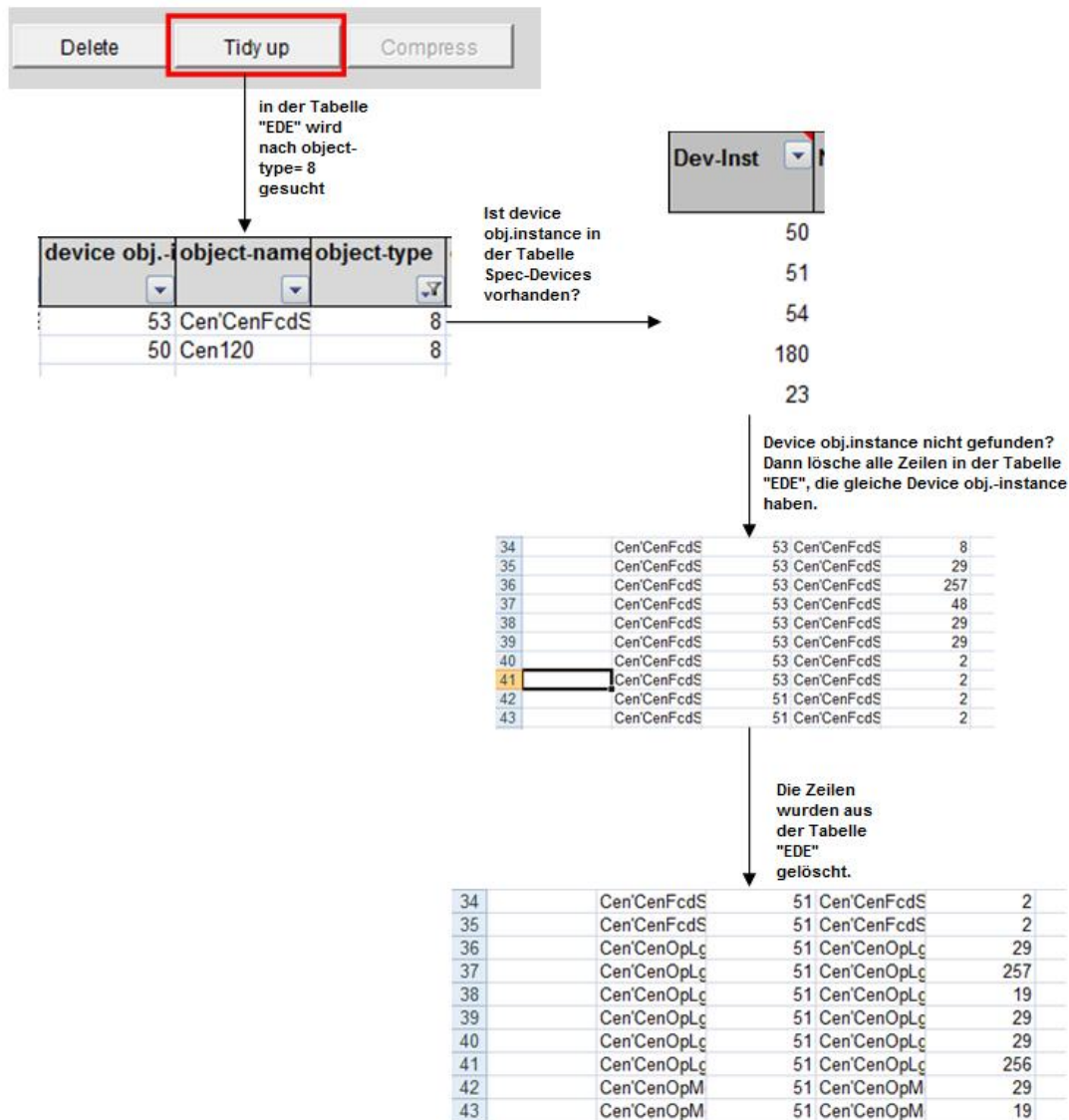


Abbildung 2-27 : TidyUp

DeleteEDE ()

Die Funktion DeleteEDE () wird durch Klicken auf den Button „Delete“ aufgerufen. Die Funktion entfernt alle EDE Daten von dem ausgewählten Gerät aus der Tabelle „EDE“, aber das funktioniert erst, wenn der User das Löschen von EDE Daten mit „Ja“ bestätigt hat. Zuerst muss der User eine Zeile aus der Geräteliste auswählen und danach auf den Knopf „Delete“ klicken. In der Tabelle „EDE“ werden nach EDE.DevobjInstance = SelectedTsNetDevInst gesucht. Wenn die Gleichung vorkommt, werden alle Zeilen in der Tabelle „EDE“ mit der gleichen SelectedTsNetDevInst gelöscht.

Beispiel:

SelectedTsNetDevInst = 50

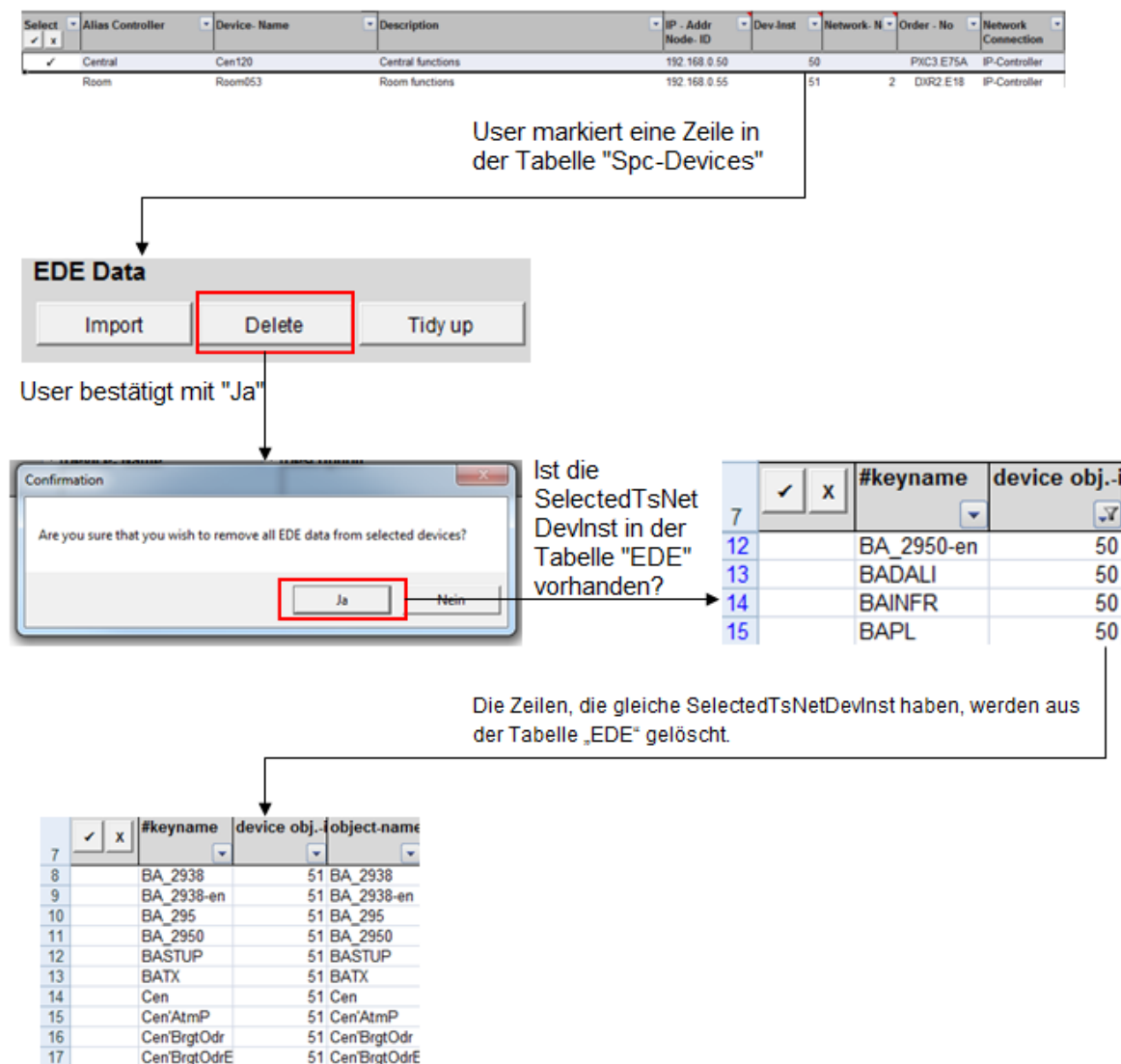


Abbildung 2-28 : DeleteEDE

ActualiseTsNet()

Die Funktion ActualiseTsNet() wird durch Klicken auf den Button „...“ aufgerufen.
Die Funktion liest die „TsNetExecutablePath“ aus der „C:\ProgramData\Siemens\tsNET\tsNET Settings.xml“ und schreibt sie in cl_TsNetExecutablePath.

Das Anzeigen von Pfadnamen ist möglich:

- Wenn tsNET installiert ist
- Wenn tsNET version ab 2.3.9493.0 ist

Das Anzeigen von Pfadnamen ist nicht möglich:

- Wenn tsNET nicht installiert ist

Beispiel:

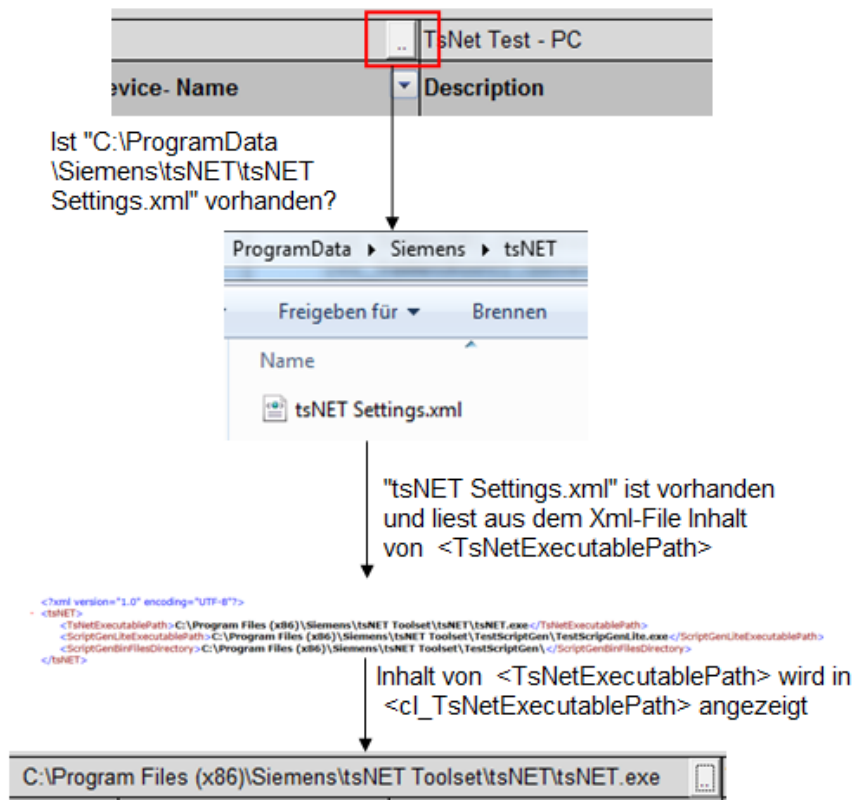


Abbildung 2-29 : ActualiseTsNET

2.3 Kontrollierung

2.3.1 Testkonzept

Testumgebung

Die Testdurchführung findet in der folgenden Entwicklungsumgebung, ausgelesen aus dem Microsoft Hilfsprogramm „msinfo32“, statt:

Element	Wert
Betriebssystemname	Microsoft Windows 7 Enterprise
Version	6.1.7601 Service Pack 1 Build 7601
Zusätzliche Betriebssystemesc...	Nicht verfügbar
Betriebssystemhersteller	Microsoft Corporation
Systemname	MD12TJKC
Systemhersteller	FUJITSU
Systemmodell	CELSIUS H700
Systemtyp	x64-basierter PC
Prozessor	Intel(R) Core(TM) i7 CPU M 620 @ 2.67GHz, 2667 MHz, 2 Kern(e), 4 logische(r) Prozessor(en)
BIOS-Version/-Datum	FUJITSU // Phoenix Technologies Ltd. Version 1.18, 26.10.2010
SMBIOS-Version	2.6
Windows-Verzeichnis	C:\WINDOWS
Systemverzeichnis	C:\WINDOWS\system32
Startgerät	\Device\HarddiskVolume2
Gebietsschema	Vereinigte Staaten von Amerika
Hardwareabstraktionsebene	Version = "6.1.7601.17514"
Benutzername	WW002\z00359dn
Zeitzone	Mitteleuropäische Zeit
Installierter physikalischer Speic...	8.00 GB
Gesamter realer Speicher	7.86 GB
Verfügbare realer Speicher	4.76 GB
Gesamter virtueller Speicher	15.7 GB
Verfügbare virtueller Speicher	11.9 GB
Größe der Auslagerungsdatei	7.86 GB
Auslagerungsdatei	C:\pagefile.sys

Abbildung 2-30 : Testumgebung

Information zu Microsoft Office Excel

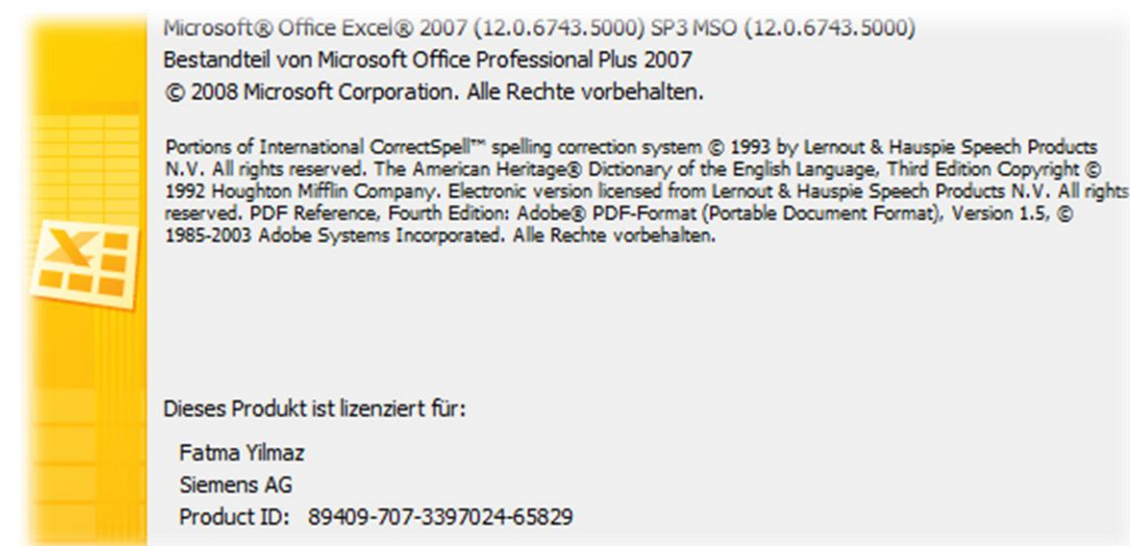


Abbildung 2-31 : Information zu Excel

Testdaten für „Import EDE“

- Cen120.csv
- Collaboration V5.1 - V6-Site01'SpvrV6.csv
- MultiController.csv

Testablauf

Innerhalb der IPA entwickelte Module oder angepasste Funktionen werden beim Testen überprüft. Falls Codestellen von den Änderungen betroffen sind, werden im Visual Basic Editor Tests ausgeführt oder allenfalls angepasst. Falls am Ende der IPA oder in der Phase nach dem Abschluss dieser Tätigkeit noch Zeitreserven vorhanden sind, werden neue Funktionen oder weitere Modultests implementiert. Da für diese Arbeit aber nicht viel Zeit eingeplant wurde und die Tests trotzdem wichtig sind, werden diese von Hand getestet. Das Testen besteht aus zwei Komponenten:

- White - Box Testfälle
 - o Die einzelnen Testfälle wurden innerhalb der IPA von der IPA-Ausführenden ermittelt. Um die neuen Funktionen genau zu testen und dabei auch alle erstellten Funktionen gegen Fehlbedienungen und fehlerhafte Daten und Dateien zu überprüfen, werden White Box Test verwendet. Diese Tests werden während der Realisierungsphase durchgeführt, werden auch Fehlerfälle genau getestet. Bei Bedarf kann man im Visual Basic Editor mit verschiedenen Möglichkeiten von Debuggen profitieren.
- Akzeptanz – Testfälle
 - o Die Testfälle wurden in der Entscheidungsphase mit Hilfe der gewünschten Anforderungen von Michael Speckien ermittelt und in der Kontrollphase von einer Siemens Mitarbeiterin, die eine kurze Einführung durch den Auftraggeber erhalten wird, durchgeführt. Frau Klinger, die Testperson, wird ein Testprotokoll erhalten, das sie während der Durchführung von einzelnen Testfällen ausfüllen soll.

Nur die IPA relevante Tabelle „Spec-Devices“, mit den folgenden Funktionen der Buttons, wird mit Testfällen abgedeckt:

- Modify
- Check List
- Check Devices
- Import
- Delete
- TidyUp
- Select
- Deselect
- Help

Die folgenden Funktionen der Buttons sind nicht IPA relevant und wurden somit nicht mit Testfällen abgedeckt:

- Compress
- Check comm.

Im nachfolgenden Kapitel sind die detaillierten Erwartungen und die Testergebnisse tabellarisch festgehalten.

2.3.2 White-Box-Test

Testfälle

Test ID : 1.0	
Name	Startbedingung
Testvoraussetzungen	Die Tabelle „Spec-Devices“ ist aktiviert
Testablauf	<ul style="list-style-type: none"> - Benutzer wählt die Zelle K21 - Benutzer tippt etwas in die Zelle K21
Erwartetes Resultat	<ul style="list-style-type: none"> - Tabelle ist schreibgeschützt, Benutzer sollte mit einem Dialog Fenster informiert werden
Test ID : 1.1	
Name	Bearbeitung von Daten
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec-Devices“ ist aktiviert
Testablauf	<ul style="list-style-type: none"> - Benutzer klickt auf Modify Button
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte Änderungen machen können, die Input Felder sind weiss
Test ID : 1.2	
Name	Alias Controller : Leere Zellen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle B22 ist leer
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf CheckList
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es leere Zellen in der Spalte Alias Controller gibt.
Test ID : 1.3	
Name	Alias Controller : Verdoppelungen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle B22 ist Segment - Zelle B23 ist Segment
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf CheckList
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es auf der Tabelle Spec-Devices Verdoppelungen gibt.

Test ID : 1.4	
Name	Device-Name : Leere Zellen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle D20 ist leer
Testablauf	<ul style="list-style-type: none"> - User klickt auf „CheckList“
Erwartetes Resultat	<ul style="list-style-type: none"> - Keine Fehlermeldung. Sollte ignoriert werden
Test ID : 1.5	
Name	Device-Name : Verdoppelungen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle D20 ist Cen120 - Zelle D21 ist Cen120
Testablauf	<ul style="list-style-type: none"> - User klickt auf „CheckList“
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es auf der Tabelle Spec-Devices Verdoppelungen gibt.
Test ID : 1.6	
Name	Ip-Addr: Falsche Ip –Format
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle J20 hat die Ip Adresse 192.168.0
Testablauf	<ul style="list-style-type: none"> - User klickt auf „CheckList“
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es auf der Tabelle Spec-Device , falsche Ip-Format gibt.
Test ID : 1.7	
Name	Ip-Addr: leere Zellen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle J21 ist leer
Testablauf	<ul style="list-style-type: none"> - User klickt auf „CheckList“
Erwartetes Resultat	<ul style="list-style-type: none"> - Die leere Zelle sollte ignoriert werden

Test ID : 1.8	
Name	Dev-Inst : Verdoppelungen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle K20 ist 50 - Zelle K21 ist 50
Testablauf	<ul style="list-style-type: none"> - User klickt auf „CheckList“
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es auf der Tabelle Spec-Devices Verdoppelungen gibt.
Test ID : 1.9	
Name	Ip-Format : nicht passende Network Connection
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle J20 hat die IP – Format 0.0.0.0 - Network Connection ist : MSTP Controller
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf den Button „Check List“.
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass das IP – Format falsch ist
Test ID : 2.0	
Name	Import EDE Daten : keine Selektion
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Status : Ok – List
Testablauf	<ul style="list-style-type: none"> - Benutzer klickt auf „Import“
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer hat kein Gerät ausgewählt, so sollte ein Import nicht möglich sein

Test ID : 2.1	
Name	Import EDE Daten : Sub Menu öffnen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Status : Ok – List - Zeile A20 ist selektiert - Device Name = Cen120 - Dev-Inst = 50
Testablauf	<ul style="list-style-type: none"> - Benutzer klickt auf „Import“
Erwartetes Resultat	<ul style="list-style-type: none"> - Die Tabelle „EDE“ wird nach Zeilen gesucht, die Object-Type als 8 haben und Dev-Inst sollte gleich sein wie die ausgewählte TsNet DevInst - Die Zeilen sollten gefunden werden - Button: Import ist deaktiviert - Sub Menu wird geöffnet
Test ID : 2.2	
Name	Import EDE Daten : Richtige .csv File öffnen
Testvoraussetzungen	<ul style="list-style-type: none"> - Sub Menu ist geöffnet <p>Selected Device:</p> <ul style="list-style-type: none"> - Device Name = Cen120 - Dev-Inst = 50 <p>Device in EDE:</p> <ul style="list-style-type: none"> - Device Name = Cen120 - Dev-Inst = 50
Testablauf	<ul style="list-style-type: none"> - Benutzer klickt auf „[...]“ - Benutzzer wählt ein richtigs EDE-Format
Erwartetes Resultat	<ul style="list-style-type: none"> - In EDE Datei wird nach object-type 8 gesucht, Dev-instance in Ede Datei ist gleich wie die ausgewählte TsNet DevInst - im Bereich Message steht eine Information - Im Bereich Symbol steht ein grüner Hacken. - Import möglich

Test ID : 2.3	
Name	Sub Menu : Import Button
Testvoraussetzungen	<ul style="list-style-type: none"> - Sub Menu ist geöffnet - Richtiges EDE File ist ausgewählt - Im Bereich „Selected Device“ steht : Device Name = Cen120 Dev-Inst = 50 - Im Bereich „Device In EDE“ steht : Device Name = Cen120 Dev-Inst = 50 - Im Bereich „Import File“ steht: Device Name = Cen120 Dev-Inst = 50 - Import Button ist aktiviert - Message : Data is consistent: Import replaces actual EDE for device - Symbol : tick := green
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf „Import“
Erwartetes Resultat	<ul style="list-style-type: none"> - Alle Zeilen in Tabelle „EDE“ wo die EDE.device-obj.-instance = selected TsNet device ist , werden gelöscht - Alle Zeilen aus dem Import File wo import.device-obj.-instance = selected TsNet device.Dev-Inst, werden in die Tabelle EDE hinzugefügt. - Import fertig, sub menü wird abgeschlossen - Date EDE import = aktuelle Datum - Status von ausgewählte TsNet device = Imported - Status = Import – Ok - Info = Import is done
Test ID : 2.4	
Name	Sub Menu : Falsche EDE format auswählen
Testvoraussetzungen	<ul style="list-style-type: none"> - Test ID : 2.2 ist durchgeführt - Selektiert die Zeile A20 - Sub Menu ist geöffnet
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf „Import“ - Klickt auf [...]
Erwartetes Resultat	<ul style="list-style-type: none"> - Device in EDE ist leer Device Name = Dev-Inst = - Import File ist leer Device Name = Dev-Inst = - Fehlermeldung durch falsches EDE Format - Import nicht möglich

Test ID : 2.5	
Name	Sub Menu : Selected Device ist leer
Testvoraussetzungen	<ul style="list-style-type: none"> - Test ID : 2.4 ist durchgeführt - die Zeile A21 ist selektiert - Die Zelle K21 ist leer - Sub Menu ist geöffnet
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf „Import“
Erwartetes Resultat	<ul style="list-style-type: none"> - Fehlermeldung, Import ist nicht möglich
Test ID : 2.6	
Name	Sub Menu : Cancel Button
Testvoraussetzungen	<ul style="list-style-type: none"> - Test ID : 2.1 ist durchgeführt
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf „Cancel“
Erwartetes Resultat	<ul style="list-style-type: none"> - Date Ede import = Datum - Status Ede Import = Import stopped
Test ID : 2.7	
Name	Delete EDE data
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert
Testablauf	<ul style="list-style-type: none"> - Benutzer selektiert Zelle A20 - Der Benutzer klickt auf „Delete “
Erwartetes Resultat	<ul style="list-style-type: none"> - Ein Dialog Fenster für die Bestätigung wird geöffnet
Test ID : 2.8	
Name	Delete EDE data
Testvoraussetzungen	<ul style="list-style-type: none"> - Test ID 2.8 ist ausgeführt - Ein Dialog Fenster für die Bestätigung ist geöffnet - Benutzer bestätigt mit Yes
Testablauf	<ul style="list-style-type: none"> - Benutzer bestätigt mit Yes
Erwartetes Resultat	<ul style="list-style-type: none"> - Alle Daten aus der Tabelle «EDE», die nicht in der Device List sind, werden gelöscht

Test ID : 2.9	
Name	Tidy Up data
Testvoraussetzungen	- Die Tabelle „Spec- Devices“ ist aktiviert
Testablauf	- Der Benutzer klickt auf «Tidy Up»
Erwartetes Resultat	- Ein Dialog Fenster für die Bestätigung wird geöffnet - Daten sind gelöscht
Test ID : 3.0	
Name	TsNet executable Path
Testvoraussetzungen	- Die Tabelle „Spec- Devices“ ist aktiviert - tsNET Settings.xml ist nicht vorhanden
Testablauf	- Der Benutzer klickt auf den «...» Button
Erwartetes Resultat	- Benutzer wird informiert durch eine MessageBox
Test ID : 3.1	
Name	Auswählen von Zeilen
Testvoraussetzungen	Die Tabelle „Spec-Devices“ ist aktiviert - Eine Zeile[ab 20] ist markiert
Testablauf	- Der Benutzer klickt auf den Select Button (√)
Erwartetes Resultat	- Die Zelle wird ausgewählt
Test ID : 3.2	
Name	Abwählen von Zeilen
Testvoraussetzungen	Die Tabelle „Spec- Devices“ ist aktiviert - Eine Zeile, die ein (√) – Symbol beinhaltet, ist markiert
Testablauf	- Der Benutzer klickt auf Deselect (X) Button
Erwartetes Resultat	- Die Zelle ist abgewählt

Test ID : 3.3	
Name	Check – Devices
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Device liste wurde überprüft-> Status: Ok- List - Import wurde gemacht -> Status: Import – OK
Testablauf	<ul style="list-style-type: none"> - Benutzer klickt auf Check-Devices
Erwartetes Resultat	<ul style="list-style-type: none"> - User wird informiert, dass alle Imports getätigt wurden
Test ID : 3.4	
Name	Alias Controller : Leere Zellen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle B22 ist leer
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf „Check Devices “
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es leere Zellen in der Spalte Alias Controller gibt.
Test ID : 3.5	
Name	Alias Controller : Verdoppelungen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle B22 ist Segment - Zelle B23 ist Segment
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf „Check Devices “
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es auf der Tabelle Spec-Devices Verdoppelungen gibt.
Test ID : 3.6	
Name	Device-Name : Leere Zellen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle D20 ist leer
Testablauf	<ul style="list-style-type: none"> - User klickt auf „Check Devices “
Erwartetes Resultat	<ul style="list-style-type: none"> - Zeigt eine Fehlermeldung. Sollte nicht ignoriert werden

Test ID : 3.7	
Name	Device-Name : Verdoppelungen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle D20 ist Cen120 - Zelle D21 ist Cen120
Testablauf	<ul style="list-style-type: none"> - User klickt auf „Check Devices “
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es auf der Tabelle Spec-Devices Verdoppelungen gibt.
Test ID : 3.8	
Name	Ip-Addr: Falsche Ip –Format
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle J20 hat die Ip Adresse 192.168.0
Testablauf	<ul style="list-style-type: none"> - User klickt auf „Check Devices “
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es auf der Tabelle Spec-Devices, falsche Ip-Format gibt.
Test ID : 3.9	
Name	Ip-Addr: leere Zellen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle J21 ist leer
Testablauf	<ul style="list-style-type: none"> - User klickt auf „Check Devices “
Erwartetes Resultat	<ul style="list-style-type: none"> - Die leere Zelle dürfen nicht ignoriert werden - Zeigt Fehlermeldung
Test ID : 4.0	
Name	Dev-Inst : Verdoppelungen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert - Zelle K20 ist 50 - Zelle K21 ist 50
Testablauf	<ul style="list-style-type: none"> - User klickt auf „Check Devices “
Erwartetes Resultat	<ul style="list-style-type: none"> - Benutzer sollte informiert werden, dass es auf der Tabelle Spec-Devices Verdoppelungen gibt.

Test ID : 4.1	
Name	Tidy Up – Unused EDE Data
Testvoraussetzungen	- Test-ID 2.9 ist durchgeführt
Testablauf	- Der Benutzer klickt auf „Yes“
Erwartetes Resultat	- Daten, von einer nicht verwendeten EDE Data sollten gelöscht sein

Testergebnis

Datum		05.04.2016	
Testperson		Fatma Yilmaz	
ID	Erwartung erfüllt?	Kommentar	Weitere Schritte
1.0	Ja	Info Fenster ist geöffnet: Die Zelle oder das Diagramm, die bzw. das Sie ändern möchten, ist schreibgeschützt.	Keine
1.1	Ja	<ul style="list-style-type: none"> - Ab Zeile 20 : die Spalten sind weiss, ausser die Spalte O un P - Die Zelle : cl_TsNetlp ist weiss - Die Zelle : cl_TsNetPort ist weiss - Button: Import ist deaktiviert - Button: Delete ist deaktiviert - Button: Tidy uP ist deaktiviert - Status: Working - Info: In Progress 	Keine
1.2	Ja	<ul style="list-style-type: none"> - Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Check the alias controller: empty column - Status: Error-List - Datum: Aktuelle Datum und Uhrzeit 	Keine
1.3	Ja	<ul style="list-style-type: none"> - Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Not Unique: Check the red cells - Status: Error-List - Datum: Aktuelle Datum und Uhrzeit - Zelle B22 ist rot - Zelle B23 ist rot 	Keine
1.4	Ja	Die leere Zeile in der Spalte Device-Name wird ignoriert	Keine
1.5	Ja	<ul style="list-style-type: none"> - Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Not Unique: Check the red cells - Status: Error-List - Datum: Aktuelle Datum und Uhrzeit 	Keine

		<ul style="list-style-type: none"> - Zelle D20 ist rot - Zelle D21 ist rot 	
1.6	Ja	<ul style="list-style-type: none"> - Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Wrong IP Format in Column Ip-Addr - Status: Error List - J20 ist rot markiert 	Keine
1.7	Nein	<p>Die Zelle wird nicht ignoriert und zeigt trotzdem ein Fehlermeldung</p> <ul style="list-style-type: none"> - Info: Wrong IP Format in Column Ip-Addr - Status: Error List <p>Fehler behoben Code hinzugefügt: im CheckList()</p> <pre>'it doesnt have the format of an ip If Cells(x, 10) <> vbNullString Then Cells(x, 10).Font.Color = RGB(255, 0, 0) GoTo BadIpFormat End If</pre>	
1.8	Ja	<ul style="list-style-type: none"> - Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Not Unique: Check the red cells - Status: Error-List - Datum: Aktuelle Datum und Uhrzeit - Zelle K20 ist rot - Zelle K21 ist rot 	Keine
1.9	Ja	<ul style="list-style-type: none"> - Info: Wrong IP Format in Column Ip-Addr - Status: Error List 	Keine
2.0	Ja	<ul style="list-style-type: none"> - Import ist nicht möglich - Info : No device is selected - Status: Error Import 	Keine
2.1	Ja	<p>*Im Bereich „Selected Device“ steht :</p> <ul style="list-style-type: none"> - Device Name = Cen120 Dev-Inst = 50 <p>*Im Bereich „Device In EDE“ steht :</p> <ul style="list-style-type: none"> - Device Name = Cen120 Dev-Inst = 50 <p>*Bereich Import File ist leer</p> <p>*Import ist deaktiviert</p>	Keine
2.2	ja	<p>*Im Bereich „Selected Device“ steht :</p> <ul style="list-style-type: none"> - Device Name = Cen120 Dev-Inst = 50 <p>*Im Bereich „Device In EDE“ steht :</p> <ul style="list-style-type: none"> - Device Name = Cen120 Dev-Inst = 50 <p>*Im Bereich „Import File“ steht:</p> <ul style="list-style-type: none"> - Device Name = Cen120 Dev-Inst = 50 <p>*Import Button ist aktiviert</p> <p>*Message : Data is consistent: Import replaces actual EDE for device</p> <p>*Symbol : tick := green</p>	Keine

2.3	Ja	<ul style="list-style-type: none"> - In EDE wurden Zeilen gelöscht , die 50 als DevInst haben - Neue Daten wurden hinzugefügt - Neue zeilen die DevInst als 50 haben - Info: Import is done - Status: Import – OK - Datum : Aktuelle Datum und Uhrzeit - Date EDE import: DaTUM - Status: Imported 	Keine
2.4	Ja	<ul style="list-style-type: none"> - Import nicht möglich - Device in EDE ist leer - Device Name = Dev-Inst = - Import File ist leer - Device Name = Dev-Inst = - Message: No EDE file , no import possible - Symbol : X := red 	Keine
2.5	Nein	<ul style="list-style-type: none"> - Keine Fehlermeldung <p>Fehler behoben Code hinzugefügt: im ImportEDE()</p> <pre>' if selected TsNet Devinst is empty or selected TsNet Dev name is empty If strSelectedDevInst = vbNullString Or strSelectedDevName = vbNullString Then 'shows the negative message frmImportEDE.txt_result.Text = "Selected device is not correctly defined, define devicename and device-instance" 'Sets the x symbol to red frmImportEDE.txt_Symbol.ForeColor = vbRed 'Shows the x symbol frmImportEDE.txt_Symbol.Text = "X" 'the button import is disabled frmImportEDE.cmd_ImportEDE.Enabled = False End If</pre>	
2.6	Ja	<ul style="list-style-type: none"> - Date EDE import = datum - Status = import stopped - Info = Import is not done - Status = Import Error 	Keine
2.7	Ja	<ul style="list-style-type: none"> - Confirm: Are you sure that you wish to remove all EDE data from selected devices? 	Keine
2.8	Ja	<ul style="list-style-type: none"> - Alle Daten, die EDE.device-obj.-instance = selected TsNet device.DevInst sind aus der EDE Tabelle gelöscht 	Keine
2.9	Nein	<ul style="list-style-type: none"> - Nach keiner Bestätigung wurde gefragt <p>Fehler behoben Code hinzugefügt: im cmd_TidyUp_Click ()</p> <pre>'Removes all EDE data from controllers which are not in the devicelist If MsgBox("Are you sure that you wish to remove all EDE data ?", vbYesNo, "Confirmation") = vbYes Then Call TidyUp End If</pre>	
3.0	Ja	<ul style="list-style-type: none"> - Message: It doesn't contain xml file 	Keine
3.1	Ja	<ul style="list-style-type: none"> - In der Spalte A wird bei der ausgewählten Zeile ein Symbol (√) gesetzt. 	Keine

3.2	Ja	- Die ausgewählte Zelle beinhaltet kein (✓) – Symbol mehr	Keine
3.3	Ja	- Status = "OK-EDE" - Info = "All EDE imports done and correct"	Keine
3.4	Ja	- Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Check the alias controller is empty - Status: Error- Devices - Datum: Aktuelle Datum und Uhrzeit	Keine
3.5	Ja	- Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Check the alias controller: empty column - Status: Error-Devices - Datum: Aktuelle Datum und Uhrzeit	Keine
3.6	Ja	- Zeigt Fehlermeldung - Info: There are empty cells - Status: Error Devices	Keine
3.7	Ja	- Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Not Unique: Check the red cells - Status: Error-Devices - Datum: Aktuelle Datum und Uhrzeit - Zelle D20 ist rot - Zelle D21 ist rot	Keine
3.8	Ja	- Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Wrong IP Format in Column Ip-Addr - Status: Error-Devices - J20 ist rot markiert	Keine
3.9	Ja	- Zeigt Fehlermeldung - Info: There are empty cells - Status: Error-Devices	Keine
4.0	Ja	- Die Tabelle „Spec-Devices“ ist schreibgeschützt - Info: Not Unique: Check the red cells - Status: Error-List - Datum: Aktuelle Datum und Uhrzeit - Zelle K20 ist rot - Zelle K21 ist rot	Keine

4.1	Nein	- Daten, von einer nicht verwendeten EDE wurde nicht gelöscht
		Fehler behoben, Code hinzugefügt: im TidyUp()
		<pre> For i = strLastRowEDE To 8 Step by - 1 Rows(i).Select If WbsEdeData.Range("C" & i).Value = strDevInstValue Then 'select entirerow from ede Rows(i).EntireRow.Select 'delete entirerow from ede Rows(i).Delete End If Next i </pre>

Tabelle 9 : WhiteBox-Testergebnisse

Nachtest

Datum		07.04.2016	
Testperson		Fatma Yilmaz	
ID	Erwartung erfüllt?	Kommentar	Weitere Schritte
1.7	Ja	- Die leere Zelle wurde ignoriert werden	Keine
2.5	Ja	- Fehlermeldung, Import ist nicht möglich	Keine
2.9	Ja	- Ein Dialog Fenster für die Bestätigung wird geöffnet	Keine
4.1	Ja	- Daten, von einer nicht verwendeten EDE Data wurden gelöscht	Keine

Tabelle 10 : WhiteBox-Nachtest

2.3.3 Akzeptanz-Test

Testfälle

Test ID : 1.0	
Name	Startbedingung
Testvoraussetzungen	Die Bedienoberfläche wird korrekt geöffnet.
Testablauf	-
Erwartetes Resultat	- Die Excel File „TestTemplate.xlsm“ wird im Read-Only Modus geöffnet.
Test ID : 1.1	
Name	„Spec-Devices“ bearbeiten
Testvoraussetzungen	Die Tabelle „Spec-Devices“ ist aktiviert.
Testablauf	- Benutzer klickt auf „Modify“.
Erwartetes Resultat	<ul style="list-style-type: none"> - Die Tabelle wird vom Read-Only Modus zum Read/Write Modus gewechselt. - Alle Eingabefelder und alle Funktionen sind aktiviert. - Status wird auf „Working“ gesetzt. - Button CheckList ist aktiviert. - Button Check Devices ist deaktiviert - Button Import ist deaktiviert - Button Delete ist deaktiviert - Button Tidy Up ist deaktiviert
Test ID : 1.2	
Name	Help-File
Testvoraussetzungen	Die Tabelle „Spec- Devices“ ist aktiviert.
Testablauf	- Der Benutzer klickt auf blau markierten „ Help “ Text.
Erwartetes Resultat	<ul style="list-style-type: none"> - Die .pdf - Datei „PD103267-T-0104_EN_TestManual_TsNet“ wird geöffnet. - Die Seite „9“ wird angezeigt.

Test ID : 1.3	
Name	Die „Overview“ Tabelle aktivieren
Testvoraussetzungen	Die Tabelle „Spec- Devices“ ist aktiviert.
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf blau markierten „ GoTo Overview“ Text.
Erwartetes Resultat	<ul style="list-style-type: none"> - Die Tabelle Overview wird aktiviert.
Test ID : 1.4	
Name	Daten überprüfen
Testvoraussetzungen	<p>Die Tabelle „Spec- Devices“ ist aktiviert.</p> <ul style="list-style-type: none"> - Alias Controller: eindeutig und nicht leer. - Device-Name: eindeutig und nicht leer. - IP-Addr, Node-ID : eindeutig. - Dev-Inst: eindeutig.
Testablauf	<ul style="list-style-type: none"> - User klickt auf „Check List“
Erwartetes Resultat	<ul style="list-style-type: none"> - Info: List is okay. - Status: auf OK-List gesetzt. - Datum: aktuelle Datum und Uhr Zeit. - Wechselt zu Read Only Mode. - Import – Button: aktiviert. - Delete – Button: aktiviert. - TidyUp – Button: aktiviert.
Test ID : 1.5	
Name	Anzeigen von TsNetExecutablePath
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert. - TsNet- Programm ist installiert.
Testablauf	<ul style="list-style-type: none"> - Benutzer klickt auf „Modify“ - Der Benutzer klickt auf den Button „Actualise TsNet [...]“.
Erwartetes Resultat	<ul style="list-style-type: none"> - Auf der Tabelle „Spec-Devices“ in der Zeile „TsNetExecutablePath“ steht C:\Program Files (x86)\Siemens\tsNET Toolset\tsNET\tsNET.exe.

Test ID : 1.6	
Name	EDE Daten importieren
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert. - Eine EDE .csv Datei ist vorhanden. - Eine Zeile ist markiert.
Testablauf	<ul style="list-style-type: none"> - Benutzer Klickt auf Modify. - Benutzer markiert eine Zeile. - Benutzer klickt auf den Button „Import“.
Erwartetes Resultat	<ul style="list-style-type: none"> - Import EDE Sub Menü wird geöffnet.
Test ID : 1.7	
Name	Richtige EDE-Format importieren
Testvoraussetzungen	<ul style="list-style-type: none"> - Import EDE, Sub Menu, ist geöffnet.
Testablauf	<ul style="list-style-type: none"> - Eine EDE .csv Datei ist ausgewählt. - Der Testperson klickt auf „Import“.
Erwartetes Resultat	<ul style="list-style-type: none"> - Symbol ist der Grüne Hacken (✓) - die Daten werden in dem Arbeitsblatt „EDE“ hinzugefügt.
Test ID : 1.8	
Name	Date EDE Import
Testvoraussetzungen	<ul style="list-style-type: none"> - die aktuelle EDE Daten sind im Arbeitsblatt „EDE“. - Sub Menü ist abgeschlossen.
Testablauf	-
Erwartetes Resultat	<ul style="list-style-type: none"> - In Spalte „Date EDE Import“ steht das aktuelle Datum. - In Spalte „Status“ steht „Imported“. - Button „Check Devices“ ist aktiviert. - Status: Import – OK. - Info: Import is done.

Test ID : 1.9	
Name	Geräte überprüfen
Testvoraussetzungen	<p>Die Tabelle „Spec- Devices“ ist aktiviert.</p> <ul style="list-style-type: none"> - Alias Controller : eindeutig und nicht leer. - Device-Name: eindeutig und nicht leer. - IP- Addr, Node-ID: eindeutig und nicht leer. - Dev-Inst: eindeutig und nicht leer.
Testablauf	<ul style="list-style-type: none"> - Der Benutzer klickt auf den Button „Check Device“.
Erwartetes Resultat	<ul style="list-style-type: none"> - Status: „OK-EDE“. - Info: „All EDE imports done and correct“ - Datum: aktuelle Datum und Zeit.
Test ID : 2.0	
Name	EDE Daten löschen
Testvoraussetzungen	<p>Die Tabelle „Spec- Devices“ ist aktiviert.</p> <ul style="list-style-type: none"> - Eine Zeile, z.B. A20 ist selektiert.
Testablauf	<ul style="list-style-type: none"> - Benutzer klickt auf „Delete EDE“.
Erwartetes Resultat	<ul style="list-style-type: none"> - Ein Dialogfenster wird für die Bestätigung geöffnet.
Test ID : 2.1	
Name	EDE Daten zum Löschen bestätigen
Testvoraussetzungen	<p>Die Tabelle „Spec-Devices“ ist aktiviert.</p> <ul style="list-style-type: none"> - eine Zeile in der Spalte A ist selektiert. - Ein Dialogfenster für Bestätigung ist geöffnet.
Testablauf	<ul style="list-style-type: none"> - Der User klickt auf „Yes“.
Erwartetes Resultat	<ul style="list-style-type: none"> - In der Tabelle „EDE“ werden die Daten, die selektierten Dev-Inst haben, gelöscht.

Test ID : 2.2	
Name	EDE Daten aufräumen
Testvoraussetzungen	- Die Tabelle „Spec- Devices“ ist aktiviert.
Testablauf	- Der User klickt auf „Tidy Up“.
Erwartetes Resultat	- Ein Dialogfenster wird für die Bestätigung geöffnet.
Test ID : 2.3	
Name	EDE Daten zum Aufräumen bestätigen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec-Devices“ ist aktiviert - „Tidy Up“ wurde geklickt. - Ein Dialogfenster für die Bestätigung ist geöffnet.
Testablauf	- Der User bestätigt mit „Yes“.
Erwartetes Resultat	- Alle Daten, deren Device obj.-instance nicht mit einer Dev-inst in Tabelle Spec-Devices übereinstimmt werden aus der „EDE“ Tabelle gelöscht.
Test ID : 2.4	
Name	Auswählen von Zeilen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec-Devices“ ist aktiviert. - Eine Zeile[ab 20] ist durch Klicken in Spalte A markiert.
Testablauf	- Der Testperson klickt auf den Select Button (✓)
Erwartetes Resultat	- In der Spalte A wird bei der ausgewählten Zeile ein Symbol (✓) gesetzt.
Test ID: 2.5	
Name	Abwählen von Zeilen
Testvoraussetzungen	<ul style="list-style-type: none"> - Die Tabelle „Spec- Devices“ ist aktiviert. - Eine Zelle, die ein (✓) – Symbol beinhaltet, ist durch Klicken in Spalte selektiert.
Testablauf	- Der Benutzer klickt auf den Deselect (X) Button.
Erwartetes Resultat	- Die ausgewählte Zelle beinhaltet kein (✓) – Symbol mehr.

Testergebnis

Datum		06.04.2016	
Testperson		Berit Klinger	
ID	Erwartung erfüllt?	Kommentar	Weitere Schritte
1.0	ja	Die Tabelle „Spec-Devices“ wird angezeigt	Keine
1.1	ja	Felder aktiviert gemäss Erwartungen	Keine
1.2	Ja	Doku wird auf der richtigen Seite geöffnet	Keine
1.3	Ja	Navigation wie erwartet	Keine
1.4	Ja	Funktion wie erwartet	Keine
1.5	ja	Pfad wird korrekt ausgelesen	Keine
1.6	Ja	Dialog „Import EDE“ wird geöffnet	Keine
1.7	Ja	Import korrekt ausgeführt	Keine
1.8	Ja	Status korrekt nachgeführt	Keine
1.9	Ja	Status korrekt nachgeführt	Keine
2.0	Ja	Dialogfenster öffnet sich	Keine
2.1	Ja	Daten der Dev.Inst 50 sind aus der EDE-Tabelle gelöscht	Keine
2.2	Ja	Dialogfenster öffnet sich	Keine
2.3	Ja	Daten des nicht verwendeten Controllers sind gelöscht	Keine
2.4	Ja	Haken vor ausgewählter Zeile gesetzt	Keine
2.5	Ja	Haken vor ausgewählter Zeile gelöscht	Keine

Tabelle 11 : Akzeptanztest - Testergebnisse

2.4 Auswertung

2.4.1 Schlusswort

Ich habe bereits im Vorfeld der IPA an anderen Teilen von TsNet mit der gleichen Arbeitsumgebung gearbeitet. Das Programmieren mit VBA war für mich deshalb nicht neu. Es stellte sich aber heraus, dass die vorherigen Arbeiten nie so ausführlich waren wie jetzt während der IPA, da ich bisher nie einen Zeitplan oder ein Testkonzept erstellen musste.

Das Implementieren ist relativ gut verlaufen und ich denke, dass ich alle Anforderungen erfüllt und getestet habe. Ich finde, die Bedienoberfläche wurde benutzerfreundlich und leicht verständlich umgesetzt, so dass dem Benutzer die Bedienung leichtfallen wird.

Für mich war die Zeit während der IPA streng. Ich musste zum ersten Mal eine Arbeit so ausführlich dokumentieren und selbständig organisieren. Dies war eine grosse Herausforderung für mich. Beim Zeitplan habe ich darauf geachtet, täglich ein Zehntel der Zeit als Reserve einzuplanen. Diese Zeit habe ich auch gebraucht. Während der IPA tauchten immer wieder Fragen im Sinne von "Was ist wichtiger, Qualität oder Quantität?" auf, weil ich es schwierig fand, abzuschätzen, wie genau ich dokumentieren sollte.

Alles in allem war die IPA eine gute Erfahrung und ich freue mich, sie in dieser Form nun abzuschliessen.

2.5 Glossar

ID	Begriff	Erklärung
1	ABT	ABT (Automation Building Tool) Gebäudeautomation Tool für Engineering, Inbetriebnahme und Wartung von BACS. Es basiert auf TIA Portal (Total Integrated Automation).
2	Akzeptanztest	Bei dem Akzeptanztest wird überprüft ob die Software die funktionalen Erwartungen und Anforderungen im Gebrauch erfüllt.
3	Applications	Gesamte Applikation zum Steuern und Regeln einer Aufgabenstellung der Gebäudeautomation in einem Controller. Die Applikation umfasst auch das Erfassen und Ausgeben von Daten von/an Feldgeräte, Bedienpanels, Leitsysteme und andere Controller. Beispiel: Lichtsteuerung und Temperaturregelung für mehrere Räume. Eine Applikation besteht aus Applikationsfunktionen (AFs) und Charts
5	BACnet	Building Automation and Control Networks, ist ein Netzwerkprotokoll für die Gebäudeautomation
6	Controller	Unter Controller versteht man frei programmierbares Automationsgerät für HLK. Die Firmware beinhaltet bereits alle Funktionen zur Kommunikation mit den Feldgeräten und mit dem Anwender. Eine spezifische HLK Applikation wird mit dem ABT engineert und in den Controller runtergeladen.
7	EDE	BACnet Engineering Data Exchange wird verwendet, um Daten von einem BACnet Engineering System auf ein anderes zu übertragen. Der Inhalt der Datei ist standardisiert
8	IPA	IPA (Individuelle Praktische Arbeit) wird am Ende der Erstausbildung im Lehrbetrieb gemacht und bildet einen Teil im Qualifikationsbereich "Praktische Arbeiten".
9	IPERKA	IPERKA ist eine Projektmanagementmethode, die aus sechs Schritten besteht. (Informieren, Planen, Entscheiden, Realisieren, Kontrollieren, Auswerten)
10	Modul	Module sind Container für Variablen und für Code

11	Network Topology	Bei einem Computernetzwerk charakterisiert die Topologie die Struktur der Verbindungen mehrerer Geräte zueinander, damit ein Datenaustausch möglich ist.
12	Netzwerkprotokoll	ist eine Vereinbarung für den Datenaustausch zwischen Computern/Prozessen
13	Prozeduren /Funktionen	Prozeduren: Gruppen von Anweisungen haben aber kein Rückgabewert Funktionen: wie Prozeduren aber mit Rückgabewert, Beide können Parameter übergeben --> Standard byRef, sonst angeben mit ByVal
14	Struktogramm	Struktogramme werden auch als Nassi Schneiderman-Diagramme bezeichnet. Sie stellen Programmstrukturen dar.
15	TRA	Design TRA: Total Room Automation ist ein Teil des des Design Systems zur Raumautomation. Es ermöglicht eine nahtlose Interaktion von HLK, Beleuchtung und Beschattung mechanischen und elektrischen Anlagen. Dies führt zu einem energieoptimierten Betrieb auf ein Maximum Komfort in den Zimmern.
16	TsNet	Testtool zum Open-Loop Test für Applikationen. Es besteht unter anderem aus einem Excel-File zur Testspezifikation mit der Definition von Eingangsbedingungen und den erwarteten Reaktionen. Testscripts ermöglichen das automatisierte Abfahren und Auswerten dieser Testspezifikation.
17	VBA	VBA (Visual Basic Application), ist die Programmiersprache von Microsoft Office. Und wird beim Automatisieren von wiederholten Aufgaben verwendet
18	Workflow	Ein Workflow ist ein Block von Aktivitäten und Ergebnisse. Es fasst die wichtigsten Prozessschritte für eine bestimmte Disziplin zusammen. Somit stellt ein Workflow immer nur eine begrenzte Sicht des Gesamtprozesses und muss nicht vollständig sein. Die Ansicht von Phasen müssen immer für eine vollständige Darstellung aller Aktivitäten verwendet werden
19	White-Box-Test	White-Box-Test ist eine Methode für Software- Tests. Man hat Zugriff auf den Code

Tabelle 12 : Glossar

2.6 Quellen

Nr.	Thema	Quelle	Datum
1	Projektauftrag	https://extranet.pkorg.ch/	21.03.2016
2	IPERKA	http://tgabathuler.ch/IPERKA/Index.html	21.03.2016
3	HUS Struktogrammer	https://www.zg.ch/behoerden/volkswirtschaftsdirektion/gibz/berufe/informatiker-in-efz	22.03.2016

Tabelle 13 : Quellenverzeichnis
[n] = Verweis auf Quellennummer

2.7 Anhang

2.7.1 Code

2.7.2 Tabelle 19 (Spec-Devices)

```
' -----
' Name:           Tabelle19(Spec-Devices)
' File:           TestTemplate.xlsm
' -----
' Copyright @2016, SIEMENS Building Technologies
' -----
' OS:             Win64
' Lang:           Microsoft Visual Basic 6.5
' -----
' Project:        TsNet
' Author:         Fatma Yilmaz
' Version         V.1
' Date            24-03-2016
' -----
' Description:     Aus diesem Makro werden die einzelne Funktionen/Prozeduren aufgerufen,
'                  die im Modul "mdl_SpecDevices" liegen
' -----
' History:         24-03-2016 Fatma Yilmaz
'                  V.1 Document Creation for IPA
' -----

Private Sub cmd_Modify_Click()
    'switches the sheet form read only to read write mode
    Call ModifyWBS()
End Sub

Private Sub cmd_ActualiseTsNet_Click()
    'read the TsNet direction
    Call ActualiseTsNet()
End Sub

Private Sub cmd_Select_Click()
    'Unicode character Check Mark
    ActiveCell.Value = ChrW(&H2713)
End Sub
```

```
Private Sub cmd_Deselect_Click()  
    'Shift:=xlUp tells excel to shift up the cells not to delete entire row  
    Selection.Delete(Shift:=xlUp)  
End Sub  
  
Private Sub cmd_CheckList_Click()  
    'Checks the data and consistence of input fields  
    Call CheckList()  
End Sub  
  
Private Sub cmd_Import_Click()  
    'Opens userform Import EDE  
    Call ImportEDE()  
End Sub  
  
Public Sub cmd_Delete_Click()  
    'asks user for confirmation before deleting  
    If MsgBox("Are you sure that you wish to remove all EDE data from selected devices?", vbYesNo, "Confirmation") = vbYes Then  
        Call DeleteEDE(ActiveCell.EntireRow.Cells(1, 11).Value)  
    End If  
End Sub  
  
Private Sub cmd_TidyUp_Click()  
    'Removes all EDE data from controllers whic are not in the devicelist  
    If MsgBox("Are you sure that you wish to remove all unused EDE data ?", vbYesNo, "Confirmation") = vbYes Then  
        Call TidyUp()  
    End If  
End Sub  
  
Private Sub cmd_CheckDevices_Click()  
    'Check the data and the consistence of all input fields and EDE  
    Call CheckDevices()  
End Sub  
  
Private Sub Worksheet_SelectionChange(ByVal Target As Excel.Range)  
    'execute Open_Pdf from mdl_Helpfile  
    If Target.Address = "$0$3" Then  
        Call GetHelp()  
    End If  
End S
```

2.7.3 mdl_SpecDevices

Option Explicit On

Name:	mdl_SpecDevices
File:	TestTemplate.xlsm
Copyright @2016,	SIEMENS Building Technologies
OS:	Win64
Lang:	Microsoft Visual Basic 6.5
Project:	TsNet
Author:	Fatma Yilmaz
Version	V.1
Date	24-03-2016
Description:	The functions for the table spec-devices is implemented in mdl_SpecDevices
Parameters:	
p_hwnd	the parent window will be used for displaying a UI or error messages
p_Operation	defines actions like open, edit , print etc. that will take place on execution
p_File	defines the file to open
p_Parameters	defines the parameters to be passed
p_Directory	defines the default directory
p_ShowCmd	defines how the application should be shown
History:	24-03-2016 Fatma Yilmaz
	V.1 Document Creation for IPA
	V.1 ModifyWBopen()
	V.1 ModifyWBS()
	V.1 CheckDevices()
	V.1 CheckList()
	V.1 CheckDuplication()
	29-03-2016 Fatma Yilmaz
	V.1 Select()
	V.1 Deselect()
	V.1 ActualiseTsNet()
	V.1 ImportEDE()
	V.1 CheckFrmImportEDE()

```

'           31-03-2016  Fatma Yilmaz
'           V.1 appendingEDE()
'           V.1 replacingEDE()
'           V.1 CheckDevicesInEDE()
'-----
'           01-04-2016  Fatma Yilmaz
'           V.1 DeleteEDE()
'           V.1 TidyUp()
'-----

'the value 1 change the pdf file to original size or position
Public Const G_SW_SHOWNORMAL = 1

'ShellExecute starts a PDF File, it will called when a lbl_Help is clicked
Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" ( _
    ByVal p_hwnd As Long, _
    ByVal p_Operation As String, _
    ByVal p_File As String, _
    ByVal p_Parameters As String, _
    ByVal p_Directory As String, _
    ByVal p_ShowCmd As Long) As Long
'-----
' Description:      Opens the sheet in read only mode and disables all functions
'-----

Sub ModifyWBopen()

    'Variable Declaration
    Dim objCheckList As OLEObject
    Dim objCheckDevices As OLEObject
    Dim objImport As OLEObject
    Dim objDelete As OLEObject
    Dim objTidyUp As OLEObject
    Dim objCompress As OLEObject
    Dim objCheckComm As OLEObject
    Dim objActualiseTsNet As OLEObject

    'Assigning object to variable
    objCheckList = Sheets("Spec-Devices").OLEObjects("cmd_CheckList")
    objCheckDevices = Sheets("Spec-Devices").OLEObjects("cmd_CheckDevices")
    objImport = Sheets("Spec-Devices").OLEObjects("cmd_Import")
    objDelete = Sheets("Spec-Devices").OLEObjects("cmd_Delete")
    objTidyUp = Sheets("Spec-Devices").OLEObjects("cmd_TidyUp")

```



```
objCompress = Sheets("Spec-Devices").OLEObjects("cmd_Compress")
objCheckComm = Sheets("Spec-Devices").OLEObjects("cmd_CheckComm")
objActualiseTsNet = Sheets("Spec-Devices").OLEObjects("cmd_ActualiseTsNet")

'Set Properties
objCheckList.Enabled = False
objCheckDevices.Enabled = False
objImport.Enabled = False
objCheckList.Enabled = False
objDelete.Enabled = False
objTidyUp.Enabled = False
objCompress.Enabled = False
objCheckComm.Enabled = False
objActualiseTsNet.Enabled = False

ActiveSheet.Unprotect()
'Set the editable fields to grey
[cl_Editablefields].Interior.Color = RGB(216, 216, 216)
[cl_TsNetIP].Interior.Color = RGB(216, 216, 216)
[cl_TsNetPort].Interior.Color = RGB(216, 216, 216)

'Protect the editable fields
ActiveSheet.Protect(DrawingObjects:=False, Contents:=True, Scenarios:=True)
End Sub
```

```
'-----'  
' Description:      Switches the sheet from ReadOnly to ReadWrite mode. Enables all input fields and '  
'                  all functions. Sets status to Working and info to "in Progress" '  
'-----'  
Sub ModifyWBS()  
  'Variable Declaration  
  Dim objCheckList As OLEObject  
  Dim objActualiseTsNet As OLEObject  
  
  'Assigning object to variable  
  objCheckList = Sheets("Spec-Devices").OLEObjects("cmd_CheckList")  
  objActualiseTsNet = Sheets("Spec-Devices").OLEObjects("cmd_ActualiseTsNet")  
  
  'Set Properties  
  objCheckList.Enabled = True  
  objActualiseTsNet.Enabled = True  
  
  'Unprotect the table Spec-Devices  
  ActiveSheet.Unprotect()  
  
  'set the editable fields to white  
  [cl_Editablefields].Interior.Color = RGB(255, 255, 255)  
  [cl_IsNetIP].Interior.Color = RGB(255, 255, 255)  
  [cl_IsNetPort].Interior.Color = RGB(255, 255, 255)  
  [cl_Status].Value = "Working"  
  [cl_Info].Value = "In Progress"  
End Sub
```

```
'-----  
' Description:      Reads the TsNetExecutablePath from  
'                  "C:\ProgramData\Siemens\tsNet\tsNET Settings.xml" and writes it into  
'                  cl_TsNetExecuteablePath  
'-----  
Sub ActualiseTsNet()  
    'Variable Declaration  
    Dim objXmlDoc As Object  
    Dim strFile As String  
    Dim bFileExists As Boolean  
    Dim vrtXmlNode As Object  
    Dim vrtTsNetDirec As Object  
  
    'Assigning object to variable  
    objXmlDoc = CreateObject("MSXML2.DOMDocument")  
    'Direction name of tsNET Settings.xml  
    strFile = "C:\ProgramData\Siemens\tsNet\tsNET Settings.xml"  
  
    'file is existing then bFileExists = True  
    bFileExists = objXmlDoc.Load(strFile)  
    'if bFileExists = True  
    If bFileExists = True Then  
        'Get the first document element  
        vrtXmlNode = objXmlDoc.DocumentElement  
        'Get the first child element  
        vrtTsNetDirec = vrtXmlNode.FirstChild  
        'Print first child Text  
        Range("cl_TsNetExecutablePath").Value = vrtTsNetDirec.Text  
    Else  
        'the xml file doesnt exist  
        MsgBox("It doesnt contain xml file")  
    End If  
    objXmlDoc = Nothing  
End Sub
```

```
' .....  
' Description:      Checks for the duplications in sheet "Spec-Devices" and marks them with red color '  
' .....  
Sub CheckDuplication()  
    'Variable Declaration  
    Dim rngAliasController As Range  
    Dim rngDeviceName As Range  
    Dim rngIpAddr As Range  
    Dim rngDevInst As Range  
    Dim cell As Range  
    Dim lngCheckCells As Long  
    Dim intStartList As Integer  
  
    On Error GoTo ErrHandler  
    Application.ScreenUpdating = False  
  
    'Get the name of Column  
    rngAliasController = Range("B20:B" & Cells(Rows.Count, "B").End(xlUp).Row)  
    rngDeviceName = Range("D20:D" & Cells(Rows.Count, "D").End(xlUp).Row)  
    rngIpAddr = Range("J20:J" & Cells(Rows.Count, "J").End(xlUp).Row)  
    rngDevInst = Range("K20:K" & Cells(Rows.Count, "K").End(xlUp).Row)  
    lngCheckCells = Cells(Rows.Count, "B").End(xlUp).Row  
  
    'Check for duplications in column Alias Controller  
    For Each cell In rngAliasController  
        'set the default color to black  
        cell.Offset(0, 0).Font.Color = vbBlack  
        ' Find the duplicate values in range Alias Controller  
        If Application.Evaluate("COUNTIF(" & rngAliasController.Address & "," & cell.Address & ")") > 1 Then  
            ' mark the duplicate values to red  
            cell.Offset(0, 0).Font.Color = vbRed  
        End If  
    Next cell  
    'Check for duplications in column Device Name  
    For Each cell In rngDeviceName  
        'set the default color to black  
        cell.Offset(0, 0).Font.Color = vbBlack  
        'Find the duplicate values in range Device Name  
        If Application.Evaluate("COUNTIF(" & rngDeviceName.Address & "," & cell.Address & ")") > 1 Then  
            'mark the duplicate values to red  
            cell.Offset(0, 0).Font.Color = vbRed  
        End If  
    Next cell
```

```
'Check for duplications in column IP-Addr, Node-ID
For Each cell In rngIpAddr
    'set the default color to black
    cell.Offset(0, 0).Font.Color = vbBlack
    ' Find the duplicate values in range IP-Addr, Node-ID
    If Application.Evaluate("COUNTIF(" & rngIpAddr.Address & "," & cell.Address & ")") > 1 Then
        ' mark the duplicate values to red
        cell.Offset(0, 0).Font.Color = vbRed
    End If
Next cell
'Check for duplications in column DEV-inst
For Each cell In rngDevInst
    'set the default color to black
    cell.Offset(0, 0).Font.Color = vbBlack
    ' Find the duplicate values in range DEV-inst
    If Application.Evaluate("COUNTIF(" & rngDevInst.Address & "," & cell.Address & ")") > 1 Then
        ' mark the duplicate values to red
        cell.Offset(0, 0).Font.Color = vbRed
    End If
Next cell
'Check if there is red font in editable areas
For intStartList = 20 To lngCheckCells
    If Range("B" & intStartList).Font.Color = vbRed Or Range("D" & intStartList).Font.Color = vbRed _
    Or Range("J" & intStartList).Font.Color = vbRed Or Range("K" & intStartList).Font.Color = vbRed Then
        'if there is duplication,set the status to "Error-List"
        GoTo ErrorDuplicate
        'if there is duplication, exit sub
    Else
        'no duplication,set the status to "OK-List"
        Range("cl_Status").Value = "OK-List"
        Range("cl_Info").Value = "Check list is OK"
    End If
Next intStartList

rngAliasController = Nothing
rngDeviceName = Nothing
rngIpAddr = Nothing
rngDevInst = Nothing

Exit Sub
```

```
'mark the duplicate values to red
ErrorDuplicate:
    Range("cl_Status").Value = "Error List"
    Range("cl_Info").Value = "Not Unique: check the red cells"
    Call ModifyWBopen()
End

ErrHandler:
    MsgBox("There is a failure in modul CheckDuplication", vbAbortRetryIgnore + vbCritical, "TsNet V2")
    Application.EnableEvents = True
    Application.ScreenUpdating = True
End Sub

'-----
' Description:      Checks the data and the consistence of all input fields
'-----

Sub CheckList()
    'Variable Declaration
    Dim x As Integer
    Dim y As Integer
    Dim z As Integer
    Dim strArrIpAddr() As String
    Dim strIpAddr As String
    Dim lngCounter As Long
    Dim vrtIsNumeric As Object
    Dim objImport As OLEObject
    Dim objDelete As OLEObject
    Dim objTidyUp As OLEObject
    Dim vrtlastRowAlias As Object
    Dim vrtlastrowNetConn As Object

    'Assigning object to variable
    objImport = Sheets("Spec-Devices").OLEObjects("cmd_Import")
    objDelete = Sheets("Spec-Devices").OLEObjects("cmd_Delete")
    objTidyUp = Sheets("Spec-Devices").OLEObjects("cmd_TidyUp")
    'find the last row in column B
    vrtlastRowAlias = Cells(Rows.Count, 2).End(xlUp).Row
    'find the last row in column network connection
    vrtlastrowNetConn = Cells(Rows.Count, 14).End(xlUp).Row

    'Check for Duplications
    Call CheckDuplication()
```

```

For x = 20 To vrtlastRowAlias
    'if range alias controller not empty
    If Range("B" & x).Value <> vbNullString Then
        'if network connection is IP-Controller or MSTP-Router or LON-Router
        If Range("N" & x).Value = "IP-Controller" Or Range("N" & x).Value = "MSTP-Router" Or Range("N" & x).Value = "LON-Router" Then
            'select the active Cell IP-Addr,Node ID
            Cells(x, 10).Select()
            'get the IP of active Cell
            strIpAddr = ActiveCell.Value
            'Returns an array containing IP with dot
            strArrIpAddr = Split(strIpAddr, ".")
            'contains the ip address 3 dots e.g 8.8.8.8
            If (UBound(strArrIpAddr) = 3) Then
                'split the IP address e.g "8" "8" "8" "8"
                For lngCounter = 0 To UBound(strArrIpAddr)
                    'the characters in cells are numeric then true
                    vrtIsNumeric = IsNumeric(strArrIpAddr(lngCounter))
                    If vrtIsNumeric Then
                        Select Case lngCounter
                            Case 0, 3
                                'is it greater than zero and less than 255
                                vrtIsNumeric = (Val(strArrIpAddr(lngCounter)) > 0 And Val(strArrIpAddr(lngCounter)) < 255)
                            Case 1, 2
                                ' is it greater than or equal zero and less than or equal 255
                                vrtIsNumeric = (Val(strArrIpAddr(lngCounter)) >= 0 And Val(strArrIpAddr(lngCounter)) <= 255)
                        End Select
                        Cells(x, 10).Font.Color = vbBlack
                    End If
                    'if it isnt numeric
                    If Not vrtIsNumeric Then GoTo BadIpFormat
                    'next split part of ip
                    Cells(x, 10).Font.Color = vbBlack
                Next lngCounter
            Else
                'it doesnt have the format of an ip
                If Cells(x, 10) <> vbNullString Then
                    Cells(x, 10).Font.Color = RGB(255, 0, 0)

                    GoTo BadIpFormat
                End If
            End If
            'Else: GoTo EmptyCells
        End If
    End If

```

```
'the network connection is MSTP-Controller or LON-Controller
If Range("N" & x).Value = "MSTP-Controller" Or Range("N" & x).Value = "LON-Controller" Then
    'select the active Cell IP-Addr,Node ID
    Cells(x, 10).Select()
    'get the IP of active Cell
    strIpAddr = ActiveCell.Value
    'is it numeric and contains three characters
    If IsNumeric(strIpAddr) Then
        'less than or equal 255
        If strIpAddr <= 255 Then
            'greater than or equal zero
            If strIpAddr >= 0 Then
                'Set the color to black
                Range("J" & x).Font.Color = vbBlack
                'Dev-Inst equal to IP-Addr,Node ID
                Range("K" & x).Value = Range("J" & x).Value
            End If
            Range("J" & x).Font.Color = vbBlack
            Range("cl_Status").Value = "OK-List"
            Range("cl_Info").Value = "List ist okay"
            Range("cl_Date").Value = Date & " " & Time
        Else
            'set color to red
            Range("J" & x).Font.Color = RGB(255, 0, 0)
            'Ip-Format is OK
            GoTo BadIpFormat
        End If
    Else
        'Wrong IP Format then go to BadIpFormat
        Range("J" & x).Font.Color = RGB(255, 0, 0)
        GoTo BadIpFormat
    End If
End If
'if network connection is MSTP-Controller
If Range("N" & x).Value = "MSTP-Controller" Then
    For y = 20 To vrtlastrowNetConn
        'if network connection is MSTP-Router
        If Range("N" & y).Value = "MSTP-Router" Then
            'if not empty
            If Range("N" & x).Value <> vbNullString Then
                'network no of mstp-controller is identical to mstp-router
                Range("L" & x).Value = Range("L" & y).Value
            End If
        End If
    End For
End If
```



```

        End If
    Next y
End If
'if network connection is LON-Controller
If Range("N" & x).Value = "LON-Controller" Then
    For z = 20 To vrtlastrowNetConn
        'if network connection is LON-Router
        If Range("N" & z).Value = "LON-Router" Then
            'if not empty
            If Range("N" & x).Value <> vbNullString Then
                'network no of LON-Controller is identical to LON-Router
                Range("L" & x).Value = Range("L" & z).Value
            End If
        End If
    Next z
End If
'all conditions are fulfilled status = "OK-List"
Range("cl_Status").Value = "OK-List"
Range("cl_Info").Value = "List is okay"
Range("cl_Date").Value = Date & " " & Time

Else : GoTo EmptyCells
End If
Next x
'if status is Ok-List
If Range("I4").Value = "OK-List" Then
    Call ModifyWBopen()
    objDelete.Enabled = True
    objTidyUp.Enabled = True
    objImport.Enabled = True
End If

Exit Sub
'If there is a empty cell then error list
EmptyCells:
Range("cl_Status").Value = "Error List"
Range("cl_Info").Value = "Check the alias controller empty column"
Range("cl_Date").Value = Date & " " & Time
Call ModifyWBopen()

Exit Sub
'If there is wrong Ip Format in Column IP-Addr
BadIpFormat:

```

```

Range("cl_Status").Value = "Error List"
Range("cl_Info").Value = "Wrong Ip Format in Column IP-Addr"
Range("cl_Date").Value = Date & " " & Time
Call ModifyWBopen()

```

```
End Sub
```

```

' -----
' Description:      Checks the data and the consistence of all input fields and EDE
' -----

```

```

Sub CheckDevices()
'Variable Declaration
Dim x As Integer
Dim y As Integer
Dim z As Integer
Dim strArrIpAddr() As String
Dim strIpAddr As String
Dim lngCounter As Long
Dim vrtIsNumeric As Object
Dim vrtlastRowAlias As Object
Dim vrtlastRowConn As Object

'find the last row in column B
vrtlastRowAlias = Cells(Rows.Count, 2).End(xlUp).Row
'find the last row in column network connection
vrtlastRowConn = Cells(Rows.Count, 14).End(xlUp).Row

'Check for Duplications
Call CheckDuplication()

For x = 20 To vrtlastRowAlias
'Range alias controller, Device name,IP-Add,Node-ID and Dev-Str m not empty
If Range("B" & x).Value <> vbNullString And Range("D" & x).Value <> vbNullString _
And Range("J" & x).Value <> vbNullString And Range("K" & x).Value <> vbNullString Then
'if network connection is IP-Controller or MSTP-Router or LON-Router
If Range("N" & x).Value = "IP-Controller" Or Range("N" & x).Value = "MSTP-Router" Or Range("N" & x).Value = "LON-Router" Then
'select the active Cell IP-Addr,Node ID
Cells(x, 10).Select()
'get the IP of active Cell
strIpAddr = ActiveCell.Value
'Returns an array containing IP with dot
strArrIpAddr = Split(strIpAddr, ".")
'contains the ip address 3 dots e.g 8.8.8.8

```

```
If (UBound(strArrIpAddress) = 3) Then
    'split the IP address e.g "8" "8" "8" "8"
    For lngCounter = 0 To UBound(strArrIpAddress)
        'the characters in cells are numeric then true
        vrtIsNumeric = IsNumeric(strArrIpAddress(lngCounter))
        If vrtIsNumeric Then
            Select Case lngCounter
                Case 0, 3
                    'is it greater than zero and less than 255
                    vrtIsNumeric = (Val(strArrIpAddress(lngCounter)) > 0 And Val(strArrIpAddress(lngCounter)) < 255)
                Case 1, 2
                    ' is it greater than or equal zero and less than or equal 255
                    vrtIsNumeric = (Val(strArrIpAddress(lngCounter)) >= 0 And Val(strArrIpAddress(lngCounter)) <= 255)
            End Select
            Cells(x, 10).Font.Color = vbBlack
        End If
        'if it isnt numeric
        If Not vrtIsNumeric Then GoTo BadIpFormat
        'next split part of ip
        Cells(x, 10).Font.Color = vbBlack
    Next lngCounter
Else
    'it doesnt have the format of an ip
    Cells(x, 10).Font.Color = RGB(255, 0, 0)
    GoTo BadIpFormat
End If
End If
'the network connection is MSTP-Controller or LON-Controller
If Range("N" & x).Value = "MSTP-Controller" Or Range("N" & x).Value = "LON-Controller" Then
    'select the active Cell IP-Addr,Node ID
    Cells(x, 10).Select()
    'get the IP of active Cell
    strIpAddress = ActiveCell.Value
    'is it numeric and contains three characters
    If IsNumeric(strIpAddress) Then
        'less than or equal 255
        If strIpAddress <= 255 Then
            'greater than or equal zero
            If strIpAddress >= 0 Then
                'Set the color to black
                Range("J" & x).Font.Color = vbBlack
                'Dev-Inst equal to IP-Addr,Node ID
                Range("K" & x).Value = Range("J" & x).Value
            End If
        End If
    End If
End If
```

```
End If
'Ip-Format is OK
Range("j" & x).Font.Color = vbBlack
Range("cl_Status").Value = "OK-Device"
Range("cl_Info").Value = "Devices are okay"
Range("cl_Date").Value = Date
End If
Else
'Wrong IP Format then go to BadIpFormat
Range("j" & x).Font.Color = RGB(255, 0, 0)
GoTo BadIpFormat
End If
End If
'if the network - no Is Not Empty
If Range("L" & x).Value <> vbNullString Then
'select the active Cell IP-Addr,Node ID
Cells(x, 12).Select()
'get the IP of active Cell
strIpAddr = ActiveCell.Value
'Dev-Inst equal to IP-Addr,Node ID
If IsNumeric(strIpAddr) Then
'less than or equal 255
If strIpAddr <= 255 Then
'greater than or equal zero
If strIpAddr >= 0 Then
'Set the color to black
Range("j" & x).Font.Color = vbBlack
End If
'Ip-Format is OK
Range("j" & x).Font.Color = vbBlack
Range("cl_Status").Value = "OK-Device"
Range("cl_Info").Value = "Devices are okay"
Range("cl_Date").Value = Date
End If
Else
'Wrong IP Format then go to BadIpFormat
Range("j" & x).Font.Color = RGB(255, 0, 0)
GoTo BadIpFormat
End If
End If
'if network connection is MSTP-Controller
If Range("N" & x).Value = "MSTP-Controller" Then
```

```
For y = 20 To vrtlastRowConn
    'if network connection is MSTP-Router
    If Range("N" & y).Value = "MSTP-Router" Then
        'if not empty
        If Range("N" & x).Value <> vbNullString Then
            'network no of mstp-controller is identical to mstp-router
            Range("L" & x).Value = Range("L" & y).Value
        End If
    End If
Next y
End If
'if network connection is LON-Controller
If Range("N" & x).Value = "LON-Controller" Then
    For z = 20 To vrtlastRowConn
        'if network connection is LON-Router
        If Range("N" & z).Value = "LON-Router" Then
            'if not empty
            If Range("N" & x).Value <> vbNullString Then
                'network no of LON-Controller is identical to LON-Router
                Range("L" & x).Value = Range("L" & z).Value
            End If
        End If
    Next z
End If
'all conditions are fulfilled status = "OK-List"
Range("cl_Status").Value = "OK-List"
Range("cl_Info").Value = "List ist okay"
Range("cl_Date").Value = Date
Else : GoTo EmptyCells
End If

If Range("N" & x).Value = "MSTP-Router" Then
    If Range("L" & x).Value <> vbNullString Then
        Else : GoTo EmptyCells
    End If
End If

If Range("N" & x).Value = "LON-Router" Then
    If Range("L" & x).Value <> vbNullString Then
        Else : GoTo EmptyCells
    End If
End If
Next x
```

```
'Check for matched in EDE and Spec-Devices  
Call CheckDevicesInEDE()
```

```
EdeDevices:
```

```
Range("cl_Status").Value = "OK-Devices"  
Range("cl_Info").Value = "all devices are OK"  
Range("cl_Date").Value = Date & " " & Time  
Call ModifyWBopen()  
End
```

```
Exit Sub
```

```
EmptyCells:
```

```
Range("cl_Status").Value = "Error-Devices"  
Range("cl_Info").Value = "There are empty cells"  
Range("cl_Date").Value = Date & " " & Time  
Call ModifyWBopen()  
End
```

```
Exit Sub
```

```
BadIpFormat:
```

```
Range("cl_Status").Value = "Error-Devices"  
Range("cl_Info").Value = "Wrong Ip Format in Column IP-Addr"  
Range("cl_Date").Value = Date & " " & Time  
Call ModifyWBopen()
```

```
End Sub
```

```
' -----  
' Description:      Checks the data and the consistence of EDE  
' -----  
Sub CheckDevicesInEDE()  
    'Variable Declaration  
    Dim intStartListEDE As Integer  
    Dim intStartListSpec As Integer  
    Dim strSpecData As String  
    Dim strEdeData As String  
    Dim WbsSpecDevices As Worksheet  
    Dim WbsEde As Worksheet  
  
    'Get the name of active sheet  
    WbsSpecDevices = ActiveWorkbook.ActiveSheet  
    'Get the name of EDE  
    WbsEde = ActiveWorkbook.Sheets("EDE")  
  
    'find the last row in column K in Spec Devices  
    With WbsSpecDevices  
        strSpecData = ActiveSheet.Cells(Rows.Count, "K").End(xlUp).Row  
    End With  
    'find the last row in column E in EDE  
    With WbsEde  
        strEdeData = Worksheets("EDE").Cells(Rows.Count, "E").End(xlUp).Row  
    End With  
  
    For intStartListSpec = 20 To strSpecData  
        For intStartListEDE = 8 To strEdeData  
            'if network connection is IP-Controller or MSTP-Controller or LON-Controller  
            If WbsSpecDevices.Range("N" & intStartListSpec).Value = "IP-Controller" Or _  
                WbsSpecDevices.Range("N" & intStartListSpec).Value = "MSTP-Controller" Or _  
                WbsSpecDevices.Range("N" & intStartListSpec).Value = "LON-Controller" Then  
                'Check if in sheet("EDE") Object-type = 8  
                If WbsEde.Range("E" & intStartListEDE).Value = "8" Then  
                    'Check if EDE!Device-object-instance = Spec-Devices!Dev-Inst  
                    If WbsEde.Range("C" & intStartListEDE).Value = WbsSpecDevices.Range("K" & intStartListSpec).Value Then  
                        'Check if EDE!Object-name = Spec-Devices!Dev-Name  
                        If WbsEde.Range("D" & intStartListEDE).Value = WbsSpecDevices.Range("D" & intStartListSpec).Value Then  
                            'if EDE!Object-name = Spec-Devices!Dev-Name then exit  
                            GoTo EdeMatched  
                        End If  
                    End If  
                End If  
            End If  
        End For  
    End For  
End Sub
```

```

        End If
    Next intStartListEDE
Next intStartListSpec
'if there are no matches in Ede
ErrorObjectType:
Worksheets("Spec-Devices").Range("cl_Status").Value = "Error Device"
Worksheets("Spec-Devices").Range("cl_Info").Value = "There are no matches with EDE"
Worksheets("Spec-Devices").Range("cl_Date").Value = Date & " " & Time
End
'if matches found
EdeMatched:
Worksheets("Spec-Devices").Range("cl_Status").Value = "Ok - Devices"
Worksheets("Spec-Devices").Range("cl_Info").Value = "Eda data and spec data are matched"
Worksheets("Spec-Devices").Range("cl_Date").Value = Date & " " & Time
Exit Sub

```

End Sub

```

' -----
' Description:      Selected TsNet: Dev-Inst and Device-Name of the selected TsNet device is shown
'                  EDE area:In Worksheet EDE the line with object-type 8 and where device-instance
'                  equal to selected TsNet- device Dev-Inst
' -----

```

```

Sub ImportEDE()
'Variable Declaration
Dim strSelectedDevInst As String
Dim strSelectedDevName As String
Dim strSelectedNetConn As String
Dim strSelectedRow As String
Dim intStartListEDE As Integer
Dim strEdeData As String
Dim WbsEde As Worksheet
'Get the value of the selected Device Name
strSelectedDevName = ActiveCell.EntireRow.Cells(1, 4).Value
'Get the value of the selected Dev-Inst
strSelectedDevInst = ActiveCell.EntireRow.Cells(1, 11).Value
'Get the value of the selected network connection
strSelectedNetConn = ActiveCell.EntireRow.Cells(1, 14).Value
'Get the value of the selected row range A
strSelectedRow = ActiveCell.EntireRow.Cells(1, 1).Value
'Assigning object to variable
WbsEde = ActiveWorkbook.Sheets("EDE")
With WbsEde

```



```
' find the last row of range E in EdE
strEdeData = Worksheets("EDE").Cells(Rows.Count, "E").End(xlUp).Row
End With
' check the selected row in TsNET, if selected cell not empty
If strSelectedRow <> vbNullString Then
    ' the selected device is type of IP-Controller or MSTP-Controller, or MSTP-Controller
    If strSelectedNetConn = "IP-Controller" Or strSelectedNetConn = "MSTP-Controller" Or strSelectedNetConn = "LON-Controller" Then
        For intStartListEDE = 8 To strEdeData
            'check if in worksheet line with object-type = 8 and device-instance selected TsNet-device Dev-Inst
            If WbsEde.Range("E" & intStartListEDE).Value = "8" And WbsEde.Range("C" & intStartListEDE).Value = strSelectedDevInst Then
                'show the found DEV-INST of EDE in frmImportEde
                frmImportEde.txt_EdeDevInst.Text = WbsEde.Range("C" & intStartListEDE).Value
                'show the found DEVICE-Name of EDE in frmImportEde
                frmImportEde.txt_EdeDevName.Text = WbsEde.Range("D" & intStartListEDE).Value
                'if the line are found then exit for
                Exit For
            End If
        Next intStartListEDE
        'show the selected DEV-Inst in frmImportEde
        frmImportEde.txt_SelectedDevInst.Text = strSelectedDevInst
        'show the selected Device Name in frmImportEde
        frmImportEde.txt_SelectedDevName.Text = strSelectedDevName
        ' if selected TsNet Devinst is empty or selected TsNet Dev name is empty
        If strSelectedDevInst = vbNullString Or strSelectedDevName = vbNullString Then
            'shows the negative message
            frmImportEde.txt_result.Text = "Selected device is not correctly defined, define devicename and device-instance"
            'Sets the x symbol to red
            frmImportEde.txt_Symbol.ForeColor = vbRed
            'Shows the x symbol
            frmImportEde.txt_Symbol.Text = "X"
            'the button import is disabled
            frmImportEde.cmd_ImportEde.Enabled = False
        End If
        'Open the sub menu Import EDE
        frmImportEde.Show()
        'if no device type
    Else : GoTo NoDeviceType
End If
'if no selected row
Else : GoTo NotSelected
End If

Exit Sub
```

```

' if no selected row
NotSelected:
Worksheets("Spec-Devices").Range("c1_Status").Value = "Error Import"
Worksheets("Spec-Devices").Range("c1_Info").Value = "No device is selected"
Worksheets("Spec-Devices").Range("c1_Date").Value = Date & " " & Time
End
'if no device type
NoDeviceType:
Worksheets("Spec-Devices").Range("c1_Status").Value = "Error Import"
Worksheets("Spec-Devices").Range("c1_Info").Value = "Please check the network connection"
Worksheets("Spec-Devices").Range("c1_Date").Value = Date & " " & Time
End
End Sub

' -----
' Description:      Checks userform Import Ede for making a decision which message should be shown
' -----

Sub CheckFrmImportEDE()
'Checks the import file information with selected TsNet information
If frmImportEDE.txt_ImportDevInst.Text = frmImportEDE.txt_SelectedDevInst.Text Then
    If frmImportEDE.txt_ImportDevName.Text = frmImportEDE.txt_SelectedDevName.Text Then
        Else : GoTo ImportFailed
    End If
Else : GoTo ImportFailed
End If
'Checks the Selected TsNet information, it should be not empty
If frmImportEDE.txt_SelectedDevInst.Text <> vbNullString Then
    If frmImportEDE.txt_SelectedDevName.Text <> vbNullString Then
        Else : GoTo SelDevFailed
    End If
Else : GoTo SelDevFailed
End If
'Data should be consistent
If frmImportEDE.txt_ImportDevName.Text = frmImportEDE.txt_SelectedDevName.Text And _
frmImportEDE.txt_ImportDevInst.Text = frmImportEDE.txt_SelectedDevInst.Text And _
frmImportEDE.txt_EdeDevInst.Text = frmImportEDE.txt_ImportDevInst.Text Then
    'shows the message
    frmImportEDE.txt_result.Text = "Data is consistent: Import replaces actual EDE for device"
    'Sets the tick symbol to green
    frmImportEDE.txt_Symbol.ForeColor = vbGreen
    'shows the tick symbol
    frmImportEDE.txt_Symbol.Text = ChrW(&H2713)
    'the button import is enabled

```

```
    frmImportEDE.cmd_ImportEDE.Enabled = True
    frmImportEDE.cmd_ImportEDE.Width = "93"
    Exit Sub
End If
'Data Consistent and import is possible
If frmImportEDE.txt_ImportDevName.Text = frmImportEDE.txt_SelectedDevName.Text And _
    frmImportEDE.txt_ImportDevInst.Text = frmImportEDE.txt_SelectedDevInst.Text And _
    frmImportEDE.txt_EdeDevInst.Text = vbNullString Then
    'shows the message
    frmImportEDE.txt_result.Text = "Data is consistent: Imported data will be appended to worksheet EDE"
    'Sets the tick symbol to green
    frmImportEDE.txt_Symbol.ForeColor = vbGreen
    'shows the tick symbol
    frmImportEDE.txt_Symbol.Text = ChrW(&H2713)
    'the button import is enabled
    frmImportEDE.cmd_ImportEDE.Enabled = True
    frmImportEDE.cmd_ImportEDE.Width = "93"
End If

Exit Sub
'if selected device is not equal to Import device, no import possible
ImportFailed:
    'shows the negative message
    frmImportEDE.txt_result.Text = "Selected device is not equal to import device"
    'Sets the x symbol to red
    frmImportEDE.txt_Symbol.ForeColor = vbRed
    'Shows the x symbol
    frmImportEDE.txt_Symbol.Text = "X"
    'the button import is disabled
    frmImportEDE.cmd_ImportEDE.Enabled = False
    Exit Sub
    'if selected device is not correctly defined, no import possible
SelDevFailed:
    'shows the negative message
    frmImportEDE.txt_result.Text = "Selected device is not correctly defined"
    'Sets the x symbol to red
    frmImportEDE.txt_Symbol.ForeColor = vbRed
    'Shows the x symbol
    frmImportEDE.txt_Symbol.Text = "X"
    'the button import is disabled
    frmImportEDE.cmd_ImportEDE.Enabled = False
End Sub
```

```

' -----
' Description:      When CheckFrmImportEDE is successfully executed, and when the user clicks import
'                  button, imported data will be appended to worksheet EDE
' -----
Sub appendingEDE()
    'Variable Declaration
    Dim strImportCsv As String
    Dim intStartListImport As Integer
    Dim objCheckDevices As OLEObject
    Dim objImportStatus As OLEObject
    Dim vrtStartList As Object
    Dim x As Integer

    'Assigning object to variable
    objImportStatus = Sheets("Spec-Devices").OLEObjects("lbl_ImportStatus")
    objCheckDevices = Sheets("Spec-Devices").OLEObjects("cmd_CheckDevices")
    'Find the last row in the InsertedData
    strImportCsv = Sheets("InsertedData").Cells(Rows.Count, "B").End(xlUp).Row
    Application.ScreenUpdating = False
    'is the value equal?
    If frmImportEDE.txt_ImportDevName.Text = frmImportEDE.txt_SelectedDevName.Text And _
        frmImportEDE.txt_ImportDevInst.Text = frmImportEDE.txt_SelectedDevInst.Text And frmImportEDE.txt_EdeDevInst.Text = vbNullString Then
        'Activate the sheet InsertedData
        Sheets("InsertedData").Activate()
        For intStartListImport = 1 To strImportCsv
            'Check if the file has a Ede format
            If Range("A" & intStartListImport).Value = "# mandatory" Then
                If Range("A" & intStartListImport + 1).Value = "# keyname" Then
                    'Select the row of the data
                    Cells(intStartListImport + 1, "D").Select()
                    'get the value of activeCell
                    vrtStartList = ActiveCell.Row
                    For x = vrtStartList To strImportCsv
                        'All lines in the import, where import.device-obj-instance equal to selected TsNet device.Dev-Inst
                        If Cells(x, 2) = frmImportEDE.txt_SelectedDevInst.Text Then
                            'copy the cell which is equal
                            Range(Cells(x, 1), Cells(x, 14)).Copy()
                            'paste it in Sheet EDE
                            Worksheets("EDE").Range("B" & Rows.Count).End(xlUp).Offset(1).PasteSpecial()
                        End If
                    Next x
                    ' if everything is pasted then go to weiter
                    GoTo UnloadUserFrm
                End If
            End If
        Next x
    End If
End Sub

```

```
        End If
    End If
    Next intStartListImport
End If
UnloadUserFrm:
'Close the userform submenu
Unload(frmImportEDE)
Worksheets("Spec-Devices").Activate()
'Set status: import Ok
Range("cl_Status").Value = "Import - OK"
'Set info: import is done
Range("cl_Info").Value = "Import is done"
Range("cl_Date").Value = Date & " " & Time

If Range("I4").Value = "Import - OK" Then
    'Check devices is enabled
    objCheckDevices.Enabled = True
    ActiveCell.Cells(1, 16).Value = "Imported"
    ActiveCell.Cells(1, 15).Value = Date
Else
    'Check devices is enabled
    objCheckDevices.Enabled = False
    ActiveCell.Cells(1, 16).Value = "Import Failed"
    ActiveCell.Cells(1, 15).Value = Date
End If

Application.ScreenUpdating = True
End Sub
```

```
' -----  
' Description:      replacing EDE Data is carried out, if already for the same device-obj-instance  
'                  as the selected TsNet device exists in Worksheet EDE  
' -----  
  
Sub replacingEDE()  
    'Variable Declaration  
    Dim objCheckDevices As OLEObject  
    Dim WbsEde As Worksheet  
    Dim WbsCSV As Worksheet  
    Dim strStartListEDE As String  
    Dim strStartListCSV As String  
    Dim objImportStatus As OLEObject  
    Dim i As Integer  
    Dim x As Integer  
    Dim by As Object  
    Dim vrtStartList As Object  
    'Assigning object to variable  
    objCheckDevices = Sheets("Spec-Devices").OLEObjects("cmd_CheckDevices")  
    objImportStatus = Sheets("Spec-Devices").OLEObjects("lbl_ImportStatus")  
    WbsEde = ThisWorkbook.Worksheets("EDE")  
    WbsCSV = ActiveWorkbook.Sheets("InsertedData")  
    'finds the end of the row in the EDE sheet  
    With WbsEde  
        strStartListEDE = WbsEde.Cells(Rows.Count, "C").End(xlUp).Row  
    End With  
    'finds the end of the row in the InsertedData sheet  
    With WbsCSV  
        strStartListCSV = WbsCSV.Cells(Rows.Count, "B").End(xlUp).Row  
    End With  
  
    Application.ScreenUpdating = False  
    'Activate the sheet EDE  
    Worksheets("EDE").Activate()  
    'is the value equal?  
    If frmImportEDE.txt_ImportDevName.Text = frmImportEDE.txt_SelectedDevName.Text And _  
        frmImportEDE.txt_ImportDevInst.Text = frmImportEDE.txt_SelectedDevInst.Text And _  
        frmImportEDE.txt_EdeDevInst.Text = frmImportEDE.txt_ImportDevInst.Text Then  
        For i = strStartListEDE To 8 Step by - 1  
            'Select the row  
            Rows(i).Select()  
            'is it equal
```

```
If ThisWorkbook.Worksheets("EDE").Cells(i, 3).Value = frmImportEDE.txt_SelectedDevInst.Text Then
    'Select the row
    Rows(i).Select()
    'Delete
    Rows(i).Delete()
End If
Next
For i = 1 To strStartListCSV
    'Activate the sheet InsertedData
    Sheets("InsertedData").Activate()
    'check the ede file format
    If Range("A" & i).Value = "# mandatory" Then
        If Range("A" & i + 1).Value = "# keyname" Then
            'Select the Start
            Cells(i + 1, "D").Select()
            'get the value of the activeCell
            vrtStartList = ActiveCell.Row
            For x = vrtStartList To strStartListCSV
                'is it equal?
                If Cells(x, 2) = frmImportEDE.txt_SelectedDevInst.Text Then
                    'Copy it
                    Range(Cells(x, 1), Cells(x, 14)).Copy()
                    'Paste it into ede
                    Worksheets("EDE").Range("B" & Rows.Count).End(xlUp).Offset(1).PasteSpecial()
                End If
            Next x
        End If
    End If
End If
Next i
End If
'Close the userform submenu
Unload(frmImportEDE)
Worksheets("Spec-Devices").Activate()
'Set status: import Ok
Range("cl_Status").Value = "Import - OK"
'Set info: import is done
Range("cl_Info").Value = "Import is done"
Range("cl_Date").Value = Date & " " & Time

If Range("I4").Value = "Import - OK" Then
    ActiveCell.Cells(1, 16).Value = "Imported"
    objCheckDevices.Enabled = True
    ActiveCell.Cells(1, 15).Value = Date
```

```
Else
    ActiveCell.Cells(1, 16).Value = "Import Failed"
    'Check devices is enabled
    objCheckDevices.Enabled = False
    ActiveCell.Cells(1, 15).Value = Date
End If

Application.ScreenUpdating = True
End Sub

' -----
' Description:      Deletes all lines in EDE where EDE.device-obj.-instance equal to selected TsNet
'                  Dev-Inst
' -----

Sub DeleteEDE(ByVal vrtSelectedDevInst As Object)
    'Variable Declaration
    Dim i As Integer
    Dim strLastRowEDE As String
    Dim by As Object
    Application.ScreenUpdating = False
    'Activate Ede
    Worksheets("EDE").Activate()
    'get the last row of ede file
    strLastRowEDE = ThisWorkbook.Worksheets("EDE").Cells(Rows.Count, 3).End(xlUp).Row
    For i = strLastRowEDE To 8 Step by - 1
        'select the last row
        Rows(i).Select()
        'is it equal?
        If ThisWorkbook.Worksheets("EDE").Cells(i, 3).Value = vrtSelectedDevInst Then
            'select the last row
            Rows(i).Select()
            'select the last row
            Rows(i).Delete()
        End If
    Next
    'activate spec-devices
    Worksheets("Spec-Devices").Activate()
    Application.ScreenUpdating = True
End Sub
```



```
' .....  
' Description:      The Ede file (.csv) can be selected. It will be checked for the ede file format  
' .....  
Sub openFilecsv(ByVal DevInst As Object, ByVal DevName As Object)  
    'Variable Declaration  
    Dim ObjFolder As Object  
    Dim strFileName As String  
    Dim WbCSV As Object  
    Dim WbsCSV As Object  
    Dim strImportCsv As String  
    Dim i As Integer  
    Dim x As Integer  
    Dim vrtStartList As Object  
  
    'Assigning object to variable  
    ObjFolder = Application.FileDialog(msoFileDialogFilePicker)  
    'delete other options  
    ObjFolder.Filters.Clear()  
    'only .csv files to choose  
    ObjFolder.Filters.Add("Only .CSV Files", "*.csv")  
    'no multi select options  
    ObjFolder.AllowMultiSelect = False  
    'Title of the openfile dialog  
    ObjFolder.Title = "Select a .csv file to import"  
  
    Application.ScreenUpdating = False  
  
    If ObjFolder.Show Then  
        Application.ScreenUpdating = False  
        Application.DisplayAlerts = False  
        'Delete InsertedData  
        Sheets("InsertedData").Delete()  
        Application.DisplayAlerts = True  
        'Get the filename  
        strFileName = ObjFolder.SelectedItems(1)  
        Application.ScreenUpdating = False  
        'show the file name in userform  
        frmImportEDE.txt_FileName.Text = strFileName  
        'converting the csv file  
        Workbooks.Open(Filename:=strFileName, local:=True)  
        WbCSV = ActiveWorkbook.Name  
        WbsCSV = ActiveSheet.Name  
        'Copy the content of the file
```

```

Sheets(WbCSV).Rows.Copy()
ThisWorkbook.Activate()
'Add a new sheet
Sheets.Add()
ActiveSheet.Name = "InsertedData"
'Paste the content into InsertedData
ActiveSheet.Paste()
Application.DisplayAlerts = False
'Close the opened csv-file
Workbooks(WbCSV).Close(False)
Application.DisplayAlerts = True
'find the end of the inserted data
strImportCsv = ThisWorkbook.Sheets("InsertedData").Cells(Rows.Count, "A").End(xlUp).Row
'Check the csv format
If Left(Range("A1").Value, 1) = "#" Then
    For i = 1 To strImportCsv
        If Range("A" & i).Value = "# mandatory" And Range("B" & i).Value = "mandatory" And Range("C" & i).Value = "mandatory" Then
            If Range("A" & i + 1).Value = "# keyname" And Range("B" & i + 1).Value = "device-obj-instance" And Range("C" & i +
1).Value = "object-name" Then
                'get the start of the data
                Cells(i + 1, "D").Select()
                'get the value of active cell
                vrtStartList = ActiveCell.Row
                For x = vrtStartList To strImportCsv
                    'get the line with object-type 8 and device-instance equal to selected TsNet device dev- inst
                    If Range("D" & x).Value = "8" And Range("B" & x).Value = DevInst Then
                        frmImportEDE.txt_ImportDevInst.Text = Range("B" & x).Value
                        frmImportEDE.txt_ImportDevName.Text = Range("C" & x).Value
                        GoTo ImportFound
                    End If
                Next x
            End If
        End If
    Next i
Else : GoTo NoEdeFile
End If
Else
End If
ObjFolder = Nothing

On Error GoTo 0
'

If frmImportEDE.txt_ImportDevInst.Text <> vbNullString And frmImportEDE.txt_ImportDevName.Text <> vbNullString Then

```

```
Else
    frmImportEDE.txt_result.Text = "Please choose an other Import file"
    frmImportEDE.txt_Symbol.ForeColor = vbRed
    frmImportEDE.txt_Symbol.Text = "X"
End If

Exit Sub
Application.ScreenUpdating = True
ImportFound:
Call CheckFrmImportEDE()
Exit Sub

Application.ScreenUpdating = True

NoEdeFile:
    frmImportEDE.txt_result.Text = "No EDE file, no import possible"
    frmImportEDE.txt_Symbol.ForeColor = vbRed
    frmImportEDE.txt_Symbol.Text = "X"
End Sub

' -----
' Description:      Removes all EDE data from controllers which are not in the devicelist
' -----

Sub TidyUp()
    'Variable declaration
    Dim strDevInstValue As String
    Dim rng As Range
    Dim WbsSpecData As Object
    Dim WbsEdeData As Object
    Dim strLastRowSpec As String
    Dim strLastRowEDE As String
    Dim x As Integer
    Dim i As Integer
    Dim by As Integer

    Application.ScreenUpdating = False
    WbsSpecData = ThisWorkbook.Worksheets("Spec-Devices")
    WbsEdeData = ThisWorkbook.Worksheets("EDE")

    With WbsSpecData
        'Find the last row in Column Dev-Inst
        strLastRowSpec = ThisWorkbook.Worksheets("Spec-Devices").Cells(Rows.Count, "K").End(xlUp).Row
    End With
```

```
With WbsEdeData
    'Find the last row in Column device obj.-instance
    strLastRowEDE = ThisWorkbook.Worksheets("EDE").Cells(Rows.Count, "C").End(xlUp).Row
End With

For x = strLastRowEDE To 8 Step by - 1
    'Is object type 8?
    If WbsEdeData.Range("E" & x).Value = "8" Then
        'get the value of device obj.-instance
        strDevInstValue = WbsEdeData.Range("C" & x).Value
        'Search for a strDevInstValue
        WbsSpecData.Columns("K:K").Select()
        'Set find strDevInstValue
        rng = Selection.Find(strDevInstValue)
        ' is it not found
        If rng Is Nothing Then
            Sheets("EDE").Activate()
            'select entire row from ede
            Rows(x).EntireRow.Select()
            'delete entire row from ede
            Rows(x).Delete()
            GoTo GoToI
        End If
    Else : GoTo GoToX
End If

GoToI:
    For i = strLastRowEDE To 8 Step by - 1
        Rows(i).Select()
        If WbsEdeData.Range("C" & i).Value = strDevInstValue Then
            Rows(i).EntireRow.Select()
            'delete entire row from ede
            Rows(i).Delete()
        End If
    Next i

GoToX:
    Next x
    Application.ScreenUpdating = True
    Sheets("Spec-Devices").Activate()
End Sub
```

```
' ----- '
' Description:      Opens pdf File with page number in Pdf document that should be called '
' ----- '

Sub GetHelp()
    'the full name of pdf file
    Dim strHelpFile As String
    'the page number in Pdf document that should be called
    Dim intPage As Integer
    'Parameter for Acrobat reader
    Dim strParam As String
    'Enable error-handling routine
    On Error GoTo NoReader
    'The name of the path
    strHelpFile = "\\ch021012\SIM_MOD\IPA Fatma Yilmaz\04_Realisieren\TestDaten\PD103267-T-0104_EN_TestManual_TsNet.pdf"
    'Selection of the sheet name and their help pages
    Select Case ActiveSheet.Name
        Case "Config V.1"
            intPage = 1
        Case "Overview"
            intPage = 5
        Case "Spec-Devices"
            intPage = 9
        Case "Spec-Objects"
            intPage = 10
    End Select
    'Set the parameters together
    strParam = "/A " & Chr(34) & "page=" & intPage & "&navpanes=0=OpenActions" & Chr(34) & " " & strHelpFile
    'The previously defined parameters will be called
    ShellExecute(0, vbNullString, "acrord32", strParam, " ", G_SW_SHOWNORMAL)
    Exit Sub

NoReader:
    'if there is no pdf programm
    MsgBox("Sorry, the File is not able to open" & vbCrLf & "Error Number: " & Err.Number & vbCrLf & "Error Description: " & Err.Description)
End Sub
```

2.7.4 frmImportEDE

```

' -----
' Name:          frmImportEDE
' File:          TestTemplate.xlsm
' -----
' Copyright @2016, SIEMENS Building Technologies
' -----
' OS:           Win64
' Lang:         Microsoft Visual Basic 6.5
' -----
' Project:      TsNet
' Author:       Fatma Yilmaz
' Version      V.1
' Date         24-03-2016
' -----
' Description:   Aus diesem Makro werden die einzelne Funktionen/Prozeduren aufgerufen,
'               die im Modul "mdl_SpecDevices" liegen
' -----
' History:      24-03-2016  Fatma Yilmaz
'               V.1 Document Creation for IPA
' -----
Private Sub cmd_ImportEDE_Click()
    'Check the content of the message box
    If frmImportEDE.txt_result.Text = "Data is consistent: Import replaces actual EDE for device" Then
        strSelectedDevInst = frmImportEDE.txt_SelectedDevInst.Text
        'Call the function replacingEDE
        Call replacingEDE()
        Exit Sub
    End If
    'Check the content of the message box
    If frmImportEDE.txt_result.Text = "Data is consistent: Imported data will be appended to worksheet EDE" Then
        'Call the function appendingEDE
        Call appendingEDE()
        Exit Sub
    End If
End Sub

```

```
Private Sub cmd_OpenFileName_Click()  
    'Variable Declaration  
    Dim ObjFolder As FileDialog  
    Dim strFileName As String  
    Dim intStartList As Integer  
  
    'Assigning object to variable  
    strSelectedDevName = ActiveCell.EntireRow.Cells(1, 4).Value  
    strSelectedDevInst = ActiveCell.EntireRow.Cells(1, 11).Value  
    'Create a FileDialog object as a Folder Picker dialog box  
    Call openFilecsv(strSelectedDevInst, strSelectedDevName)  
End Sub  
  
Private Sub cmd_CancelEDE_Click()  
    Unload(frmImportEDE)  
    Worksheets("Spec-Devices").Activate()  
    'Set status: import Ok  
    Range("cl_Status").Value = "Import - Error"  
    'Set info: import is done  
    Range("cl_Info").Value = "Import is not done"  
    Range("cl_Date").Value = Date & " " & Time  
  
    If Range("I4").Value = "Import - Error" Then  
        ActiveCell.Cells(1, 16).Value = "Import stopped"  
        ActiveCell.Cells(1, 15).Value = Date  
    Else  
        ActiveCell.Cells(1, 16).Value = "Imported"  
        ActiveCell.Cells(1, 15).Value = Date  
    End If  
End Sub
```

2.7.5 DieserArbeitsmappe

Name:	DieserArbeitsmappe
File:	TestTemplate.xlsm
Copyright @2016,	SIEMENS Building Technologies
OS:	Win64
Lang:	Microsoft Visual Basic 6.5
Project:	TsNet
Author:	Fatma Yilmaz
Version	V.1
Date	24-03-2016
Description:	Beim Öffnen von dieser Arbeitsmappe wird die Funktion ModifyWbopen aufgerufen, um die Tabelle Spec-Devices im ReadOnly Modus zu öffnen
History:	24-03-2016 Fatma Yilmaz V.1 Document Creation for IPA

```

Private Sub workbook_open()
    'Sheet Spec Devices in ReadOnly Mode
    Worksheets("Spec-Devices").Activate()
    'set workbook unprotect
    Sheets("Spec-Devices").Unprotect()
    'Set the status on "working"
    [cl_Status].Value = "Working"
    'Set info on "In Progress"
    [cl_Info].Value = "In Progress"
    Call ModifyWBopen()
End Sub

```