

BACeye. Professional device control 2.1.0.14 - MBS GmbH - Standard Project

File Edit Data Transfer Window View Diagnose BACnet Test Help

**BACnet Test**

Netzwerk  
Netzen  
Eigenschaften  
BACnet Test

**Testplan-Baum**

- Prüfung Existenz von Properties
  - Objekte Analog Input
    - Property PresentValue
    - Property Unit
    - Property Notification Class
    - Property LowLimit
    - Property High Limit
  - Objekte Binary Input
    - Property Present Value
    - Assert BACnet-Property ElapsedActiveTime exists
    - Assert BACnet-Property ChangeOfStateCount exists
    - Assert BACnet-Property StateText exists
    - Assert BACnet-Property FeedBackValue exists
- Prüfung Existenz von Properties, 03.09.2018 09:56:59
- Prüfung korrekter Objekt Namen
  - Device 2010
    - Assert BACnet-Property charstring value equals
    - Assert BACnet-Property charstring value equals
    - Assert BACnet-Property charstring value equals
    - Assert BACnet-Property charstring value equals
- Prüfung Schreibbarkeiten von Properties
  - Schreibbarkeit MO Properties

Löschen nur Testpläne Bearbeiten Neu

**Test Run**

Testplan Prüfung Existenz von Properties  
Datum Montag, 3. September 2018 09:56:59  
Anmerkung  
Dauer 0:00:02.592

**Testliste**

Suche Filter 10 / 10

Status	Name	Online/Offline
✓	Property PresentValue	Both
✓	Property Unit	Both
✓	Property Notification Class	Both
✓	Property LowLimit	Both
✓	Property High Limit	Both
✓	Property Present Value	Both
✓	Assert BACnet-Property ElapsedActiveTime ex	Both
✓	Assert BACnet-Property ChangeOfStateCount	Both
✗	Assert BACnet-Property StateText exists	Both
✗	Assert BACnet-Property FeedBackValue exists	Both

**Test-Eigenschaften**

Name Property PresentValue  
 Beende Test Case...

Anmerkung

Parameter  
Property 85 PresentValue  
Objectype 0 Analog In  
Obj.Inst.No.  
Obj.Name  
Obj.Descr.  
Dev.Inst.No.  
von bis

**Testausgaben**

Suche Schnellfilter Selektion in Testliste 4 / 35

Status	Test	Dev. Inst. Nr.	Obj. Typ	Obj. Inst. Nr.	Property	Beschreibung
✓	Property PresentValue	2000	AI	1	PresentValue	Property PresentValue ist vorhanden in (Analogir
✓	Property PresentValue	2010	AI	1	PresentValue	Property PresentValue ist vorhanden in (Analogir
✓	Property PresentValue	2020	AI	1	PresentValue	Property PresentValue ist vorhanden in (Analogir
✓	Property PresentValue	1	AI	133	PresentValue	Property PresentValue ist vorhanden in (Analogir

Bereit

# BACeye BACnet-Test Plug-In

# Inhalt

<b>1</b>	<b>Vorwort</b>	<b>3</b>
1.1	Copyright	3
1.2	Kontaktinfo	3
1.3	Aufgabe und Funktion des BACnet-Test Plug-Ins	3
<b>2</b>	<b>Installieren des Test-Plug-Ins</b>	<b>5</b>
<b>3</b>	<b>Neue Test Plug-In-Elemente in der BACeye-Oberfläche</b>	<b>6</b>
3.1	Neue Register, Menüs und Kontextmenüs	6
3.2	Komponenten	8
3.2.1	Testplan	8
3.2.2	Test Case	9
3.2.3	Assertion (Test)	9
<b>4</b>	<b>Exemplarische Einführung</b>	<b>11</b>
4.1	Workflow	11
4.2	Eingebundene BACnet-Geräte (Online Testbeispiel)	11
4.3	Verwenden des Testplan-Editors	11
4.3.1	Neuen Testplan anlegen	11
4.3.2	Neuen Test Case anlegen	12
4.3.3	Neuen Test anlegen und konfigurieren	13
4.4	Test Run ausführen	16
4.4.1	Test Run starten	17
4.4.2	Anzeige von Testergebnissen	18
4.5	Test Run offline ausführen	20
4.6	Beispiele für weitere komplexere Testpläne	20
<b>5</b>	<b>Datenhaltung und -transfer</b>	<b>24</b>
5.1	Speichern und Löschen von Testplänen und Test Runs	24
5.2	Speichern und Laden des BACeye-Projekts	24
5.3	Export und Import von Testplänen, Test Runs und Test Cases	24

# 1 Vorwort

Vielen Dank für den Einsatz des BACeye BACnet-Test Plug-In zur Prüfung von BACnet-Netzwerken.

Diese Kurzdokumentation gibt eine exemplarische Einführung in das BACnet-Test Plug-In für die Software BACeye.

## 1.1 Copyright

©2018 MBS GmbH | Römerstraße 15 | D-47809 Krefeld

Telefon: +49 / 21 51 / 72 94 – 0

Telefax: +49 / 21 51 / 72 94 – 50

E-Mail: [info@mbs-solutions.de](mailto:info@mbs-solutions.de)

Internet: <http://www.mbs-solutions.de/>

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form (Druck, Fotokopie oder in einem anderen Verfahren) ohne schriftliche Genehmigung der MBS GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

## 1.2 Kontaktinfo

MBS GmbH  
Römerstraße 15  
47809 Krefeld

Telefon: +49 21 51 72 94 – 0

[info@mbs-solutions.de](mailto:info@mbs-solutions.de)

<http://www.baceye.com>

### **Support**

Telefon: +49 21 51 72 94 – 0

[support-baceye@mbs-solutions.de](mailto:support-baceye@mbs-solutions.de)

## 1.3 Aufgabe und Funktion des BACnet-Test Plug-Ins

### **BACeye**

BACeye ermöglicht einen einfachen Zugriff auf BACnet-Geräte und auf deren Objekte und Eigenschaften. Typische Einsatzfälle von BACeye sind die Inbetriebnahme von BACnet-Installationen, die Analyse bei beobachteten Problemen sowie die Überwachung und Bedienung von BACnet-Geräten.

### **BACeye BACnet-Test Plug-In**

*Aufgabe und Funktion:*

Mit dem BACnet-Test Plug-In soll die Fehlerquelle durch den *Medienbruch* an der Trennstelle zwischen Planung und Umsetzung beseitigt werden. Das Plug-In bietet die Möglichkeit einer automatisierten Prüfung auf Einhaltung von ausgeschriebenen Anforderungen (Pflichtenheft) bezüglich eines BACnet-Netzwerkes. Komplette Anlagen oder auch nur einzelne Geräte oder Objekte können auf einen gewünschten Status hin überprüft werden.

### *Hintergrund:*

Im Prozess der Erstellung eines BACnet-Netzwerkes kommen diverse *Medienübergänge* vor, die oft zu Informationsverlust und Ungenauigkeiten führen. Medienübergänge gibt es zwischen Planung/Definition des Lastenheftes, Ausschreibung, Auftragsvergabe und Umsetzung. So plant ein Planer (Auftraggeber, Bauherr, Endkunde) z. B. ein BACnet-Netzwerk mit Anlagen zur Gebäudeautomation und übergibt die Unterlagen an den Projektierer. Der Projektierer übernimmt die Vorgaben und projiziert das BACnet-Netzwerk (z. B. mit einem Planungssystem). An dieser Trennstelle zwischen Planung und Umsetzung liegt ein Medienbruch mit gewissen Fehlerrisiken vor.

Die Endabnahme durch den Planer enthält in der Regel die Prüfung auf Funktion, Vollständigkeit und Einhalten der vorgegebenen Regeln. Nicht selten stellt der Planer fest, dass eine Liegenschaft nach Übergabe durch den Errichter die angeforderte Funktion nicht oder nicht vollständig erfüllt.

Statt einer mühsamen manuellen Prüfung kann der auftraggebende Planer - aber auch der auftragnehmende Projektierer vor der Abnahme - automatische, zuverlässige Tests für das gesamte Netzwerk oder für Teile davon durchführen (Workflow siehe Kap. 4.1).

Die Tests werden in *Testplänen* und *Test Cases* konfiguriert und gesichert. Als Ergebnis wird im Fehlerfall eine detaillierte Fehlerliste generiert. Fehler werden danach korrigiert und das BACnet-Netzwerk oder Teile davon erneut geprüft. Durch mehrere Test-Korrektur-Iterationen wird somit das BACnet-Netzwerk bis zum optimalen Zustand gebracht und kann danach fehlerfrei in Betrieb genommen werden.

Folgende einzelnen BACnet-Elemente können auf verschiedene Eigenschaften hin überprüft werden:

- Netzwerk
- Device
- Object
- Property.

Ob das Prüfergebnis im jeweiligen Fall als positiv oder negativ zu werten ist, hängt vom entsprechenden Test Case ab. Zum einen kann das Vorhandensein eines bestimmten Objekts als positiv gewertet werden, in einem anderen Fall als negativ.

## 2 Installieren des Test-Plug-Ins

Installieren Sie wie folgt das BACnet-Test Plug-In:

- Starten Sie die BACeye-Software.
- Klicken Sie im Menü **Hilfe** auf den Menüpunkt **Plug-Ins verwalten** (Abbildung 1).  
Der Dialog **BACeye – Plug-Ins verwalten** öffnet sich (Abbildung 2).

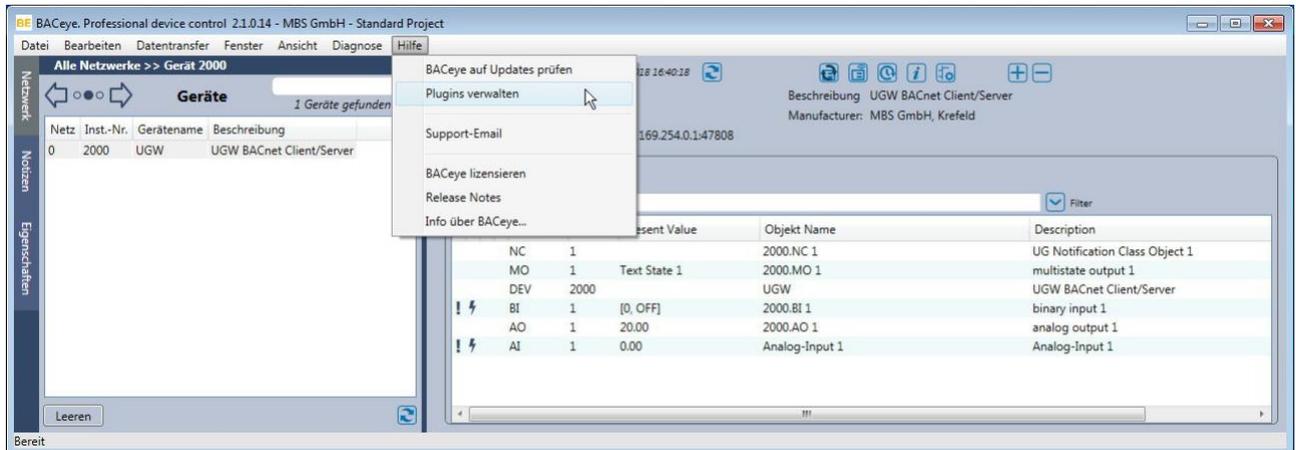


Abbildung 1

- Klicken Sie im Dialog **BACeye – Plug-Ins verwalten** auf den Button **[Installieren]**.  
Der Auswahl-Dialog öffnet sich.



Abbildung 2

- Wählen Sie im Auswahl-Dialog die für das BACnet-Test Plug-In gelieferte zip-Installationsdatei und bestätigen Sie mit **[Öffnen]**.  
Das Plug-In wird danach installiert und im Dialog **BACeye – Plug-Ins verwalten** angezeigt (Abbildung 3).
- Schließen Sie den Dialog mit **[Schließen]**.  
Ein Hinweisfenster fordert Sie danach zum Neustart der BACeye-Software auf.

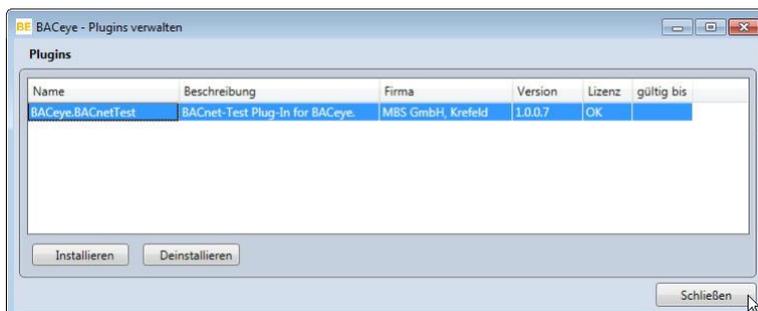


Abbildung 3

- Starten Sie BACeye neu.  
Danach können Sie die Funktionen des BACeye-Test Plug-Ins verwenden.

## 3 Neue Test Plug-In-Elemente in der BACeye-Oberfläche

### 3.1 Neue Register, Menüs und Kontextmenüs

In der grafischen Oberfläche der BACeye-Software (GUI) gibt es durch das Test Plug-In folgende neue Elemente (Abbildung 4 bis Abbildung 6):

- Registerkarte **BACnet Test** in der Navigationsleiste (a)
- Menü **BACnet Test** in der BACeye-Menüleiste (b)
- Neue Funktionen in den Kontextmenüs von strukturellen Elementen

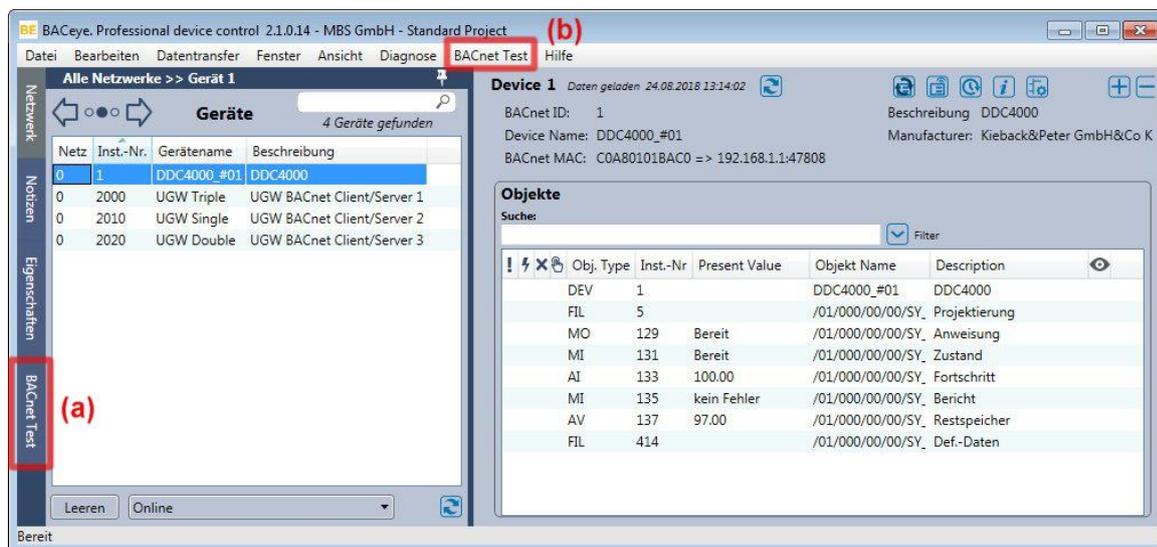


Abbildung 4

Durch Klick auf den Reiter **BACnet Test** in der Navigationsleiste wird links die Navigationsansicht (1) des Plug-Ins mit dem Testplan-Baum und rechts dessen Detailansicht (2) angezeigt (Abbildung 5). Im Testplan-Baum können ein oder mehrere Testpläne angelegt und angezeigt werden. Mit den Steuerelementen am unteren Rand können bestimmte Funktionen zur Testplanverwaltung ausgeführt werden.

Das neue Menü **BACnet Test** enthält die Menüpunkte:

- Testplan-Export... (siehe auch Kap. 5.3)
- Testplan-Import... (siehe auch Kap. 5.3)
- Auswahl von Tests und Test Cases (siehe Kap. 4.3.3)

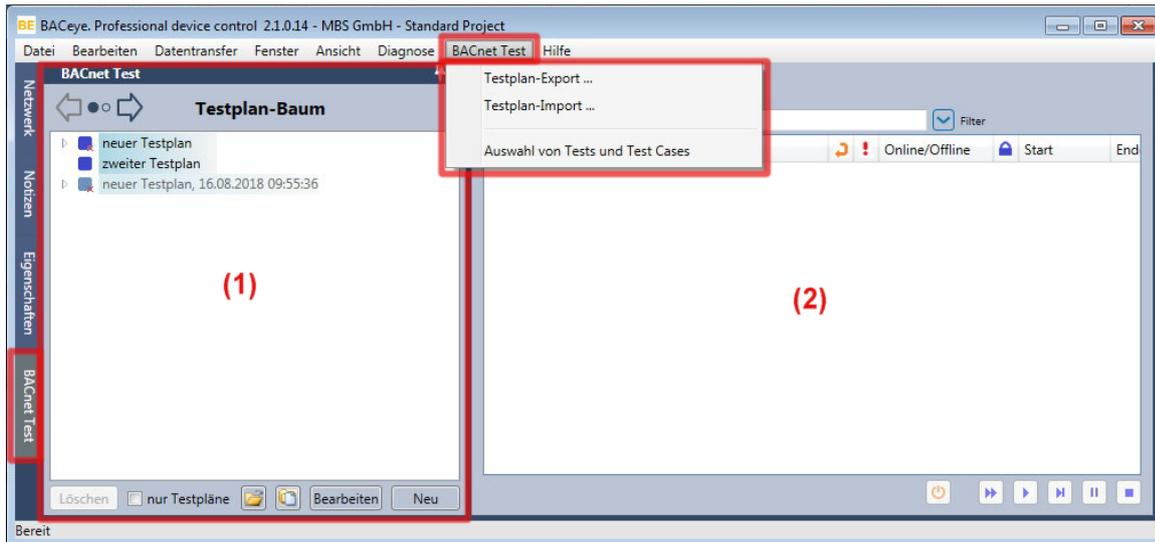


Abbildung 5

Bei der Auswahl eines Elementes (Testplan, Test Case oder Test/Assertion) im Testplan-Baum öffnet sich unterhalb der Navigationsansicht ein Konfigurationsbereich für das ausgewählte Element (3) (Abbildung 6). In der Detailansicht wird nach der Auswahl eines Testplans oder Test Cases die Liste aller im Testplan / Test Case enthaltenen Tests angezeigt (Testliste).

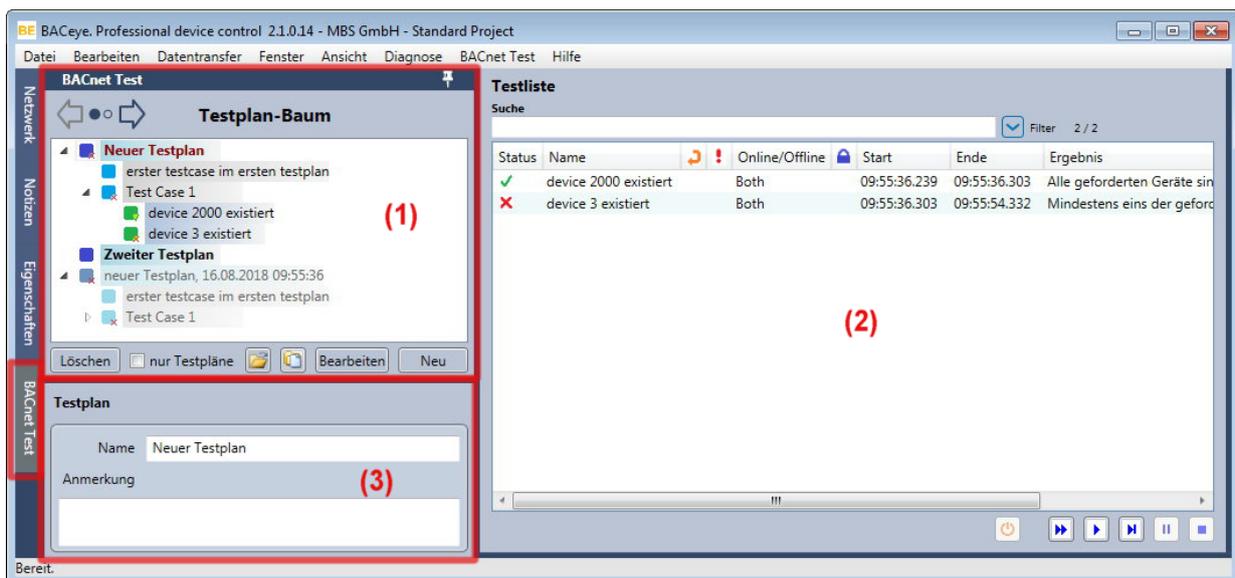


Abbildung 6

Auch über Kontextmenüs von strukturellen Elementen können Sie auf neue Funktionen des Test Plug-Ins zugreifen. Abbildung 7 zeigt das Kontextmenü eines Gerätes in der Navigationsansicht **Netzwerk**.

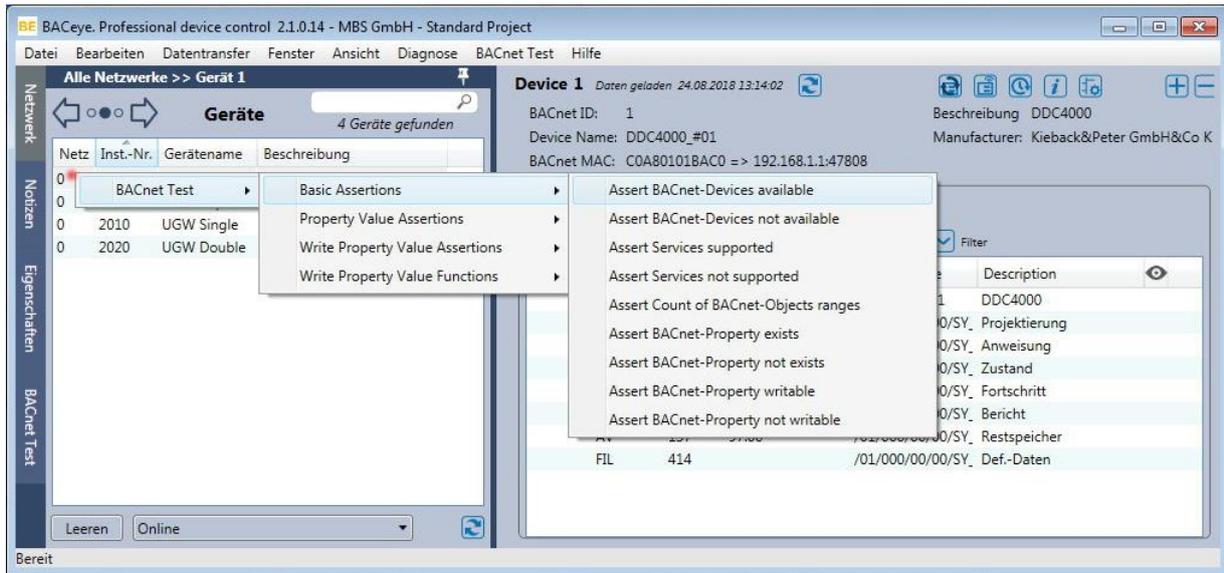


Abbildung 7

## 3.2 Komponenten

Der Testplan-Baum hat drei Verzweigungsebenen und enthält folgende hierarchisch gegliederte Elemente (Abbildung 8):

1. Testpläne
2. Instanzen von Test Cases
3. Instanzen von Assertions (Tests)

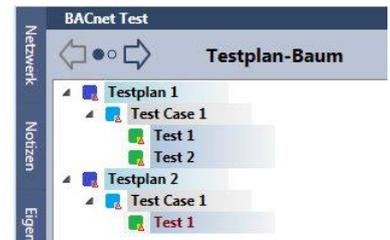


Abbildung 8

### 3.2.1 Testplan

Ein Testplan ist eine Sammlung von Test Cases. Konfigurierte Testpläne können als Test Runs ausgeführt werden. Jeder Testplan besteht aus mindestens einem Test Case. Jeder Test Case verfügt über einen Titel und eine Beschreibung und kann in einer Datei gespeichert und daraus wieder geladen werden.

Kontextmenü eines Testplan-Elementes im Testplan-Baum:

Funktion	Beschreibung
Neu	Legt einen neuen Test Case an
Test Case importieren	Importiert einen Test Case aus einer wählbaren Datei
Leeren	Löscht alle Test Cases im Testplan
Löschen	Löscht den Testplan
Exportieren	Exportiert den Testplan in eine wählbare Datei
Alle reduzieren	Klappt die unter diesem Testplan sichtbare Struktur ein

### Funktionsleiste für den Testplan-Baum

Unter dem Testplan-Baum mit den Testplänen befindet sich eine Zeile mit Steuerelementen für folgende Funktionen:

Funktion	Typ	Beschreibung
Löschen	Button	Löscht das ausgewählte Element
Nur Testpläne	Checkbox	Ausblenden von Test Runs im Testplan-Baum
Exportieren	Button/Symbol 	Exportiert Testpläne oder Test Runs in eine wählbare Datei
Importieren	Button/Symbol 	Importiert Testpläne oder Test Runs aus einer wählbaren Datei

Bearbeiten	Button	Ruft den Dialog <b>Auswahl von Tests und Test Cases</b> auf
Neu	Button	Legt einen neuen Testplan an

### Testplan-Editor

Der im Plug-In integrierte Testplan-Editor bietet die Möglichkeit, Testpläne zu erstellen. Dazu werden ihm die vordefinierten Assertions und Test Cases zur Auswahl gestellt. Sie können aber auch eigene Test Cases erstellen und speichern. Testpläne und Test Cases sind auch in gespeicherter Form (XML) editierbar.

### Test Run

Ein Test Run ist die vollständige oder teilweise Ausführung eines Testplans. Für jede ausgeführte Assertion im Test Run werden das Ergebnis, die Uhrzeit und der von der Assertion zurückgelieferte Log-Eintrag dokumentiert. Ein Test Run kann nur einmal ausgeführt werden. Jede weitere Ausführung führt zu einem neuen Test Run. Die Ergebnisse eines Test Runs können gespeichert, geladen und visualisiert werden.

### 3.2.2 Test Case

Ein Test Case ist eine Gruppierung von Assertions (Tests). Jeder Test Case verfügt über einen Titel und eine Beschreibung. Sie können aber auch eigene Test Cases definieren. Test Cases können als Datei gespeichert werden.

Kontextmenü eines Test Case-Elementes im Testplan-Baum:

Funktion	Beschreibung
Leeren	Löscht alle Tests im Test Case
Löschen	Löscht den Test Case
Exportieren	Exportiert den Test Case in eine wählbare Datei
Alle reduzieren	Klappt die unter diesem Test Case sichtbare Struktur ein

### 3.2.3 Assertion (Test)

Eine Assertion ist eine vorgegebene Prüfroutine (Testvorlage), die in einer Assertion-Library mit BACeye mitgeliefert wird. Die Assertion wird zu einem Test, sobald sie in einen Test Case eingefügt wurde (siehe Kapitel 4.3.3). Eine Assertion besitzt einen Satz ggf. optionaler Parameter, die die Ausführung steuern. Als ein in sich abgeschlossener Testschritt verfügt eine Assertion stets über einen booleschen Rückgabewert (True, False) und eine textuelle Mitteilung zu Testdetails. Nach der Auswahl eines Tests im Testplan-Baum werden im Konfigurationsbereich die Eigenschaften des Tests angezeigt. Hier können der Name des Tests oder seine Parameter geändert werden.

Kontextmenü eines Test-Elementes im Testplan-Baum:

Funktion	Beschreibung
Löschen	Löscht den Test

### Assertions-Library

Eine Assertions-Library stellt einen Satz von Assertions zur Verfügung. Die Summe der mit dem Plug-In installierten Assertion-Libraries definiert den Umfang der verfügbaren Assertions. Eine Assertions-Library ist eine .NET-Library die eine definierte Schnittstelle per MEF (Managed Extensibility Framework) exportiert.

### Drag-and-Drop im Testplan-Baum

Die Komponenten des Testplan-Baums können innerhalb des Baums per Drag&Drop verschoben oder kopiert werden. Beim Kopieren verwenden Sie gleichzeitig die Ctrl-Taste.

Mit dem Verschieben von Tests oder Test Cases im Testplan wird deren Ausführungsreihenfolge festgelegt. Mit dem Verschieben oder der Ablage von Testplänen im Testplan-Baum wird nur deren Anordnung im Baum geändert bzw. festgelegt. Ein Testplan wird hinter allen anderen Testplänen abgelegt, wenn sein Ziel ein freier Bereich im Testplan-Baum, also kein Testplan oder eine seiner Komponenten ist.

## 4 Exemplarische Einführung

### 4.1 Workflow

Das Plug-In wird mit einem Satz Test Cases und Assertions ausgeliefert. Aus den mitgelieferten Test Cases und Assertions können Sie unter Verwendung des Testplan-Editors oder durch Editieren der XML-Datei eigene Test Cases und Testpläne erstellen.

Für den BACnet-Test bietet sich folgender Workflow an:

1. Anlegen eines neuen Testplanes als Container für Test Cases
2. Definieren eines oder mehrerer Test Cases im Testplan als Container für Tests
3. Definieren mindestens einer Prüfung / Assertion im Test Case
4. Auswählen eines Testplanes zur Ausführung
5. Starten eines Test Runs
6. Auswertung der Testergebnisse

Dieser BACnet Test-Workflow ist sowohl für den Online- als auch für den Offline-Modus von BACeye möglich.

### 4.2 Eingebundene BACnet-Geräte (Online Testbeispiel)

Das BACnet- Netzwerk enthält im nachfolgenden Beispiel folgende BACnet-Geräte (Abbildung 9):

- eine Automationsstation DDC4000 (BACnet Device-ID: 1)
- ein MBS Universal Gateway UGW Triple (BACnet Device-ID: 2000)
- ein MBS Universal Gateway UGW Single (BACnet Device-ID: 2010)
- ein MBS Universal Gateway UGW Double (BACnet Device-ID: 2020).

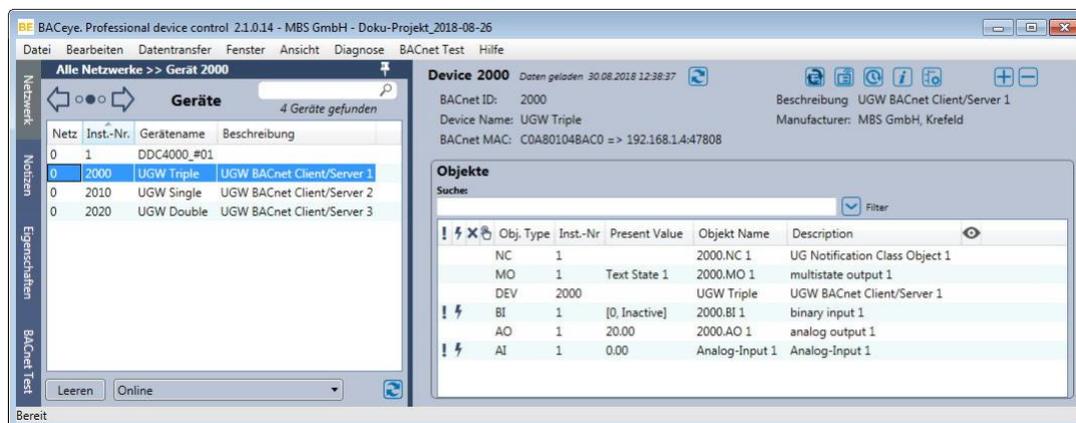


Abbildung 9

### 4.3 Verwenden des Testplan-Editors

#### 4.3.1 Neuen Testplan anlegen

- Aktivieren Sie die Registerkarte **BACnet Test** in der Navigationsleiste.
- Legen Sie einen neuen, leeren Testplan an, indem Sie in der Funktionsleiste des Testplan-Baums auf **[Neu]** klicken. Dabei wird ein nummerierter Testplan erzeugt. Unterhalb des Testplan-Baums öffnet sich der Konfigurationsbereich mit den Eigenschaften des Testplans (Abbildung 10).
- Akzeptieren Sie den vom Programm vorgeschlagenen Namen oder geben Sie dem Testplan im Konfigurationsbereich einen neuen eindeutigen Namen. Im Textfeld *Anmerkung* können Sie optional einen erläuternden Text zu diesem Testplan eingeben.

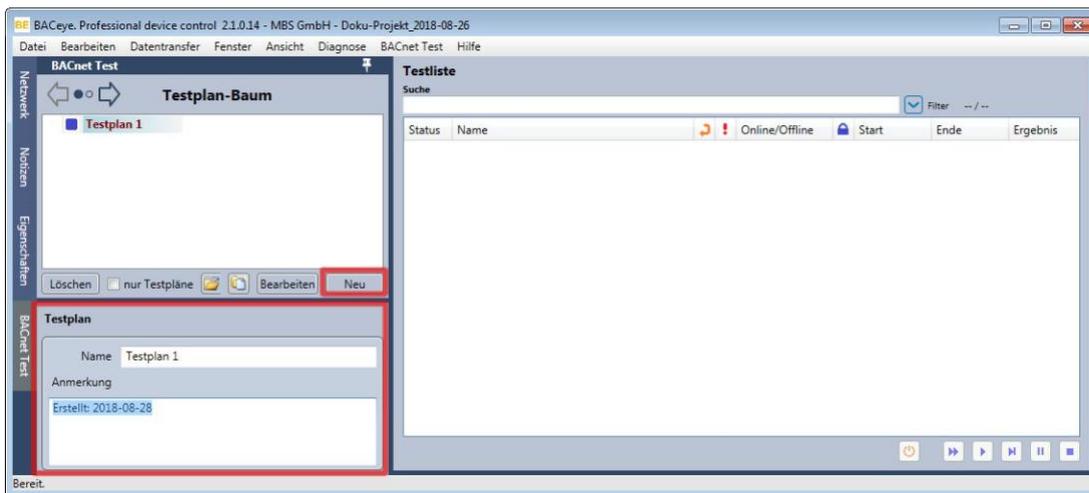


Abbildung 10

### Hinweis:

Über den Button **[Löschen]** oder über seinen Kontextmenüpunkt *Löschen* kann der ausgewählte Testplan wieder gelöscht werden. Bestätigen Sie danach den Rückfragedialog mit **[OK]**. Dies gilt auch für Test Cases.

### 4.3.2 Neuen Test Case anlegen

- Als zweite Strukturebene legen Sie einen neuen Test Case an, indem Sie im Kontextmenü des Testplanes auf **Neu** klicken (Abbildung 11). Weiterhin kann über den Testplan-Kontextmenüpunkt **Testcase importieren** ein Test Case aus einer wählbaren Datei importiert werden.



Abbildung 11

- Akzeptieren Sie den vom Programm vorgeschlagenen Namen oder geben Sie dem Test Case im Konfigurationsbereich einen neuen eindeutigen Namen (Abbildung 12). Im Textfeld *Anmerkung* können Sie optional einen erläuternden Text zu diesem Test Case eingeben.

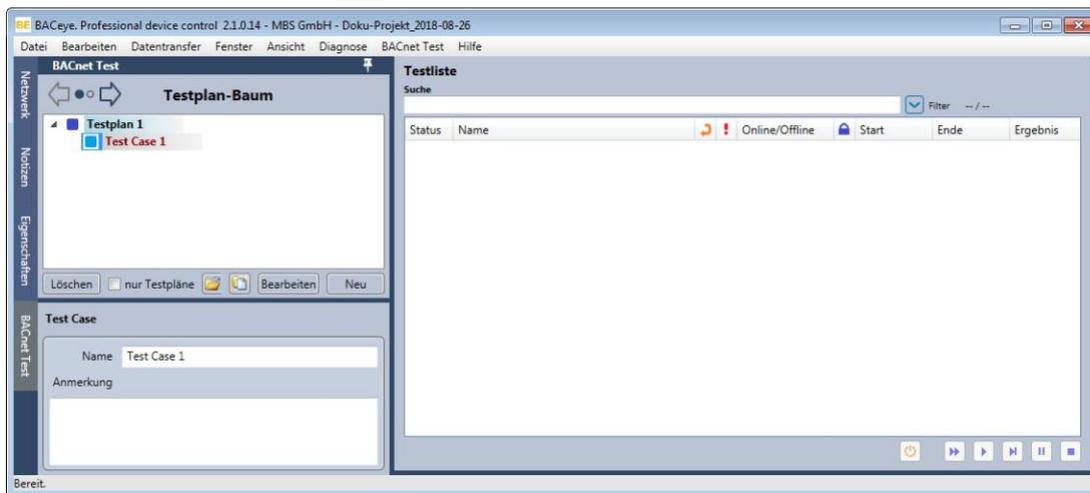


Abbildung 12

### 4.3.3 Neuen Test anlegen und konfigurieren

- Als dritte Strukturebene legen Sie einen neuen Test an, indem Sie in der Funktionsleiste des Testplan-Baums auf **[Bearbeiten]** klicken. Alternativ nutzen Sie das Menü **BACnet Test > Auswahl Tests und Test Cases**. Dabei öffnet sich das Fenster **Auswahl Tests und Test Cases** (Abbildung 13).

Das Fenster **Auswahl Tests und Test Cases** enthält eine vordefinierte Auswahl von Test-Vorlagen (Assertions). Im linken Bereich zeigt ein aufklappbarer Strukturbaum verschiedene Assertion-Libraries und die darin enthaltenen, vordefinierten Assertions und Test Cases. Im rechten Bereich werden die Eigenschaften des im Strukturbaum ausgewählten Elements dargestellt.

- Klappen Sie die Assertion-Library im Strukturbaum des Fensters **Auswahl Tests und Test Cases** auf und markieren Sie die gewünschte Assertion.
- Ziehen Sie die markierte Assertion per Drag-and-Drop in den Testplan-Baum auf den Test Case, der den Test enthalten soll (Abbildung 13).

Dabei werden Instanzen bzw. Kopien aus den jeweiligen Vorlagen erzeugt und im Testplan eingetragen. Der Test wird unterhalb des Test Cases eingefügt und in der Testliste angezeigt (Abbildung 14).

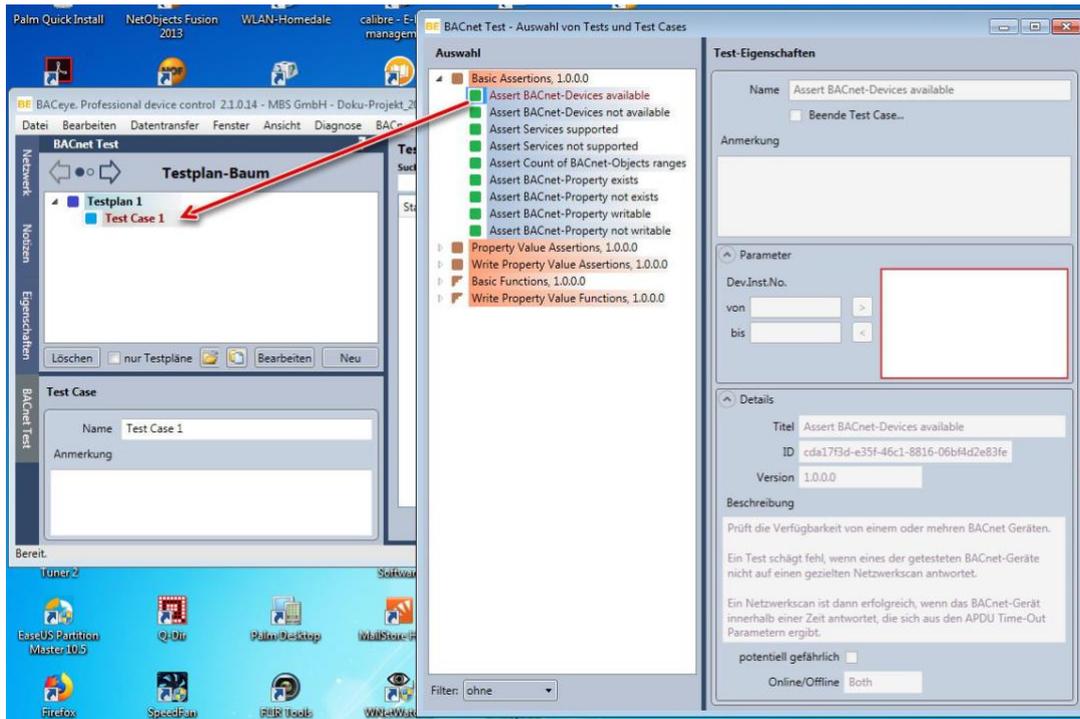


Abbildung 13

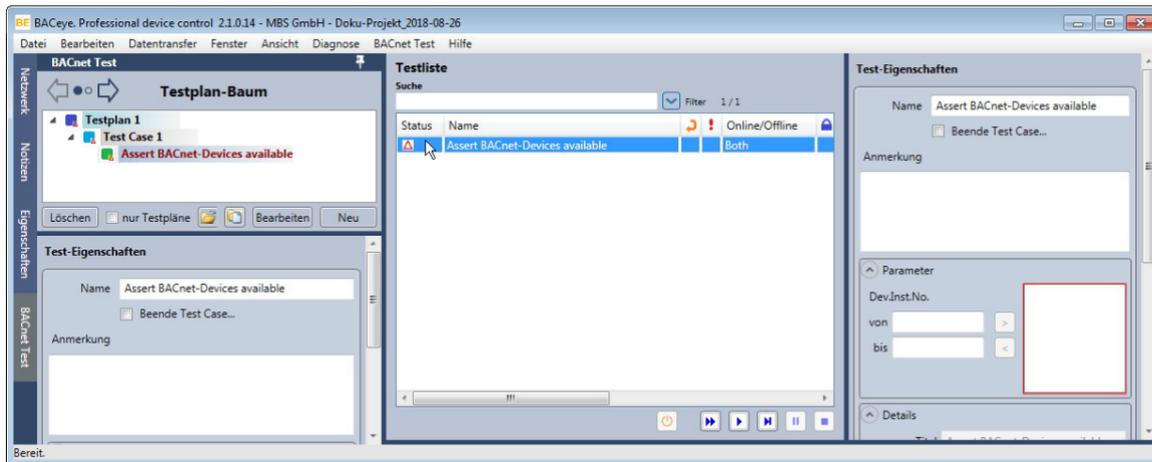


Abbildung 14

In unserem konkreten Fall (Abbildung 13 und Abbildung 14) wurde die Assertion *Assert BACnet-Devices available* aus der Assertion-Library *Basic Assertion 1.0.0.0* per Drag-and-Drop in das Test Case 1 eingefügt. Diese Assertion ermöglicht es, ein oder mehrere BACnet-Geräte auf ihre Verfügbarkeit im BACnet Netzwerk zu testen. Die Instanzen dieser Assertion stellen für nachfolgende Tests sicher, dass alle benötigten BACnet-Geräte im BACnet-Netzwerk vorhanden und in BACeye erfasst sind.

### Testliste

Die Testliste zeigt die Liste der Tests an, die dem im Testplan-Baum aktuell ausgewählten Element zugeordnet ist. Der in der ersten Spalte der Testliste ausgegebene Teststatus unterscheidet folgende Zustände:

Zustand	Symbol	Beschreibung
Parameter fehlen	⚠	Der Test ist nicht oder nur unzureichend parametrisiert.
Deaktiviert	⊘	Der Test ist deaktiviert.
Ohne Ergebnis	●	Test wurde nicht ausgeführt.
Test wird ausgeführt	⌚	Test wird ausgeführt.

Fehler	✘	Bei der Testausführung ist mindestens ein Fehler aufgetreten.
Erfolgreich	✔	Bei der Testausführung ist kein Fehler aufgetreten.

### Test-Eigenschaften

Wenn Sie den eingefügten Test in der Testliste markieren, werden rechts daneben im Konfigurationsbereich die Eigenschaften dieses Tests angezeigt. Wenn der Test nicht vordefiniert ist, werden Eingabefelder mit fehlenden Angaben rot umrandet.

Der Bereich Test-Eigenschaften enthält folgende Eingabefelder und Editierbereiche:

Eigenschaft	Typ	Beschreibung
Name	Eingabefeld	Name des Tests Wird mit dem Titel der Assertion vorbelegt.
Anmerkung	Textfeld	Anmerkung zum Text Wird mit der Beschreibung der Assertion vorbelegt.
Parameter	Aus- und einklappbarer Bereich	Hier wird das User-Control eingeblendet, welches die Assertion zum Editieren seiner testspezifischen Parameter zur Verfügung stellt.
Details	Aus- und einklappbarer Bereich	Hier können alle Eigenschaften der Assertion, aus der die Testinstanz gebildet wurde, eingesehen werden: Titel, Guid, Version, Beschreibung, potentiell gefährlich, Online/Offline

- ✚ Um den Namen des Tests zu modifizieren, tragen Sie den neuen Namen in das Eingabefeld *Name* ein.
- ✚ Tragen Sie optional eine beschreibende Anmerkung in das Textfeld *Anmerkung* ein.
- ✚ Geben Sie im Bereich *Parameter* in die Felder *Dev.Inst.No.* die gesuchte Geräte-ID ein (hier: 2000).
- ✚ Übertragen Sie den ID-Wert mit dem  Button in das rot umrandete Feld (Abbildung 15).

Danach ist dieser einfache Test parametrisiert und für den Test Run vorbereitet (Abbildung 16).

Um den Fall für im Netzwerk nicht gefundene BACnet-Geräte zu zeigen, wurde im Beispiel-Testprojekt ein weiterer Test aus der Assertion *Assert BACnet-Devices available* erzeugt und eingefügt. Dadurch soll geprüft werden, ob ein BACnet-Gerät mit der ID = 2030 im Netzwerk vorhanden ist (Abbildung 17).

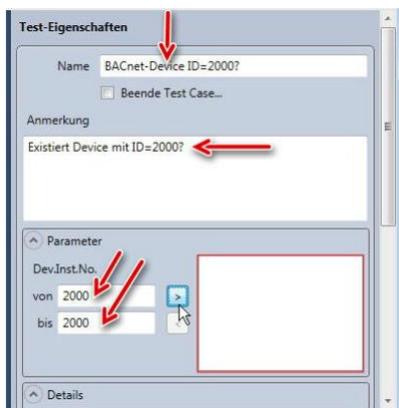


Abbildung 15

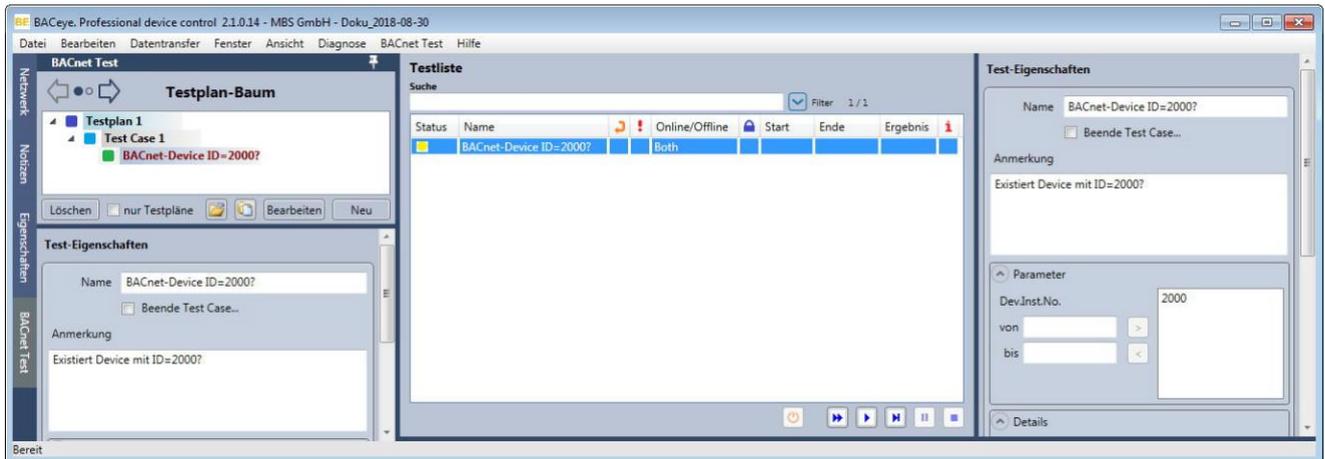


Abbildung 16

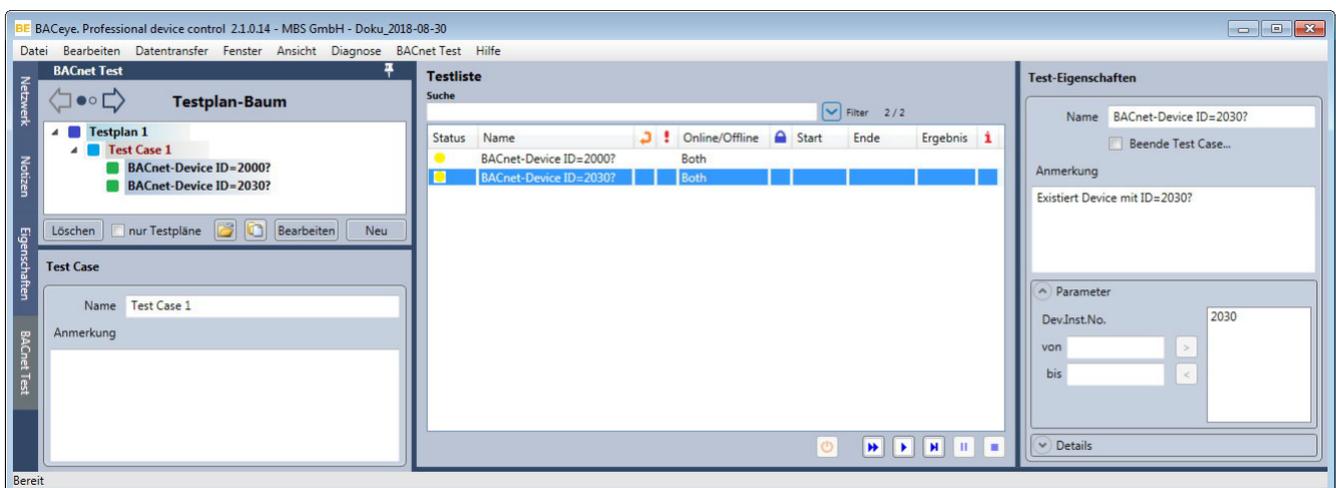


Abbildung 17

### Speichern von Änderungen im Testplan-Baum

Die Speicherung von Änderungen im Testplan-Baum erfolgt mit dem Speichern des aktuellen BACeye-Projekts (siehe Kapitel 5). Seit dem letzten Speichern oder dem Laden geänderte oder neu angelegte Komponenten werden im Baum durch eine fett hervorgehobene Beschriftung gekennzeichnet.

## 4.4 Test Run ausführen

In der Testliste werden alle Tests angezeigt, die der im Testplan-Baum ausgewählten Komponente zugeordnet sind. Die Reihenfolge der Tests in der Testliste entspricht derjenigen im Testplan-Baum und bestimmt die Reihenfolge bei der Ausführung der Tests im Test Run.

### Steuerleiste zur Steuerung des Test Runs

Unter der Testliste befindet sich eine Steuerleiste mit folgenden Buttons:

Funktion	Typ	Beschreibung
Start	Button 	Ein Test Run wird gestartet/fortgesetzt.
Einzelschritt	Button 	Im Test Run wird nur der erste/nächste Test ausgeführt und danach beim nächsten auszuführenden Test Halt gemacht. Ist kein weiterer Test auszuführen, wird der Test Run beendet.
Nächster	Button 	Überspringt den nächsten auszuführenden Test. Ist dieser der letzte im Test Run, wird der Test Run beendet. Wenn sich der Test Run im Halt befindet, wird der nächste Test übersprungen und beim darauffolgend auszuführenden Test Halt gemacht.
Pause	Button	Hält den Test Run nach Ausführung des aktuellen Tests beim nächsten auszuführenden Test an.

		Wenn kein weiterer Test auszuführen ist, wird der Test Run beendet.
Stop	Button 	Beendet nach Rückfrage und nach Ausführung des aktuellen Tests den Test Run. Wenn sich der Test Run im Halt beindet, wird dieser direkt beendet.
Not Halt	Button 	Bricht den aktuellen Test ohne Wertung ab und hält die Ausführung des Test Runs an.

#### 4.4.1 Test Run starten

- Um alle Tests des Testplans auszuführen, markieren Sie den entsprechenden Testplan im Testplan-Baum. Wenn gezielt nur ein bestimmter Test ausgeführt werden soll, markieren Sie diesen im Testplan-Baum, so dass nur dieser Test in der Testliste erscheint.

Um einen bestimmten Testumfang zu erreichen, können die Tests in der Liste auch über die oben sichtbaren Filter- und Suchfunktionen eingeschränkt werden.

- Starten Sie über den Start-Button  einen Test Run auf Basis der in der Testliste aufgeführten Tests.

#### Hinweis:

Sie können auch über den Einzelschritt-Button  die Tests im Test Run einzeln ausführen.

Ein Test Run wird nicht gestartet, wenn sich Tests in der Liste befinden, die noch nicht ausreichend parametrisiert sind (Status: ). Ein entsprechender Hinweisdialog weist darauf hin. Über das Kontextmenü können Sie Tests in der Testliste auch deaktivieren. Deaktivierte Tests werden im Test Run ignoriert.

Nach dem Start des Test Runs werden zunächst für alle Tests im Testplan ggf. noch von einem vorherigen Test Run gesetzte Ergebnisse entfernt.

Danach werden nacheinander die Tests in der Testliste abgearbeitet. Der Fortschritt des Vorgangs wird durch einen Fortschrittsbalken unter der Testliste angezeigt (Abbildung 18).

#### Hinweis:

Über den Pause-Button  können Sie den Test Run anhalten. Dabei wird der nächste auszuführende Test mit einem Statussymbol  gekennzeichnet. Ein pausierender Test Run kann er über den Start-Button  normal fortgesetzt oder über den Einzelschritt-Button  schrittweise fortgesetzt werden. Über den Stop-Button  wird ein Test Run vorzeitig beendet. Die bis dahin gewonnen Testresultate bleiben dabei erhalten.

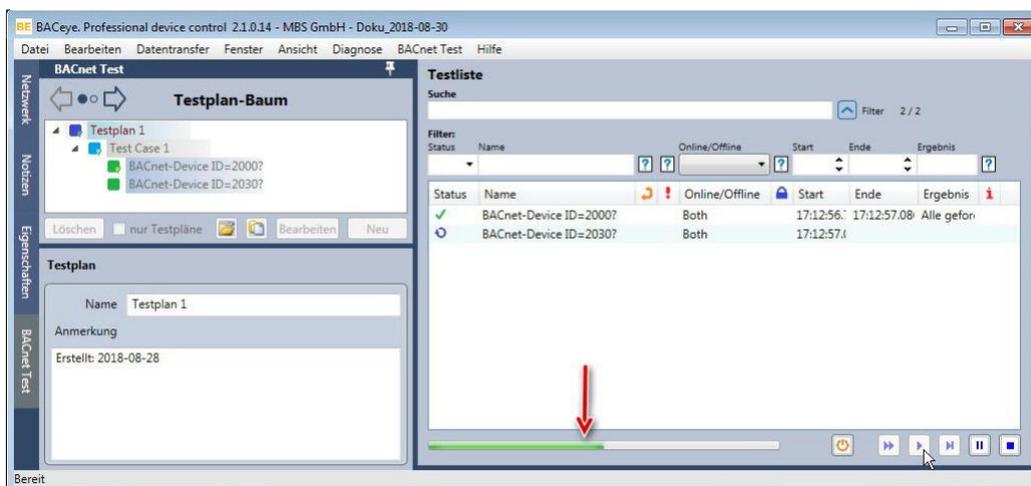


Abbildung 18

#### 4.4.2 Anzeige von Testergebnissen

Während der Ausführung eines Test Runs wird der Status der einzelnen Tests entsprechend dem Erfolg der Ausführung in der Testliste aktualisiert ( **✗** bei Fehler oder **✓** wenn erfolgreich). Start- und Endzeitpunkte sowie Ergebnistexte werden eingetragen (Abbildung 19).

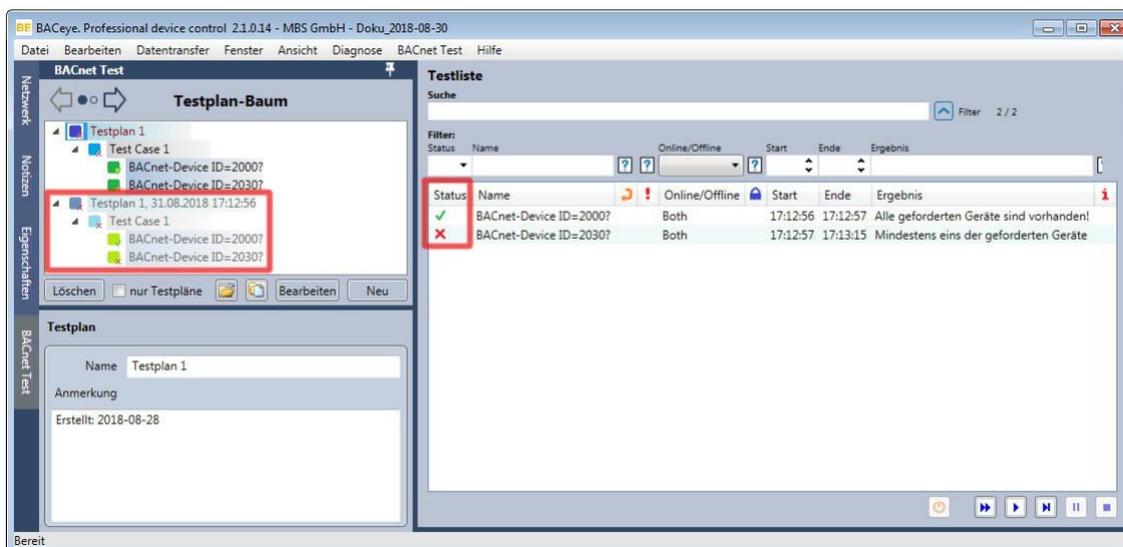


Abbildung 19

- Da das BACnet-Device mit der ID = 2000 im BACnet-Netzwerk gefunden wurde, verlief dieser Test erfolgreich **✓**.
- Da das BACnet-Device mit der ID = 2030 - bis zum Ablauf des Timeouts - im BACnet-Netzwerk nicht gefunden wurde, verlief dieser Test nicht erfolgreich **✗**.

Reaktion auf diesen Test:

- ✚ Wenn das BACnet-Device mit der ID = 2030 benötigt wird, ergänzen Sie das BACnet-Netzwerk noch mit diesem Gerät.

#### Weitere Möglichkeiten zur Auswertung des Test-Runs

Im Testplan-Baum werden die Ergebnisse der Tests ebenfalls mit den beiden Symbolen **✗** und **✓** an den Komponenten visualisiert.

Mit Beendigung eines Test Runs wird im Testplan-Baum unterhalb des Testplans eine Kopie des Testplans eingefügt (graue Textfarbe). Die Komponenten dieser Kopie (= Test Run) sind Abbildungen ihrer Originale im Testplan. Ein Test Run im Testplan-Baum ist eine schreibgeschützte Kopie des ausgeführten Testplans.

Das Kontextmenü eines Test Runs im Testplan-Baum enthält folgende Funktionen (Abbildung 20):

Funktion	Beschreibung
Löschen	Löschen des ausgewählten Test Runs (auch über den Button <b>[Löschen]</b> )
Exportieren	Export des ausgewählten Test Runs in eine wählbare Datei
Alle reduzieren	Klappt die unter diesem Test Run sichtbare Struktur ein

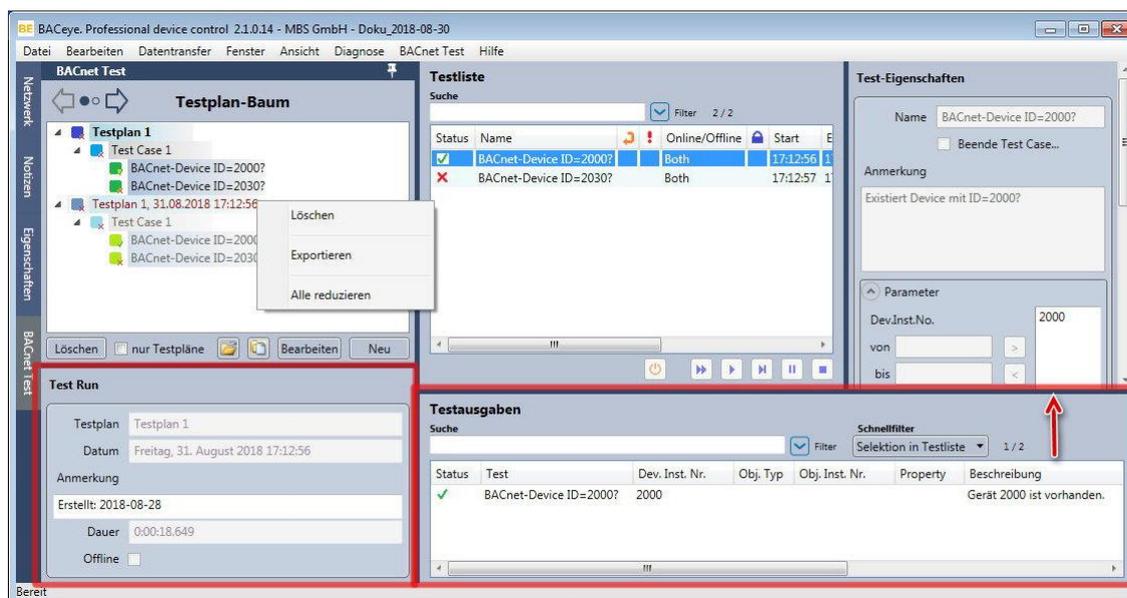


Abbildung 20

Der Konfigurationsbereich zum Test Run unter dem Testplan-Baum enthält folgende Eigenschaften:

Eigenschaft	Typ	Beschreibung
Testplan	Eingabefeld	Name des Testplans, auf dem der Test Run ausgeführt wurde (nicht editierbar)
Datum	Eingabefeld	Datum und Uhrzeit der Ausführung des Test Run (nicht editierbar)
Anmerkung	Eingabefeld	optionale Anmerkung zum Testplan auf dem der Test Run ausgeführt wurde (editierbar).
Dauer	Eingabefeld	Entweder Dauer oder Endzeitpunkt der Ausführung des Test Runs (nicht editierbar)
Offline	Check-Box	Kennzeichnet, ob der Test Run offline ausgeführt wurde (nicht editierbar)
Snapshot	Eingabefeld	Wenn der Test Run offline ausgeführt wurde, wird hier der verwendete Snapshot angegeben (nicht editierbar).

Bei der Auswahl einer Test Run-Komponente im Testplan-Baum wird in der Detailansicht analog wie beim Testplan die Liste der zugehörigen Tests angezeigt.

Im unteren Bereich der Detailansicht werden durch eine dunkelblaue Trennlinie getrennt im Bereich *Testausgaben* die getesteten Komponenten aufgeführt (Abbildung 20). Zu jeder Komponente wird das Testergebnis angegeben.

### Hinweis:

Falls der Bereich *Testausgaben* nicht sichtbar ist, ziehen Sie die dunkelblaue Trennlinie per Drag-and-Drop nach oben (siehe Abbildung 20).

## 4.5 Test Run offline ausführen

Sind im aktuellen BACeye Projekt Snapshots importiert (siehe BACeye-Handbuch) kann ein Test Run offline auf einem dieser Snapshots ausgeführt werden. Dazu ist der zu verwendende Snapshot in BACeye zu selektieren.

Wird ein Test Run offline ausgeführt, werden nur Tests ausgeführt die offline ausgeführt werden können. Alle anderen werden übersprungen z.B. Ein Test, der versucht über BACnet zu schreiben.

Mit dem Abschluss des Test Runs wird mit diesem abgelegt, dass er offline ausgeführt wurde und welcher Snapshot verwendet wurde.

## 4.6 Beispiele für weitere komplexere Testpläne

Nach der Prüfung auf benötigte BACnet-Geräte, können Sie anschließend mit dem BACnet-Test Plug-In weitere BACnet-Eigenschaften testen.

Testen Sie z. B. mit separaten Testplänen auf folgende Eigenschaften (Abbildung 21):

- Existenz von Properties
- Korrekte Objekt-Namen
- Beschreibbarkeiten von Properties

+ Erstellen Sie dafür, wie oben gezeigt, die einzelnen Testpläne und Test Cases. Für die Prüfung auf existierende Properties bietet sich die Aufteilung in zwei Test Cases mit Tests für Objekte mit analogem und digitalem Input an (siehe Abbildung 21).

+ Die Assertion-Vorlagen für die Tests entnehmen Sie aus folgenden Assertion Libraries:

- Basic Assertions (Tests *Assert BACnet-Property exists*, *Assert BACnet-Property writable*)
- Property Value Assertions (Test *Assert BACnet-Property charstring value equals*)

+ Stellen Sie die gewünschten Test-Eigenschaften in der rechten Detail-Ansicht ein, z. B. das genaue zu testende Property und den Objekttyp im Bereich *Parameter* (siehe Abbildung 21).

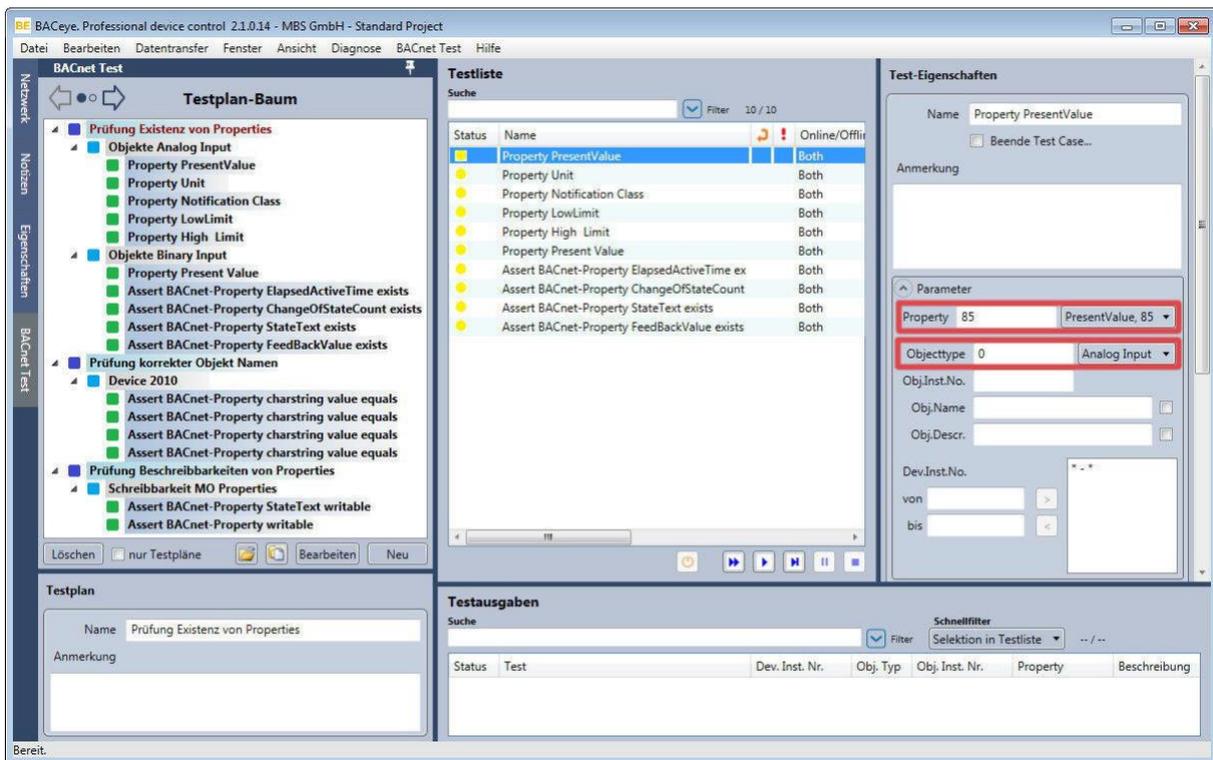


Abbildung 21

### Hinweis:

Mehr Informationen zu den bisher bereitgestellten Assertions und zu deren Eigenschaften finden Sie im MBS-Dokument „Spezifikation - BACnet-Test Plug-In Assertions.pdf“.

Starten Sie über den Start-Button  nacheinander die drei Test Runs.

Das Ergebnis z. B. für den ersten Test Run zeigt Abbildung 22. Im Bereich *Testausgaben* werden die Testergebnisse zu jeder Test-Komponente aufgelistet.

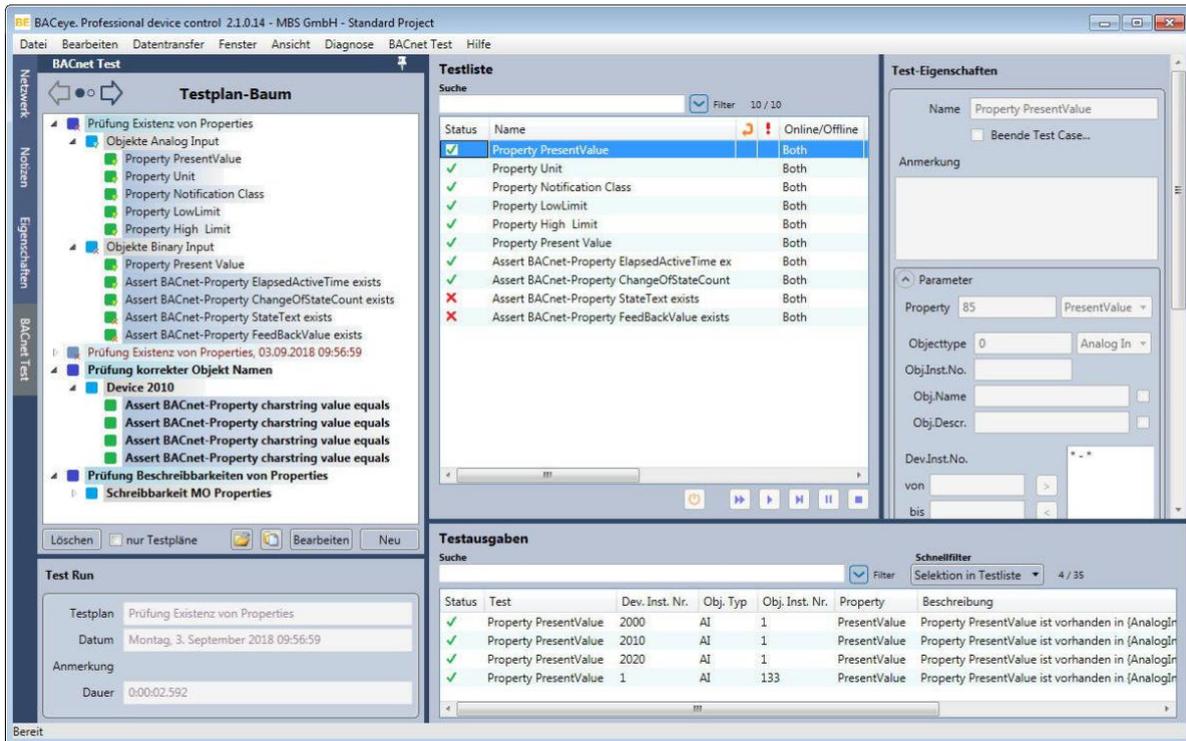


Abbildung 22

Nach dem Start von Tests für die Beschreibbarkeit von BACnet-Elementen (z. B. von Properties) wird durch einen Rückfragedialog nochmals abgefragt, ob diese Tests durchgeführt werden sollen (Abbildung 23).

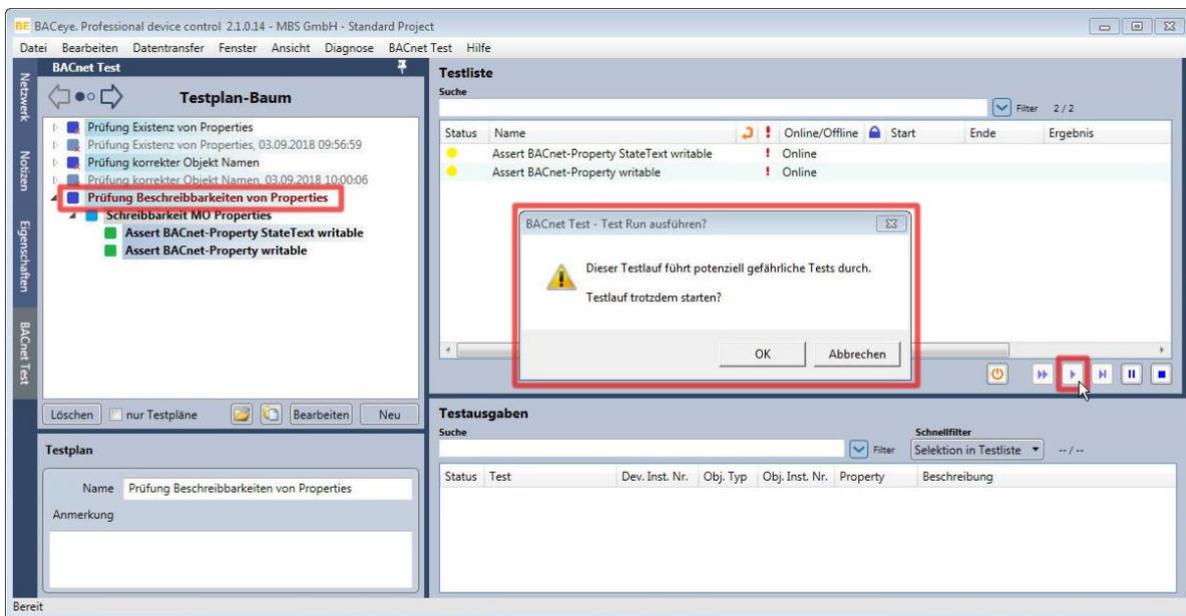


Abbildung 23

### Navigation zur Detailansicht von BACnet-Objekten

Um zur Detailansicht eines BACnet-Objekts zu navigieren gibt es zwei Möglichkeiten (Abbildung 24):

1. Markieren Sie den Eintrag des BACnet-Objekts in der Liste der Testausgaben und doppelklicken Sie auf diese Zeile.
2. Markieren Sie den Eintrag des BACnet-Objekts in der Liste der Testausgaben und klicken Sie auf den rechten Navigationspfeil in der Navigationsansicht .

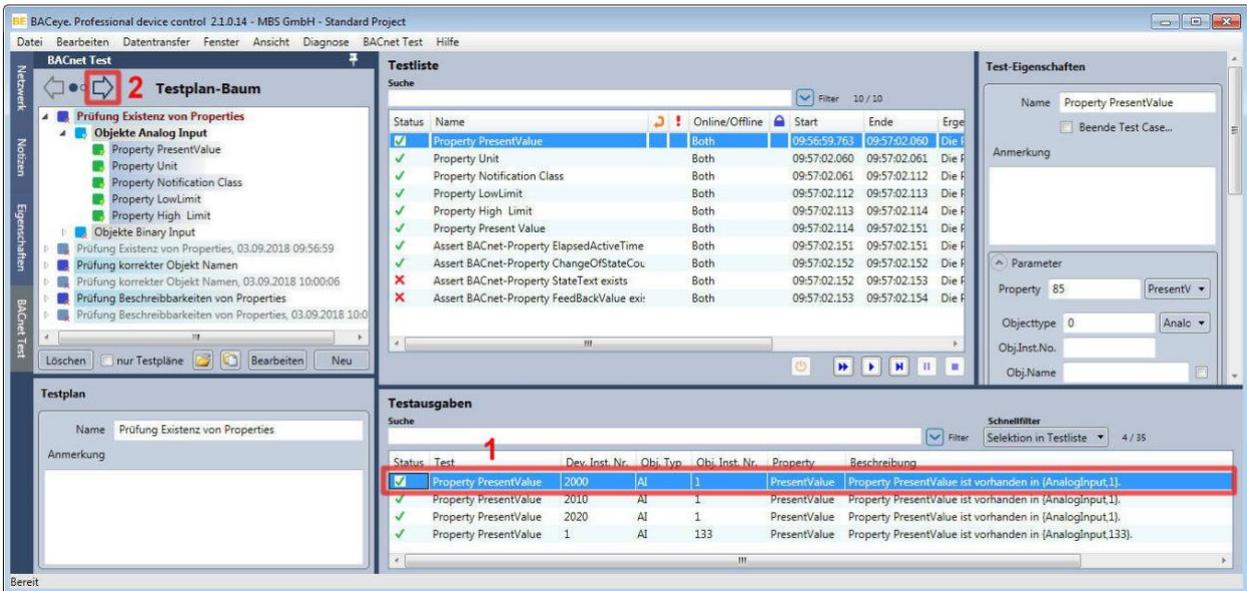


Abbildung 24

Im Navigationsfenster des BACnet-Test Plug-In wird danach die Testausgaben-Liste eingeblendet. Rechts im Detailfenster erscheint die Detailansicht des referenzierten BACnet-Objekts mit detaillierten Eigenschaften (Abbildung 25). Über den linken Navigationspfeil gelangen Sie zurück zum Testplan-Baum.

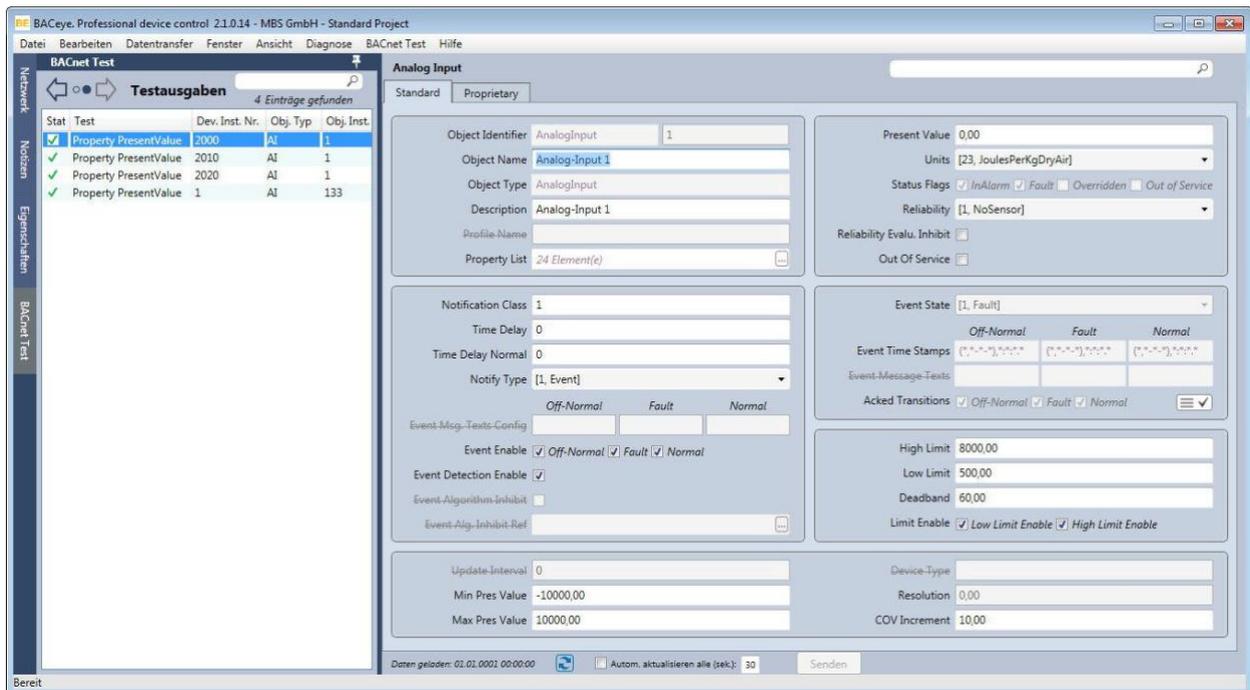


Abbildung 25

### Hinweis:

Wenn vorher nicht auf ein BACnet-Objekt verwiesen wurde oder bei Nichtverfügbarkeit der Daten, informiert rechts im Detailbereich ein Hinweistext, dass erst eine Testausgabe ausgewählt werden muss.

# 5 Datenhaltung und -transfer

## 5.1 Speichern und Löschen von Testplänen und Test Runs

Jeder Testplan oder Test Run wird in jeweils einer eigenen Datei gespeichert. Die Dateien werden in einem separaten Verzeichnis im Projektordner des aktuellen BACeye-Projekts gespeichert (XML-Format mit zugeordnetem XSD-Schema).

Jeder neue Testplan oder Test Run erhält eine ID (Guid) zur Identifikation. Diese ID wird zur Bildung des Dateinamens herangezogen. Mit dem Löschen eines Testplans oder Test Runs wird die zugehörige Datei gelöscht.

## 5.2 Speichern und Laden des BACeye-Projekts

Mit dem Speichern des aktuellen BACeye-Projekts legt das BACnet-Test Plug-In in der Projektsicherung eine Liste mit den IDs der vorhandenen Testpläne und Test Runs in der Reihenfolge ihres Auftretens im Testplan-Baum ab.

Beim Laden wird aus diesen Informationen der Testplan-Baum wieder rekonstruiert. Mit der Projektsicherung speichert das BACnet-Test Plug-In ebenfalls die Anordnung der Spalten der Testliste und der Liste der Testausgaben in der Detail- und Navigationsansicht.

## 5.3 Export und Import von Testplänen, Test Runs und Test Cases

Das BACnet-Test Plug-In ermöglicht den Export und Import von Testplänen oder Test Runs in eine Datei. Dabei werden Testpläne (\*.tpf) und Test Runs (\*.trf) als XML-Datei exportiert. Das verwendete Schema (XSD) ist das gleiche wie das beim Speichern verwendete.

### Export von Testplänen und Test Runs

Einen Export von Testplänen oder Test Runs können Sie wie folgt ausführen:

- a) Über das Kontextmenü eines Testplans oder Test Runs im Testplan-Baum
- b) Über den Button  unterhalb des Testplan-Baums
- c) Über das Menü **BACnet Test > Testplan-Export...**

Wenn Sie für den Export den Kontextmenüpunkt **Exportieren** im Testplan-Baum benutzen, enthält die Liste im Export-Dialog nur den markierten Eintrag für den Kontext (Testplan oder Test Run).

 Vergeben Sie im sich öffnenden Speichern-Dialog einen Dateinamen und klicken Sie auf **[Speichern]**.

In den Fällen b) und c) öffnet sich ein Dialog, der eine Liste mit den wählbaren Komponenten enthält (Abbildung 26). Zu exportierende Komponenten können per Checkbox an- bzw. abgewählt werden. Über die Checkbox *alle wählen* werden alle Testpläne ausgewählt. Über die Checkbox *Test Runs* wird festgelegt, dass nur Test Runs ausgewählt und exportiert werden sollen.

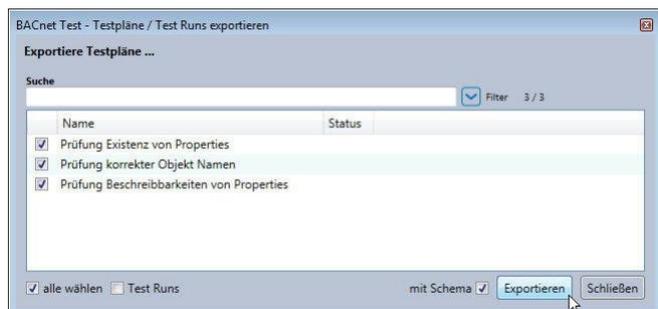


Abbildung 26

Die Liste hat Such-, Filter und Sortierfunktionen. Über die Checkbox *mit Schema* legen Sie fest, ob mit dem Export der Daten auch die zugehörige Schema-Datei mit exportiert werden soll oder nicht.

- ✚ Mit dem Button **[Exportieren]** starten Sie den Export.  
Vergeben Sie im Speichern-Dialog einen Dateinamen und klicken Sie auf **[Speichern]**.  
Der Fortschritt des Exports wird angezeigt.  
Ein Hinweisfenster zeigt den erfolgreichen Export an.

### Import von Testplänen und Test Runs

Einen Import von Testplänen oder Test Runs können Sie wie folgt ausführen:

- Über den Button  unterhalb des Testplan-Baums
- Über das Menü **BACnet Test > Testplan-Import...**

- ✚ Wählen Sie in beiden Fällen im sich öffnenden Dialog die Quelldatei aus und klicken Sie auf **[Öffnen]**.

Danach werden die enthaltenen Testpläne oder Test Runs im Import-Dialog gelistet (Abbildung 27). In der Liste der Registerkarte *Testpläne* können die zu importierenden Testpläne oder Test Runs per Checkbox markiert und danach über **[Importieren]** importiert werden.

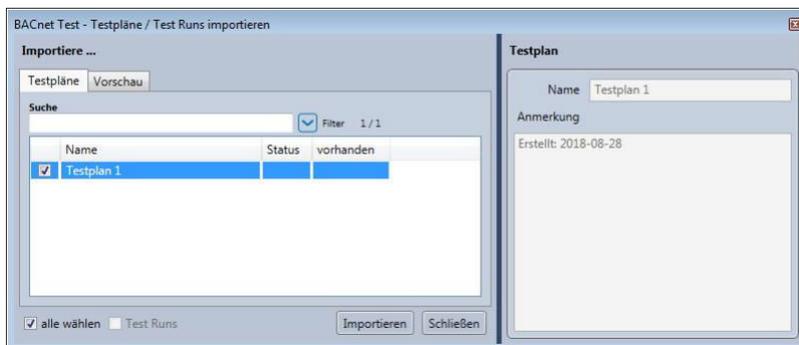


Abbildung 27

Die Liste hat Such-, Filter und Sortierfunktionen. Die Spalte *vorhanden* zeigt an, ob die Komponente schon im System vorhanden ist (Häkchen). Rechts von der Liste im Bereich *Testplan* werden die Eigenschaften der ausgewählten Komponente als Vorschau eingeblendet.

In der Registerkarte *Vorschau* wird der Testplan-Baumstruktur der ausgewählten Komponente angezeigt (Abbildung 28).

- ✚ Starten Sie den Import mit dem Button **[Importieren]**.

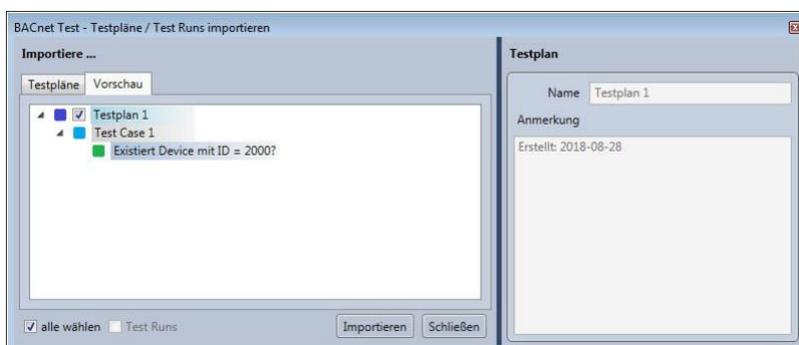


Abbildung 28

### Hinweis:

Wenn Sie schon vorhandene Komponenten importieren, weist ein Rückfragedialog darauf hin, dass die bestehenden Komponenten ersetzt werden (Abbildung 29).



Abbildung 29

### Export und Import von Test Cases

Test Cases exportieren Sie über das Kontextmenü eines Test Cases im Testplan-Baum.

- ☛ Klicken Sie auf den Kontextmenüpunkt **Exportieren**, vergeben Sie im sich öffnenden Speichern-Dialog einen Dateinamen (\*.tcf) und klicken Sie auf **[Speichern]**.

Test Cases importieren Sie über das Kontextmenü eines Test Plans im Testplan-Baum.

- ☛ Klicken Sie auf den Kontextmenüpunkt **Test Case importieren**, wählen Sie im sich öffnenden Dialog die Quelldatei aus und klicken Sie auf **[Öffnen]**.

Danach wird der enthaltene Test Case im Import-Dialog als Vorschau angezeigt (Abbildung 30).

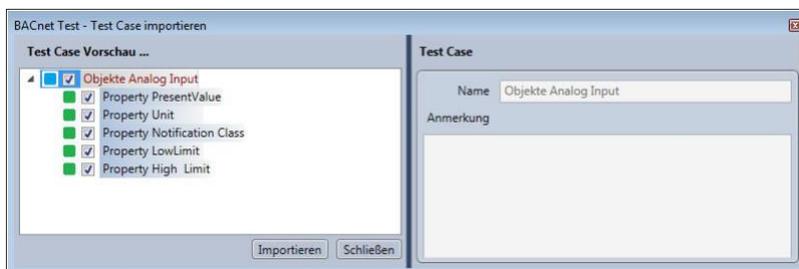


Abbildung 30

In der Liste können die zu importierenden Komponenten – z. B. einzelne Tests – per Checkbox an- und abgewählt werden.

- ☛ Starten Sie den Import mit dem Button **[Importieren]**.