



Bedienerhandbuch
netX Studio Engineering Tool
IOT-Konfiguration
V1.400

Hilscher Gesellschaft für Systemautomation mbH
www.hilscher.com

DOC180404OI01DE | Revision 1 | Deutsch | 2018-11 | Freigegeben | Öffentlich

Inhaltsverzeichnis

1	Allgemeine Informationen.....	4
1.1	Über dieses Handbuch.....	4
1.1.1	Änderungsübersicht	4
1.1.2	Online-Hilfe	4
1.2	Konventionen in diesem Dokument	5
1.3	Rechtliche Hinweise	6
1.4	Warenmarken.....	10
2	IOT-Funktionen OPC UA, MQTT	11
2.1	IOT-Funktionen mit netPROXY	11
2.2	Kommunikationskanäle	13
2.3	Labels.....	15
3	netX Studio Engineering Tool.....	16
3.1	Über das Werkzeug	16
3.1.1	Anwendungsbereich	16
3.1.2	Basic- und Engineering-Version	16
3.1.3	Systemvoraussetzungen.....	16
3.1.4	Voraussetzungen	16
3.2	Installation	17
3.2.1	Installationsschritte	17
3.2.2	Schritte zur Deinstallation	20
3.3	Kurzbeschreibung	21
3.3.1	Programm starten und Projekt anlegen	21
3.3.2	Projekt öffnen und schließen	23
3.3.3	Grafische Benutzeroberfläche	24
3.4	Zugriff auf Objektdaten.....	27
3.4.1	Beispiele	28
4	Übersicht zur IOT-Konfiguration	29
4.1	Projektkonfiguration erstellen und IOT-Konfiguration spezifizieren	29
4.2	Offline-Betrieb und Voraussetzungen	30
4.3	Schritte zur Vorgehensweise	30
4.3.1	Erläuterung der Schritte	31
4.3.2	Labels und Kommunikationseinstellungen für OPC UA, MQTT bzw. WebServer 33	
4.3.3	Beispiel C100	34
5	IOT-Konfiguration	35
5.1	Objekte erstellen	35
5.1.1	Kategorie anlegen	36
5.1.2	Objekt anlegen	40
5.1.3	Eigenschaften (Properties) des Objektes anpassen	44
5.1.4	Elemente (Elements) anlegen.....	48
5.2	Standardobjekte verwenden	52
5.2.1	Standardobjekte ergänzen und anzeigen	52
5.2.2	Standardobjekte kopieren oder löschen	54
5.2.3	Standardobjekte zu OPC UA, MQTT und WebServer zuordnen	55
5.3	Applikationsvariante konfigurieren (Funktionsumfang)	57

5.3.1	Funktionsumfang festlegen.....	57
5.3.2	Applikationsvariante konfigurieren.....	58
5.4	Kommunikationseinstellungen vornehmen	65
5.4.1	Kommunikationseinstellungen ergänzen	65
5.4.2	OPC UA, MQTT bzw. WebServer ergänzen.....	68
5.5	Produktdetails konfigurieren.....	78
5.5.1	Ordner für Produktdetails anlegen	79
5.5.2	Produktinformationen eingeben	83
5.6	Herstellerangaben eintragen.....	85
5.7	Benutzerverwaltung konfigurieren.....	87
5.8	Webinhalte erstellen und verwalten	91
5.9	Produktdateien erzeugen	91
5.9.1	Build-Prozess	91
5.9.2	Produktdateien kopieren, entfernen.....	95
5.9.3	Bedeutung der Produktdateien	96
5.10	Produktdateien herunterladen.....	97
6	Import, Export	98
6.1	Import.....	98
6.2	Export.....	103
6.2.1	Objektdefinitionen exportieren	103
6.2.2	Projektbeschreibungen exportieren	107
7	Anhang.....	111
7.1	Referenzdaten für netPROXY	111
7.1.1	Objekt-IDs	111
7.1.2	Element-Datentypen, Größe, Anzahl, Grenzen	112
7.2	Referenzen.....	113
	Glossar.....	118
	Kontakte.....	119

1 Allgemeine Informationen

1.1 Über dieses Handbuch

Dieses Handbuch beschreibt die IOT-Konfiguration mithilfe von netX Studio Engineering Tool zu folgenden Fragen:

- Einführung zur IOT-Konfiguration mit netPROXY: IOT-Funktionen OPC UA oder MQTT bei Produkten mit netX-Firmware ergänzen (bzw. WebServer kann ergänzt werden),
- Installation und Bedienung von netX Studio Engineering Tool,
- Übersicht und Schrittfolge zur IOT-Konfiguration,
- Vorgehensweise zur IOT-Konfiguration anhand eines Use Cases.

Die in diesem Handbuch empfohlene Vorgehensweise bei der IOT-Konfiguration richtet sich nach der Reihenfolge der Ordner in der Bedienoberfläche von netX Studio Engineering Tool, die sich an den Prinzipien von netPROXY orientiert.



Zu anderen Fragestellungen zu netX Studio Engineering Tool stehen weitere Handbücher zur Verfügung (siehe Abschnitt *Referenzen* [► Seite 113]), zum Beispiel zum Web Builder [5].

1.1.1 Änderungsübersicht

Index	Datum	Änderungen
1	29.11.2018	Erstellt

Tabelle 1: Änderungsübersicht

1.1.2 Online-Hilfe

Das Programm netX Studio Engineering Tool enthält eine integrierte Online-Hilfe.

- Um die Online-Hilfe aufzurufen, klicken Sie **View > Help** oder drücken Sie F1.

1.2 Konventionen in diesem Dokument

Hinweise, Handlungsanweisungen und Ergebnisse von Handlungen sind wie folgt gekennzeichnet:

Hinweise

**Wichtig:**

<Wichtiger Hinweis, der befolgt werden muss, um Fehlfunktionen auszuschließen>

**Hinweis:**

<Allgemeiner Hinweis >



<Hinweis, wo Sie weitere Informationen finden können>

Handlungsanweisungen

1. Handlungsziel
2. Handlungsziel
 - Handlungsanweisung

Ergebnisse

↻ Zwischenergebnis

⇒ Endergebnis

1.3 Rechtliche Hinweise

Copyright

© Hilscher Gesellschaft für Systemautomation mbH

Alle Rechte vorbehalten.

Die Bilder, Fotografien und Texte der Begleitmaterialien (in Form eines Benutzerhandbuchs, Bedienerhandbuchs, Statement of Work Dokument sowie alle weiteren Dokumenttypen, Begleittexte, Dokumentation etc.) sind durch deutsches und internationales Urheberrecht sowie internationale Handels- und Schutzbestimmungen geschützt. Sie sind ohne vorherige schriftliche Genehmigung nicht berechtigt, diese vollständig oder teilweise durch technische oder mechanische Verfahren zu vervielfältigen (Druck, Fotokopie oder anderes Verfahren), unter Verwendung elektronischer Systeme zu verarbeiten oder zu übertragen. Es ist Ihnen untersagt, Veränderungen an Copyrightvermerken, Kennzeichen, Markenzeichen oder Eigentumsangaben vorzunehmen. Darstellungen werden ohne Rücksicht auf die Patentlage mitgeteilt. Die in diesem Dokument enthaltenen Firmennamen und Produktbezeichnungen sind möglicherweise Marken bzw. Warenzeichen der jeweiligen Inhaber und können warenzeichen-, marken- oder patentrechtlich geschützt sein. Jede Form der weiteren Nutzung bedarf der ausdrücklichen Genehmigung durch den jeweiligen Inhaber der Rechte.

Wichtige Hinweise

Vorliegende Dokumentation in Form eines Benutzerhandbuchs, Bedienerhandbuchs sowie alle weiteren Dokumenttypen und Begleittexte wurden/werden mit größter Sorgfalt erarbeitet. Fehler können jedoch nicht ausgeschlossen werden. Eine Garantie, die juristische Verantwortung für fehlerhafte Angaben oder irgendeine Haftung kann daher nicht übernommen werden. Sie werden darauf hingewiesen, dass Beschreibungen in dem Benutzerhandbuch, den Begleittexte und der Dokumentation weder eine Garantie, noch eine Angabe über die nach dem Vertrag vorausgesetzte Verwendung oder eine zugesicherte Eigenschaft darstellen. Es kann nicht ausgeschlossen werden, dass das Benutzerhandbuch, die Begleittexte und die Dokumentation nicht vollständig mit den beschriebenen Eigenschaften, Normen oder sonstigen Daten der gelieferten Produkte übereinstimmen. Eine Gewähr oder Garantie bezüglich der Richtigkeit oder Genauigkeit der Informationen wird nicht übernommen.

Wir behalten uns das Recht vor, unsere Produkte und deren Spezifikation, sowie zugehörige Dokumentation in Form eines Benutzerhandbuchs, Bedienerhandbuchs sowie alle weiteren Dokumenttypen und Begleittexte jederzeit und ohne Vorankündigung zu ändern, ohne zur Anzeige der Änderung verpflichtet zu sein. Änderungen werden in zukünftigen Manuals berücksichtigt und stellen keine Verpflichtung dar; insbesondere besteht kein Anspruch auf Überarbeitung gelieferter Dokumente. Es gilt jeweils das Manual, das mit dem Produkt ausgeliefert wird.

Die Hilscher Gesellschaft für Systemautomation mbH haftet unter keinen Umständen für direkte, indirekte, Neben- oder Folgeschäden oder Einkommensverluste, die aus der Verwendung der hier enthaltenen Informationen entstehen.

Haftungsausschluss

Die Hard- und/oder Software wurde von der Hilscher Gesellschaft für Systemautomation mbH sorgfältig erstellt und getestet und wird im reinen Ist-Zustand zur Verfügung gestellt. Es kann keine Gewährleistung für die Leistungsfähigkeit und Fehlerfreiheit der Hard- und/oder Software für alle Anwendungsbedingungen und -fälle und die erzielten Arbeitsergebnisse bei Verwendung der Hard- und/oder Software durch den Benutzer übernommen werden. Die Haftung für etwaige Schäden, die durch die Verwendung der Hard- und Software oder der zugehörigen Dokumente entstanden sein könnten, beschränkt sich auf den Fall des Vorsatzes oder der grob fahrlässigen Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen vorhersehbaren Schaden begrenzt.

Insbesondere wird hiermit ausdrücklich vereinbart, dass jegliche Nutzung bzw. Verwendung von der Hard- und/oder Software im Zusammenhang

- der Luft- und Raumfahrt betreffend der Flugsteuerung,
- Kernschmelzungsprozessen in Kernkraftwerken,
- medizinischen Geräten die zur Lebenserhaltung eingesetzt werden
- und der Personenbeförderung betreffend der Fahrzeugsteuerung

ausgeschlossen ist. Es ist strikt untersagt, die Hard- und/oder Software in folgenden Bereichen zu verwenden:

- für militärische Zwecke oder in Waffensystemen;
- zum Entwurf, zur Konstruktion, Wartung oder zum Betrieb von Nuklearanlagen;
- in Flugsicherungssystemen, Flugverkehrs- oder Flugkommunikationssystemen;
- in Lebenserhaltungssystemen;
- in Systemen, in denen Fehlfunktionen der Hard- und/oder Software körperliche Schäden oder Verletzungen mit Todesfolge nach sich ziehen können.

Sie werden darauf hingewiesen, dass die Hard- und/oder Software nicht für die Verwendung in Gefahrumgebungen erstellt worden ist, die ausfallsichere Kontrollmechanismen erfordern. Die Benutzung der Hard- und/oder Software in einer solchen Umgebung geschieht auf eigene Gefahr; jede Haftung für Schäden oder Verluste aufgrund unerlaubter Benutzung ist ausgeschlossen.

Gewährleistung

Die Hilscher Gesellschaft für Systemautomation mbH übernimmt die Gewährleistung für das funktionsfehlerfreie Laufen der Software entsprechend der im Pflichtenheft aufgeführten Anforderungen und dafür, dass sie bei Abnahme keine Mängel aufweist. Die Gewährleistungszeit beträgt 12 Monate beginnend mit der Abnahme bzw. Kauf (durch ausdrückliches Erklärung oder konkludent, durch schlüssiges Verhalten des Kunden, z.B. bei dauerhafter Inbetriebnahme).

Die Gewährleistungspflicht für Geräte (Hardware) unserer Fertigung beträgt 36 Monate, gerechnet vom Tage der Lieferung ab Werk. Vorstehende Bestimmungen gelten nicht, soweit das Gesetz gemäß § 438 Abs. 1 Nr. 2 BGB, § 479 Abs.1 BGB und § 634a Abs. 1 BGB zwingend längere Fristen vorschreibt. Sollte trotz aller aufgewendeter Sorgfalt die gelieferte Ware einen Mangel aufweisen, der bereits zum Zeitpunkt des Gefahrübergangs vorlag, werden wir die Ware vorbehaltlich fristgerechter Mängelrüge, nach unserer Wahl nachbessern oder Ersatzware liefern.

Die Gewährleistungspflicht entfällt, wenn die Mängelrügen nicht unverzüglich geltend gemacht werden, wenn der Käufer oder Dritte Eingriffe an den Erzeugnissen vorgenommen haben, wenn der Mangel durch natürlichen Verschleiß, infolge ungünstiger Betriebsumstände oder infolge von Verstößen gegen unsere Betriebsvorschriften oder gegen die Regeln der Elektrotechnik eingetreten ist oder wenn unserer Aufforderung auf Rücksendung des schadhafte Gegenstandes nicht umgehend nachgekommen wird.

Kosten für Support, Wartung, Anpassung und Produktpflege

Wir weisen Sie darauf hin, dass nur bei dem Vorliegen eines Sachmangels kostenlose Nachbesserung erfolgt. Jede Form von technischem Support, Wartung und individuelle Anpassung ist keine Gewährleistung, sondern extra zu vergüten.

Weitere Garantien

Obwohl die Hard- und Software mit aller Sorgfalt entwickelt und intensiv getestet wurde, übernimmt die Hilscher Gesellschaft für Systemautomation mbH keine Garantie für die Eignung für irgendeinen Zweck, der nicht schriftlich bestätigt wurde. Es kann nicht garantiert werden, dass die Hard- und Software Ihren Anforderungen entspricht, die Verwendung der Hard- und/oder Software unterbrechungsfrei und die Hard- und/oder Software fehlerfrei ist.

Eine Garantie auf Nichtübertretung, Nichtverletzung von Patenten, Eigentumsrecht oder Freiheit von Einwirkungen Dritter wird nicht gewährt. Weitere Garantien oder Zusicherungen hinsichtlich Marktgängigkeit, Rechtsmangelfreiheit, Integrierung oder Brauchbarkeit für bestimmte Zwecke werden nicht gewährt, es sei denn, diese sind nach geltendem Recht vorgeschrieben und können nicht eingeschränkt werden.

Vertraulichkeit

Der Kunde erkennt ausdrücklich an, dass dieses Dokument Geschäftsgeheimnisse, durch Copyright und andere Patent- und Eigentumsrechte geschützte Informationen sowie sich darauf beziehende Rechte der Hilscher Gesellschaft für Systemautomation mbH beinhaltet. Er willigt ein, alle diese ihm von der Hilscher Gesellschaft für Systemautomation mbH zur Verfügung gestellten Informationen und Rechte, welche von der Hilscher Gesellschaft für Systemautomation mbH offen gelegt und zugänglich gemacht wurden und die Bedingungen dieser Vereinbarung vertraulich zu behandeln.

Die Parteien erklären sich dahin gehend einverstanden, dass die Informationen, die sie von der jeweils anderen Partei erhalten haben, in dem geistigen Eigentum dieser Partei stehen und verbleiben, soweit dies nicht vertraglich anderweitig geregelt ist.

Der Kunde darf dieses Know-how keinem Dritten zur Kenntnis gelangen lassen und sie den berechtigten Anwendern ausschließlich innerhalb des Rahmens und in dem Umfang zur Verfügung stellen, wie dies für deren Wissen erforderlich ist. Mit dem Kunden verbundene Unternehmen gelten nicht als Dritte. Der Kunde muss berechnigte Anwender zur Vertraulichkeit verpflichten. Der Kunde soll die vertraulichen Informationen ausschließlich in Zusammenhang mit den in dieser Vereinbarung spezifizierten Leistungen verwenden.

Der Kunde darf diese vertraulichen Informationen nicht zu seinem eigenen Vorteil oder eigenen Zwecken, bzw. zum Vorteil oder Zwecken eines Dritten verwenden oder geschäftlich nutzen und darf diese vertraulichen Informationen nur insoweit verwenden, wie in dieser Vereinbarung vorgesehen bzw. anderweitig insoweit, wie er hierzu ausdrücklich von der offen legenden Partei schriftlich bevollmächtigt wurde. Der Kunde ist berechnigt, seinen unmittelbaren Rechts- und Finanzberatern die Vertragsbedingungen dieser Vereinbarung unter Vertraulichkeitsverpflichtung zu offenbaren, wie dies für den normalen Geschäftsbetrieb des Kunden erforderlich ist.

Exportbestimmungen

Das gelieferte Produkt (einschließlich der technischen Daten) unterliegt gesetzlichen Export- bzw. Importgesetzen sowie damit verbundenen Vorschriften verschiedener Länder, insbesondere denen von Deutschland und den USA. Das Produkt/Hardware/Software darf nicht in Länder exportiert werden, in denen dies durch das US-amerikanische Exportkontrollgesetz und dessen ergänzender Bestimmungen verboten ist. Sie verpflichten sich, die Vorschriften strikt zu befolgen und in eigener Verantwortung einzuhalten. Sie werden darauf hingewiesen, dass Sie zum Export, zur Wiederausfuhr oder zum Import des Produktes unter Umständen staatlicher Genehmigungen bedürfen.

Geschäftsbedingungen

Beachten Sie auch die Hinweise zu weiteren relevanten rechtlichen Themen auf unserer netIOT-Website unter <http://www.netiot.com/netiot/netiot-edge/terms-and-conditions/>.

1.4 Warenmarken

Windows® 7, Windows® 8 und Windows® 10 sind registrierte Warenmarken der Microsoft Corporation.

Alle anderen erwähnten Marken sind Eigentum ihrer jeweiligen rechtmäßigen Inhaber. Die in diesem Dokument enthaltenen Firmennamen und Produktbezeichnungen sind möglicherweise Marken (Unternehmens- oder Warenmarken) der jeweiligen Inhaber und können marken- oder patentrechtlich geschützt sein.

2 IOT-Funktionen OPC UA, MQTT

2.1 IOT-Funktionen mit netPROXY

Real-Time-Ethernet-Produkte für die zyklische E/A-Datenkommunikation bzw. für die azyklische Kommunikation arbeiten auf Basis einer ladbaren Firmware „Real-Time-Ethernet“. Das Real-Time-Ethernet-Produkt enthält bereits eine entwickelte Applikation, welche diese zyklische bzw. azyklische Kommunikation nutzt.

Soll dieses Real-Time-Ethernet-Produkt auch eine IOT-Kommunikation mit den Protokollen OPC UA oder MQTT unterstützen, erfordert dies den Einsatz einer erweiterten ladbaren Firmware. Zusätzlich kann ein integrierter WebServer für das Produkt genutzt werden.

Mithilfe von netX Studio Engineering Tool werden die IOT-Funktionen OPC UA oder MQTT konfiguriert, bzw. WebServer kann hinzugefügt werden. Diese Erweiterung basiert auf netPROXY. Die notwendigen Konfigurationsschritte beschreibt dieses Dokument. Die bereits entwickelte Applikation kann erweitert werden und Datenkommunikation über ein IOT-Protokoll nutzen.

netPROXY-Objektmodell

netPROXY ist ein objektorientiertes Datenmodell und verfügt über eine protokollunabhängige Anwendungsprogramm-Schnittstelle (API) für den Zugriff auf die Objekte.

Das netPROXY-Objektmodell ermöglicht es, Geräteeigenschaften auf Objekte abzubilden. Jedes Objekt und damit die im Objekt definierten Geräteeigenschaften sind über eine ID identifizierbar. Für jedes Objekt kann der Anwender Festlegungen treffen, für den Zugriff auf das Objekt, sowie zur Weiterverarbeitung der Objektdaten.

Das netPROXY-Objektmodell beinhaltet auf der obersten Ebene die Objekt-ID. Darunter folgen die Instanz-ID und die Element-ID. Diese drei Ebenen charakterisieren zusammen ein Objekt. Die Instanz-ID legt die Häufigkeit fest, mit der ein Objekt vorkommt. Die Element-ID beinhaltet den eigentlichen Wert für die im Objekt definierte Geräteeigenschaft. Mehrere Objekte werden über eine Gruppen-ID zu einer Gruppe zusammengefasst.

Anwendungsbeispiel zur IOT-Konfiguration

Zum Beispiel wird ein Betriebsstundenzähler über eine Objekt-ID einem Objekt zugeordnet. Die Instanz-ID kennzeichnet wie häufig ein Betriebsstundenzähler vorkommt: Betriebsstundenzähler 1, Betriebsstundenzähler 2, etc. Die Betriebsstundenzahl wird in einem Element als „Status“ abgelegt, oder es könnten Einstellungen als „Parameter“ in einem Element abgelegt werden.

Spezifikation für die IOT-Konfiguration

Folgende Anwendungsfälle sind denkbar:

- OPC UA konfigurieren
- MQTT konfigurieren
- OPC UA und/oder MQTT konfigurieren
- Gegebenenfalls zusätzlich WebServer konfigurieren

Objekteigenschaften und -funktionen

Im netPROXY-Objektmodell verfügt jedes Objekt über Eigenschaften und Funktionen. Eine Eigenschaft ist durch einen Wert, einen Parameter sowie ein Kommunikationsattribut gekennzeichnet.

Wert
Parameter
Kommunikationsattribut (Kommunikationsweg)

Tabelle 2: Kennzeichnung von Objekteigenschaften

Ein Objekt kann die Funktionen Erstellen, Lesen, Schreiben und Löschen beinhalten.

Create (Erstellen)
Read (Lesen)
Write (Schreiben)
Delete (Löschen)

Tabelle 3: Objektfunktionen

Die Objekteigenschaften und -funktionen werden dem Objekt mithilfe des Werkzeugs netX Studio Engineering Tool mitgegeben.

2.2 Kommunikationskanäle

Die folgende Grafik gibt eine Übersicht zur Kommunikation zwischen der Applikation und dem Netzwerk. Neben der zyklischen und azyklischen Kommunikation und der Ethernet-Kommunikation zum Real-Time-Ethernet-Master ist dies die IOT-Kommunikation mit OPC UA oder MQTT vom Objekt zur Applikation bzw. zum Netzwerk. Zusätzlich kann ein integrierter WebServer genutzt werden.

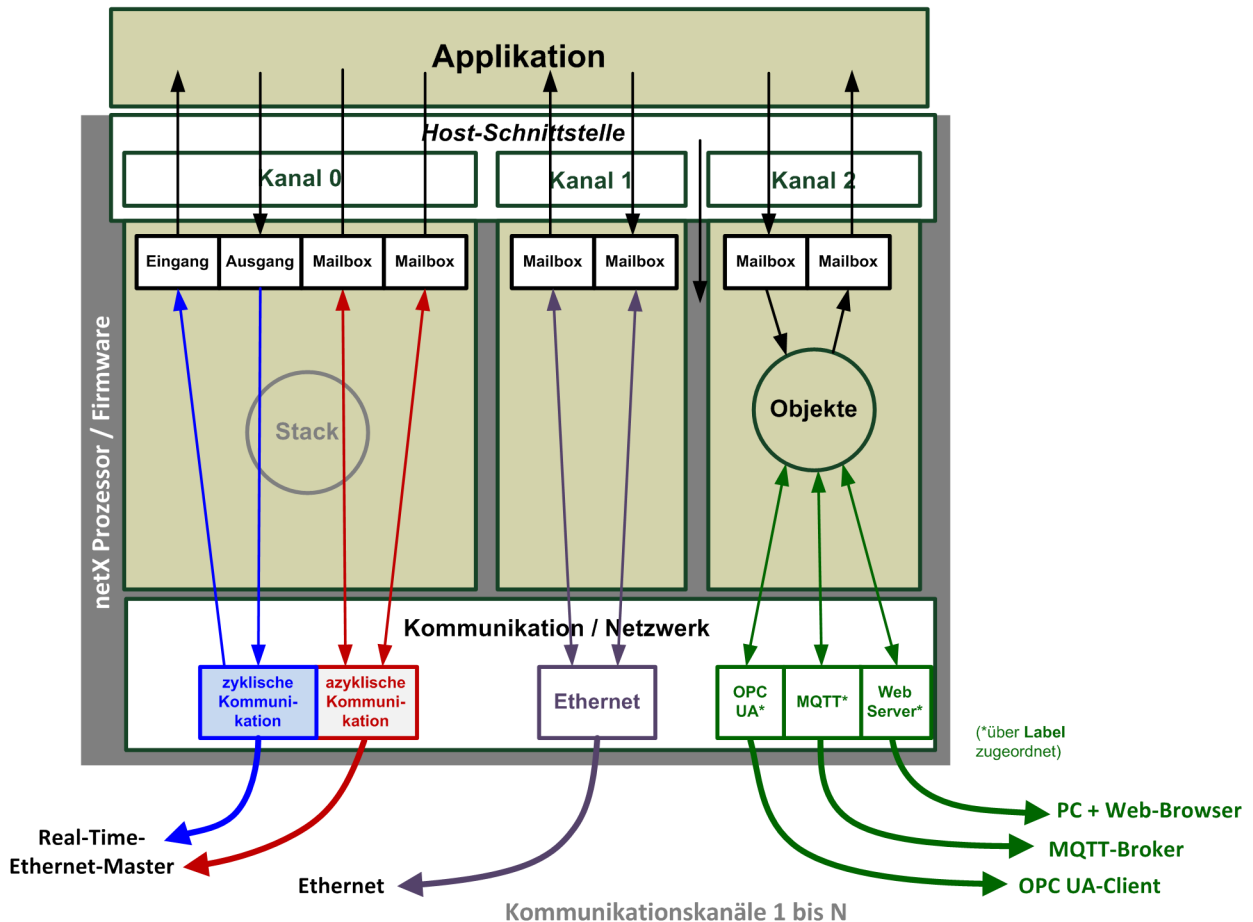


Abbildung 1: netX Studio Engineering Tool IOT - Kommunikationskanäle

Der Datentransfer zwischen Applikation und Kommunikation / Netzwerk wird im netX von der Geräte-Firmware mit IOT-Funktionen (dunkelgrau dargestellt) entsprechend der gültigen Konfiguration und den Einstellungen ausgeführt.

Bei der IOT-Kommunikation erfolgt der Datenfluss vom Objekt zum Netzwerk oder umgekehrt über:

- OPC UA*
- MQTT*
- WebServer**

**IOT-Funktion, **zusätzliche WebServer-Funktion*

Die Objekte mit dem Kommunikationsattribut **None** können über Labels zu OPC UA, MQTT bzw. WebServer zugeordnet werden. Weitere Angaben zur Konfiguration des Zugriffs auf Objekte bei der Objektdefinition finden Sie im Abschnitt *Zugriff auf Objektdaten* [► Seite 27].

Komponente	Beschreibung
Anwendungsprogramm (Applikation)	<p>Zyklische Kommunikation:</p> <ul style="list-style-type: none"> Eingangsdaten der Applikation: Die Applikation liest zyklische Daten vom Netzwerk. Ausgangsdaten der Applikation: Die Applikation schreibt zyklische Ausgangsdaten zum Netzwerk. <p>Azyklische Kommunikation:</p> <ul style="list-style-type: none"> Die Applikation bearbeitet die azyklische Kommunikation. <p>Ethernet-Kommunikation:</p> <ul style="list-style-type: none"> Die Applikation bearbeitet die Ethernet-Kommunikation. <p>Zugriff auf IOT-Objekte:</p> <ul style="list-style-type: none"> Die Applikation liest Daten aus den Objekten aus, die vom Netzwerk über OPC UA, MQTT oder WebServer empfangen wurden. Die Applikation schreibt Daten in die Objekte und diese Daten werden über OPC UA, MQTT oder WebServer zum Netzwerk gesendet.
Host-Schnittstelle	<p>Kanal 0: Eingang bzw. Ausgang für die zyklische und azyklische Real-Time-Ethernet-Kommunikation</p> <p>Kanal 1: Mailboxen für die Standard-Ethernet-Kommunikation</p> <p>Kanal 2: Protokollunabhängige Host-Schnittstelle für netPROXY</p> <p>Die Host-Schnittstelle für netPROXY kann ein serielles oder ein paralleles Dual-Port-Memory (DPM) sein und kann z. B. auf SPI basieren. Bestandteile sind Mailboxen.</p>
Objektverzeichnis im netPROXY-Anteil	<p>Entsprechend dem netPROXY-Datenmodell strukturiertes Verzeichnis der Objekte</p> <p>Über die Objekte werden einzelne Geräteeigenschaften zur Verfügung gestellt.</p> <p>Das zugrundeliegende netPROXY-Datenmodell ist unabhängig von dem später verwendeten Kommunikationsprotokoll oder der angeschlossenen Hardware.</p>
Kommunikation / Netzwerk	<p>Hardware-abhängige Kommunikationsschnittstelle bzw. Verbindung zum Netzwerk, über die Kommunikationskanäle 1 ... N.</p> <ul style="list-style-type: none"> Die zyklische, die azyklische und die Ethernet-Kommunikation zum Master sind als Bestandteile der netX-Firmware vorhanden. Die Kommunikation über OPC UA, MQTT bzw. WebServer kann von der Applikation zusätzlich genutzt werden.
Objektdaten	<p>Über OPC UA, MQTT und WebServer können Objekte gelesen bzw. geschrieben werden. OPC UA, MQTT und WebServer stellen aus Sicht der Objekte eine „Applikation“ dar.</p>
Kommunikationsweg	<p>Die Kommunikation erfolgt von der Applikation zum Netzwerk bzw. vom Netzwerk zur Applikation. Weitere Angaben finden Sie in dem hier nachfolgenden Abschnitt sowie im Abschnitt <i>Zugriff auf Objektdaten</i> [► Seite 27].</p>

Tabelle 4: Komponenten netX Firmware mit IOT

2.3 Labels

Die Konfiguration des Zugriffs auf die Objektdaten erfolgt über Labels.

Labels definieren den Zugriff auf die Objektdaten zum/vom Netzwerk über

- OPC UA,
- MQTT und
- WebServer.

Die Applikation greift über die Mailbox auf die Objektdaten zu.

Bei der Definition der Objekte wird festgelegt, auf welchen Kommunikationswegen die Firmware die von den Objekten bereitgestellten Geräteeigenschaften übertragen soll.

Kommunikationsattribut „None“



Hinweis:

Für die IOT-Konfiguration von netX-Firmware-Produkten ist bei den Objekten standardmäßig das Kommunikationsattribut **None** festgelegt, da die Eingangs- und Ausgangsdaten nicht über netPROXY abgebildet werden.

3 netX Studio Engineering Tool

3.1 Über das Werkzeug

Das netX Studio Engineering Tool ist ein Eclipse-basiertes Werkzeug, welches für Produkte mit netX-Firmware bei der Konfiguration der IOT-Kommunikation für folgende Zwecke eingesetzt werden kann:

- Objektkonfiguration (Objekte erstellen und verwalten),
- Konfiguration für OPC UA und MQTT,
- Einstellungen für Benutzerrechte und Webinhalte,
- Build-Prozess der IOT-Kommunikation (Produktdateien erzeugen, mit Objektkonfiguration und allen Einstellungen der IOT-Protokolle).

3.1.1 Anwendungsbereich

Der Anwendungsbereich für das netX Studio Engineering Tool ist auf Slave-Geräte auf Basis von netX begrenzt, die mit einer ladbaren Firmware arbeiten. Das Werkzeug wird hauptsächlich für Real-Time-Ethernet-Systeme eingesetzt.

3.1.2 Basic- und Engineering-Version

netX Studio Engineering Tool können Sie als kostenfreie Basisversion mit eingeschränktem Funktionsumfang erhalten. Für die Vollversion müssen Sie eine Lizenz erwerben. Die beiden Versionen unterscheiden sich wie folgt:

- **netX Studio (netPROXY Basic)** ermöglicht Ihnen die Konfiguration
 - einer Applikationsvariante und
 - eines Kommunikationsprotokolls (Netzwerk).
- **netX Studio (netPROXY Engineering)** ermöglicht Ihnen die Konfiguration
 - beliebig vieler Applikationsvarianten sowie
 - aller verfügbaren Kommunikationsprotokolle (Netzwerke).

3.1.3 Systemvoraussetzungen

- PC mit Standardausstattung (Monitor, Tastatur, Maus, DVD-Laufwerk, gegebenenfalls USB-Anschluss etc.)
- Windows® 7 (32-Bit) SP1, Windows® 7 (64-Bit) SP1, Windows® 8 (32-Bit), Windows® 8 (64-Bit) oder Windows® 10 (64-Bit)

3.1.4 Voraussetzungen



Hinweis:

Um ein Gerät über USB (USB/RS232) mit netX Studio Engineering Tool verbinden zu können, darf keine andere serielle Verbindung zum Gerät bestehen.

3.2 Installation

In den folgenden Abschnitten finden Sie Beschreibungen zur Installation sowie zur Deinstallation des Programms netX Studio Engineering Tool unter Windows® 7, Windows® 8 bzw. unter Windows® 10 auf Ihrem PC.

Voraussetzungen und Hinweise

Sie benötigen die Installationsdatei nxsbuilder-win32.win32.x86.msi.



Hinweis:

Vor der Installation bzw. Deinstallation müssen Sie alle Programme auf Ihrem PC schließen, außerdem benötigen Sie Administratorrechte unter Windows®.

Wenn Sie bereits eine ältere Programmversion installiert haben, deinstallieren Sie diese, bevor Sie eine neue Version installieren.

Bei der Programmdeinstallation werden vom Anwender angelegte Dateien und Ordner nicht gelöscht.

3.2.1 Installationsschritte

- Machen Sie einen Doppelklick auf die Installationsdatei.
- Die Installation wird gestartet.
- Klicken Sie auf **Next**.

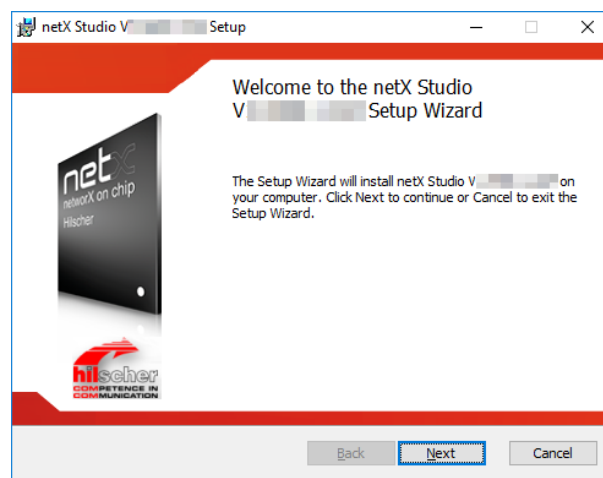


Abbildung 2: Installationsinformation (Beispiel Windows® 10)

- Haken Sie **I accept the terms in the License Agreement** an.

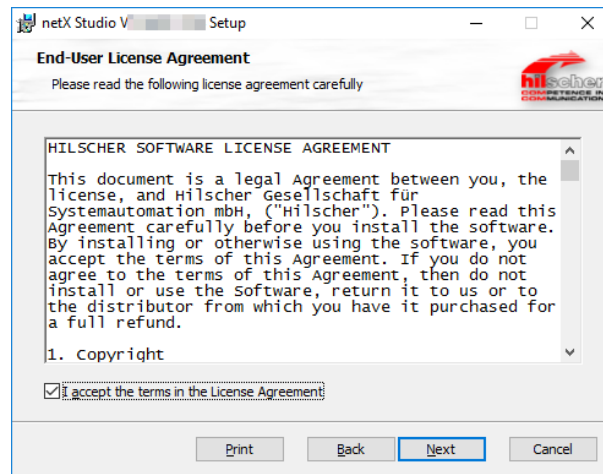


Abbildung 3: Endbenutzer-Lizenzvereinbarung akzeptieren (Beispiel Windows® 10)

- Klicken Sie auf **Next**.

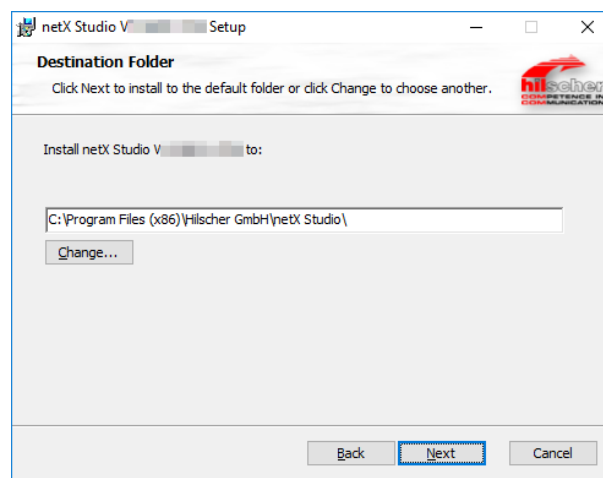


Abbildung 4: Zielordner für die Installation (Beispiel Windows® 10)

- Legen Sie den Zielordner fest.

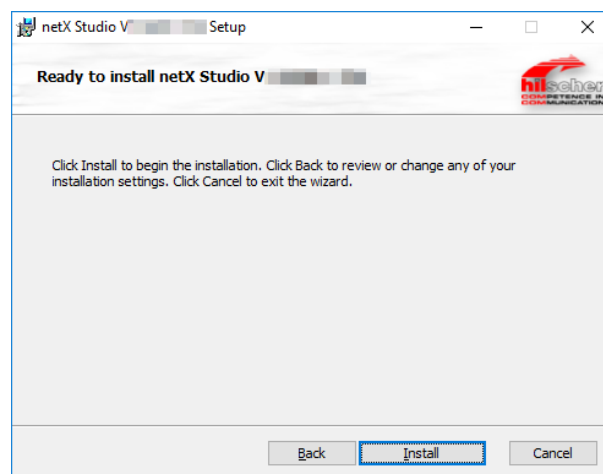


Abbildung 5: Installation durchführen (Beispiel Windows® 10)

- Klicken Sie auf **Install**.

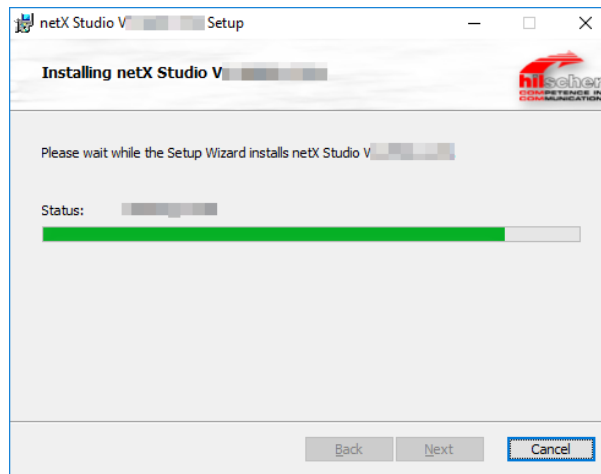


Abbildung 6: Installationsfortschritt (Beispiel Windows® 10)

- ⇒ Es erscheint eine Windows® Sicherheitsabfrage.
- **Ja** anklicken.
- ⇒ Die Komponenten werden installiert. Dies dauert einige Minuten.
- Klicken Sie auf **Finish**.

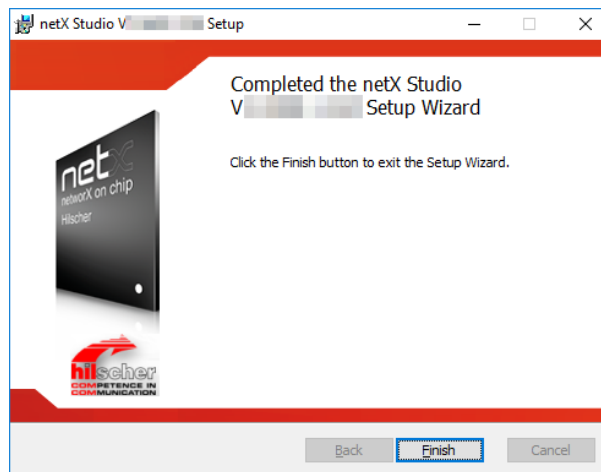


Abbildung 7: Installationsabschluss (Beispiel Windows® 10)

- ⇒ Die Installation ist abgeschlossen.

3.2.2 Schritte zur Deinstallation

- Öffnen Sie das Windows® Menüfenster **Programme deinstallieren oder ändern**.
- Machen Sie einen Rechtsklick auf den Eintrag **netX Studio VX.XXXX.XXXXX**.
- Klicken Sie **Deinstallieren** an.
- ⇒ Die Sicherheitsabfrage erscheint, ob Sie netX Studio deinstallieren möchten.
- Klicken Sie **Ja** an.
- ⇒ Der Windows Installer bereitet die Deinstallation vor.

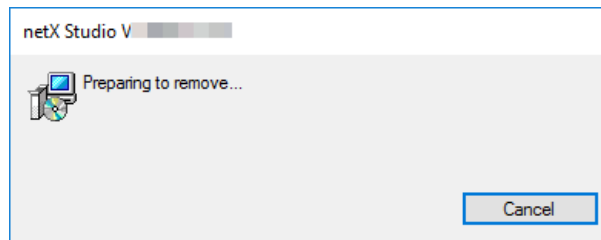


Abbildung 8: Deinstallation wird vorbereitet (Beispiel Windows® 10)

- ⇒ Eine Windows® Sicherheitsabfrage erscheint.
- Klicken Sie **Ja** an.
- ⇒ Die Komponenten werden deinstalliert. Dies dauert einige Minuten.

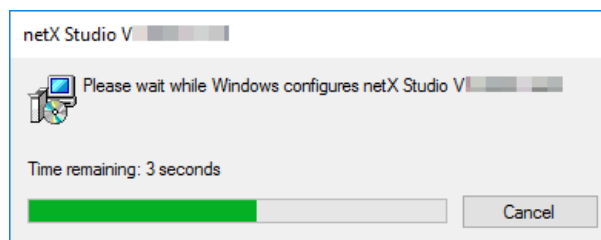


Abbildung 9: Fortschrittsanzeige zur Deinstallation (Beispiel Windows® 10)

- ⇒ Die Deinstallation ist abgeschlossen. Das Programm wurde von Ihrem PC entfernt.

3.3 Kurzbeschreibung

3.3.1 Programm starten und Projekt anlegen

**Hinweis:**

Die Geräteentwicklung mit netX Studio Engineering Tool ist ein Offline-Prozess. Eine Verbindung zu einem Gerät ist nicht erforderlich.

- Um netX Studio Engineering Tool zu starten, klicken Sie **Start > netX Studio**.
- ⇒ Das Programm wird mit dem Startbildschirm **Welcome** geöffnet.

In der Basisversion **netX Studio (netPROXY Basic)** erscheint zusätzlich die Meldung:

Die kostenfreie „netPROXY Basic“-Version von netX Studio begrenzt die Gerätekonfiguration auf:

- ein Objektmodell (Applikation)
- ein Kommunikationsprotokoll (Kommunikationseinstellungen).

Für die unlimitierte „netPROXY Engineering“-Version wenden Sie sich bitte an die Vertriebsabteilung der Hilscher GmbH unter <http://www.hilscher.com/en/sales/sales-contact/>.

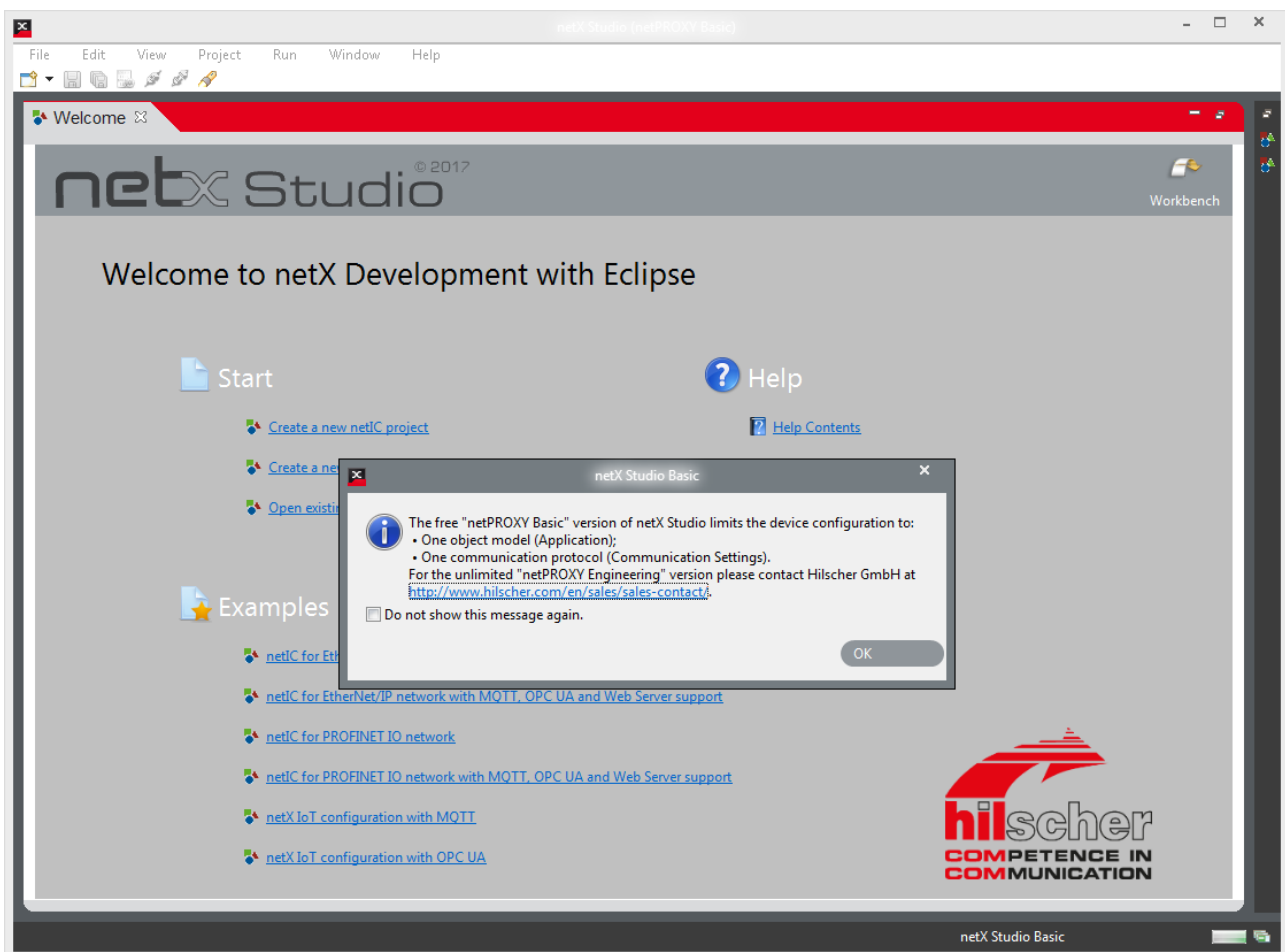


Abbildung 10: Startbildschirm netX Studio (netPROXY Basic) mit Meldungsfenster

- Bestätigen Sie die Meldung mit **OK**.

Bei der Vollversion **netX Studio (netPROXY Engineering)** erscheint der Startbildschirm ohne eine Meldung.

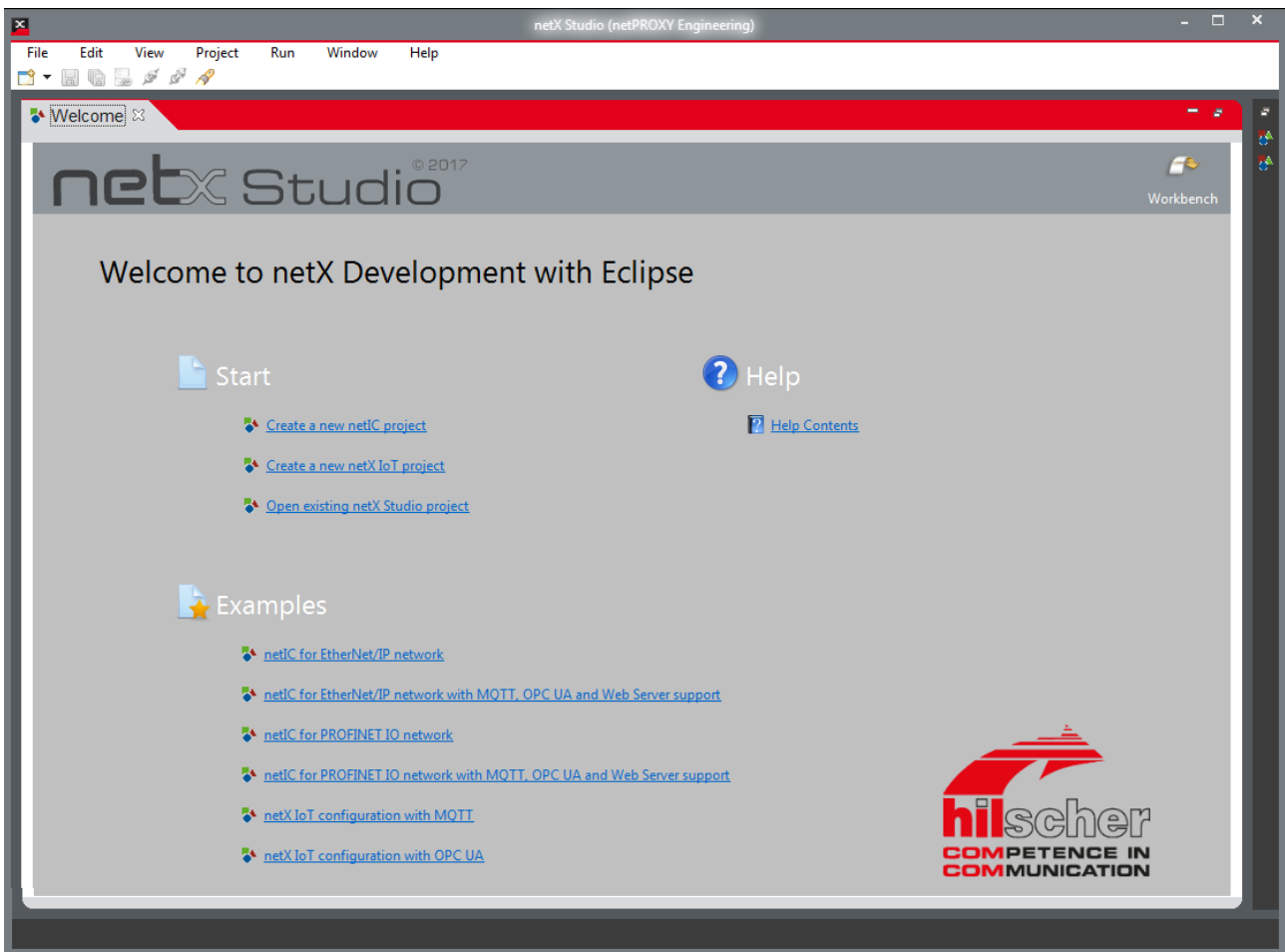


Abbildung 11: Startbildschirm netX Studio (netPROXY Engineering)

Auf dem Startbildschirm können Sie ein Projekt erstellen. Wenn Sie vor dem Beenden des Programms Ihr Projekt nicht schließen, wird das Projekt beim nächsten Programmstart automatisch angezeigt.

Projekt anlegen

- Um ein Projekt anzulegen, klicken Sie **File > New > New netX IoT project ...** (oder wählen Sie auf der Registerkarte **Welcome > Start > Create a new netX IoT project**).

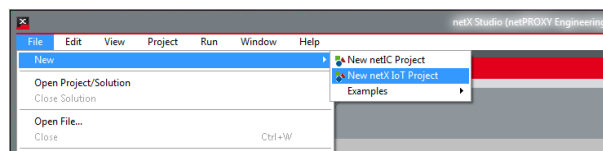


Abbildung 12: Neues Projekt anlegen

- Das Fenster **New netX IoT Project** wird geöffnet:

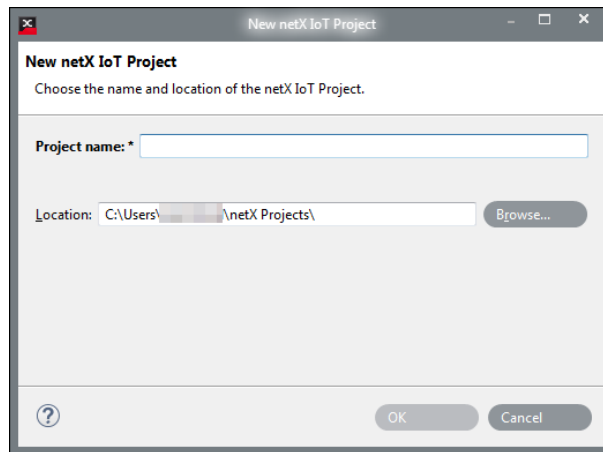


Abbildung 13: Projektname und Speicherpfad eingeben

- Geben Sie unter **Project Name*** (Pflichtfeld) einen Projektnamen ein.
- Stellen Sie unter **Location** den Speicherpfad für Ihr Projekt ein und klicken Sie **OK** an.
- ⇒ Das Projekt wird angelegt.



Wichtig:

Beachten Sie, dass netX Studio Engineering Tool alle Einstellungen und Änderungen in Ihrem Projekt automatisch in die Projektdatei übernimmt.

3.3.2 Projekt öffnen und schließen

Projekt öffnen

- Um in netX Studio Engineering Tool ein Projekt zu öffnen, klicken Sie **File > Open Project/Solution**.
- Oder wählen Sie auf der Registerkarte **Welcome > Start > Open existing netX IoT project**.
- ⇒ Das Fenster **Öffnen** erscheint.
- Wählen Sie die Datei **.project** (Typ: PROJECT-Datei) oder die Datei **.solproject** (Typ: SOLPROJECT-Datei).
- Klicken Sie **Öffnen**.
- ⇒ Das Projekt wird geöffnet.



Hinweis:

Beachten Sie, dass Sie in der Basisversion von netX Studio Engineering Tool „netX Studio (netPROXY Basic)“ nur Projekte aufrufen können, deren Konfiguration auf *eine* Applikationsvariante sowie *ein* Kommunikationsprotokoll (Netzwerk) beschränkt ist.

Projekt schließen

- Um in netX Studio Engineering Tool ein Projekt zu schließen, klicken Sie **File > Close Solution**.
- ⇒ Das Projekt wird geschlossen.

3.3.3 Grafische Benutzeroberfläche

In der folgenden Abbildung sehen Sie die grafische Benutzeroberfläche von netX Studio Engineering Tool mit dem geöffneten Projekt:

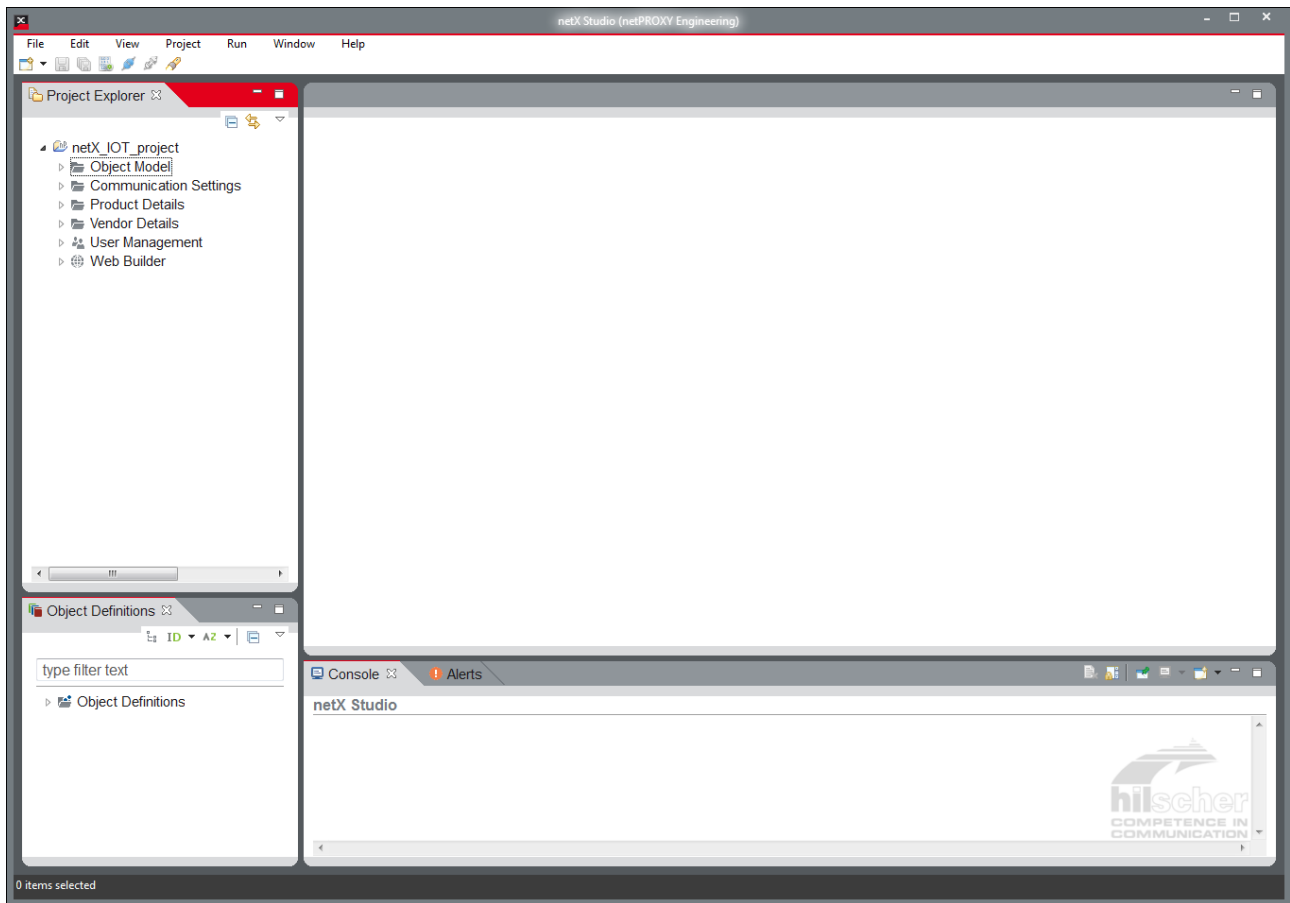


Abbildung 14: Grafische Benutzeroberfläche von netX Studio Engineering Tool mit geöffnetem Projekt

Hauptkomponenten

Die grafische Benutzeroberfläche des Programms gliedert sich in:

- das Menü,
- die Symbolleiste und
- die beweglichen Fenster:
 - **Project Explorer** und Editor (die Anzeige ist abhängig vom gewählten Ordner),
 - **Object Definitions, Console** und **Alerts**.



Hinweis:

Um die IOT-Konfiguration korrekt durchzuführen, müssen Sie die Ordnerstruktur im Fenster **Project Explorer** von oben nach unten abarbeiten. Abschließend wird der Build-Prozess durchgeführt.

Symbole und Dialoge

Hinzu kommen die in den einzelnen Fenstern angeordneten Symbole zur Programmbedienung bzw. Dialoge, die bei den unterschiedlichen Schritten zur IOT-Konfiguration erscheinen. Die Bedeutung der Symbole können Sie mithilfe der in den Schnellinfos gegebenen Beschreibungen erfassen.

Weitere Informationen



Eine Beschreibung der verschiedenen Programmkomponenten und Bedienelemente finden Sie bei den Beschreibungen der Schritte zur IOT-Konfiguration im Kapitel *IOT-Konfiguration* [► Seite 35].

3.3.3.1 Symbolleiste

Die netX Studio Engineering Tool-Symbolleiste enthält Symbole zum Aufruf verschiedener Programmooptionen.

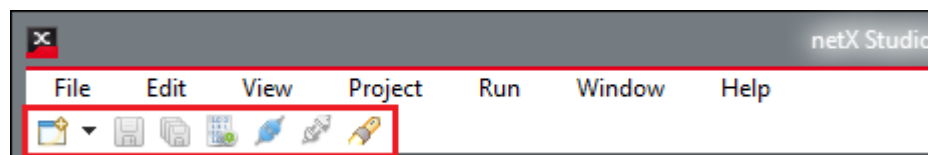


Abbildung 15: Symbolleiste (aktiviert)

Symbol	Name	Menüpfad	Bedeutung	Weitere Informationen siehe Abschnitt
	New Project	File > New > New Project	Neues Projekt anlegen	<i>Programm starten und Projekt anlegen</i> [► Seite 21]
	Save	File > Save	Nicht aktivierbar in netX Studio Engineering Tool	-
	Save as	File > Save as		
	Build	Project > Build	Build-Prozess starten	<i>Build-Prozess</i> [► Seite 91]
	Connect	Run > Connect	Online-Verbindung zum Gerät herstellen	Ohne Funktion
	Disconnect	Run > Disconnect	Online-Verbindung zum Gerät trennen	
	Search	-	Standard-Suchfunktionen in Eclipse	-

Tabelle 5: Symbole in der netX Studio Engineering Tool-Symbolleiste

Beispielprojekte aufrufen

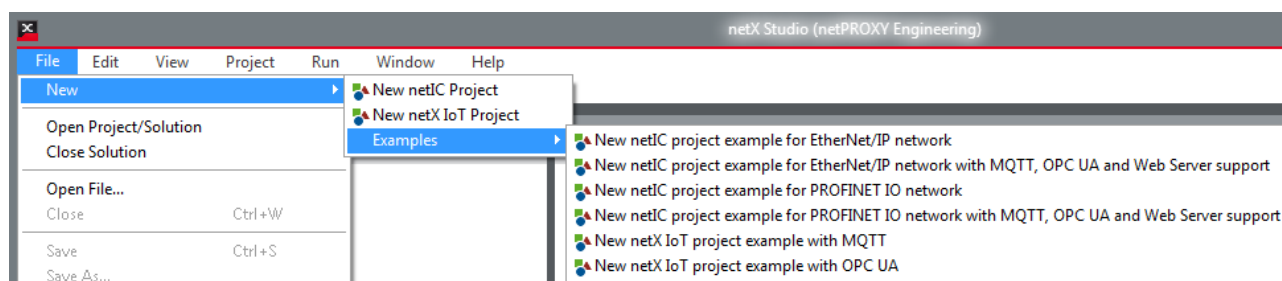


Abbildung 16: Menü New Project

Beispielprojekte können Sie aufrufen über **File > New > Examples >**

- **New netX IoT project example with MQTT,**
- **New netX IoT project example with OPC UA.**

3.3.3.2 Allgemeine Bedienelemente

In der nachfolgenden Tabelle finden Sie Erläuterungen zu Bedienelementen, die in netX Studio Engineering Tool wiederholt in verschiedenen Dialogen vorkommen.



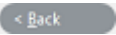
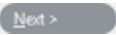
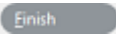
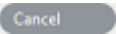
Parameter	Bedeutung
Filter [type filter text] 	Feld zur Eingabe eines Filtertextes.
	Hilfeinformationen aufrufen.
	Einen Schritte zurück gehen.
	Einen Schritte vor gehen.
	Den aktuellen Schritt abschließen.
	Den Vorgang abbrechen.

Tabelle 6: Allgemeine Bedienelemente

3.4 Zugriff auf Objektdaten

In netX Studio Engineering Tool können Sie für die IOT-Kommunikation über OPC UA bzw. MQTT konfigurieren, wie auf die Objektdaten zugegriffen werden soll.

OPC UA, MQTT, WebServer

Der Zugriff auf Objektdaten über OPC UA, MQTT bzw. WebServer wird über die Zuweisung von Labels auf Objektebene konfiguriert.

Datenzuordnung auf Objektebene

Die Konfiguration der Kommunikation über OPC UA, MQTT bzw. WebServer erfolgt auf Objektebene. Dabei wird der Zugriff auf die Daten *aller* Elemente eines Objektes für den jeweiligen Kommunikationsweg möglich.

Object Definitions-Editor und Communication Settings

Die Konfiguration der Zugriffsmöglichkeiten auf die Objektdaten erfolgt in netX Studio Engineering Tool für OPC UA, MQTT bzw. WebServer

- bei der Definition der Objekte im **Object Definitions**-Editor, durch Zuweisung von Labels auf Objektebene,
- sowie bei den Kommunikationseinstellungen unter **Communication Settings**, durch Ergänzung des jeweiligen Editors bzw. die erneute Zuweisung von Labels darin und die Durchführung der entsprechenden Konfiguration.

Kombinationsmöglichkeiten verschiedener Zugriffswege auf Objektdaten



Wichtig:

Die Zugriffsmöglichkeiten auf Objektdaten über OPC UA, MQTT und WebServer können auf verschiedene Weise miteinander kombiniert werden. In den Kommunikationseinstellungen für eine Applikationsvariante dürfen OPC UA und MQTT jedoch nicht gleichzeitig genutzt werden.



Wichtig:

1. Ein Objekt oder ein Element darf nur von einem Anwender (Master, OPC-Client etc.) aus geschrieben werden.
2. Ein Objekt oder Element darf von mehreren Anwendern gelesen werden.

3.4.1 Beispiele

Die Tabelle *Objektdaten nach Konfiguration des Zugriffs (Beispiele)* zeigt Beispiele für Kombinationsmöglichkeiten verschiedener Zugriffswege vom Netzwerk auf Objektdaten. Betrachtet werden die Zugriffsmöglichkeiten **OPC UA**, **MQTT** und **WebServer**.

Objektdaten*	Kommunikations- attribut	Label		
		OPC UA	MQTT	WebServ
• Objektdaten, auf die vom WebServer aus zugegriffen werden soll.	None	-	-	www
• Objektdaten, auf die vom OPC UA-Server aus zugegriffen werden soll.	None	OPC UA	-	-
• Objektdaten, auf die vom OPC UA-Server aus und zusätzlich vom WebServer aus zugegriffen werden soll.	None	OPC UA	-	OPC UA
• Objektdaten, auf die über MQTT zugegriffen werden soll.	None	-	MQTT	-
• Objektdaten, auf die über MQTT und zusätzlich vom WebServer aus zugegriffen werden soll.	None	-	MQTT	MQTT

Tabelle 7: Objektdaten nach Konfiguration des Zugriffs (Beispiele)

*Auf alle in der Spalte Objektdaten genannten Daten hat die Applikation Lese- und Schreibzugriff.

4 Übersicht zur IOT-Konfiguration

Das Ziel der IOT-Konfiguration ist es, Produkte mit netX-Firmware um die IOT-Funktionen OPC UA oder MQTT zu erweitern. Auf Basis von netPROXY kann zusätzlich ein integrierter WebServer ergänzt werden.

4.1 Projektkonfiguration erstellen und IOT-Konfiguration spezifizieren

Als vorbereitender Schritt für die Abbildung der Geräteeigenschaften in Objekten und die IOT-Konfiguration mit netX Studio Engineering Tool ist eine Aufstellung des vorhandenen Funktionsumfangs, der Geräteeigenschaften bzw. des Gerätedesigns sowie die Spezifikation der IOT-Konfiguration der zu konfigurierenden Produkte erforderlich (im verwendeten Beispiel ‚netX_IOT_project‘ mit dem Beispiel ‚C100‘).

Gehen Sie wie folgt vor:

- **Design der Produkte und Produktfamilie**
 - Erstellen Sie für die Produkte (bzw. Produktfamilie), bei denen IOT-Funktionen ergänzt werden sollen, eine Aufstellung des vorhandenen Funktionsumfangs, der Geräteeigenschaften bzw. des Gerätedesigns.
- **Definition der Objekte und der Zuordnung zu Objektkategorien**
 - Legen Sie Objektkategorien (gegebenenfalls mit ID-Bereichen) und Objekte fest, um die Geräteeigenschaften abzubilden.

Die Objektkategorien dienen dazu, die Objekte sinnvoll zu strukturieren.

- Legen Sie die Objektkennungen, die Objektnamen und die Zuordnung der Objekte zu den Kategorien fest.

Berücksichtigen Sie auch die in netX Studio Engineering Tool verfügbaren Standardobjekte.

- **Zugriff auf die Objekte über das Netzwerk**
 - Spezifizieren Sie die zu ergänzenden IOT-Funktionen (OPC UA oder MQTT) bzw. gegebenenfalls zusätzlich die Funktion WebServer und legen Sie damit die erforderlichen Kommunikationswege fest (OPC UA oder MQTT, bzw. WebServer).
- **Anlegen der Elemente**
 - Legen Sie fest, welche Elemente Sie für die einzelnen Objekte benötigen, d. h., ob z. B. noch Elemente für Parameter angelegt werden müssen.

4.2 Offline-Betrieb und Voraussetzungen

Die IOT-Konfiguration mit netX Studio Engineering Tool einschließlich dem Build-Prozess erfolgt im Offline-Betrieb. Eine Verbindung zu einem Gerät ist dafür nicht erforderlich.

4.3 Schritte zur Vorgehensweise

Die folgende Tabelle zeigt eine Übersicht zur IOT-Konfiguration, zur Inbetriebnahme bzw. zum Test.

Aufgabe	Schritt	Beschreibung	Menü bzw. Ordner in netX Studio Engineering Tool	Weitere Informationen in Abschnitt
IOT-Konfiguration (Offline-Prozess)	1.1	Projekt erstellen	File > New > New Project ...	<i>Programm starten und Projekt anlegen</i> [► Seite 21]
	1.2	Objekt erstellen	Object Model / Object Definitions / User	<i>Objekte erstellen</i> [► Seite 35]
	1.3	Standardobjekte verwenden	Object Model / Object Definitions / Standard	<i>Standardobjekte verwenden</i> [► Seite 52]
	1.4	Applikationsvariante konfigurieren (Funktionsumfang)	Object Model / Object Instances	<i>Applikationsvariante konfigurieren (Funktionsumfang)</i>
	1.5	Kommunikationseinstellungen vornehmen	Communication Settings	<i>Kommunikationseinstellungen vornehmen</i>
	1.6	Produktdetails konfigurieren	Product Details	<i>Produktdetails konfigurieren</i> [► Seite 78]
	1.7	Herstellerangaben eintragen	Vendor Details	<i>Herstellerangaben eintragen</i> [► Seite 85]
	1.8	Benutzerverwaltung konfigurieren	User Management	<i>Benutzerverwaltung konfigurieren</i> [► Seite 87]
	1.9	Webinhalte erstellen und verwalten	Web Builder	<i>Webinhalte erstellen und verwalten</i> [► Seite 91]
	1.10	Produktdateien erzeugen	Product Build	<i>Produktdateien erzeugen</i> [► Seite 91]
Laden der Produktdateien	2.1	Produktdateien (mit der Konfiguration) mithilfe der Host-Applikation in das Gerät übertragen	-	<i>Produktdateien herunterladen</i> [► Seite 97]

Tabelle 8: Schrittfolge bei der Projektkonfiguration bzw. zur IOT-Konfiguration

4.3.1 Erläuterung der Schritte

Im folgenden Abschnitt finden Sie Erläuterungen der Schritte für die Projektkonfiguration zur Abbildung bestehender Produkte sowie zur IOT-Konfiguration in netX Studio Engineering Tool.

Projektkonfiguration und Ergänzung der IOT-Konfiguration (Offline-Prozess)

- **Schritt 1.1: Projekt erstellen**

- Legen Sie ein neues Projekt an.

- **Schritt 1.2: Objekt erstellen**

- Legen Sie für die Produkte mit den IOT-Funktionen den Funktionsumfang fest, einschließlich der Geräteeigenschaften.
- Spezifizieren Sie die zu ergänzenden IOT-Funktionen.
- Definieren Sie die Objekte sowie den Zugriff auf die Objekte und deren Zuordnung zum Netzwerk.

Zum Netzwerk kann jeweils eine Kommunikation über OPC UA oder MQTT eingerichtet werden, bzw. gegebenenfalls zusätzlich über WebServer.

- **Schritt 1.3: Standardobjekte verwenden**

netX Studio Engineering Tool bietet auf der Ebene der Objektdefinition die Möglichkeit vordefinierte Standardobjekte zu verwenden. Eine Zuordnung der Standardobjekte zu OPC UA-Client, MQTT bzw. WebServer ist standardmäßig nicht eingerichtet und kann (falls erforderlich) nur manuell konfiguriert werden.

- **Schritt 1.4: Applikationsvariante konfigurieren (Funktionsumfang)**

- Klären Sie anhand der Aufstellung aus Schritte 1.3 die abzubildende Produktsystematik, den vorhandenen Funktionsumfang bzw. die Geräteeigenschaften.
- Konfigurieren Sie auf Basis der in Schritt 1.3 definierten Objekte den Funktionsumfang für die zur IOT-Konfiguration vorgesehenen Produkte.



Hinweis:

Beachten Sie, dass Sie in der Basisversion von netX Studio Engineering Tool „netX Studio (netPROXY Basic)“ nur *eine* Applikationsvariante (Application) konfigurieren können.

- **Schritt 1.5: Kommunikationseinstellungen vornehmen (IOT-Konfiguration)**

- Nehmen Sie die erforderlichen Kommunikationseinstellungen vor, um den Zugriff auf Objektdaten über OPC UA, MQTT bzw. WebServer zu konfigurieren.

- **Schritt 1.6: Produktdetails konfigurieren**

- Legen Sie für jeden Gerätetyp einen Ordner für die Produktdetails an.
- Konfigurieren Sie die Produktdaten zur Hardware, zur Firmware und zum Bootloader, sowie die Daten zur Produktidentifikation und Versionierung der einzelnen Geräte.

- **Schritt 1.7: Herstellerangaben eintragen**

- **Schritt 1.8: Benutzerverwaltung konfigurieren**

- **Schritt 1.9: Webinhalte erstellen und verwalten**



Angaben zur Benutzerverwaltung und zum WebServer finden Sie in gesonderten Dokumentationen (siehe Abschnitt *Referenzen* [► Seite 113]).

- **Schritt 1.10: Produktdateien erzeugen**

- Starten Sie den Build-Prozess für das Gerät zur Erzeugung der Produktdateien mit der Konfiguration.

In den Produktdateien sind die Objektdefinition und die IOT-Konfiguration sowie alle Einstellungen zum Produkt enthalten.

Laden der Produktdateien

- **Schritt 2.1: Produktdateien mit der Konfiguration in das Gerät laden**

- Laden Sie alle mit dem Build unter **Product Build > Products > Product (Application)** erzeugten Produktdateien mithilfe von netHOST.exe bzw. der Host-Applikation in das Gerät.

4.3.2 Labels und Kommunikationseinstellungen für OPC UA, MQTT bzw. WebServer

In diesem Abschnitt finden Sie eine Übersicht zum prinzipiellen Vorgehen zur Konfiguration des Zugriffs auf die Objektdaten mithilfe von Labels bzw. Kommunikationseinstellungen für OPC UA, MQTT bzw. WebServer.

Vorbereitung und Voraussetzungen

Legen Sie ein Schema an, wie auf die für Ihr Produkt erforderlichen Objektdaten zugegriffen werden soll bzw. auf welche Objekte zugegriffen werden soll.



Wichtig:

1. Ein Objekt oder ein Element darf nur von einem Anwender (Master, OPC-Client etc.) aus geschrieben werden.
2. Ein Objekt oder Element darf von mehreren Anwendern gelesen werden.

Prinzipielle Schritte in netX Studio Engineering Tool

1. Weisen Sie bei der Objektdefinition ein oder mehrere Labels auf Objektebene zu.
 - Weisen Sie dem Objekt im **Object Definitions**-Editor über **Assign label** ein Label zu, siehe Abschnitt *Eigenschaften (Properties) des Objektes anpassen* [► Seite 44] und/oder
2. Konfigurieren Sie die Kommunikationseinstellungen für OPC UA, MQTT bzw. WebServer (**Communication Settings**), siehe Abschnitt *OPC UA, MQTT bzw. WebServer ergänzen* [► Seite 68].
 - OPC UA: Konfigurieren Sie im Editor **OPC UA** den Zugriff über den OPC UA-Server auf Objektdaten.
 - OPC UA: Legen Sie im Editor **OPC UA** fest, welche Objekte über den OPC UA-Server angezeigt werden sollen.
Hinweis: Objekte, für die ein Zugriff vom WebServer aus konfiguriert wird, können gleichzeitig über den OPC UA-Server angezeigt werden.
 - MQTT: Konfigurieren Sie im Editor **MQTT** den Zugriff über MQTT auf Objektdaten.
 - MQTT: Legen Sie im Editor **MQTT** fest, welche Objekte über MQTT angezeigt werden sollen.
Hinweis: Objekte, für die ein Zugriff vom WebServer aus konfiguriert wird, können gleichzeitig über MQTT angezeigt werden.
 - WebServer: Legen Sie im Editor **WebServer** fest, welche Objekte über den WebServer angezeigt werden sollen.
Hinweis: Objekte, für die ein Zugriff vom OPC UA-Server aus oder über MQTT konfiguriert wird, können gleichzeitig über den WebServer angezeigt werden.

4.3.3 Beispiel C100

Das in diesem Handbuch verwendete Beispiel umfasst das Produkt C100 mit einem Zähler und einem Parameter-Objekt.

Objektdaten für das Beispiel C100

Das im Beispiel C100 verwendete Produkt verfügt über die Funktion eines Betriebsstundenzählers und beinhaltet ein Parameter-Objekt.

Objektkennung	Objektname
0x40010001	Counter
0x40020001	Parameter

Tabelle 9: Objektdaten für das Beispiel C100

Applikationsvarianten für das Beispiel C100

Im Beispiel wird das bestehende Produkt C100 mit einem Zähler und einem Parameter-Objekt abgebildet (siehe Tabelle *Funktionsumfang, Beispiel C100* [► Seite 57]). Beim Produkt C100 nimmt der Betriebsstundenzähler ‚Counter‘ die Betriebsstunden auf, während im Objekt ‚Parameter‘ Einstellungen gemacht werden können.

Kommunikationseinstellungen für das Beispiel C100

Im Beispiel werden für das Produkt C100 die Kommunikationseinstellungen für OPC UA, MQTT und WebServer ergänzt.

5 IOT-Konfiguration

Zur IOT-Konfiguration müssen Sie ein Projekt für Ihr Produkt erstellen, darin Objekte erstellen und IOT-Parameter einstellen.

5.1 Objekte erstellen

Um die Geräteeigenschaften in netX Studio Engineering Tool in Objekten abzubilden, sind folgende Hauptschritte notwendig:

- Legen Sie für alle Geräteeigenschaften die Kategorien, die Objekte und die Elemente der Objekte an, so dass alle Geräteeigenschaften in den erstellten Objekten abgebildet werden.
- Definieren Sie den Zugriff auf die Objekte bzw. Elemente und deren Zuordnung zur Applikation und zum Netzwerk.

In den nachfolgenden Abschnitten finden Sie eine schrittweise Beschreibung zur Vorgehensweise.

5.1.1 Kategorie anlegen

- Um eine Kategorie anzulegen, machen Sie im Fenster **Project Explorer** einen Rechtsklick auf **Object Model** > **Object Definitions** > **User**.

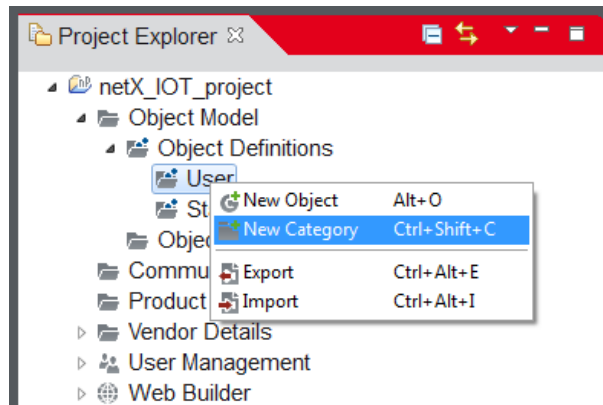


Abbildung 17: Project Explorer - New Category

- Klicken Sie **New Category** an.
- ⇒ Die Dialogbox **Create New Category** erscheint.
- Geben Sie unter **Category name*** (Pflichtfeld) den Namen für die Kategorie ein.

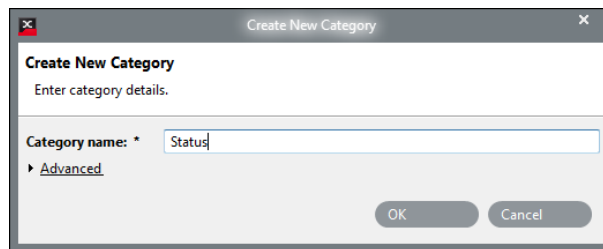


Abbildung 18: Create New Category

- Klicken Sie **OK**.
- ⇒ Der Ordner für die Kategorie erscheint (im Beispiel **Input Data**).

Auf dem gleichen Weg können Sie auch zusätzliche Unterkategorien anlegen.

- Legen Sie ebenfalls Ausgangsdaten an (im Beispiel **Output Data**).

Kategorien mit ID-Bereichen (Advanced)

Beim Anlegen einer Kategorie können Sie einen eigenen ID-Bereich für die Objekte dieser Kategorie definieren, um Objekt-IDs für zukünftige Erweiterungen zu reservieren. netX Studio Engineering Tool unterstützt bei der Verwaltung von Objekt-IDs. Alle Objekt-IDs der Objekte der entsprechenden Kategorie liegen dann innerhalb des definierten Bereichs.

- Klicken Sie **Advanced**.
- ⇒ Die Dialogbox **Create New Category** mit den zusätzlichen Eingabefeldern für den ID-Bereich erscheint.
- Geben Sie unter **Category name*** (Pflichtfeld) den Namen für die Kategorie ein.

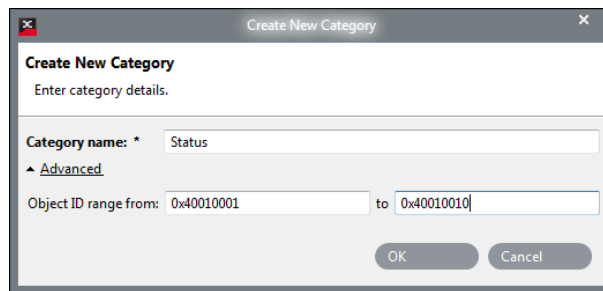


Abbildung 19: Create New Category – Advanced

- Definieren Sie unter **Object ID range from** (Untergrenze) bzw. **to** (Obergrenze) einen ID-Bereich für die Objekte der Kategorie.



Hinweis:

Es gelten die Bereichsgrenzen 0x40000000 bis 0x4FFFFFFF. Die Start-ID und die Ende-ID müssen innerhalb dieser Grenzen liegen. Die Start-ID muss kleiner sein, als die Ende-ID, andernfalls erscheint eine Fehlermeldung.

- Klicken Sie **OK**.
- ⇒ Der Ordner für die Kategorie erscheint (im Beispiel **Input Data**). Zusätzlich erscheint der festgelegte ID-Bereich.



Details zu Kategorien mit ID-Bereichen finden Sie im Abschnitt *ID-Bereich für Kategorie* [▶ Seite 38].

Rename, Remove (Kategorie umbenennen oder löschen)

- Um den Namen einer Kategorie zu ändern oder die Kategorie wieder zu löschen, machen Sie im Fenster **Project Explorer** einen Rechtsklick auf einen Ordner einer Kategorie (im Beispiel **Input data** bzw. **Output data**).
- Klicken Sie **Rename** (Umbenennen) bzw. **Remove** (Entfernen) an.

5.1.1.1 ID-Bereich für Kategorie

Bei der Vollversion von netX Studio Engineering Tool **netX Studio (netPROXY Engineering)** können Sie über die Zuordnung von Objekten zu Kategorien hinaus, Objekte definierten ID-Bereichen zuordnen und damit gruppieren. Dabei gilt:

- Zu jeder Kategorie kann optional ein ID-Bereich festgelegt werden, der für die der Kategorie zugewiesenen Objekte gültig ist. Es gelten die Bereichsgrenzen 0x40000000 bis 0x4FFFFFFF und die Start-ID muss kleiner sein, als die Ende-ID.
- Bei Veränderungen eines ID-Bereichs müssen alle Objekt-IDs der einer Kategorie zugewiesenen Objekte weiterhin innerhalb des ID-Bereichs liegen.
- Zu einer Kategorie mit festgelegtem ID-Bereich können nur Objekte mit Objekt-IDs innerhalb dieses ID-Bereichs zugeordnet werden.
- Bei Objektkategorien mit ID-Bereichen, erhält jedes in einer Kategorie erstellte Objekt direkt eine Objekt-ID aus dem für die Kategorie definierten ID-Bereich.
- Zwei oder mehr Kategorien können gleiche oder sich überlappende ID-Bereiche haben. Die Objekt-ID bleibt eindeutig, da dieser nur einmal verwendet werden darf.
- ID-Bereiche von Unterkategorien stehen nicht im Zusammenhang mit dem ID-Bereich der Hauptkategorie und dürfen außerhalb des ID-Bereichs der Hauptkategorie liegen.
- Ein Objekt kann nur in eine Kategorie verschoben werden, wenn dessen Objekt-ID innerhalb des ID-Bereichs der Zielkategorie liegt.
- Eine Kategorie mit ID-Bereich kann nur importiert werden, wenn sie keine Objekte mit IDs außerhalb des ID-Bereichs der Zielkategorie enthält. Beim Import wird die vorhandene Struktur durch die importierte Struktur ersetzt.

Objekte einem definierten ID-Bereich zuordnen

- Um einen ID-Bereich für eine Kategorie anzulegen, machen Sie im Fenster **Project Explorer** einen Rechtsklick auf **Object Model > Object Definitions > User > [Name der Kategorie]**.

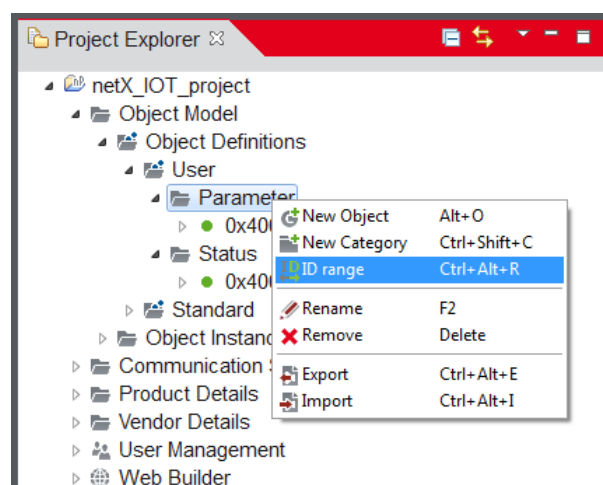


Abbildung 20: Kontextmenü ID range

- Klicken Sie **ID range** an.
- Die Dialogbox **Edit Category Object ID Range** erscheint.

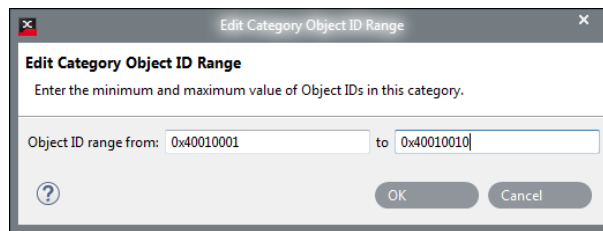


Abbildung 21: Edit Category Object ID Range

- Definieren Sie unter **Object ID range from** (Untergrenze) bzw. **to** (Obergrenze) einen ID-Bereich für die Objekte der Kategorie.



Hinweis:

Es gelten die Bereichsgrenzen 0x40000000 bis 0x4FFFFFFF und die Start-ID muss kleiner sein, als die Ende-ID, andernfalls erscheint eine Fehlermeldung.

- Klicken Sie **OK**.
- Im Ordner für die Kategorie erscheinen (im Beispiel **Input data** bzw. **Output data**) die festgelegten ID-Bereiche (im Bild blau umrahmt).

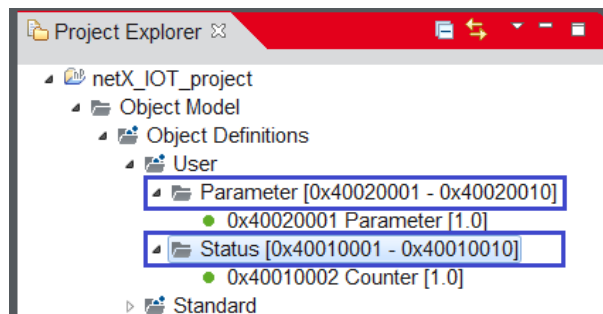


Abbildung 22: Beispiel für sich überlappende ID-Bereiche

Kategorien mit definierten ID-Bereichen importieren oder exportieren

Kategorien mit definierten ID-Bereichen können Sie über **File > Import** bzw. **Export** importieren oder exportieren oder über das Kontextmenü **Import** bzw. **Export** einer Kategorie.

5.1.2 Objekt anlegen

Objekte können Sie direkt im Ordner **Object Definitions** anlegen oder in einem Ordner einer Kategorie oder Unterkategorie.

- Um ein Objekt anzulegen, machen Sie im Fenster **Project Explorer** einen Rechtsklick auf einen Ordner einer Kategorie.

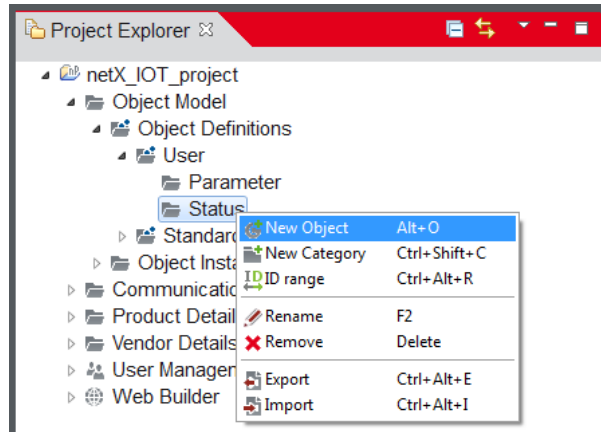


Abbildung 23: Project Explorer - New Object

- Klicken Sie **New Object** an.
- Die Dialogbox **New Object** wird geöffnet.

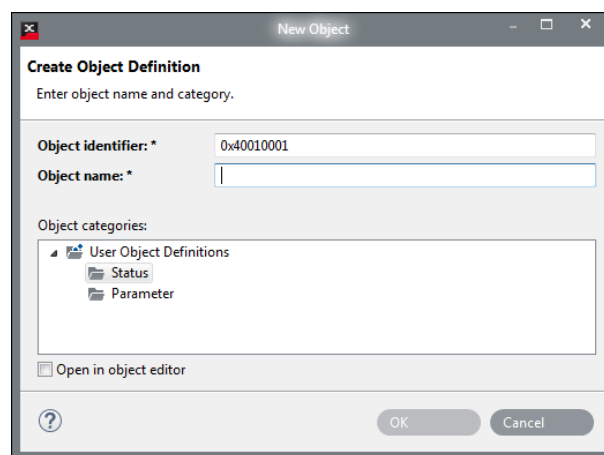


Abbildung 24: Neues Objekt anlegen

Parameter	Bedeutung	Wertebereich /Wert
Object identifier* (Pflichtfeld)	Jedes Objekt ist eindeutig über seine Objektkennung (Objekt-ID) identifizierbar. Die Objektkennung wird als fortlaufende Nummer vergeben. Dabei spielt es keine Rolle, ob ein Objekt über den Dialog New Object erstellt wurde oder durch Kopieren. Wird ein Objekt innerhalb einer Kategorie mit festgelegtem ID-Bereich erstellt, erhält jedes in dieser Kategorie erstellte Objekt automatisch eine Objekt-ID aus dem für die Kategorie definierten ID-Bereich. Bei manueller Vergabe können nur Objekt-IDs innerhalb des festgelegten ID-Bereichs gewählt werden. Siehe auch Abschnitt <i>ID-Bereich für Kategorie</i> ► Seite 38].	0x40000000 ... 0xFFFFFFFF
Object name* (Pflichtfeld)	Objektname	3 ... 50 Zeichen
Object categories (optional)	Objekte können zu einer Kategorie zugeordnet und so zusammengefasst werden.	

Tabelle 10: New Object – Parameter

**Wichtig:**

Eine Anpassung der Objektkennung (**Object identifier***) ist nur im Fenster **New Object** möglich. Die Objektkennung (**Object identifier***) kann nachträglich nicht mehr geändert werden.

Die im Fenster **New Object** vorgenommene Zuordnung eines Objektes zu einer Kategorie (**Object categories**) kann nachträglich nicht mehr geändert werden.

Im Beispiel C100 wurde die Festlegung getroffen: Objektkennungen für Status (counter): 0x4001xxxx, Objektkennungen für Parameter: 0x4002xxxx.

- Geben Sie den Objektidentifizier ein, (im Beispiel für das Objekt ‚Counter‘ 0x40010001 und für das Objekt ‚Parameter‘ 0x40020001).
- Ergänzen Sie den Objektnamen im Pflichtfeld **Object name***.
- Wählen Sie unter **Object categories** die Objektkategorie aus.

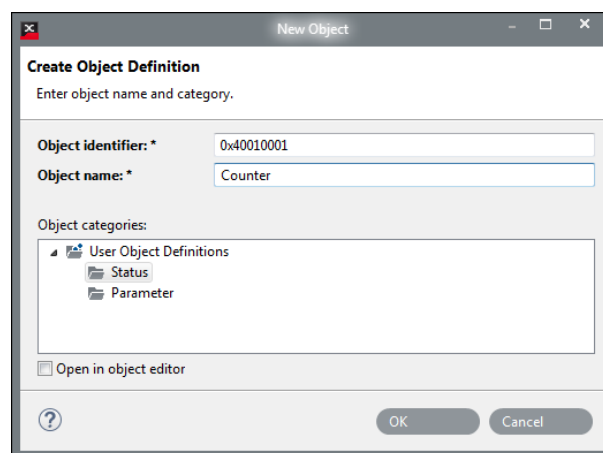


Abbildung 25: Objektkennung anpassen, Objektname eingeben, Objektkategorie auswählen

- Klicken Sie **OK** an.

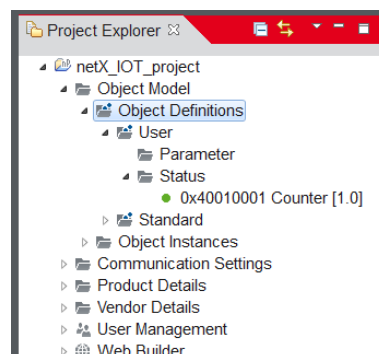


Abbildung 26: Beispielobjekt '0x40010000 Sensor – simple [1.0]'

- ⇒ Sie haben ein Objekt angelegt (im Beispiel **0x40010001 Counter**).

Angaben dazu, wie Sie ein Objekt durch Kopieren und Einfügen erzeugen bzw. wie Sie ein Objekt umbenennen oder löschen, finden Sie im Abschnitt *Objekt kopieren und einfügen, umbenennen, löschen* [► Seite 42]. Oft ist es sinnvoll zuerst die Elemente eines Objektes anzulegen und das Objekt erst anschließend zu kopieren.

5.1.2.1 Objekt kopieren und einfügen, umbenennen, löschen

Copy (Objekt kopieren)

- Um ein Objekt zu kopieren, machen Sie im Fenster **Project Explorer** einen Rechtsklick auf einen Objektordner (hier im Beispiel **0x40010001 Counter**).

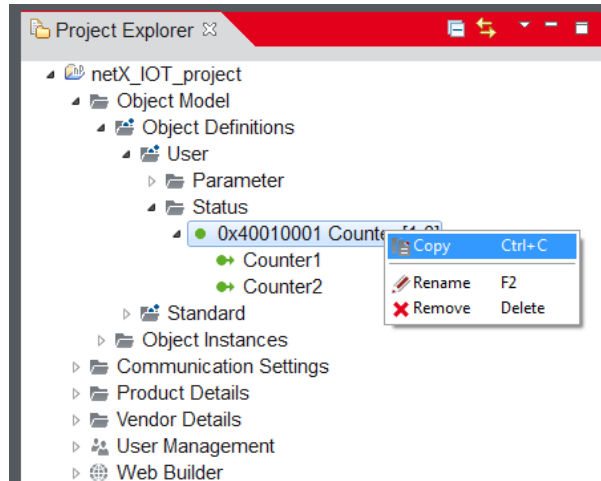


Abbildung 27: Copy (Objekt kopieren)

- Klicken Sie **Copy** (Kopieren).
- ⇒ Das Objekt wird in den Zwischenspeicher kopiert.

Paste (Objekt einfügen)

- Um ein Objekt einzufügen, machen Sie auf den Ordner in welches Sie das Objekt einfügen möchten einen Rechtsklick und klicken Sie **Paste** (Einfügen).

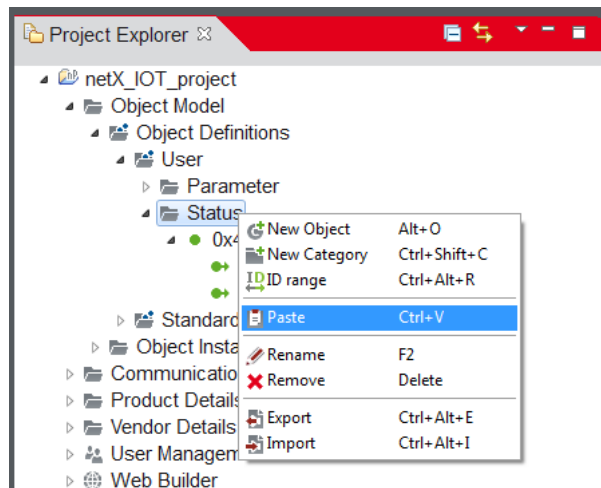


Abbildung 28: Paste (Objekt einfügen)

- ⇒ Das Objekt wird an dieser Stelle eingefügt.

netX Studio Engineering Tool erlaubt es Ihnen Objekte zu kopieren, einschließlich der darin erstellten Elemente. Legen Sie gegebenenfalls zunächst ein Objekt mit seinen Elementen an und generieren Sie daraus weitere Objektvarianten.

**Wichtig:**

Beachten Sie, dass netX Studio Engineering Tool die Objektkennungen **Object identifier*** beim Erstellen von Objekten durch **Copy** und **Paste** immer mit einer fortlaufenden Nummer vergibt, beginnend bei 0x40000000. Die Objektkennung kann nicht geändert werden.

Soll die Objektkennung nach zusätzlichen Regeln vergeben werden (systematische Vergabe von Kennungen für Objekte bestimmter Kategorien, z. B.: 0x4001XXXX für lesbare Objekte bzw. 0x4002XXXX für schreibbare Objekte), müssen Sie jedes Objekt manuell über den Befehl **New Object** und das Fenster **New Object** erstellen.

Rename (Objekt umbenennen)

- Um den Namen eines Objektes zu ändern, machen Sie im Fenster **Project Explorer** einen Rechtsklick auf einen Objektordner (hier im Beispiel **0x40010001 Counter**).
- Klicken Sie **Rename** (Umbenennen) an.
- Das Fenster **Rename** erscheint.
- Geben Sie im Feld **Enter new Name** den neuen Objektnamen ein und klicken Sie **OK**.
- ⇒ Der Objektordner erscheint mit dem geänderten Namen.

Remove (Objekt löschen)

- Um ein Objekt zu löschen, machen Sie im Fenster **Project Explorer** einen Rechtsklick auf einen Objektordner (hier im Beispiel **0x40010001 Counter**).
- Klicken Sie **Remove** (Entfernen) an.
- Bestätigen Sie die Abfrage, ob Sie das Objekt löschen wollen mit **OK**.
- ⇒ Das Objekt wird gelöscht.

5.1.3 Eigenschaften (Properties) des Objektes anpassen

- Machen Sie einen Doppelklick auf **Object Model > Object Definitions > User** und den Objektordner (im Beispiel **0x40010001 Counter [1.0]**).
- Im Editor erscheint die Registerkarte für das Objekt mit den Unterbereichen **Properties** und **Elements**.

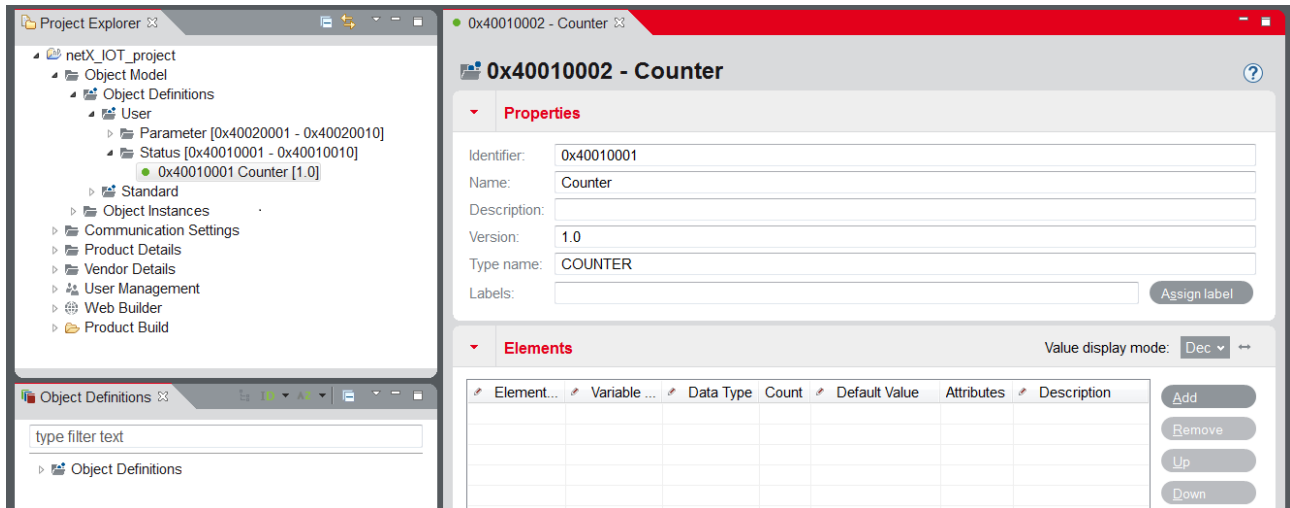


Abbildung 29: Editor mit Properties und Elements - Beispiel

Die unter **Properties** angezeigten Werte für **Identifier**, **Name**, **Version** und **Type name** werden automatisch bzw. aus den im Fenster **New Object** vorgegebenen Daten generiert. Die Objektkennung **Identifier** kann hier nicht mehr verändert werden.

- Passen Sie die Parameter unter **Properties** an.

Labels zuweisen

Über **Assign label** können Sie beliebige Labels selbst erstellen und einem Objekt ein oder mehrere Labels zuweisen. Im System sind die Labels **OPC UA**, **MQTT** und/oder **www** bereits vorhanden.

Die Zuweisung der Labels ist unabhängig davon, ob ein Zugriff auf das Objekt über Real-Time-Ethernet (zyklische und/oder azyklische Kommunikation) besteht oder nicht (siehe auch Abschnitt *Beispiele* [► Seite 28]).

Mithilfe der Zuweisung der Labels und in einem weiteren Schritt mit deren Auswahl (siehe Abschnitt *OPC UA, MQTT bzw. WebServer ergänzen* [► Seite 68]), können Sie den/die zu verwendenden Kommunikationswege OPC UA, MQTT bzw. WebServer festlegen.

- Um ein bereits vorhandenes oder selbst erstelltes Labels zuzuweisen, im **Object Definitions**-Editor **Assign label** anklicken.
- Im Fenster **Assign label** auf der linken Seite **OPC UA**, **MQTT** und/oder **www** (oder ein selbst erstelltes Label) auswählen.
- Auf den Pfeil ► klicken oder einen Doppelklick auf das Label machen, um das Label in die Liste **Assigned labels** zu verschieben.

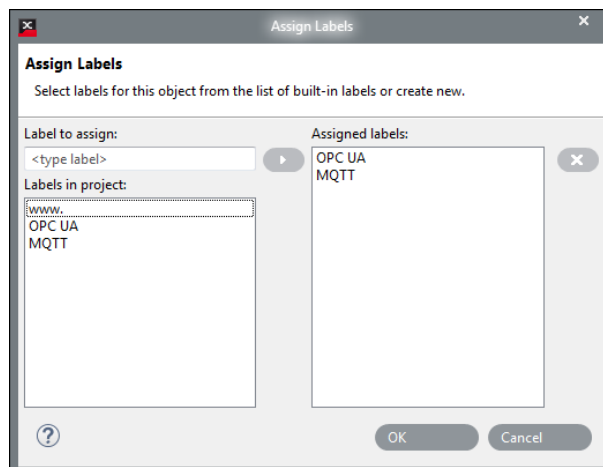


Abbildung 30: Label auswählen für Zugriff über OPC UA, MQTT oder www (Beispielauswahl www)

- Auf diese Weise können Sie weitere Labels auswählen und nach rechts verschieben, um sie dem gewählten Objekt zuzuweisen.
- **OK** klicken.
- ⇒ Die ausgewählten Labels (beispielsweise **OPC UA**, **MQTT** und **www**) werden unter **Properties** > **Labels** angezeigt.

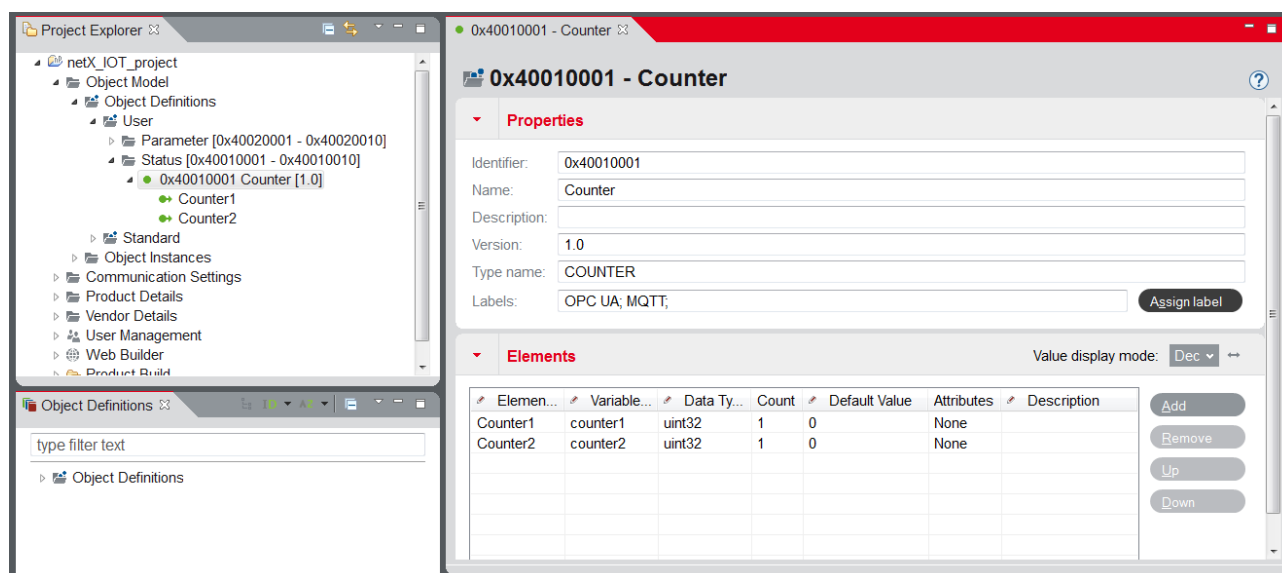


Abbildung 31: Object Definitions-Editor – ausgewählte Labels (Beispielauswahl OPC UA, MQTT und www)


Label	Bedeutung
[eigene Labels]	<p>Vom Anwender selbst zu definierende Labels. Solche Labels dienen dazu, den Zugriff vom OPC UA-Server aus, über MQTT bzw. vom WebServer auf Objekte zu konfigurieren.</p> <p>Das selbst definierte Label muss bei der Objektdefinition zugewiesen werden und in den Kommunikationseinstellungen für OPC UA, MQTT bzw. www ausgewählt werden.</p> <p>Alle Objekte mit einem selbst definierten Label können bei Bedarf zusätzlich über den OPC UA-Server, MQTT und über den WebServer angezeigt werden.</p>
OPC UA	<p>Vom System vordefiniertes Label. Das Label OPC UA dient dazu, den Zugriff vom OPC UA-Server aus auf Objekte zu konfigurieren.</p> <p>Das Label OPC UA muss bei der Objektdefinition zugewiesen werden und in den Kommunikationseinstellungen für OPC UA ausgewählt werden.</p> <p>Objekte mit dem Label OPC UA können bei Bedarf zusätzlich über MQTT und über den WebServer angezeigt werden.</p>
MQTT	<p>Vom System vordefiniertes Label. Das Label MQTT dient dazu, den Zugriff über MQTT auf Objekte zu konfigurieren.</p> <p>Das Label MQTT muss bei der Objektdefinition zugewiesen werden und in den Kommunikationseinstellungen für MQTT ausgewählt werden.</p> <p>Objekte mit dem Label MQTT können bei Bedarf zusätzlich über den OPC UA-Server und den WebServer angezeigt werden.</p>
www	<p>Vom System vordefiniertes Label. Das Label www dient dazu, den Zugriff vom WebServer aus auf Objekte zu konfigurieren.</p> <p>Das Label www muss bei der Objektdefinition zugewiesen werden und in den Kommunikationseinstellungen für WebServer ausgewählt werden.</p> <p>Objekte mit dem Label www können bei Bedarf zusätzlich über MQTT und über den OPC UA-Server angezeigt werden.</p>

Tabelle 11: Labels für OPC UA, MQTT, www bzw. eigene Labels



Die Konfiguration für den Zugriff auf die Objektdaten durch den OPC UA-Server, MQTT oder den WebServer erfordert die Auswahl der bei der Objektdefinition zugewiesenen Labels in den Kommunikationseinstellungen (**Communication Settings**), jeweils im Editor **OPC UA > Actions**, **MQTT > Connections** oder **WebServer**, sowie weitere Einstellungen (siehe Abschnitt *OPC UA, MQTT bzw. WebServer ergänzen* ► Seite 68)].

Eigenes Label erstellen

- Um ein eigenes Label zu erstellen, im **Object Definitions-Editor** **Assign label** anklicken.
- Im Fenster **Assign label** auf der linken Seite im oberen Feld einen Namen für ein eigenes Label eingeben (Beispielname „test“).
- Auf den Pfeil  klicken oder einen Doppelklick auf das Label machen, um das Label in die Liste **Assigned labels** zu verschieben.

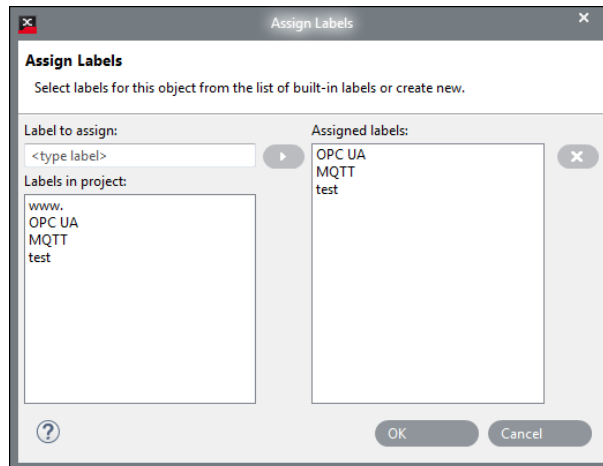


Abbildung 32: Eigenes Label erstellen (Beispielname „test“)

- Zusätzlich können Sie weitere Labels auswählen und nach rechts verschieben, um sie einem Objekt zuzuweisen.
- **OK** klicken.
- ⇒ Das neu erstellte Label (Beispielname „test“) wird unter **Properties > Labels** neben den anderen Labels angezeigt.

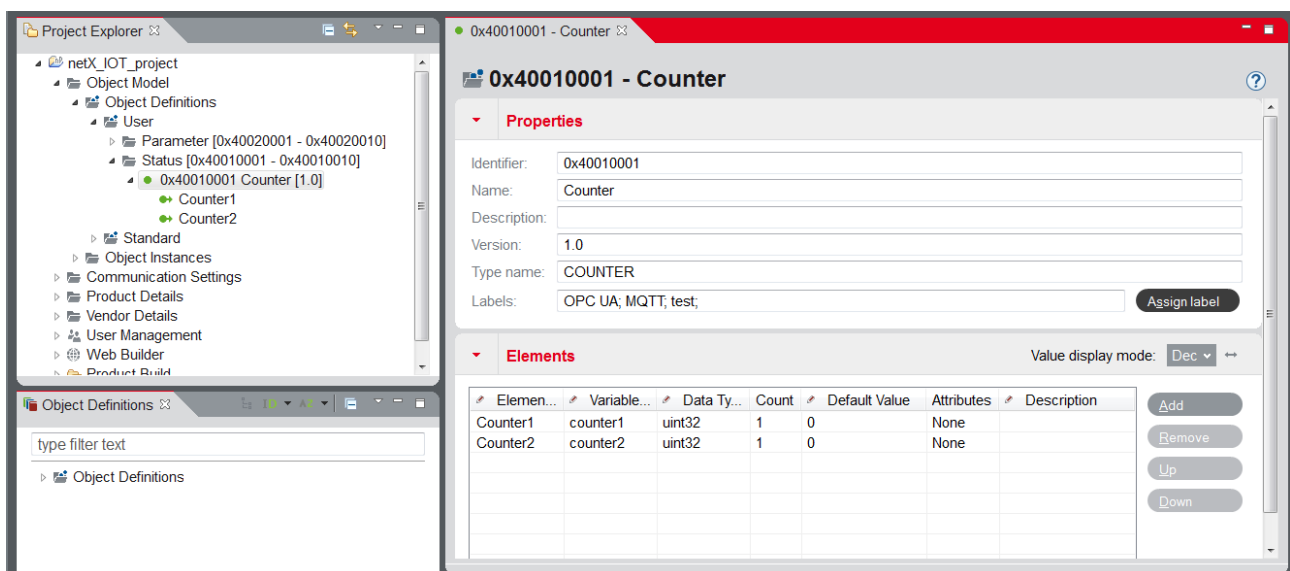



Abbildung 33: Beispiel zugeordnete Labels

Label löschen

- Sie können ein Label wieder löschen, indem Sie das Label in der Liste rechts anklicken und auf das Kreuz  klicken.

5.1.4 Elemente (Elements) anlegen

Elemente für das Beispiel C100

Im Beispiel C100 sind für die Zähler bzw. die Parameter (Objekte 0x40010001 bzw. 0x40020001) die Elemente

- **Counter** (Betriebsstundenzahl) und
- **Parameter** (Einstellungen) vorgesehen.

Vorgehen

- Klicken Sie unter **Elements** auf **Add**.
- Ein neues Element wird ergänzt.

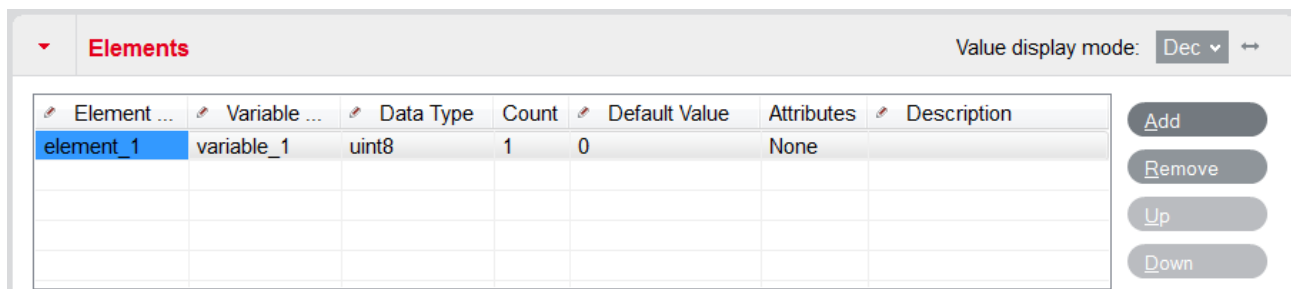


Abbildung 34: Object Definitions-Editor - Element anlegen (Beispielobjekt „0x40010001 Counter“)

Eine Beschreibung zur Bedienung der Tabelle **Elements** finden Sie im Abschnitt *Bedienelemente des Object Definitions-Editors* [► Seite 50].

Parameter	Bedeutung	Wert/Wertebereich
Element Name*	Name des Elements, der auch unter Object Instances bei dem Element der Anwendungsvariante erscheint.	
Variable Name	Hier können Sie für jedes Element einen zusätzlichen Namen hinterlegen (erscheint nicht unter Object Instances). Der Variable Name wird in der Header-Datei objects.h zur Definition der Objektstruktur in der Programmiersprache C verwendet. Empfehlung: Verwenden Sie eine C-konforme Notation, z. B. bValue.	
Data Type*	Datentyp des Elements Beispiel: Auswahl uint8	boolean, uint8, uint16, uint32, uint64, int8, int16, int32, int64, string, real32, real64
Count	Maximale Anzahl pro Datentyp: Für die Datentypen boolean, uint8, uint16, uint32, uint64, int8, int16, int32, int64, real32, real64 wird ein einzelner Datentyp verwendet, d. h., es gilt immer Count = 1. Für den Datentyp string kann Count = 1 ... 1540 sein.	Für boolean, uint8, uint16, uint32, uint64, int8, int16, int32, int64, real32, real64: Count = 1. Für string: Count = 1 ... 1540 Siehe auch Tabelle <i>Objektelement-Datentypen, Größe, Anzahl, Grenzen</i> [► Seite 112]
Default Value*	Der vom Anwender einzustellende Default-Wert legt den Initialwert für den Datenbereich fest. Die Elemente der Objekte müssen einen Initialwert bekommen. Nach einem Geräte-Reset gilt zunächst der Default-Wert. Wird im Beispiel eines Temperatursensors, der in den vier Betriebsarten 0 bis 3 arbeiten kann, ein Default-Wert von 3 eingestellt, bedeutet das, dass der Sensor beim Gerätestart in der Betriebsart 3 arbeitet.	Der Default-Wert ist abhängig vom Anwendungsfall.

Parameter	Bedeutung	Wert/Wertebereich
Attributes*	Das Kommunikationsattribut zur Zuordnung der Elementdaten zur Applikation bzw. zur zyklischen Netzwerkkommunikation ist standardmäßig auf None eingestellt. Bei der IOT-Konfiguration werden die E/A-Daten nicht über netPROXY abgewickelt und der Anwender kann hier nichts auswählen.	None
Description*	Hier können Sie eine zusätzliche Beschreibung für das Element ergänzen.	3 ... 50 Zeichen

Tabelle 12: Parameter der Elemente (Object Definitions)

**Hinweis:**

*Beachten Sie, dass der hier eingestellte Name, Datentyp, Default-Wert, sowie die Beschreibung als Vorgaben verwendet werden, wenn das Objekt in einer Applikationsvariante konfiguriert wird. Eine Änderung des Namens, Datentyps, Default-Wertes bzw. der Beschreibung bei der Objektdefinition wirkt sich auf alle Objektinstanzen aus. Im **Object Instances**-Editor in der Spalte **Element Value** vorgenommene Anpassungen für den Default-Wert werden durch nachträgliche Änderungen unter **Object Definitions > User** wieder überschrieben. Weitere Angaben finden Sie in der Tabelle *Parameter der Elemente (Object Instances)* [► Seite 62].

Element Name bzw. Variable Name

- Machen Sie einen Doppelklick in die Tabellenzelle für **Element Name** bzw. **Variable Name**, um den Text editieren zu können.
- Passen Sie die Namen für das Element und die Variable an.

Data Type

- Machen Sie einen Doppelklick in die Tabellenzelle für **Data Type** oder klicken Sie in die Tabellenzelle und drücken Sie F2.
- Die Liste mit den Datentypen erscheint.
- Wählen Sie den Datentyp aus.

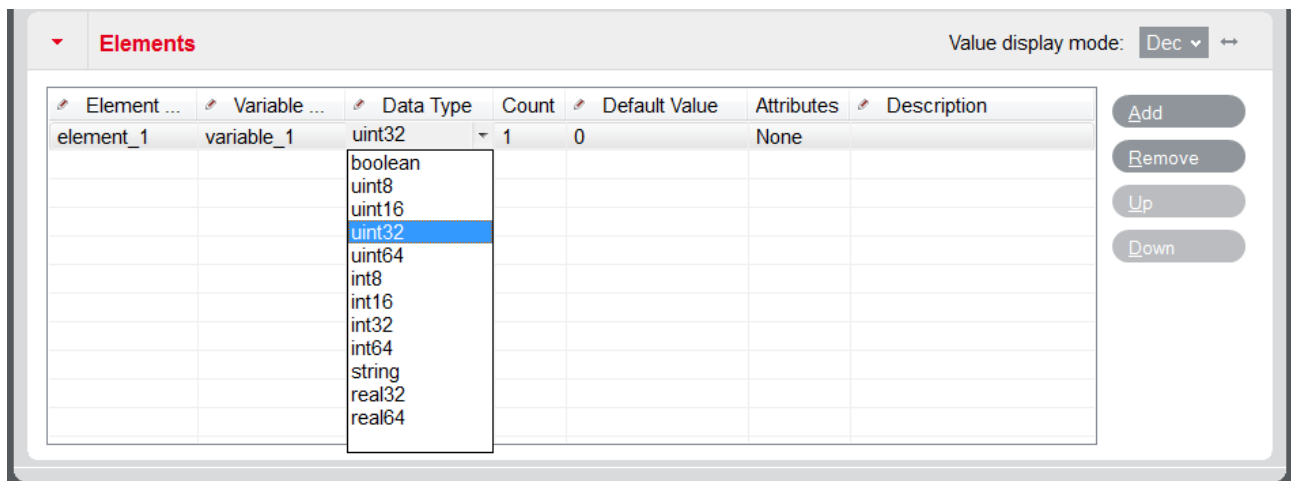


Abbildung 35: Datentyp auswählen

Count

Für den Datentyp „string“:

- Machen Sie einen Doppelklick in die Tabellenzelle für **Count**.
- Legen Sie die Anzahl der Datentypen fest.

Default Value



Hinweis:

Beachten Sie, dass eine Änderung des Default-Werts bei der Objektdefinition sich auf alle Objektinstanzen auswirkt. Werte, die bei der Konfiguration der Applikationsvariante in der Spalte **Element Value*** eingestellt wurden, werden dadurch wieder überschrieben.

*Siehe der Tabelle *Parameter der Elemente (Object Instances)* [► Seite 62].

- Um den Default-Value einzustellen, machen Sie einen Doppelklick in die Tabellenzelle für **Default Value**.
- Legen Sie den Default-Wert fest.

5.1.4.1 Bedienelemente des Object Definitions-Editors

Im **Object Definitions**-Editor können Sie in der Tabelle **Elements** die nachfolgend beschriebenen Aktionen ausführen:




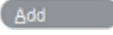
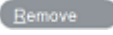


Aktion		Bedeutung
Value display mode: 		Über Value display mode können Sie eine dezimale oder hexadezimale Darstellung der Daten wählen.
	Auto Resize	Über Auto Resize können Sie die Breite der Tabellenspalten auf die Standardeinstellung zurücksetzen.
	Help	Über Help können Sie die Online-Hilfe aufrufen.
		Über Add können Sie ein Element hinzufügen.
		Über Remove können Sie ein Element entfernen.
		Über Up können Sie ein Element nach oben verschieben.
		Über Down können Sie ein Element nach unten verschieben.

Tabelle 13: Bedienelemente des Object Definitions-Editors

5.1.4.2 Beispiel für IOT-Kommunikation über OPC UA bzw. MQTT

Beim Beispiel des Betriebsstundenzählers wird die Betriebsstundenzahl ,counter1', ,counter2' sowie die Geräteeinstellung ,parameter1', ,parameter2' verwendet.

In der Spalte **Attribute** wird für alle Elemente standardmäßig der Listeneintrag **None** voreingestellt. Beschreibungen zur Datenzuordnung über Labels finden Sie im Abschnitt *Kommunikationseinstellungen vornehmen* [▶ Seite 65].

Beispielobjekte ,Counter' und ,Parameter'

Die folgenden Abbildungen zeigen die Beispielobjekte **0x40010001 Counter** und **0x40020001 Parameter** und deren Elemente:

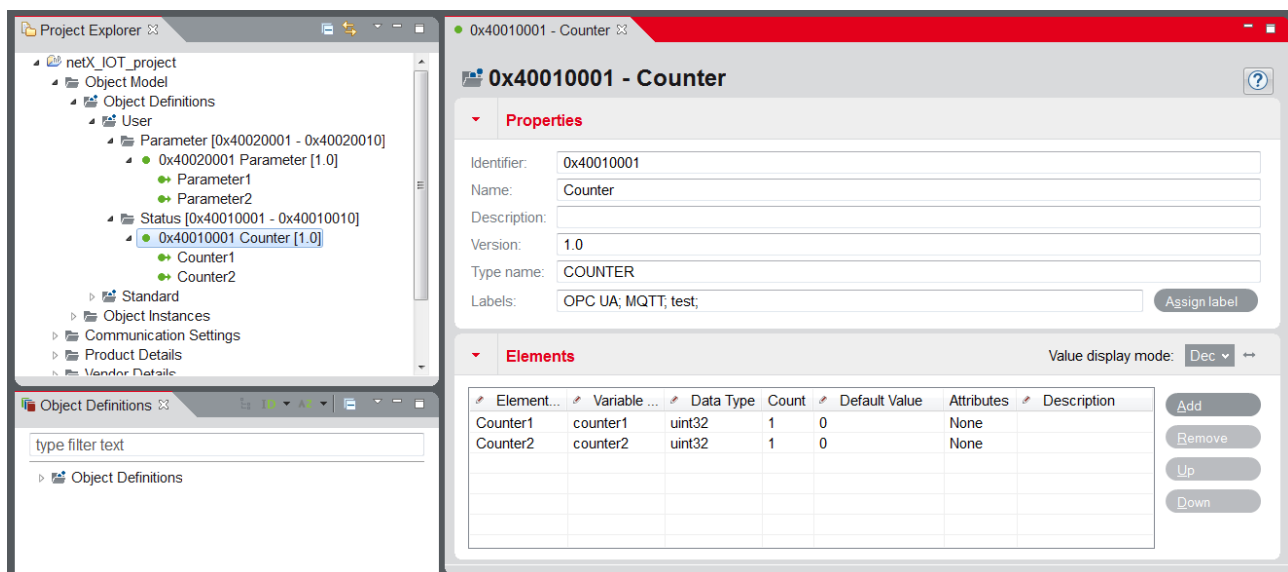


Abbildung 36: Beispielobjekt ,0x40010001 Counter [1.0]', Elemente ,Counter1' und ,Counter2'

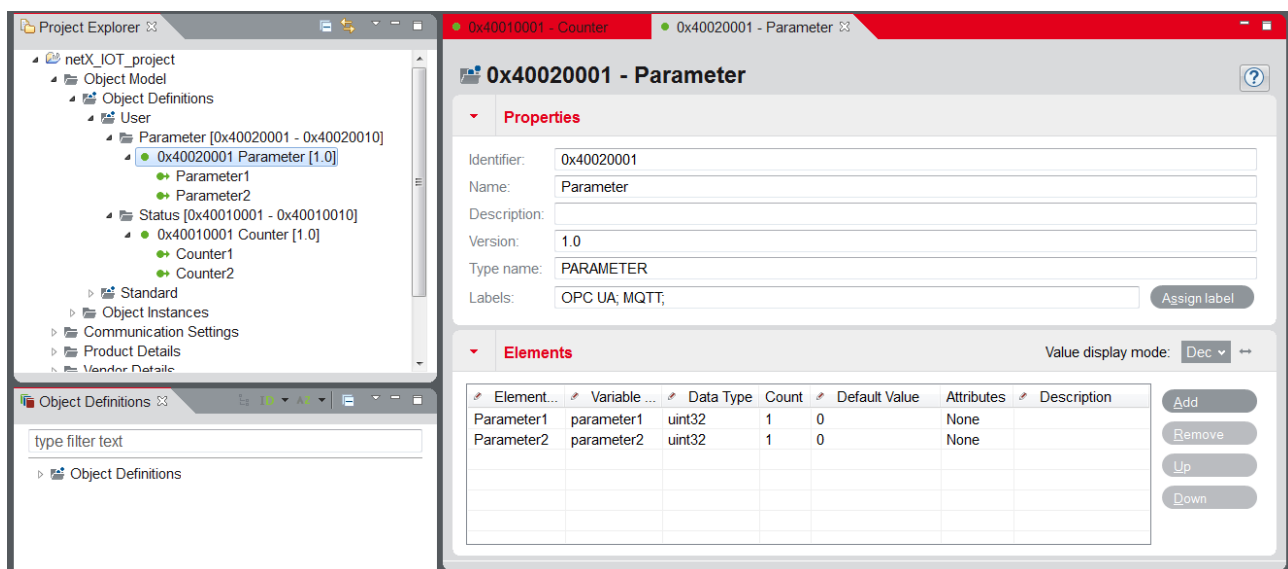


Abbildung 37: Beispielobjekt ,0x40020001 Parameter [1.0]', Elemente ,Parameter1' und ,Parameter2'

Die Elemente zur Abbildung der Geräteeigenschaften in den Objekten sind damit definiert.

5.2 Standardobjekte verwenden

netX Studio Engineering Tool bietet auf der Ebene der Objektdefinition die Möglichkeit vordefinierte Standardobjekte zu verwenden. Wie Sie Standardobjekte verwenden können ist in den nachfolgenden Abschnitten am Beispiel zum Generic Device beschrieben.



Weitere Informationen zu vordefinierten Standardobjekten erhalten Sie im Technischen Referenzhandbuch netPROXY, Objekte und Packages, DOC160204TRXXDE [2].

5.2.1 Standardobjekte ergänzen und anzeigen

- Machen Sie bei der Objektdefinition einen Rechtsklick auf den Ordner **Standard**.
- Klicken Sie **Add** an.

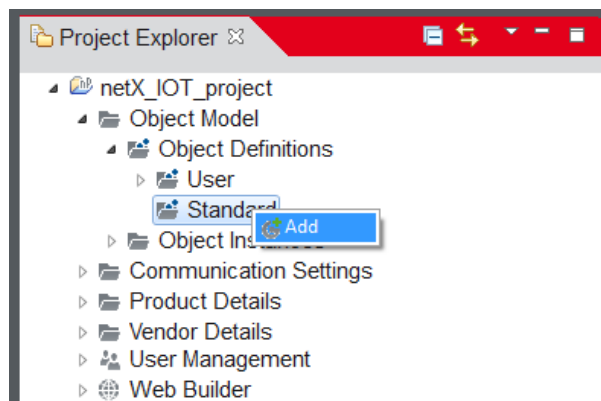


Abbildung 38: Standardobjekte verwenden

- Das Fenster Standard Objects erscheint:
- Klicken Sie die Kategorie für die Standardobjekte an und **OK**.

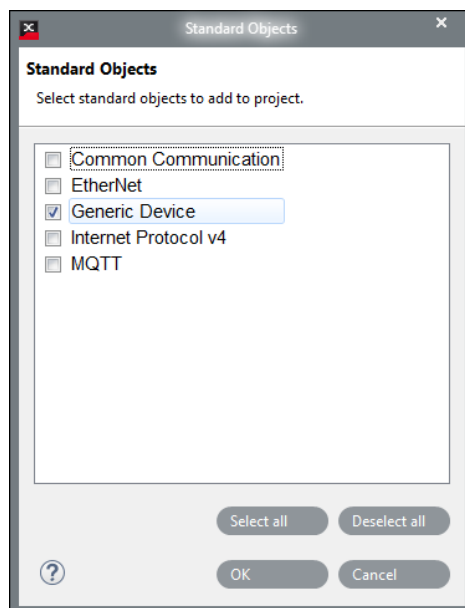


Abbildung 39: Auswahl für Standardobjekte - Beispiel Generic Device

- Im Fenster Projec Explorer erscheint die Objektkategorie Generic Device mit den darin enthaltenen Standardobjekten.

- Um ein Objekt im Editor rechts anzuzeigen, machen Sie einen Doppelklick auf den Objektordner.
- Das entsprechende Standardobjekt wird angezeigt:

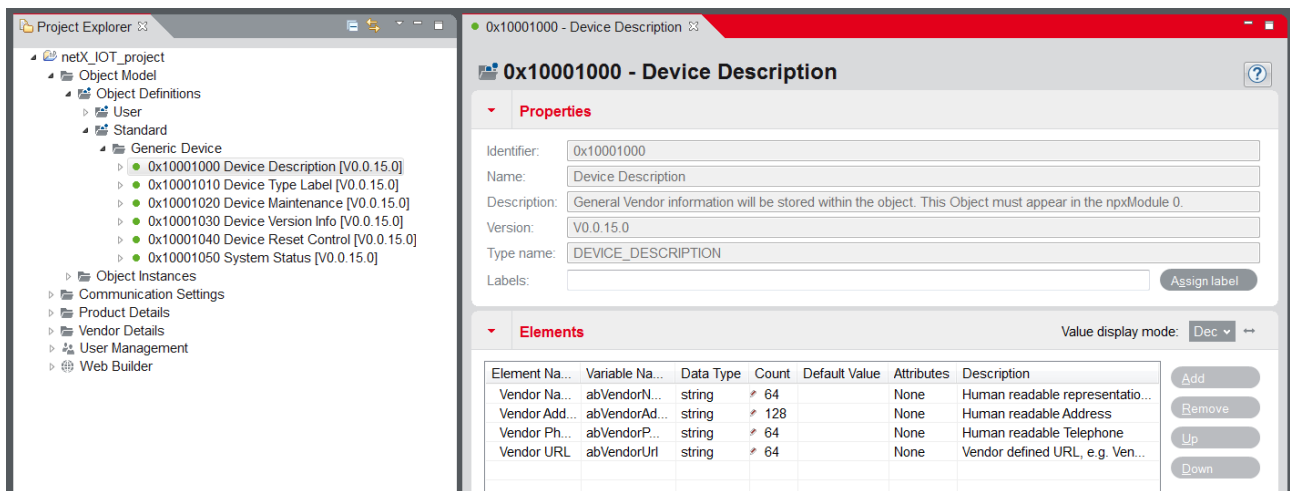


Abbildung 40: Objekteditor 0x10001000 - Device Description

5.2.2 Standardobjekte kopieren oder löschen

Standardobjekte kopieren

Standardobjekte können Sie kopieren und unter **Object Definitions > User** einfügen.

- Machen Sie einen Rechtsklick auf ein Standardobjekt.
- Klicken Sie **Copy** an.

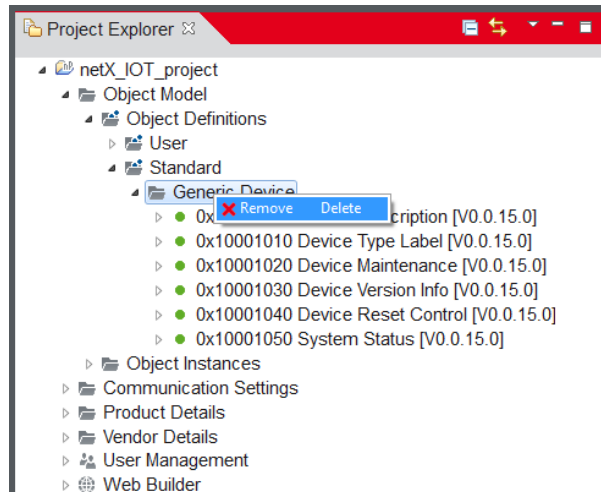


Abbildung 41: Standardobjekte kopieren (Beispiel)

- Machen Sie einen Rechtsklick auf den Ordner **Object Definitions > User** bzw. einen Unterordner davon.
- Klicken Sie **Paste** an.
- Das Objekt wird eingefügt.

Kategorie mit Standardobjekten löschen

Standardobjekte können Sie nur löschen, indem Sie die gesamte Objektkategorie löschen.

- Machen Sie einen Rechtsklick auf den Kategorieordner (im Beispiel Generic Devices).
- Klicken Sie **Remove** an.

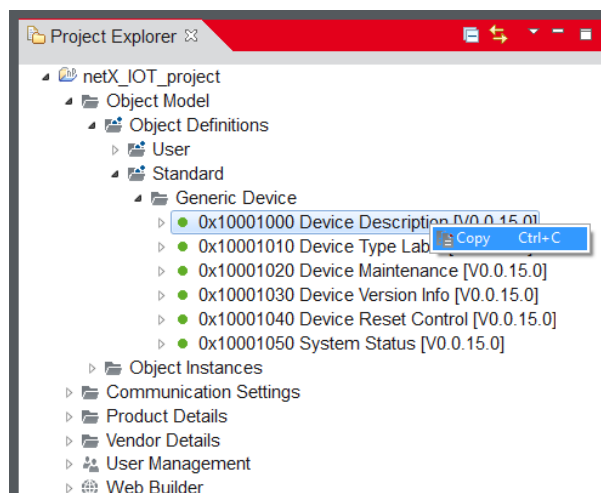


Abbildung 42: Kategorie mit Standardobjekten löschen (Beispiel)

➤ Der gewählte Ordner wird gelöscht.


5.2.3 Standardobjekte zu OPC UA, MQTT und WebServer zuordnen

Eine Zuordnung der Standardobjekte zu OPC UA, MQTT bzw. WebServer ist standardmäßig nicht eingerichtet und muss manuell durch die Zuweisung von Labels konfiguriert werden.



Im Abschnitt *Beispiele* [► Seite 28] finden Sie Erklärungen zur Kommunikation über OPC UA, MQTT bzw. WebServer und im Abschnitt *Eigenschaften (Properties) des Objektes anpassen* [► Seite 44] finden Sie weitere Details zur Zuweisung der Labels.

Gehen Sie wie folgt vor:

- Um ein bereits vorhandenes oder selbst erstelltes Label zuzuweisen, im **Object Definitions-Editor Assign label** anklicken.
- Im Fenster **Assign label** auf der linken Seite **OPC UA**, **MQTT** und/oder **www** (oder ein selbst erstelltes Label) auswählen.
- Auf den Pfeil  klicken oder einen Doppelklick auf das Label machen, um das Label in die Liste der ausgewählten Labels zu verschieben.

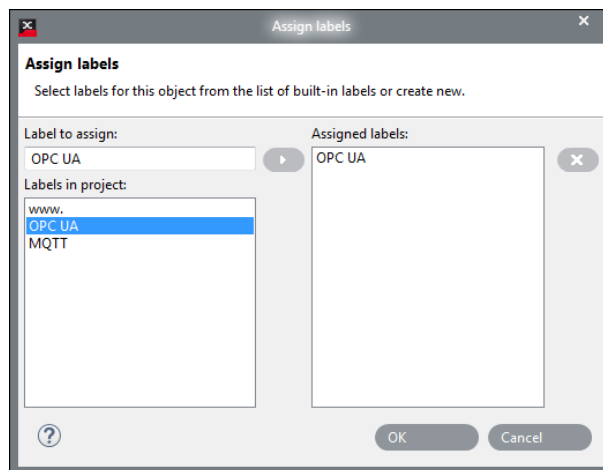


Abbildung 43: Label auswählen für Zugriff über OPC UA, MQTT oder WebServer (Beispielauswahl OPC UA)

- Auf diese Weise können Sie weitere Labels auswählen und nach rechts verschieben, um es einem Objekt zuzuweisen.
- **OK** klicken.
- ⇒ Die ausgewählten Labels werden unter **Properties > Labels** angezeigt.

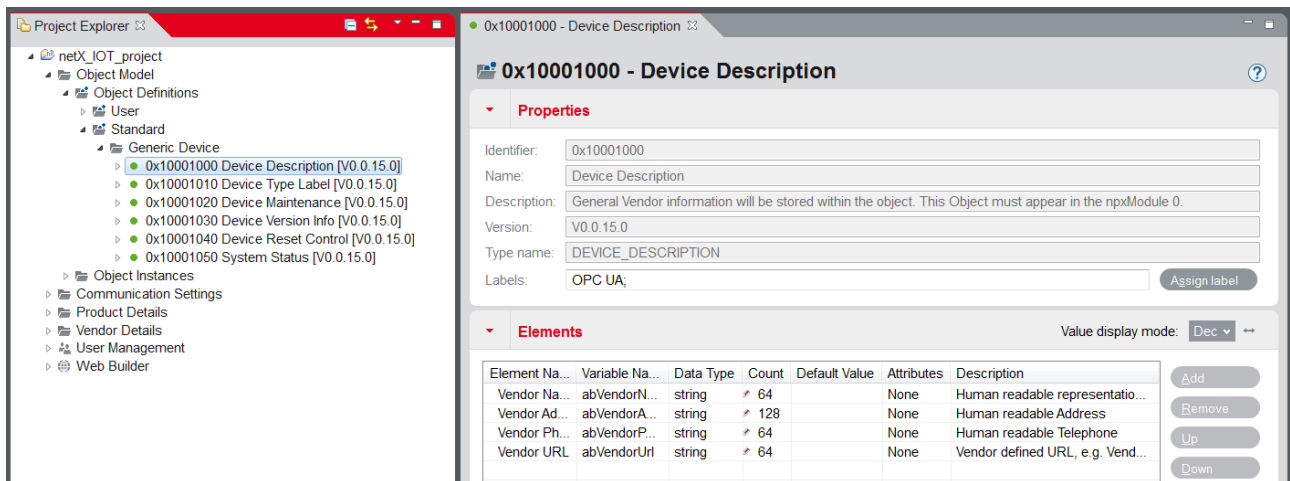


Abbildung 44: Object Definitions-Editor – ausgewählte Labels (Beispielauswahl OPC UA)

5.3 Applikationsvariante konfigurieren (Funktionsumfang)

Die mit IOT-Funktionen ausgestatteten Produkte bzw. Produktvarianten werden aus den in netX Studio Engineering Tool erstellten Objekten als Objektinstanzen zusammengestellt bzw. konfiguriert.

Dazu sind folgende Hauptschritte notwendig:

- Legen Sie den Funktionsumfang der Produkte bzw. Produktvarianten fest.
- Konfigurieren Sie auf Basis der definierten Objekte geeignete Applikationsvarianten mit dem Funktionsumfang der Produktvarianten.
- Passen Sie gegebenenfalls den Initialwert (Default-Wert) für alle in den Applikationsvarianten verwendeten Elemente an.

In den nachfolgenden Abschnitten finden Sie eine schrittweise Beschreibung zur Vorgehensweise.

5.3.1 Funktionsumfang festlegen

Als Vorbereitung für die Programmierung Ihrer Applikation bzw. der Gerätefunktionen bzw. der IOT-Funktionen, müssen Sie den Funktionsumfang festlegen, welcher durch die Ergänzung der IOT-Funktionen erreicht werden soll.

Im Beispiel C100 besteht die Objektdefinition aus dem Objekt für einen Betriebsstundenzähler ‚Counter‘, sowie einem Objekt für Geräteeinstellungen ‚Parameter‘.

Als Funktionsumfang der Applikationsvariante C100 werden Counter bzw. Parameter definiert.

Applikations-variante	Funktions-umfang	Instanz ID	Objekt-nummer	Objektname 1 (für Instanz)	Objektname 2 (für Objekt)
Application C100	2 Counter	0:	0x4001001	Counter	Counter
	2 Parameter	0:	0x4002001	Parameter	Parameter

Tabelle 14: Funktionsumfang, Beispiel C100

5.3.2 Applikationsvariante konfigurieren

Die für Ihre Produkte erforderlichen Applikationsvarianten und deren Funktionsumfang können Sie erstellen, indem Sie die unter **Object Definitions** definierten Objekte und die darin abgebildeten Geräteeigenschaften in für Ihre Produkte geeigneten Applikationsvarianten zusammenstellen und konfigurieren.

Um eine Applikationsvariante zu konfigurieren gehen Sie wie folgt vor:

- Erstellen Sie die Applikationsvariante.
- Konfigurieren Sie die Objekte für die Applikationsvariante, indem Sie die Objekte hinzufügen.
- Zur Konfiguration der Parameter der Objekte nehmen Sie folgende Anpassungen vor:
 - Registerkarte Object Instances:
 - Objektinstanznamen*
 - Default-Werte der Elemente**
 - Elementwert (Element Value)**
 - Registerkarte Details:
 - Unit**
 - Min Value und Max Value**

*auf Objektebene, **auf Elementebene

Weitere Applikationsvarianten können Sie durch Klonen der ersten Applikationsvariante erstellen oder neue Applikationsvarianten ergänzen.

Das für dieses Handbuch verwendete Beispiel ist in Tabelle *Funktionsumfang, Beispiel C100* [► Seite 57] aufgeführt.

5.3.2.1 Applikationsvariante erstellen, Objekte ergänzen

Um die erste Applikationsvarianten zu erstellen und die Objekte für dafür zu konfigurieren, gehen Sie wie folgt vor:

Applikationsvariante erstellen

- Machen Sie im Fenster **Project Explorer** einen Rechtsklick auf den Ordner **Object Model > Object Instances**.

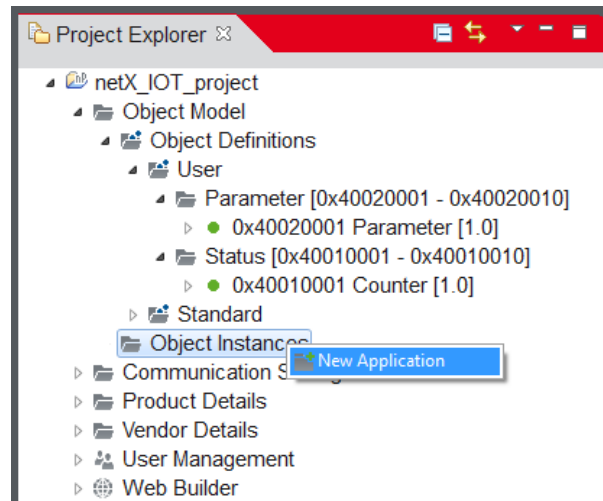


Abbildung 45: Applikationsvariante erstellen

- Klicken Sie **New Application** an.
- Das Fenster **New Application** erscheint.

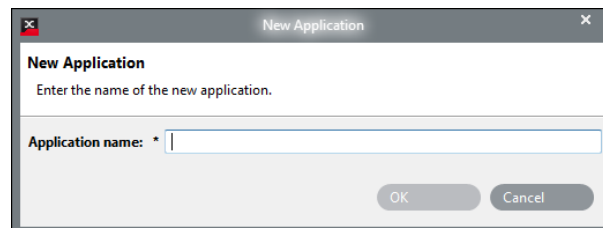


Abbildung 46: Name für erste Applikationsvariante eingeben

- Geben Sie unter **Application name** den Namen für die erste Applikationsvariante ein.

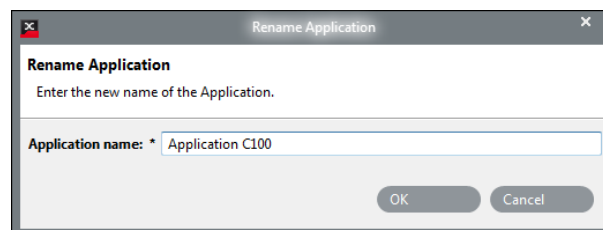


Abbildung 47: Beispiel Application C100

- Klicken Sie **OK**.
- Der Ordner für die Applikationsvariante erscheint.

Objekte der Applikationsvariante konfigurieren

Zur Konfiguration der Applikationsvariante müssen Sie die für Ihr Produkt angelegten Objektordner in den Ordner der Applikationsvariante ziehen. Im Beispiel werden die Ordner **0x40010001 Counter [1.0]** bzw. **0x40020001 Parameter [1.0]** in den Ordner **Application C100** gezogen.

- Klicken Sie die Ordner für die Objekte an, die in der ersten Applikationsvariante verwendet werden sollen.

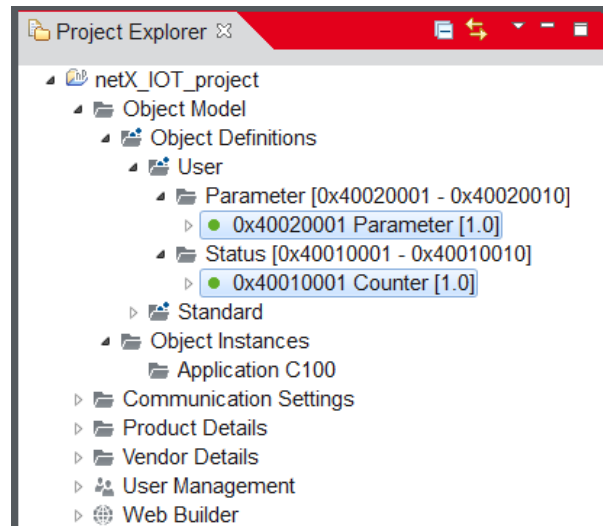


Abbildung 48: Applikationsvariante (Beispiel Application C100) - Objekte per Drag und Drop zusammenstellen

- Ziehen Sie die ausgewählten Objekte per Drag und Drop in den Ordner der Applikationsvariante.

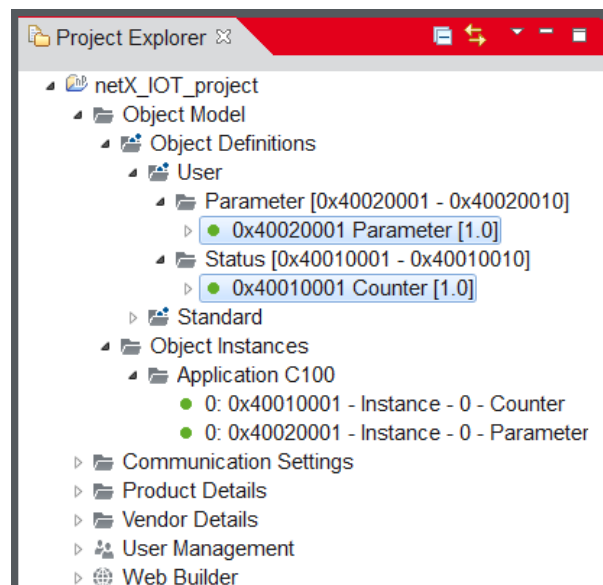


Abbildung 49: Applikationsvariante (Beispiel Application C100)

- Falls ein Objekt in der Applikationsvariante mehrfach verwendet wird, wiederholen Sie den Vorgang.

5.3.2.2 Objektparameter anpassen (Object Instances)

Zur Konfiguration der Objekte nehmen Sie in der Registerkarte Object Instances folgende Anpassungen vor:

- Objektinstanznamen*
- Default-Werte der Elemente**
- Elementwert (Element Value)**

*auf Objektebene, **auf Elementebene

Objektinstanznamen und Default-Werte der Elemente anpassen

Sie können die Namen der Objektinstanzen der Applikationsvariante anpassen, um eine für Ihre Produktvarianten geeignete Namensstrukturierung zu erhalten. Weiterhin können Sie für jedes in der Applikationsvariante konfigurierte Objekt, bzw. für die darin enthaltenen Elemente, gegebenenfalls die Default-Werte individuell anpassen.

- Um die Objektinstanzen umzubenennen, machen Sie zunächst einen Doppelklick auf einen Objektordner der Applikationsvariante.
- Im Editor wird die Applikationsvariante angezeigt (im Beispiel **Application C100**).

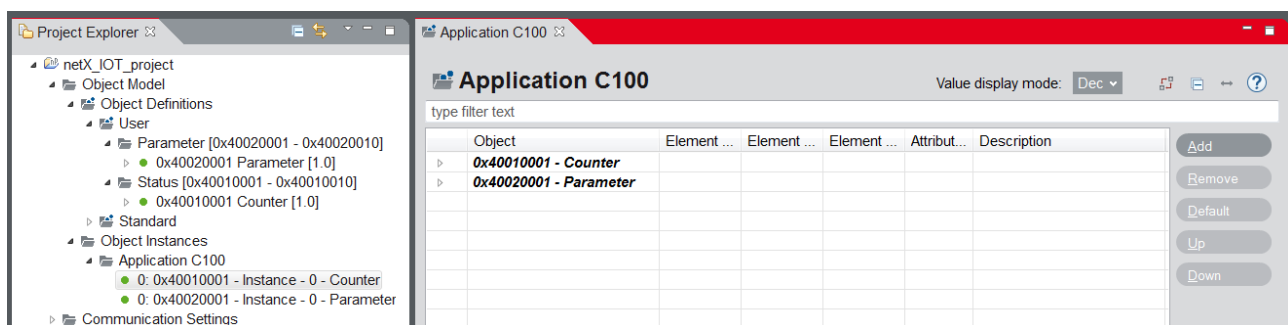


Abbildung 50: Editor der Applikationsvariante (Beispiel)

- Klicken Sie im Editor auf die Zeilen mit der Objektbezeichnung (im Beispiel **0x40010001 Counter** bzw. **0x40020001 Parameter**) und klappen Sie die darunter befindlichen Zeilen mit der Anzeige der Instanzen und den Elementdaten auf.

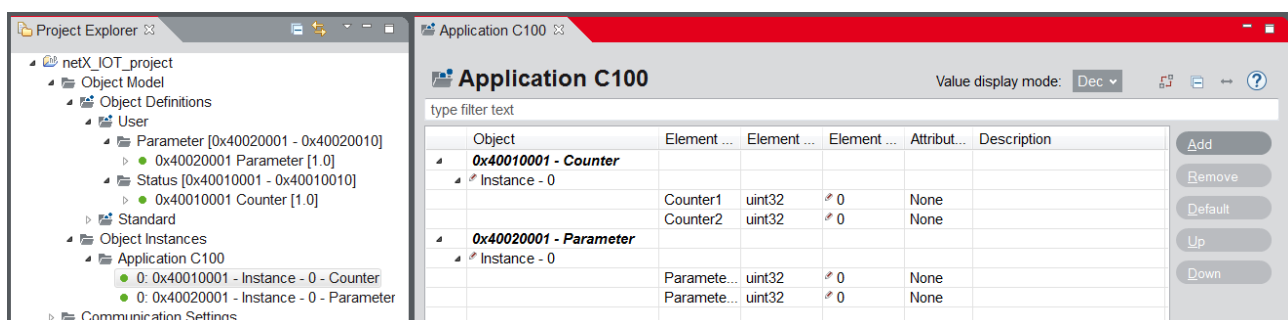


Abbildung 51: Editor der Applikationsvariante mit Zeilen für die Instanzen und Elemente (Beispiel)

Eine Beschreibung zur Bedienung der Tabelle finden Sie im Abschnitt Bedienelemente des Application Variant-Editors.

Parameter der Elemente (Object Instances)

Parameter	Bedeutung	Wert/Wertebereich
Object	<p>Objektbezeichnungen:</p> <p>In der Spalte Object erscheinen die Objektbezeichnungen, die sich aus der Objektkennung (Object identifier) und dem Objektnamen (Object name) zusammensetzen (nicht editierbar).</p> <p>Die Objektkennung (Object identifier) und der Objektnamen (Object name) werden im Fenster New Object eingestellt (siehe Tabelle <i>New Object – Parameter</i> [► Seite 40]). Der Objektnamen (Object name) wird im Object Definitions-Editor angepasst. Siehe Tabelle <i>Properties – Parameter (Object Definitions)</i>.</p>	
	<p>Instanznamen:</p> <p>Unter jeder Objektbezeichnung erscheinen die Namen der zugehörigen Objektinstanzen (editierbar).</p>	
Element Name	Name des Elements, das unter Object Definitions angelegt wurde (hier nicht editierbar). Siehe Tabelle <i>Parameter der Elemente (Object Definitions)</i> [► Seite 48].	
Element Type	Datentyp des Elements, der unter Object Definitions konfiguriert wurde (hier nicht editierbar). Siehe Tabelle <i>Parameter der Elemente (Object Definitions)</i> [► Seite 48].	Bit, uint8, uint16, uint32, uint64, int8, int16, int32, int64, string, real32, real64
Element Value	<p>Hier editierbarer Default-Wert, der für das Element als Initialwert gilt</p> <p>So lange Element Value nicht individuell angepasst wurde, erscheint der unter Object Definitions im Feld Default Value global konfigurierter Wert. Siehe Tabelle <i>Parameter der Elemente (Object Definitions)</i> [► Seite 48].</p> <p>Für jedes in der Applikationsvariante konfigurierte Objekt, bzw. die darin enthaltenen Elemente, kann der Wert für Element Value individuell angepasst werden. D. h., im Beispiel C100 können die Default-Werte für jeden Counter bzw. Parameter getrennt eingestellt werden.</p> <p>Hinweis: Beachten Sie, dass eine Änderung des globalen Default-Werts unter Object Definitions, alle individuell eingestellten Default-Werte wieder überschreibt.</p>	Der Default-Wert ist abhängig vom Anwendungsfall und wird vom Anwender festgelegt.
Attributes	Nicht verwendet (hier nicht editierbar)	None
Description	Unter Object Definitions eingegebene Beschreibung für das Element (hier nicht editierbar). Siehe Tabelle <i>Parameter der Elemente (Object Definitions)</i> [► Seite 48].	3 ... 50 Zeichen

Tabelle 15: Parameter der Elemente (Object Instances)

- Um den Namen für die Objektinstanz zu ändern, setzen Sie den Cursor mit einem Doppelklick in die Zelle in der Spalte **Object**, die den Instanznamen enthält (im Beispiel **Instance 0**).

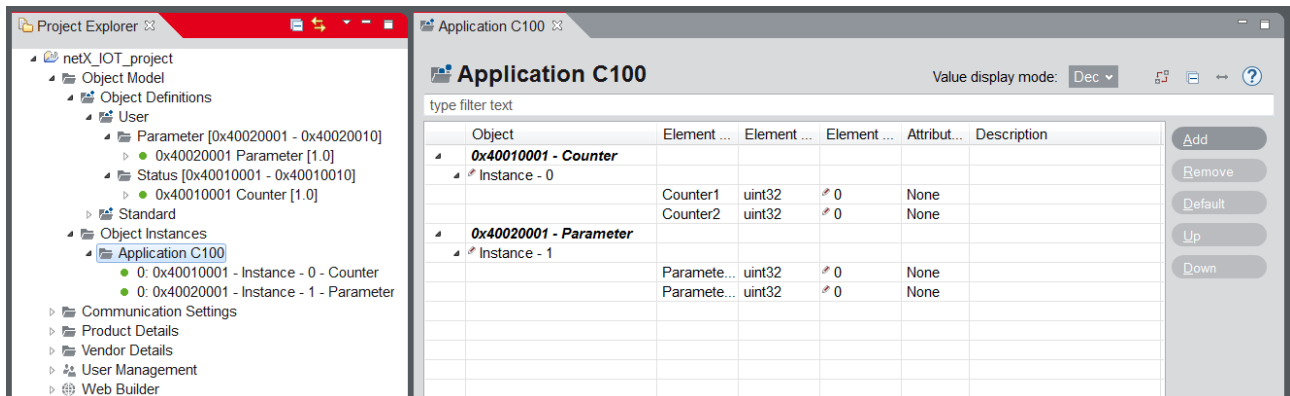


Abbildung 52: Editor der Applikationsvariante - Instanzname editieren (Beispiel)

- Geben Sie einen für Ihre Applikationsvariante geeignete Instanznamen ein (im Beispiel Status bzw. Parameter).

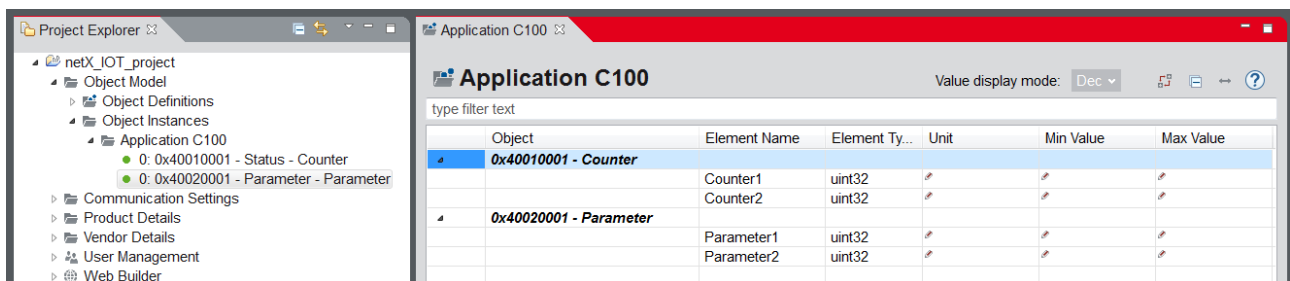


Abbildung 53: Editor der Applikationsvariante mit produktbezogenen Instanznamen (Beispiel)

Elementwert anpassen (Element value)



Wichtig:

Im Editor für eine Applikationsvariante können Sie **Element value** editieren.

Beachten Sie, dass eine Änderung des globalen Default-Werts im Feld **Default Value**, unter **Object Definitions**, alle individuell im Feld **Element Value**, unter **Object Instances**, eingestellten Default-Werte wieder überschreibt.

5.3.2.3 Objektparameter anpassen (Details)

Zur Konfiguration der Objekte nehmen Sie in der Registerkarte **Details** folgende Anpassungen vor:

- Unit**
- Min Value und Max Value**

**auf Elementebene

Unit, Min. Value und Max. Value pro Element anpassen

Für die in der Applikationsvariante konfigurierten Objekte können Sie für jedes enthaltene Element die Parameter **Unit** (Einheit) und **Min Value** (Minimalwert) bzw. **Max Value** (Maximalwert) individuell anpassen.

- Um die Anpassungen für **Unit**, **Min Value** und **Max Value** pro Element vornehmen zu können, klicken Sie auf die Registerkarte **Details**.

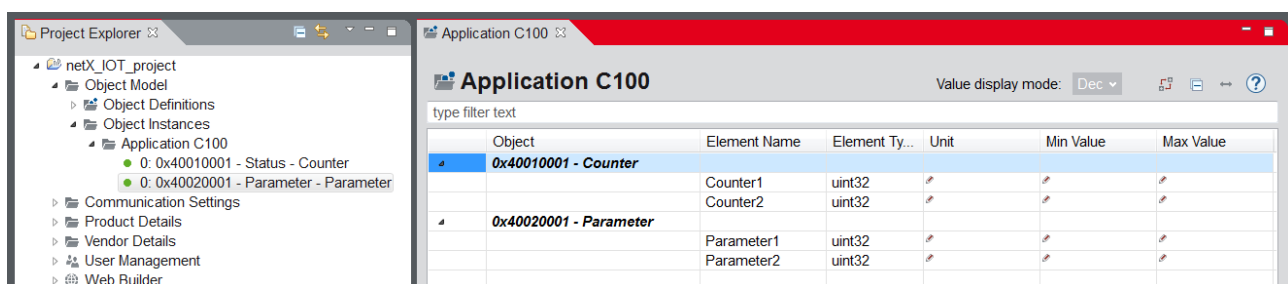


Abbildung 54: Editor der Applikationsvariante – Unit, Min./Max. Value anpassen (Beispiel)

Parameter	Bedeutung	Wert/Wertebereich
Object	In der Spalte Object erscheinen die Objektbezeichnungen, die sich aus der Objektkennung (Object identifier) und dem Objektnamen (Object name) zusammensetzen (nicht editierbar). Die Objektkennung (Object identifier) und der Objektnamen (Object name) werden im Fenster New Object eingestellt (siehe Tabelle <i>New Object – Parameter</i> [▶ Seite 40]). Der Objektnamen (Object name) kann im Object Definitions-Editor angepasst werden. Siehe Tabelle Properties – Parameter (Object Definitions).	
Element Name	Name des Elements, das unter Object Definitions angelegt wurde (hier nicht editierbar) Siehe Tabelle <i>Parameter der Elemente (Object Definitions)</i> [▶ Seite 48].	
Element Type	Datentyp des Elements, der unter Object Definitions konfiguriert wurde (hier nicht editierbar). Siehe Tabelle <i>Parameter der Elemente (Object Definitions)</i> [▶ Seite 48].	Bit, uint8, uint16, uint32, uint64, int8, int16, int32, int64, string, real32, real64
Unit	Umfangreiche Liste mit Einheiten	(z. B. m, kg, Pa, Hz, bit, byte etc.)
Min Value	Frei editierbarer Minimalwert für den Elementwert	
Max Value	Frei editierbarer Maximalwert für den Elementwert	

Tabelle 16: Parameter der Elemente – Details (Details)

- Machen Sie jeweils einen Doppelklick in die anzupassende Zelle und wählen Sie für **Unit** eine physikalische Einheit, geben Sie für **Min Value** eine Untergrenze bzw. für **Max Value** eine Obergrenze für den entsprechenden Wert ein.

5.4 Kommunikationseinstellungen vornehmen

Um in netX Studio Engineering Tool die Kommunikationseinstellungen zu konfigurieren:

- Ergänzen Sie für jede Applikationsvariante die Kommunikationseinstellungen und wählen Sie dabei das IOT-Protokoll und die Applikation aus (IoT protocol, Application).
- Konfigurieren Sie den Zugriff auf Objektdaten über OPC UA **oder** MQTT (nur erlaubte Kombinationen), bzw. gegebenenfalls zusätzlich WebServer.
- Nehmen Sie für OPC UA oder für MQTT die Einstellungen für die IOT-Kommunikation einzeln vor.

In den nachfolgenden Abschnitten finden Sie eine schrittweise Beschreibung zur Vorgehensweise.

5.4.1 Kommunikationseinstellungen ergänzen

Für jede Applikationsvariante können Sie Kommunikationseinstellungen für **ein** IoT-Protokoll (OPC UA oder MQTT) ergänzen.



Hinweis:

Jede Applikationsvariante kann mit einem IoT-Protokoll kombiniert werden.

Beachten Sie, dass Sie in der Basisversion von netX Studio Engineering Tool „netX Studio (netPROXY Basic)“ nur **ein** Kommunikationsprotokoll (Netzwerk) konfigurieren können.

Kommunikationseinstellungen ergänzen

- Um die Kommunikationseinstellungen für eine Applikationsvariante zu ergänzen, machen Sie im Fenster Project Explorer einen Rechtsklick auf den Ordner **Communication Settings**.

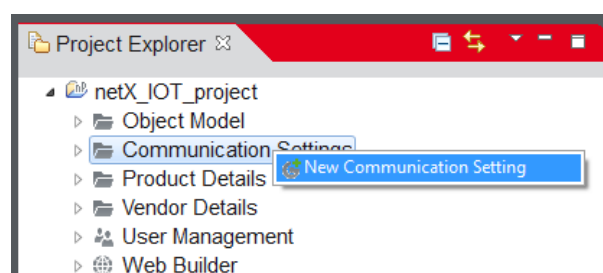


Abbildung 55: Project Explorer - New Communication Setting

- Klicken Sie **New Communication Setting** an.
- Die Dialogbox **Add Communication Settings** erscheint.
- Wählen Sie unter **IoT protocol** das Protokoll aus.

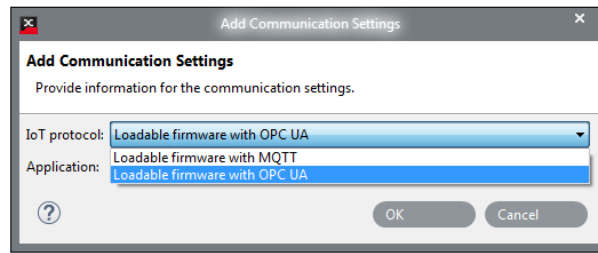


Abbildung 56: Add Communication Settings - IoT protocol

- Wählen Sie unter **Application** die Applikationsvariante aus.

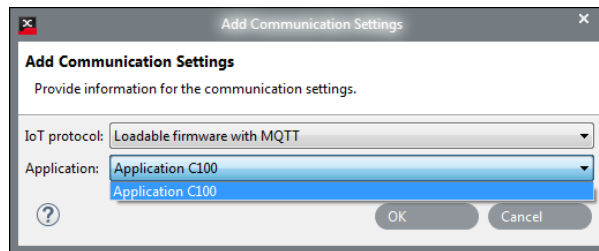


Abbildung 57: Add Communication Settings - Application (Beispiel)

- Ergänzen Sie die Kommunikationseinstellungen für weitere Applikationsvarianten.

Wenn Sie für jede Applikationsvariante die Kommunikationseinstellungen ergänzt haben (**eine** Kombination mit OPC UA oder MQTT), erscheint die Meldung, dass keine Applikation mehr für die IoT-Konfiguration zur Verfügung stehen.

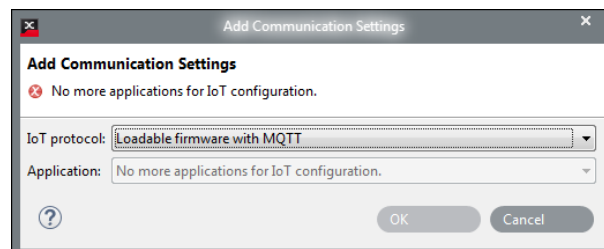


Abbildung 58: Add Communication Settings – No more applications

Kommunikationseinstellungen entfernen

- Um die Kommunikationseinstellungen für eine Applikationsvariate zu entfernen, machen Sie im Fenster Project Explorer einen Rechtsklick auf den zugehörigen Ordner, unterhalb des Ordners Communication Settings.

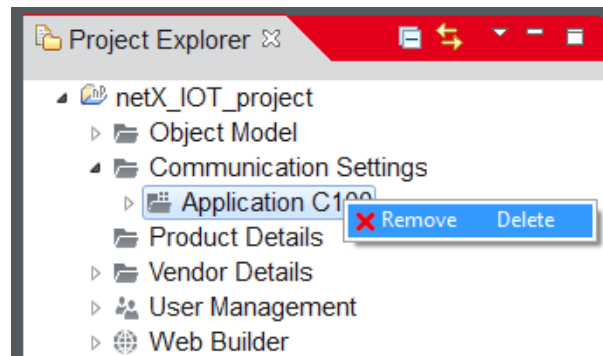


Abbildung 59: Project Explorer - Remove (Beispiel)

- Klicken Sie **Remove** an.
- ⇒ Die gewählten Kommunikationseinstellungen werden gelöscht.

5.4.2 OPC UA, MQTT bzw. WebServer ergänzen

Objektdaten können über OPC UA oder MQTT bzw. WebServer übermittelt werden (siehe dazu auch Abschnitt *Beispiele* [► Seite 28]).

Für die Konfiguration von OPC UA oder MQTT (bzw. gegebenenfalls zusätzlich WebServer) ist es erforderlich, dass Sie

- im **Object Definitions**-Editor über **Assign label** jedem Objekt ein oder mehrere vorgegebene Labels (**OPC UA** oder **MQTT** bzw. **WebServer**) oder selbst erstellte Labels zuweisen (siehe Abschnitt *Eigenschaften (Properties) des Objektes anpassen* [► Seite 44]).
- unter **Communication Settings** bei jeder Applikationsvariante den Editor **OPC UA** oder **MQTT** bzw. **WebServer** hinzufügen, im Editor die zugewiesenen Labels erneut auswählen und die Konfiguration vornehmen, wie hier nachfolgend beschrieben.



Wichtig:

In den Kommunikationseinstellungen für **ein und dieselbe** Applikationsvariante dürfen OPC UA und MQTT nicht gleichzeitig genutzt werden.

Hinweis zu den Erfordernissen Ihres Endkunden



Wichtig:

Das netX Studio Engineering Tool ist ein Geräteentwicklungswerkzeug. Es enthält keine Konfigurations- und Diagnosefunktionen für den Endanwender und steht dem Endkunden nicht zur Verfügung. Beachten Sie daher, dass Sie für Ihren Endkunden ein Werkzeug bzw. eine Lösung zur Endkonfiguration des Gerätes zur Verfügung stellen müssen!

Für die Endkonfiguration benötigt Ihr Kunde ein geeignetes Werkzeug mit Konfigurations- und Diagnosefunktionen. Die Endkonfiguration umfasst die Angabe des MQTT-Broker und des OPC UA-Port, die Benutzerverwaltung, sowie Simulationsschritte zur Vorbereitung der Inbetriebnahme des Gerätes.

Um Ihrem Kunden eine geeignete Lösung anzubieten, haben Sie zwei Möglichkeiten:

- Ergänzen Sie bei der Geräteentwicklung mit netX Studio Engineering Tool die Option für die Kommunikation über den WebServer.
- Oder stellen Sie für den Endkunden eine Alternative für die Endkonfiguration bereit.
 - Beispielsweise können Sie dem Kunden einen Service für die komplette Geräteeinrichtung anbieten,
 - Sie können Ihrem Kunden eine generische Lösung anbieten,
 - mit dem Kunden vereinbaren, dass dieser eine eigene Lösung entwickeln muss oder
 - Sie vereinbaren einen Service durch den Gerätehersteller.

Editor für OPC UA, MQTT bzw. WebServer hinzufügen

- Machen Sie unter **Communication Settings** einen Rechtsklick auf den Ordner der Kommunikationseinstellungen einer Applikationsvariante.



Wichtig:

In den Kommunikationseinstellungen für **ein und dieselbe** Applikationsvariante dürfen OPC UA und MQTT nicht gleichzeitig genutzt werden.

- Klicken Sie **Add OPC UA**, **Add MQTT** bzw. **Add WebServer**.
- Die Konfiguration erfolgt, so dass die Funktionen zur Übertragung der Objektdaten an den OPC UA-Server, MQTT bzw. WebServer genutzt werden können. Die Ordner **OPC UA**, **MQTT** bzw. **WebServer** werden im **Project Explorer** angezeigt.

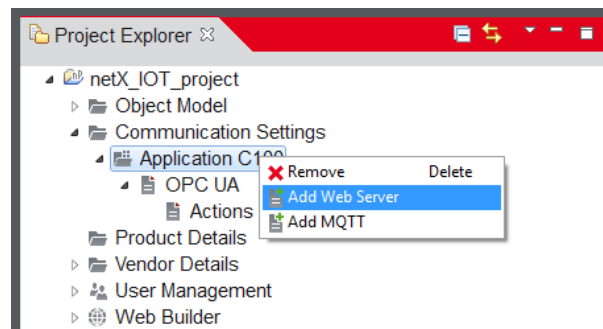


Abbildung 60: Kommunikationseinstellungen einer Applikationsvariante, Beispiel Add WebServer

5.4.2.1 Kommunikationseinstellungen für OPC UA

Um die Kommunikationseinstellungen für OPC UA zu konfigurieren, gehen Sie wie folgt vor:

- Machen Sie einen Doppelklick auf den Ordner **OPC UA**.
- Der **OPC UA**-Editor erscheint.

Unter **OPC UA Server Configuration** können Sie verschiedene Einstellungen vornehmen.

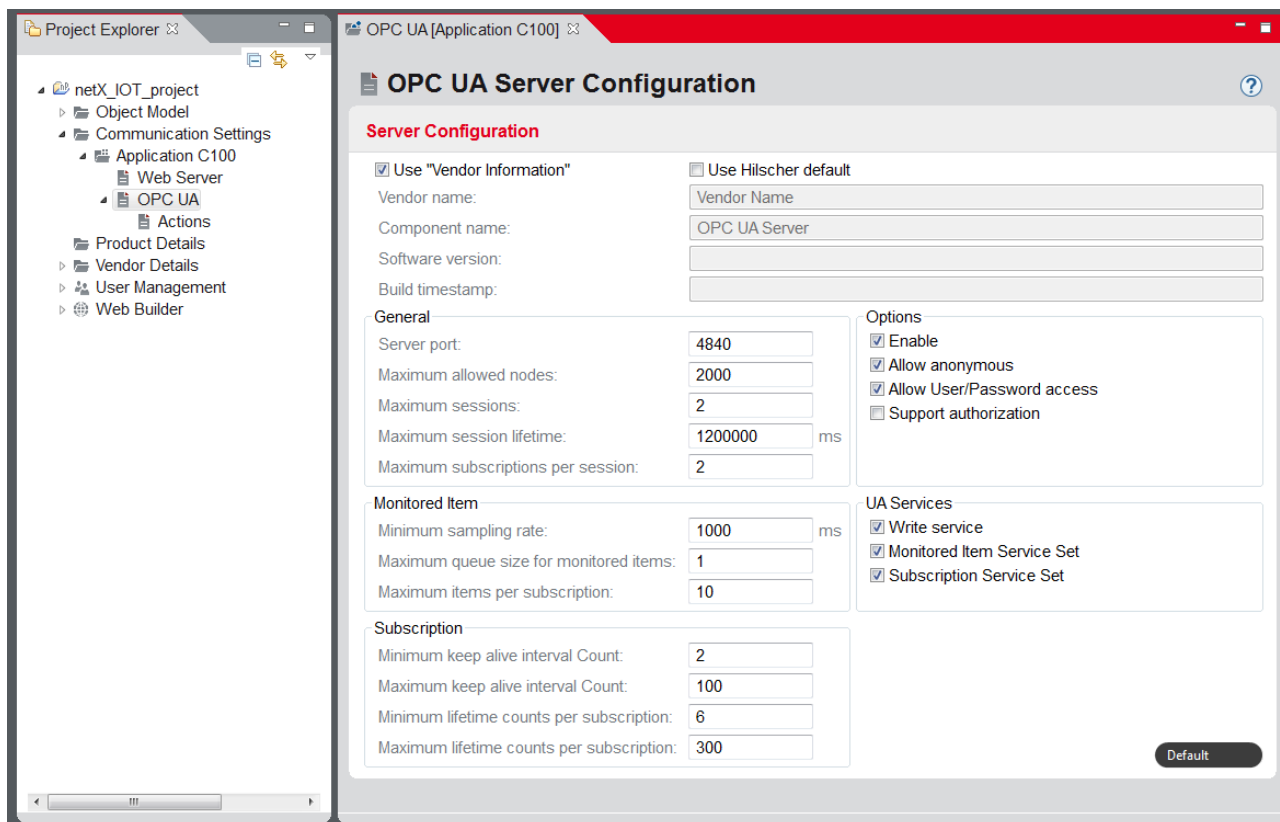


Abbildung 61: Communication Settings > OPC UA-Editor – OPC UA Server Configuration (Beispiel)

Parameter OPC UA Server Configuration

Parameter	Bedeutung	Wert/Wertebereich
Use „Vendor information“ (Herstellerinformationen verwenden)	Für Vendor name und Component name werden die Default-Werte des OPC UA-Servers bzw. für Software version und Build timestamp Hilscher-Standardwerte verwendet.	Angehakt, nicht angehakt, Default: Use „Vendor information“ angehakt
Use „Hilscher default“		
Vendor name (Herstellername)	Hier können Sie den Herstellernamen frei eintragen. Dieser kann gleich oder verschieden sein, wie unter Project Explorer > Vendor Details > Vendor Information (siehe Abschnitt <i>Herstellerangaben eintragen</i> [▶ Seite 85]) eingetragen.	Zeichenkette (max. 63 Zeichen), Default: <i>Hilscher Ges. für Systemautomation mbH</i>
Component name (Komponentenname)		Zeichenkette (max. 31 Zeichen), Default: <i>OPC UA Server</i>
Software version (Software-Version)	Zeichenkette zum Speichern der Software-Version: 'MAJOR.MINOR.REVISION.BUILD TEXT'	-

Parameter	Bedeutung	Wert/Wertebereich
Build timestamp (<i>Zeitstempel erstellen</i>)	Zeichenkette zum Speichern des Build-Zeitstempels Format: ,yyyy-MM-dd'T'HH:mm:ss.SSSXXX' yyyy: Jahr, MM: Monat, dd: Tag, T: fester Buchstabe (für Time), HH: Stunden, mm: Minuten, ss: Sekunden, SSS: Millisekunden, XXX: Zeitzone; zusätzlich: Z für UTC +00:00 Beispiel: 2016-04-07T12:34:11.123Z	-
General		
Server port (<i>Server-Port</i>)	IP-Portnummer des OPC UA-Servers Port 0 und 1 werden nicht unterstützt.	Gültiger Port, 0, 2, 3, ... 65535; Default: 4840
Maximum allowed nodes (<i>Anzahl maximal erlaubter Knoten</i>)	Erlaubte maximale Anzahl von Knoten, die der Server im definierten Adressbereich darstellen kann. Dies sind selbstdefinierte Knoten, die sowohl vom netX Studio Engineering Tool als auch von den obligatorischen Server-Knoten konfiguriert werden.	400 ... 2000, Default: 2000
Maximum sessions	Erlaubte <i>maximale Anzahl von UA-Sessions</i> , die der Server verarbeiten muss. Das Nano-Profil benötigt nur 1. UA-Kommunikation ist in Sessions aufgeteilt.	1, 2, Default: 2
Maximum session lifetime	Erlaubte maximale UA-Session-Dauer unter Lifetime-Bedingungen in Millisekunden	30000 mS ... 1200000 mS (30 S to 20 Min), Default: 1200000 mS (20 Min.)
Maximum subscriptions per session*	Maximal erlaubte Anzahl von Subscriptions pro Session	Min: 1, Max: 2, Default: 2
Options		
Optionen zur Konfiguration der Kommunikation zum OPC UA-Server.		
Enable	Freigeben	Angehakt, nicht angehakt, Default: angehakt
Allow anonymous	Anonymen Zugriff erlauben	Angehakt, nicht angehakt, Default: angehakt
Allow User/Password access	Benutzer-/Passwort-Zugriff erlauben	Angehakt, nicht angehakt, Default: angehakt
Support authorization	Autorisierung für Support	Angehakt, nicht angehakt, Default: nicht angehakt
Monitored Item		
Minimum Sampling rate in ms* (<i>Mindestabtastrate</i>)	Die für ein überwachtes Element definierte Abtastrate kann höher sein, als die Senderate für die Subscription (Empfang). Aus diesem Grund kann das überwachte Element so konfiguriert werden, dass entweder alle Nachrichten in die Warteschlange gestellt werden oder nur die letzte Nachricht zur Übertragung durch die Subscription (Empfang) in die Warteschlange gestellt wird. Im letzteren Fall ist die Warteschlangengröße eins.	Min: 50, Default: 1000
Maximum queue size for monitored items*	Unterstützte <i>maximale Warteschlangengröße für überwachte Elemente</i>	Max: 1, Default: 1
Maximum items per subscription*	Unterstützte <i>maximale Anzahl Elemente pro Subscription</i>	Max: 10, Default: 10
UA services		
Write service	<i>Schreibzugriff</i> auf OPC-UA-Objekte erlauben (Standard)	Angehakt, nicht angehakt, Default: angehakt
Monitored Item Service Set	<i>Service-Einstellung für das überwachte Element</i> erlauben (Standard)	Angehakt, nicht angehakt, Default: angehakt

Parameter	Bedeutung	Wert/Wertebereich
Subscription Service Set	Service-Einstellung für Subscription erlauben (Standard)	Angehakt, nicht angehakt, Default: angehakt
Subscription		
Minimum keep alive interval Count*	Unterstützte Mindestanzahl gezählter 'Keep-Alive-Interval-Einheiten'	Min: 1, Max: Subscription: Maximum 'Keep Alive Interval Count' Default: 2
Maximum keep alive interval Count*	Unterstützte maximale Anzahl gezählter 'Keep-Alive-Interval-Einheiten'	Min: Subscription: Minimum 'Keep Alive Interval Count', Max: 1431655765 (1/3 * UINT32_MAX) Default: 100
Minimum lifetime counts for Subscriptions*	Unterstützte Mindestanzahl gezählter 'Lebenszeit-Einheiten' für Subscriptions	Max: 4294967295 (UINT32_MAX), Default: = Min: 3 x Subscription: Minimum 'Keep Alive Interval Count'
Maximum lifetime counts for Subscriptions*	Unterstützte maximale Anzahl gezählter 'Lebenszeit-Einheiten' für Subscriptions	Min: Subscription: Minimum Lifetime Counts for Subscriptions, Max: 4294967295 (UINT32_MAX); Default: 3 x Subscription: Maximum 'Keep Alive Interval Count'
*Nur in Mikroservern verfügbar.		

Tabelle 17: Parameter OPC UA Server Configuration

Parameter OPC UA Actions Configuration

- Machen Sie einen Doppelklick auf den Ordner **OPC UA > Actions**.
- Der **OPC UA Actions**-Editor erscheint. Unter **Filter** ist die Auswahl für das Label **OPC UA** standardmäßig vorkonfiguriert. In der Tabelle darunter können Sie für jede für OPC UA konfigurierte Objektinstanz die Zugriffsart **read** oder **write** konfigurieren.

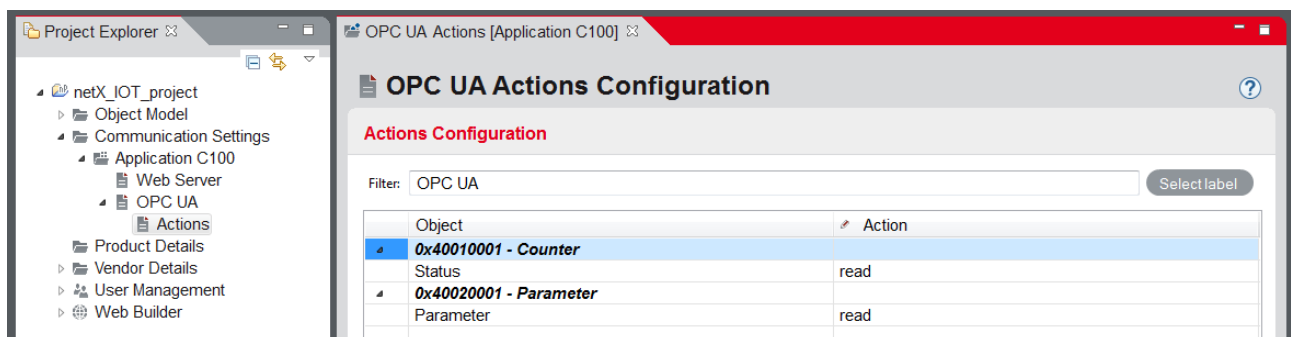


Abbildung 62: Communication Settings > OPC UA-Editor – OPC UA Actions Configuration (Beispiel)

Parameter	Bedeutung	Wert/Wertebereich
Object	Objektnamen (Object name), welcher im Fenster New Object festgelegte wurde und gegebenenfalls im Object Definitions -Editor angepasst wurde (siehe Tabelle Properties – Parameter (Object Definitions), hier nicht editierbar).	Name des Objekts
Action	Lese- und Schreibdienste; Unterfunktion der OPC-UA-Gesamtfunktionalität; siehe auch Tabelle <i>Objektfunktionen</i> [► Seite 12].	read, write, Default: read

Tabelle 18: Parameter OPC UA Actions Configuration



Hinweis:

Für Objekte mit Zugriff über WebServer, können Sie zusätzlich einen Zugriff über den OPC UA-Server konfigurieren und die für den WebServer konfigurierten Objektdaten zusätzlich am OPC UA-Server anzeigen.

5.4.2.2 Kommunikationseinstellungen für MQTT

Um die Kommunikationseinstellungen für MQTT zu konfigurieren, gehen Sie wie folgt vor:

- Machen Sie einen Doppelklick auf den Ordner **MQTT**.
- Der Editor **MQTT Component Information** erscheint.

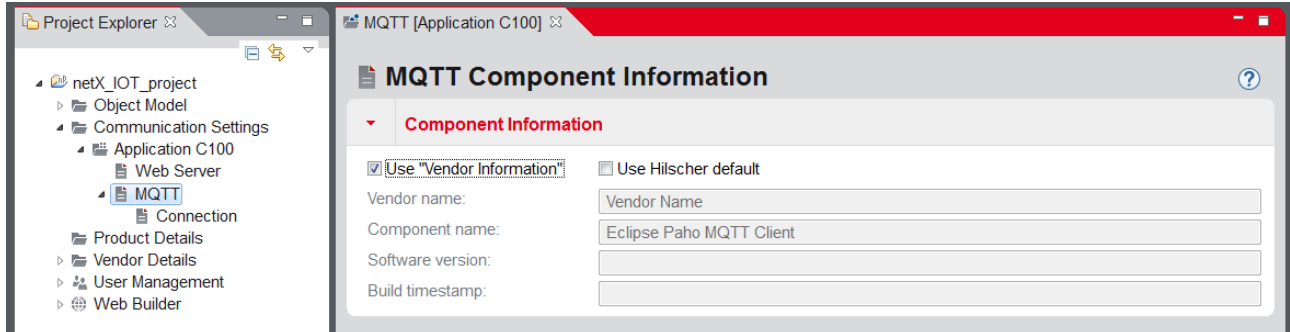


Abbildung 63: Communication Settings > MQTT-Editor - MQTT Component Information (Beispiel)

Parameter	Bedeutung	Wert/Wertebereich
Use „Vendor information“ Use „Hilscher default“	Für Vendor name und Component name werden die Default-Werte des Eclipse Paho MQTT Client bzw. für Software version und Build timestamp Hilscher-Standardwerte verwendet.	Angehakt, nicht angehakt, Default: Use „Vendor information“ angehakt
Vendor name	Herstellername. Hier können Sie den Herstellernamen frei eintragen. Dieser kann gleich oder verschieden sein, wie unter Project Explorer > Vendor Details > Vendor Information (siehe Abschnitt <i>Herstellerangaben eintragen</i> [▶ Seite 85]) eingetragen.	Zeichenkette (max. 63 Zeichen), Default: <i>Hilscher Ges. für Systemautomation mbH</i>
Component name	Komponentenname	Zeichenkette (max. 31 Zeichen), Default: <i>Eclipse Paho MQTT Client</i>
Software version	Zeichenkette zum Speichern der Software-Version: 'MAJOR.MINOR.REVISION.BUILD TEXT'	-
Build timestamp	Zeichenkette zum Speichern des Build-Zeitstempels Format: ,yyyy-MM-dd'T'HH:mm:ss.SSSXXX' yyyy: Jahr, MM: Monat, dd: Tag, T fester Buchstabe (für Time), HH: Stunden, mm: Minuten, ss: Sekunden, SSS: Millisekunden, XXX: Zeitzone; zusätzlich: Z für UTC +00:00 Beispieleingabe: 2016-04-07T12:34:11.123Z	-

Tabelle 19: Parameter MQTT Component Information

Parameter MQTT - Connection

- Machen Sie einen Doppelklick auf den Ordner **MQTT > Connections**.
- Der **MQTT**-Editor erscheint.

MQTT Connection

Parameter	Bedeutung	Wert/Wertebereich
Client ID	Eindeutiger Name des MQTT-Clients im UTF-8-Format. Alle Geräte, die mit einem Broker verbunden werden, müssen einen eindeutigen Namen besitzen. Wenn das Feld leer ist, dann vergibt der Broker einen Namen.	max. 23 Bytes für max. 23 Zeichen, Default: [Client ID]
Broker address	IP-Adresse des Brokers	Gültige IP-Adresse; Default: [BrokerAddress]
Topic prefix	Text, der jedem Topic vorangestellt wird, z. B. 'StationA'. Für jedes einzelne Topic kann konfiguriert werden, ob dieses Präfix vorangestellt werden soll oder nicht.	Text aus Groß- und Kleinbuchstaben und Unterstrich, Default: [keine Angabe]
User name	Benutzername zum Anmelden an den Broker.	[keine Angabe] oder Benutzername, Default: [keine Angabe]
Password	Passwort zum Anmelden an den Broker.	[keine Angabe] oder Passwort, Default: [keine Angabe]
Will topic	Will Topic	Text aus Groß- und Kleinbuchstaben und Unterstrich, Default: [keine Angabe]
Connect timeout	Zeit für den Versuch zum Aufbau einer Verbindung (MQTT Connect) zum Broker. War der Verbindungsaufbau erfolglos, dann wartet der MQTT-Client für die Dauer 'Connection Timeout', bis ein erneuter Verbindungsaufbau zum Broker ausgeführt wird.	Angabe in s. 0 = MQTT-Client versucht ständig eine Verbindung zum Broker aufzubauen. Default: 0
Connection idle timeout	Zeit nach der eine ungenutzte Verbindung (keine Datenübertragung) geschlossen wird.	Angabe in s. 0 = kein Timeout, d.h. Verbindung bleibt geöffnet. Default: 0
MQTT keep alive interval	Intervall in welchem der MQTT-Client ein Lebenszeichen an den Broker sendet. Der eingestellte Wert für den MQTT-Client muss kleiner sein als die im Broker eingestellte Überwachungszeit.	Angabe in s. 0 = kein Lebenszeichen an den Broker senden. Default: 0

Parameter	Bedeutung	Wert/Wertebereich
Handling of an MQTT connection	<p>Clean session: Einstellung, ob nach einem Verbindungsaufbau alle Topics an den Broker übertragen werden sollen oder nicht.</p> <p>angehakt (empfohlen): Nach einem Verbindungsaufbau zum Broker werden alle Topics vom Typ 'publish' vom MQTT-Client an den Broker übertragen.</p> <p>Nicht angehakt: Es werden nur Topics an den Broker übertragen, die sich seit der letzten Verbindung geändert haben. Beachten Sie, wenn Sie diese Einstellung verwenden, dass der Broker die Funktion 'Kontext erhalten' unterstützen muss.</p>	angehakt, nicht angehakt, Default: angehakt
	<p>Reliable: Parallele oder sequentielle Übertragung.</p> <p>nicht angehakt: parallele Übertragung von MQTT-Diensten.</p> <p>angehakt: sequentielle Übertragung von MQTT-Diensten.</p>	angehakt, nicht angehakt, Default: nicht angehakt
	<p>Will: nicht angehakt: Will-Funktion deaktiviert.</p> <p>angehakt: Will-Funktion aktiviert.</p>	angehakt, nicht angehakt, Default: nicht angehakt
	<p>Prefix will: nicht angehakt: Will-Topic ohne Präfix.</p> <p>angehakt: Will-Topic mit Präfix.</p>	angehakt, nicht angehakt, Default: angehakt
	<p>Secure: nicht angehakt: Ungesicherte Datenübertragung.</p> <p>angehakt: Mit SSL gesicherte Datenübertragung.</p>	angehakt, nicht angehakt, Default: nicht angehakt
	<p>Verify certificate: Ohne Funktion.</p>	angehakt, nicht angehakt, Default: nicht angehakt

Tabelle 20: Parameter MQTT - Connection Settings

MQTT Configuration

Unter **MQTT Configuration > Filter** ist die Auswahl für das Label **MQTT** standardmäßig vorkonfiguriert. In der Tabelle darunter erscheinen die an MQTT übertragenen Parameter Action, Retained, Prefix, Encoding QoS bzw. Topic für die Objekte der entsprechenden Applikationsvariante.

Parameter	Bedeutung	Wert/Wertebereich
Object	- Objektname (Object name) - nicht editierbar -, welcher im Fenster New Object festgelegt wurde und gegebenenfalls im Object Definitions -Editor angepasst wurde (siehe Tabelle Properties – Parameter (Object Definitions)). Darunter - Namen der Objektinstanz -, wie im Application Variant -Editor eingestellt und darunter - Name des Elements -, der unter Object Definitions angelegt wurde (siehe Tabelle <i>Parameter der Elemente (Object Definitions)</i> ► Seite 48]).	Name des Objekts
Action	Publish: MQTT-Client soll dieses Topic zum Broker übertragen. Subscribe: MQTT-Client soll dieses Topic vom Broker erhalten.	Publish, Subscribe, Default: Subscribe
Retained	Einstellung, ob der Broker die Historie eines Datenwertes speichern soll oder nicht. Nicht angehakt: Broker soll keine Historie speichern. Angehakt: Broker soll Historie speichern.	angehakt, nicht angehakt, Default: angehakt
Prefix	Nicht angehakt: Diesem Topic soll das Topic-Präfix nicht vorangestellt werden. Angehakt: Diesem Topic soll das Topic-Präfix vorangestellt werden.	angehakt, nicht angehakt, Default: nicht angehakt
Encoding	Datenformat des Topics. No encoding: Topics werden binär übertragen. JSON: Daten werden im JSON-Format übertragen, d. h. mit Informationen zu Datentyp, Datengröße und Datenwert. UTF-8/MBCS: Daten werden als Text im UTF-8/MBCS-Format übertragen.	No encoding, UTF-8/MBCS, JSON; Default: JSON
QoS	Quality of Service. Einstellung: 0 = At most once: Datenempfang wird über MQTT nicht bestätigt. 1 = At least once: Datenempfang wird über MQTT bestätigt. 2 = Exactly once: Daten werden über MQTT genau einmal mittels Zustandsmaschine übertragen und bestätigt.	At most once, At least once, Exactly once; Default: At most once
Topic	Eindeutiger Name für das Topic, editierbar.	Max. 128 Zeichen Text aus Groß- und Kleinbuchstaben und Unterstrich.

Tabelle 21: Parameter MQTT Configuration



Hinweis:

Für Objekte mit Zugriff über WebServer, können Sie zusätzlich einen Zugriff über MQTT konfigurieren und die für den WebServer konfigurierten Objektdaten zusätzlich über MQTT anzeigen.

5.4.2.3 Kommunikationseinstellungen für WebServer

Um die Kommunikationseinstellungen für WebServer zu konfigurieren, gehen Sie wie folgt vor:

- Machen Sie einen Doppelklick auf den Ordner **WebServer**.
- Der **WebServer**-Editor erscheint. Unter **Filter** ist die Auswahl für das Label **www** standardmäßig vorkonfiguriert. In der Tabelle darunter erscheinen die an den WebServer übertragenen Objektdaten.

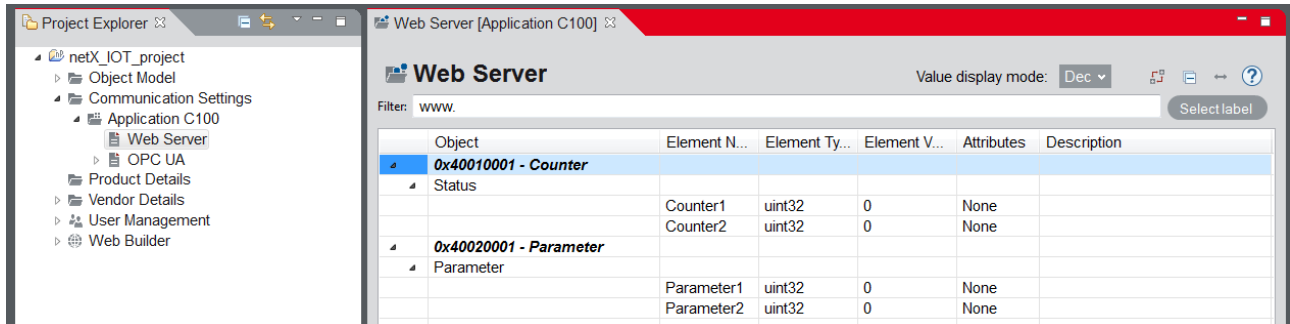


Abbildung 64: Communication Settings > Beispiel WebServer-Editor (Default-Auswahl www)

Eine Beschreibung zu der Tabelle im **WebServer**-Editor finden Sie in der Tabelle *Parameter der Elemente (Object Instances)* [▶ Seite 62]. Die Werte für Element Value sind hier jedoch nicht editierbar.

Angaben zu **Value display mode** bzw. den Incons **Toggle Expand** oder **Collapse All** etc. finden Sie Abschnitt Bedienelemente des Application Variant-Editors.



Hinweis:

Für Objekte mit Zugriff über OPC UA oder MQTT, können Sie zusätzlich einen Zugriff über den WebServer konfigurieren und die für den OPC UA-Server oder MQTT konfigurierten Objektdaten zusätzlich am WebServer anzeigen. Dafür muss das Label „www“ bei der Objektkonfiguration ausgewählt sein!

5.5 Produktdetails konfigurieren

netX Studio Engineering Tool ermöglicht unter **Product Details** die Eingabe von Daten zur Produktidentifizierung. Dazu sind folgende Hauptschritte notwendig:

1. Vorbereitung

- Schreiben Sie die Produktdetails für Ihre Geräte auf.

Diese Produktdetails beinhalten:

- Informationen zum Produkt
(Produktname, Produkt-ID, Produkttyp, Bestell-ID, Profil-ID, Profiltyp) und
- Informationen zur Produktversion
(Produktrevision, Hardware-Name, Hardware-Version, Software-Name, Software-Version, Bootloader-Name, Bootloader-Version).

2. Konfiguration in netX Studio Engineering Tool

- Legen Sie in für jedes Gerät einen Ordner für die Produktdetails an.
- Geben Sie die Daten zu den Produktdetails ein.

5.5.1 Ordner für Produktdetails anlegen

- Um einen Ordner für Produktdetails anzulegen, machen Sie einen Rechtsklick auf den Ordner **Product Details**.
- Klicken Sie **New Product**.

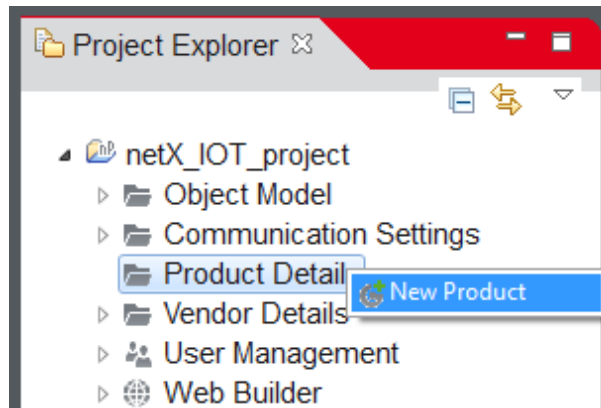


Abbildung 65: Product Details – New Product

- Die Dialogbox **Add New Product** erscheint.

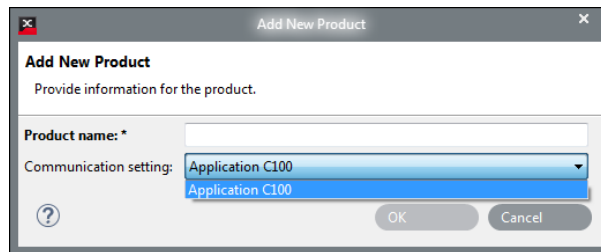


Abbildung 66: Name eingeben und Kommunikationseinstellungen auswählen

- Geben Sie im Feld **Product name *** einen Kurznamen für das Produkt ein.
- Der eingegebene Produktname erscheint unter Product Details im Ordernamen für das Produkt.



Hinweis:

Verwenden Sie für ein Produkt immer den gleichen Produktnamen, damit Sie die Daten zu diesem Produkt immer wiedererkennen!

Verwenden Sie den gleichen Produktnamen auch im Product Information-Editor unter **Product Type** im Feld **Product name ***.

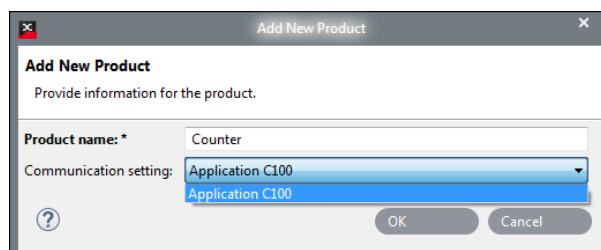


Abbildung 67: Name eingeben und Kommunikationseinstellungen auswählen

- Wählen Sie unter **Select Communication Settings** die Kommunikationseinstellungen für das Produkt.
- Klicken Sie **OK**.

- Der neue Produktordner erscheint mit den Unterordnern **Product Information** und **Product Macros**.

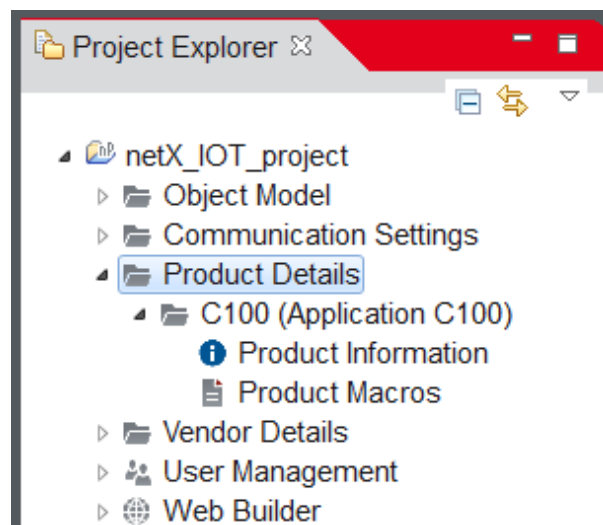


Abbildung 68: Product Details – Beispiel



Angaben zum Ordner **Product Macros** finden Sie in der Application note netX Studio Engineering Tool Web Builder [5].

- Legen Sie auf diese Weise alle Ihre Produkte an.

Sie können Ordner für Produktdetails umbenennen oder löschen (siehe dazu auch *Produktdetails umbenennen oder entfernen* [► Seite 81]).

5.5.1.1 Produktdetails umbenennen oder entfernen

Produktdetails können Sie umbenennen oder entfernen.

Produktdetails umbenennen

- Machen Sie einen Rechtsklick auf den Ordner **Product Details** > [Produktname].
- Um Ihr Produkt umzubenennen, klicken Sie **Rename**.

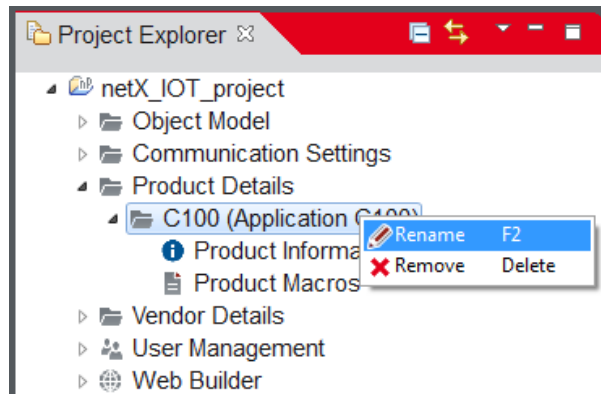


Abbildung 69: Product Details - Umbenennen (Beispiel)

- Geben Sie den neuen Produktnamen in der Dialogbox **Rename Product** (Produkt umbenennen) im Feld **Product name *** ein.
- ⇒ Der geänderte Produktnamen erscheint unter Product Details im Ordernamen für das Produkt.



Hinweis:

Verwenden Sie für ein Produkt immer den gleichen Produktnamen, damit Sie die Daten zu diesem Produkt immer wiedererkennen!

Verwenden Sie den gleichen Produktnamen auch im Product Information-Editor unter **Product Type** im Feld **Product name ***.

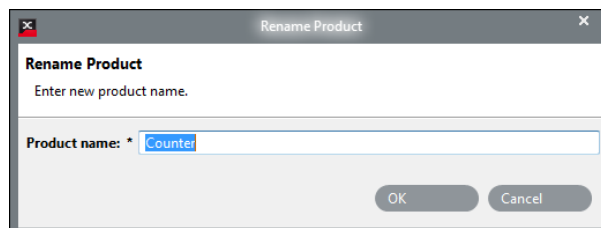


Abbildung 70: Rename Product (Produktumbenennen)

- Klicken Sie **OK**.
- ⇒ Der Produktordner erscheint mit neuem Namen.

Produktdetails entfernen

- Machen Sie einen Rechtsklick auf den Ordner **Product Details** > [Produktname].
- Um Ihr Produkt zu entfernen, klicken Sie **Remove**.

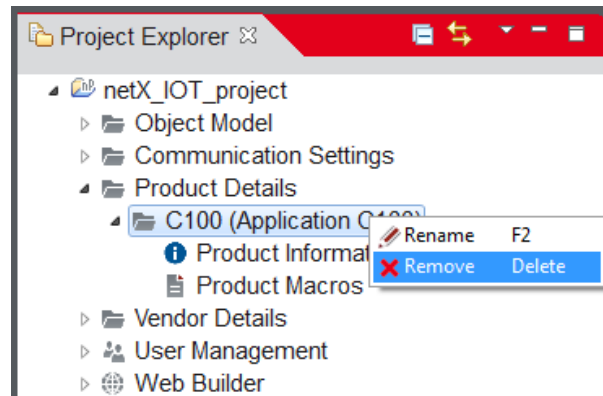


Abbildung 71: Product Details - Entfernen (Beispiel)

- ⇒ Das Produkt wird entfernt.

5.5.2 Produktinformationen eingeben



Hinweis:

Eingaben im Product Information-Editor wirken sich in verschiedenen Standardobjekten aus.

- Um die Produktinformationen einzugeben, machen Sie im Fenster **Project Explorer** einen Doppelklick auf den Ordner **Product Details** > **[Produktname]** > **Product Information**.
- ⇒ Der **Product Information**-Editor (rechts) erscheint.

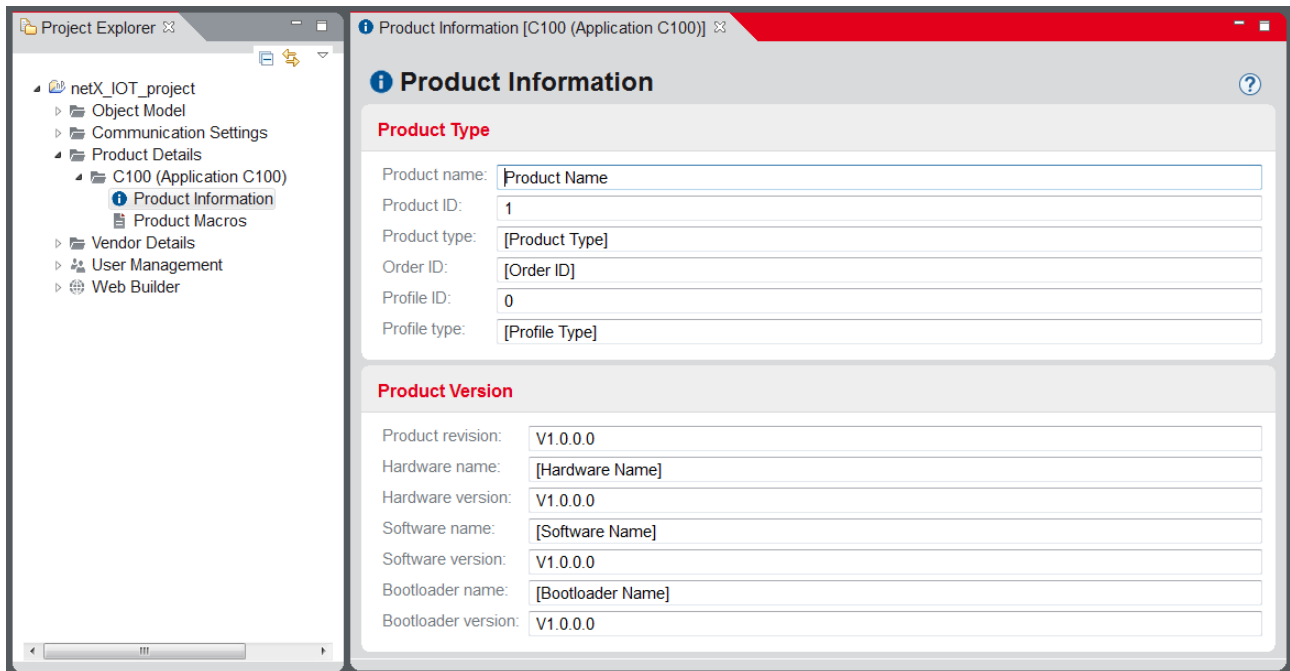


Abbildung 72: Product Details – Product Information (Beispiel)

- Wählen Sie im Product Information-Editor unter **Product Type** den Produkttyp, bzw. ergänzen Sie unter **Product Version** die Angaben zur Produktversion.



Hinweis:

Verwenden Sie für ein Produkt immer den gleichen Produktnamen, damit Sie die Daten zu diesem Produkt immer wiedererkennen!

Der unter **Product Type** im Feld **Product name** eingegebene Produktnamen wird in der Gerätebeschreibungsdatei verwendet. Verwenden Sie den gleichen Produktnamen wie bei der Eingabe in der Dialogbox **Add New Product** im Feld **Product name**, welcher unter Product Details im Ordernamen für das Produkt erscheint.

- ⇒ Die Produktinformationen werden in der Gerätebeschreibungsdatei verwendet bzw. wirken sie sich in verschiedenen Standardobjekten aus. Über die Standardobjekte können die Produktinformationen an bestimmte Orte zugeordnet bzw. gemappt werden.

Beispiel C100

Project Explorer

- netX_IOT_project
 - Object Model
 - Communication Settings
 - Product Details
 - C100 (Application C100)**
 - Product Information**
 - Product Macros
 - Vendor Details
 - User Management
 - Web Builder
 - Product Build

Product Information [C100 (Application C100)]

Product Information

Product Type

Product name:

Product ID:

Product type:

Order ID:

Profile ID:

Profile type:

Product Version

Product revision:

Hardware name:

Hardware version:

Software name:

Software version:

Bootloader name:

Bootloader version:

Abbildung 73: Product Details – Product Information (Beispiel)

5.6 Herstellerangaben eintragen

Herstellerangaben (Vendor Information)

Die Herstellerangaben sind im gesamten Projekt für alle Geräte gleich.

- Um die Herstellerangaben einzugeben, machen Sie im Fenster **Project Explorer** einen Doppelklick auf den Ordner **Vendor Details > Vendor Information**.
- ⇒ Der Vendor Information-Editor (rechts) erscheint.

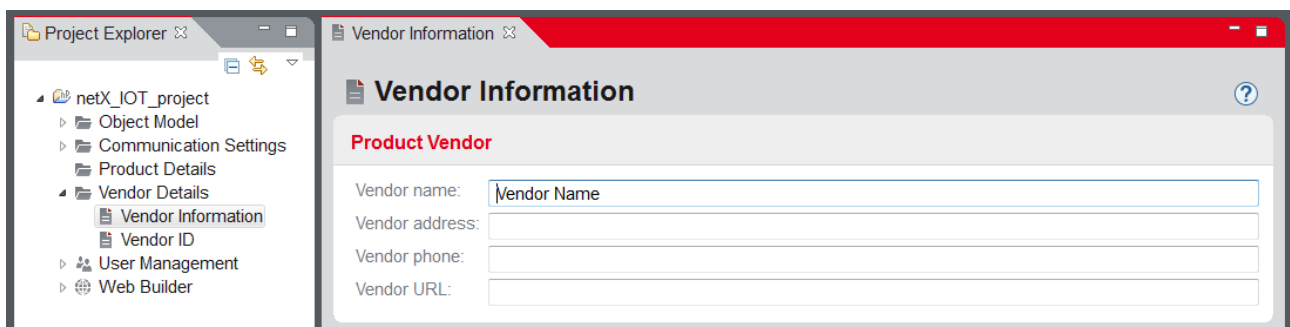


Abbildung 74: Vendor Details > Vendor Information.

In der nachfolgenden Tabelle können Sie sehen, welche Eingaben im Product Information-Editor in welchem Referenzobjekt in netX Studio Engineering Tool im **Project Explorer** unter dem Ordner **Object Model > Object Definitions > Standard > Generic** die Angaben eingetragen sind.

Herstellerinformation	Max. Anzahl Zeichen	Zeichentyp	netPROXY-Objekt			CIP-Objekt
Vendor name (Herstellernamen)	64	Printable ASCII string; nur folgende Zeichen sind erlaubt: 32 (0x20) ... 126 (0x7E)	0x10001000, Element 0	[Device] VendName	Hilscher Gesellschaft fuer Systemautomation mbH	-
Vendor address (Herstelleradresse)	128		0x10001000, Element 1	[Device] VendAdresse	Rheinstrasse 15, D-65795 Hattersheim	-
Vendor phone (Telefonnummer Hersteller)	64		0x10001000, Element 2	[Device] VendPhone	+49(0)6190/9907-0	-
Vendor URL (URL des Herstellers)	64		0x10001000, Element 3	[Device] VendURL	http://www.hilscher.com	-

Tabelle 22: Herstellerinformationen (Beispiel)

Um die Herstellerinformationen einzugeben, gehen Sie wie folgt vor:

- Geben Sie unter **Vendor name** (Pflichtfeld) den Herstellernamen an.
- Geben Sie unter **Vendor address**, **Vendor phone** und **Vendor URL** die Adresse, Telefonnummer und die Webadresse an.
- ⇒ Die Angaben erscheinen im Vendor Information-Editor.

Herstellerkennung (Vendor ID)

Die Herstellerkennung (Vendor ID) müssen Sie für jedes Kommunikationssystem separat festlegen.

- Um die Herstellerkennung einzugeben, machen Sie im Fenster **Project Explorer** einen Rechtsklick auf den Ordner **Vendor Details > Vendor ID**.
- Das Vendor Information-Editor erscheint.

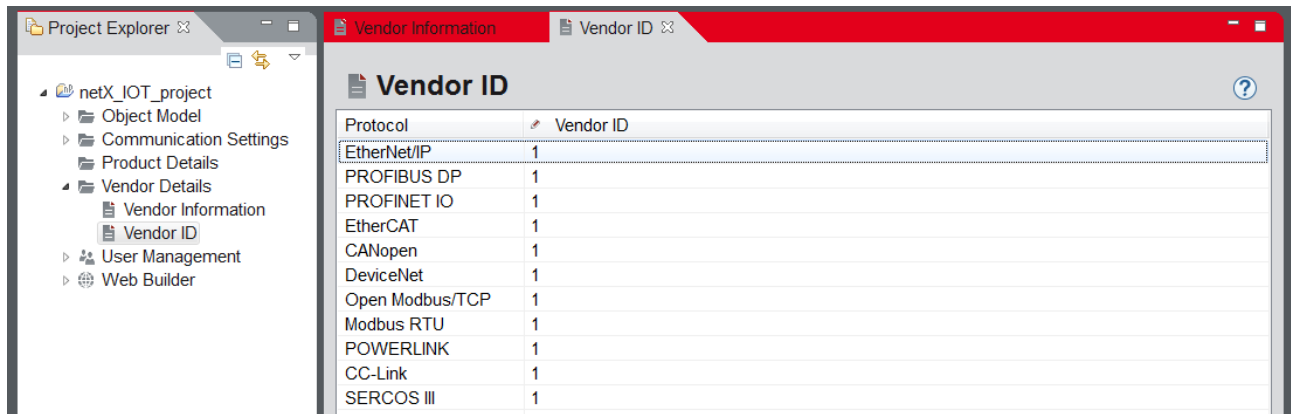


Abbildung 75: Vendor Details > Vendor ID (Beispiel mit Herstellerkennungen der Firma Hilscher)



Hinweis:

Sie benötigen eigene Herstellerkennungen (nach ODVA)!

- Geben Sie in der Spalte **Vendor ID** Ihre Herstellerkennung für die in Ihrem Projekt verwendeten Kommunikationssysteme an.

5.7 Benutzerverwaltung konfigurieren

Unter **User Management** können Sie in netX Studio Engineering Tool dem Funktionsumfang Ihres Gerätes Benutzernamen und Rollen zur Benutzerverwaltung am WebServer bzw. am FTP-Server für Ihr Gerät hinzufügen. Weiterhin können Sie eine Übersicht zur Verteilung der Zugriffsrechte auf Ihr Gerät für den WebServer bzw. den FTP-Server aufrufen. Die vorgenommenen Einstellungen zur Benutzerverwaltung werden beim Build-Prozess in der Konfigurationsdatei Security.cfg gespeichert. Die für die Rollen Administrator, Supervisor und User festgelegten Namen und Rollen können am WebServer bzw. am FTP-Server verwendet werden.

- Um ein Benutzerkonto anzulegen, machen Sie im Fenster **Project Explorer** einen Doppelklick auf den Ordner **User Management > User Accounts**.

➤ Der User Accounts-Editor erscheint.

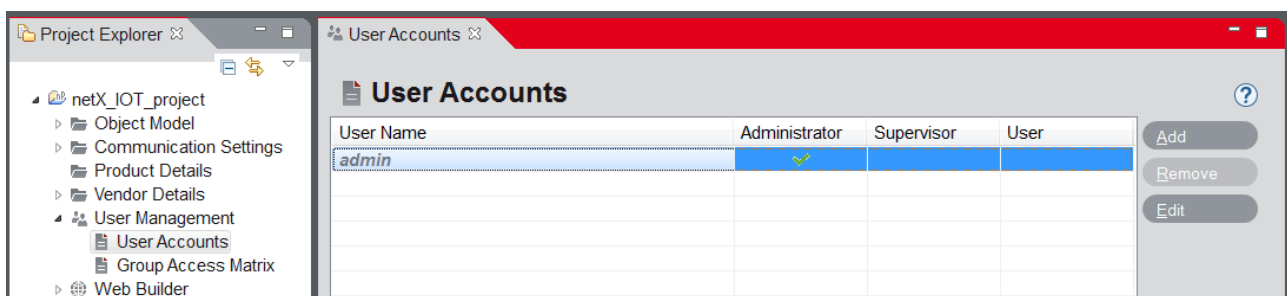


Abbildung 76: User Management - User Accounts-Editor

Aktion	Bedeutung
Add	Über Add können Sie ein neues Benutzerkonto hinzufügen.
Remove	Über Remove können Sie ein Benutzerkonto entfernen.
Edit	Über Edit können Sie ein Benutzerkonto editieren.

Tabelle 23: Bedienelemente des User Accounts-Editors

Benutzerkonto für Default-Administrator

Das Benutzerkonto für den Administrator ist standardmäßig vorkonfiguriert. Dieses kann nicht gelöscht werden. Das Administratorpasswort können Sie individuell anpassen.



Hinweis:

Ändern Sie das Default-Passwort für den Administrator.

- Klicken Sie **Edit** an.
- Der Dialog User Accounts erscheint.

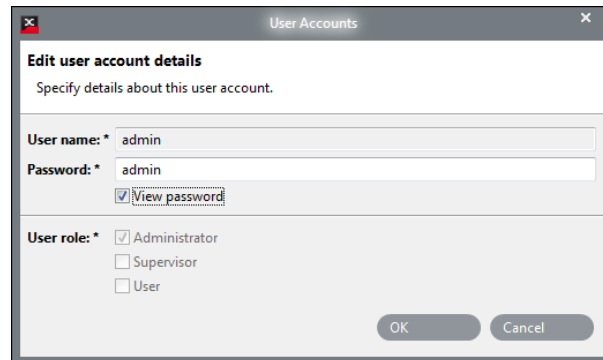


Abbildung 77: Benutzerkonto Admin editiert

- Haken Sie **View password*** an, um das Passwort zu sehen.
- Passen Sie im Feld **Password*** das Passwort an und verlassen Sie den Dialog über **OK**.

Benutzerkonten anlegen

Sie können Benutzerkonten für die Rollen Administrator, Supervisor und User anlegen. Neben dem Default-Benutzerkonto „admin“ können Sie weitere Benutzerkonten für zusätzliche Administratoren anlegen.

- Klicken Sie **Add** an.
- ⇒ Der Dialog User Accounts erscheint.

Abbildung 78: User Accounts – Angaben zum neuen Benutzerkonto hinzufügen

Parameter	Bedeutung	Wertebereich/Wert
User name*	Benutzername (Pflichtfeld)	Zeichenkette
Password*	Passwort für den Benutzer (Pflichtfeld)	Zeichenkette aus Groß- und Kleinbuchstaben, Zahlen sowie Sonderzeichen
User role*	Rolle des Benutzers (Pflichtfeld) Ein Benutzer kann nur eine oder auch mehrere Rollen in Kombination haben.	Administrator, Supervisor, User Default: User

Tabelle 24: Angaben zum Benutzerkonto

- Geben Sie im Feld **User name*** einen Benutzernamen ein.
- Geben Sie im Feld **Password*** ein Passwort für den Benutzer ein.
- Haken Sie unter **User role*** an, welche Rolle bzw. welche Rollen der Benutzer haben soll.

Abbildung 79: Neues Benutzerkonto für User anlegen (Beispiel)

- Verlassen Sie den Dialog User Accounts jedes Mal über **OK**.
- ⇒ Im User Accounts-Editor erscheinen die Namen und die Angabe der festgelegten Rollen.

Group Access Matrix

- Um die Übersicht zur Verteilung der Zugriffsrechte für den WebServer bzw. den FTP-Server aufzurufen, machen Sie im Fenster **Project Explorer** einen Doppelklick auf den Ordner **User Management > Group Access Matrix**.
- ⇒ Der Group Access Matrix-Editor erscheint. Darin sehen Sie eine Übersicht der Rollen und Rechte am WebServer und am FTP-Server.

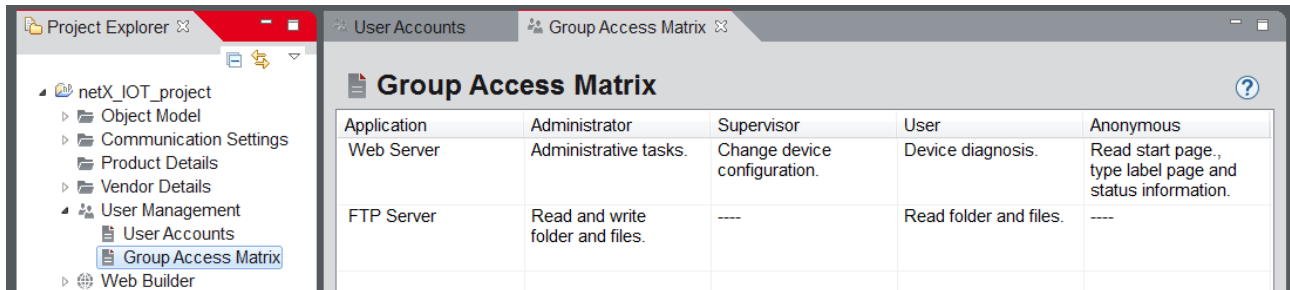


Abbildung 80: Übersicht zur Verteilung der Zugriffsrechte für den WebServer bzw. den FTP-Server

5.8 Webinhalte erstellen und verwalten



Angaben zur Erstellung und Verwaltung von Webinhalten finden Sie in der Application note, **netX Studio Engineering Tool, Web Builder** [6].

5.9 Produktdaten erzeugen

Im Build-Prozess werden die Produktdaten erzeugt, einschließlich der Konfigurationsdatei, bzw. Dateien zur Objektkonfiguration, sowie Webcontent-Dateien und die Header-Dateien. Die Produktdaten beinhalten die Konfiguration (Definition der Objekte), sowie alle zusätzlichen Einstellungen (Funktionsumfang, Kommunikation, Produktdetails, Herstellerangaben).

Eine Firmware-Image-Datei kann über die Export-Funktion von netX Studio Engineering Tool erstellt werden. Siehe Abschnitt Firmware-Image erstellen. Die Firmware-Image-Datei enthält die Einzeldateien aus dem Build-Prozess, ohne die Header-Dateien, jedoch zusätzlich den Second-Stage-Boot-Loader (SSBL).

5.9.1 Build-Prozess

Build-Prozess starten

Nach Abschluss der Objektdefinition und der Ergänzung von Einstellungen und Daten für Ihre Produkte können Sie den Build-Prozess starten.

- Gehen Sie dazu in netX Studio Engineering Tool auf **Project > Build**.

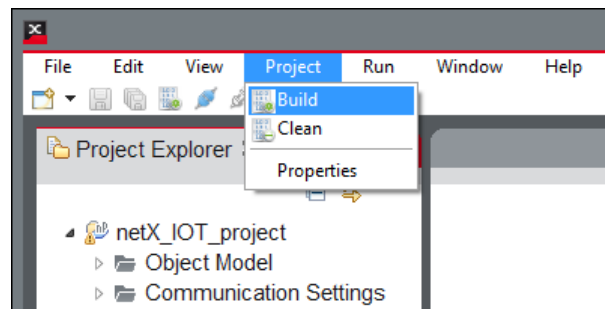


Abbildung 81: Geräteerstellungsprozess (Build-Prozess) starten

- Für Sekunden erscheint ein Dialog mit der Anzeige des Fortschritts.

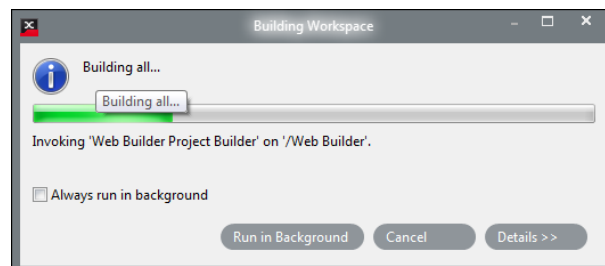


Abbildung 82: Laufender Build-Prozess

- ⇒ Im Project Explorer erscheint der Ordner **Product Build** mit den Unterordnern **Application Interface** und **Products**.

Verzeichnis Product Build mit den erzeugten Produktdateien

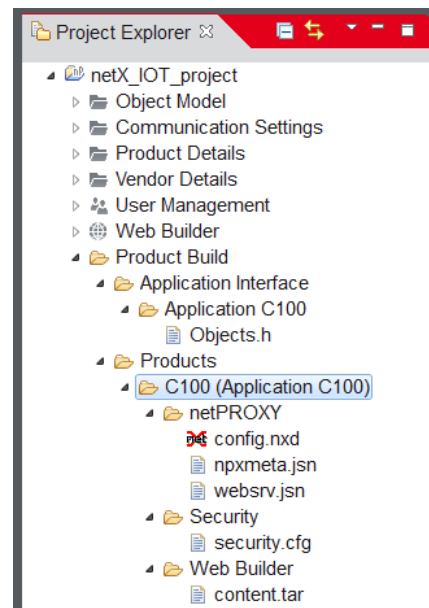
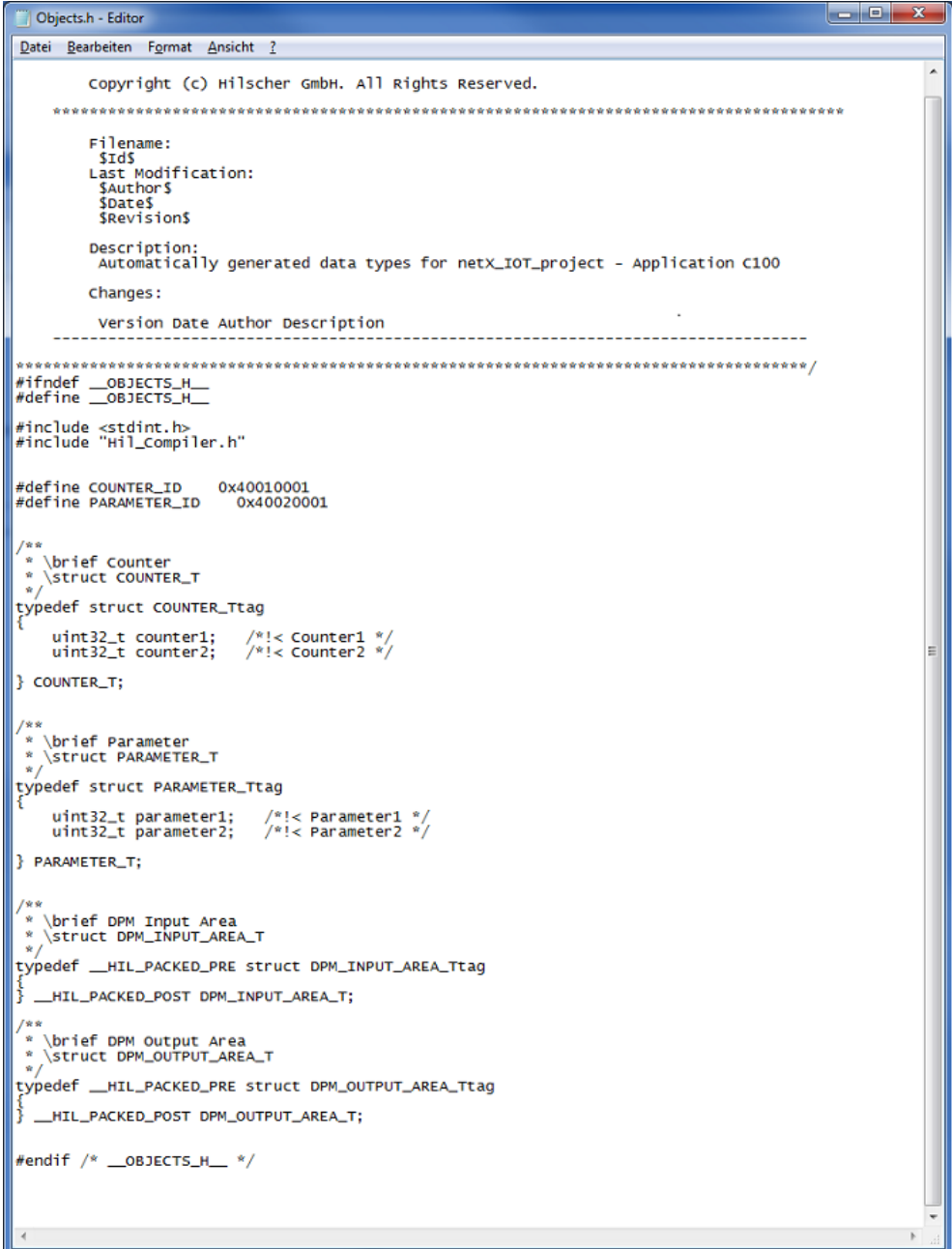


Abbildung 83: Verzeichnis Product Build

Ordner	Dateiname	Beschreibung
Application Interface-Unterordner		
Header-Datei	Objects.h	Definition der anwenderspezifischen Objekte für Programmiersprache C
Products-Unterordner		
netPROXY	config.nxd	Datei zur netPROXY-Objektkonfiguration (Objekte, OPC UA-Server-Parameter bzw. MQTT-Client-Parameter)
	npxmeta.jsn	Datei mit Meta-Informationen
	npxopcu.jsn	Datei mit Objektparametern für OPC UA
	websrv.jsn	Datei mit WebServer-Objektliste
Security	security.cfg	Konfigurationsdatei mit Inhalten zur Benutzerverwaltung am FTP-Server bzw. am WebServer
Web Builder	content.tar	Packdatei mit Webinhalte für den WebServer (optional)

Tabelle 25: Produktdateien im Build-Prozess von netX Studio Engineering Tool

- Unter dem Ordner **Application Interface** liegt für jede Applikationsvariante ein Ordner, worin sich die Header-Datei **Objects.h** befindet. Über einen Doppelklick auf die Datei öffnet sich diese.



```

Objects.h - Editor
Datei Bearbeiten Format Ansicht ?

Copyright (c) Hilscher GmbH. All Rights Reserved.

*****
Filename:
$Id$
Last Modification:
$Author$
$Date$
$Revision$

Description:
Automatically generated data types for netX_IOT_project - Application C100

Changes:
-----
Version Date Author Description
-----
*****
#ifndef __OBJECTS_H__
#define __OBJECTS_H__

#include <stdint.h>
#include "Hil_Compiler.h"

#define COUNTER_ID 0x40010001
#define PARAMETER_ID 0x40020001

/**
 * \brief Counter
 * \struct COUNTER_T
 */
typedef struct COUNTER_Ttag
{
    uint32_t counter1; /*!< Counter1 */
    uint32_t counter2; /*!< Counter2 */
} COUNTER_T;

/**
 * \brief Parameter
 * \struct PARAMETER_T
 */
typedef struct PARAMETER_Ttag
{
    uint32_t parameter1; /*!< Parameter1 */
    uint32_t parameter2; /*!< Parameter2 */
} PARAMETER_T;

/**
 * \brief DPM Input Area
 * \struct DPM_INPUT_AREA_T
 */
typedef __HIL_PACKED_PRE struct DPM_INPUT_AREA_Ttag
{
    __HIL_PACKED_POST DPM_INPUT_AREA_T;
}

/**
 * \brief DPM Output Area
 * \struct DPM_OUTPUT_AREA_T
 */
typedef __HIL_PACKED_PRE struct DPM_OUTPUT_AREA_Ttag
{
    __HIL_PACKED_POST DPM_OUTPUT_AREA_T;
}

#endif /* __OBJECTS_H__ */

```

Abbildung 84: Beispiel, Header-Datei Objects.h

**Hinweis:**

Die Header-Datei **Objects.h** enthält nur Objekte aus dem Object-ID-Bereich 0x4xxxxxxx (siehe Abschnitt *Objekt-IDs* ► Seite 111).

- Unter dem Ordner **Products** befindet sich für jedes Gerät ein Unterordner mit einer Bezeichnung der Applikationsvariante mit den weiteren Unterordnern **netPROXY**, **Security** und **Web Builder**.
 - Der Unterordner **netPROXY** enthält die Datei **config.nxd** mit der Objektkonfiguration und den Kommunikationseinstellungen für das entsprechende Gerät, die Datei **npjmeta.jsn** mit Meta-Informationen, sowie die Datei **websrv.jsn** mit der WebServer-Objektliste.
 - Der Unterordner **Security** enthält die Datei **security.cfg** mit Einstellungen zur Benutzerverwaltung.

- Der Unterordner **Web Builder** enthält die Datei **content.tar** mit Inhalten für den WebServer für das entsprechende Gerät.

Clean – Build-Prozessprobleme beseitigen

Mithilfe der Clean-Funktion können Sie Build-Prozessprobleme beseitigen und ein Projekt-Build von grundauf neu generieren.

- Klicken Sie **Project > Clean** an.
- Der **Clean**-Dialog erscheint.

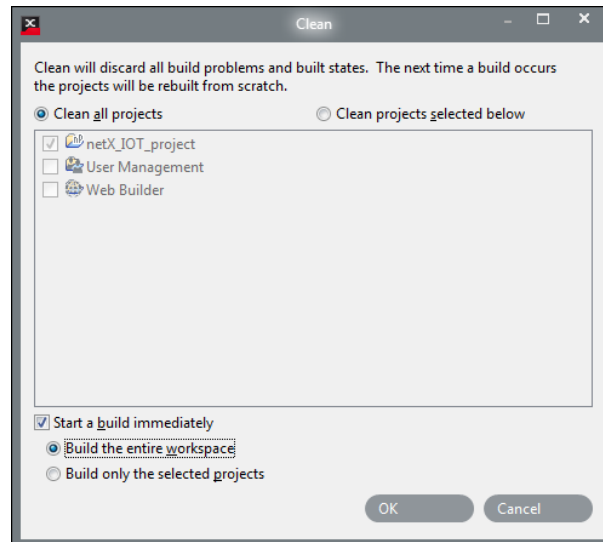


Abbildung 85: Project > Clean

- Im Build-Dialog können Sie konfigurieren, ob alle Projekte bereinigt werden sollen (Clean all projects) oder nur bestimmte von Ihnen ausgewählte Projekte (Clean projects selected below).
- Weiterhin können Sie konfigurieren, ob sofort ein neuer Build-Prozess gestartet werden soll (Start a build immediately) und wenn ja, ob der Build-Prozess für alle Projekte (Build the entire workspace) oder nur für ausgewählte Projekte (Build only the selected projects) erfolgen soll.

5.9.2 Produktdateien kopieren, entfernen

Windows-Verzeichnis mit Produktdateien öffnen

- Machen Sie in netX Studio Engineering Tool einen Rechtsklick auf **Product Build**.
- Das Kontextmenü **Show in Windows Explorer** erscheint.

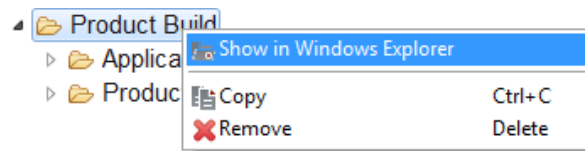


Abbildung 86: Kontextmenü Product Build > Show in Windows Explorer, Copy, Remove

- Klicken Sie **Show in Windows Explorer** an.
- Der Windows Explorer mit dem Ordner **Product Build** wird geöffnet. Die Produktdateien befinden sich in den Unterverzeichnissen Application Interface und Products.

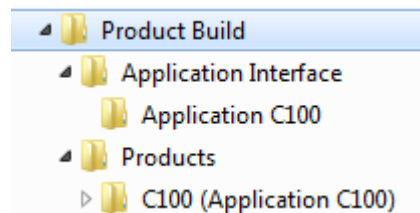


Abbildung 87: Ordner Product Build, Unterverzeichnisse Application Interface und Products

Produktdateien kopieren, entfernen

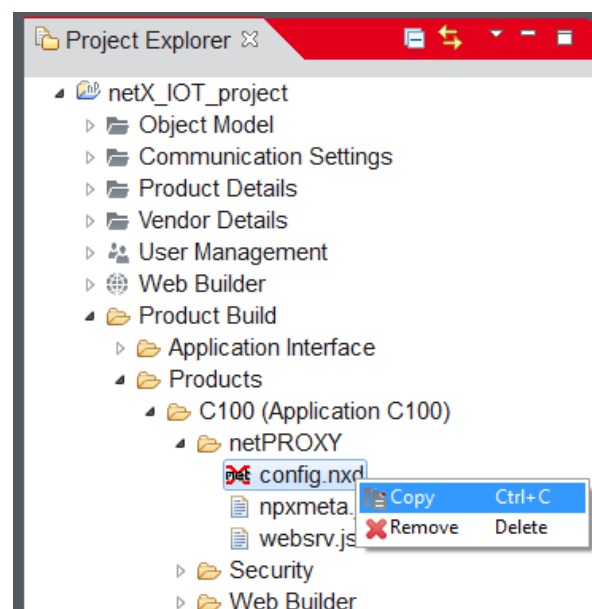


Abbildung 88: Kontextmenübefehle Copy, Remove

- Über **Copy** können Sie ein Unterverzeichnis oder eine Datei aus dem Product Build-Verzeichnis kopieren und im Windows Explorer ablegen.
- Über **Remove** können Sie ein Unterverzeichnis oder eine Datei aus dem Product Build-Verzeichnis entfernen.

5.9.3 Bedeutung der Produktdateien

Die im Build-Prozess erstellten Produktdateien sind Einzeldateien, worin die Objektdefinition und die Gerätekonfiguration sowie alle Einstellungen zum Gerät enthalten sind.

Datei	Dateiname	Beschreibung
Header-Datei	Objects.h	Enthält Informationen für den Anwendungsentwickler darüber, welche Objekte im Gerät vorhanden sind und wie auf die Objekte zugegriffen werden kann
netPROXY-Objektkonfiguration	config.nxd	Darin enthalten sind die Konfigurationsinformationen zur Objektdefinition mit der Definition der Geräteeigenschaften sowie die Festlegungen zu den Objektinstanzen zum Funktionsumfang der Geräte, sowie die Kommunikationseinstellungen.
	npxmeta.jsn	Datei mit Meta-Informationen
	npxopcu.jsn	Datei mit Objektparametern für OPC UA
	websrv.jsn	Datei mit WebServer-Objektliste
Benutzerverwaltung	Security.cfg	Konfigurationsdatei mit Inhalten zur Benutzerverwaltung am FTP-Server bzw. am WebServer
Webcontent (optional)	Content.tar	Packdatei mit Webinhalte für den WebServer (einschließlich Firmenlogos, Zähler etc.); findet Anwendung bei Ethernet-Systemen

Tabelle 26: Produktdateien des Build-Prozesses

5.10 Produktdateien herunterladen

Alle beim Product Build unter **Product (Application)** erzeugten Dateien (siehe in der Tabelle *Produktdateien im Build-Prozess von netX Studio Engineering Tool* [► Seite 92]) müssen in das Gerät übertragen werden. Dabei muss jede Datei einzeln heruntergeladen werden.



Hinweis:

Bei einem Product Build entstehen nicht in jedem Fall alle in der Tabelle *Produktdateien im Build-Prozess von netX Studio Engineering Tool* [► Seite 92] aufgeführte Dateien.

Um die Dateien in das Gerät zu übertragen, gehen Sie wie folgt vor:

- Machen Sie einen Rechtsklick auf **Product Build** und wählen Sie **Show in Windows Explorer**.
- Übertragen Sie mithilfe von netHOST.exe bzw. der Host-Applikation alle beim Product Build unter **Product Build > Products > Product (Application)** erzeugten Produktdateien einzeln nacheinander in das Gerät.



Details zur Benutzeroberfläche von netHOST bzw. zur Host-Applikation lesen Sie im zugehörigen Software-Handbuch nach.



Wichtig:

Nachdem alle beim Product Build unter **Product Build > Products > Product (Application)** erzeugten Produktdateien in das Gerät geladen worden sind, ist ein Geräte-Reset erforderlich.

- Führen Sie nach Abschluss des Downloads der Produktdateien einen Geräte-Reset aus.
- ⇒ Im Anschluss daran ist Ihr Gerät vorbereitet. D. h., ein Zugriff auf das Gerät mit OPC UA, MQTT oder dem WebBrowser ist möglich.

6 Import, Export

netX Studio Engineering Tool beinhaltet Import- und Exportfunktionen, die Sie für verschiedene Zwecke einsetzen können. In den nachfolgenden Abschnitten finden Sie weitere Detailangaben dazu.

Die Dialoge zu den Import- und Exportfunktionen können Sie an verschiedenen Stellen aufrufen.

- Allgemein über **File > Import ...** bzw. **File > Export ...**
- Im Fenster **Project Explorer** über einen Rechtsklick auf den Projektordner
- Nach Ausführung der **Build**-Funktion über einen Rechtsklick auf eine der Produktdateien (Objects.h, config.nxd, security.cfg etc.), die sich in den Unterverzeichnissen des **Product Build**-Ordners befinden.

6.1 Import

Über die Importfunktion in netX Studio Engineering Tool können Sie die aus einem bestehenden Projekt exportierten Objektdefinitionen in ein bzw. mehrere neue Projekte importieren.

Die Beschreibungen zur Vorgehensweise finden Sie hier nachfolgend.



Hinweis:

Vor einem Import ist es empfehlenswert, dass Sie bereits erstellte Objektdefinitionen durch einen Export (siehe Abschnitt *Export* ► Seite 103)) sichern, um ein versehentliches Überschreiben von Objektdefinitionen zu vermeiden.

Importdialog öffnen und Importinhalt festlegen

- Machen Sie im Fenster **Project Explorer** einen Rechtsklick auf den Projektordner.

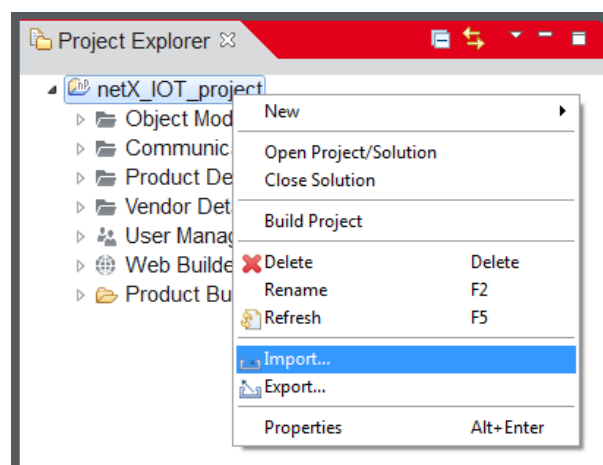


Abbildung 89: Importfunktion aufrufen (Beispiel)

- Klicken Sie **Import ...** an.
- Das Fenster **Import (Select)** erscheint.

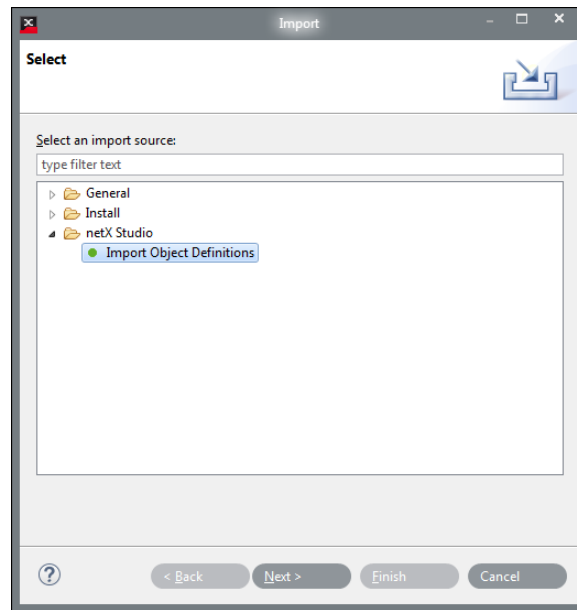


Abbildung 90: Importinhalt festlegen (Beispiel)

Parameter	Bedeutung	Wertebereich / Default-Wert
Select an import source	Anzeigefenster mit einer Ordnerstruktur zur Auswahl der Importinhalte. Relevant ist der Ordner netX Studio. Die übrigen Ordner sind nicht relevant. Im Ordner netX Studio können Sie den Importinhalt festlegen: <ul style="list-style-type: none"> • Import Object Definitions (Objektdefinitionen) 	General, Install, netX Studio Default: Ordner netX Studio
Filter [type filter text]	Feld zur Eingabe eines Filtertextes zur Auswahl eines Unterordners.	-

Tabelle 27: Erläuterungen zum Fenster Import (Select)

- Wählen Sie den Ordner **netX Studio**.
- Wählen Sie **Import Object Definitions** und klicken Sie **Next >** an.
- Das Fenster **Import Object Definitions (Object Definitions Import)** erscheint.

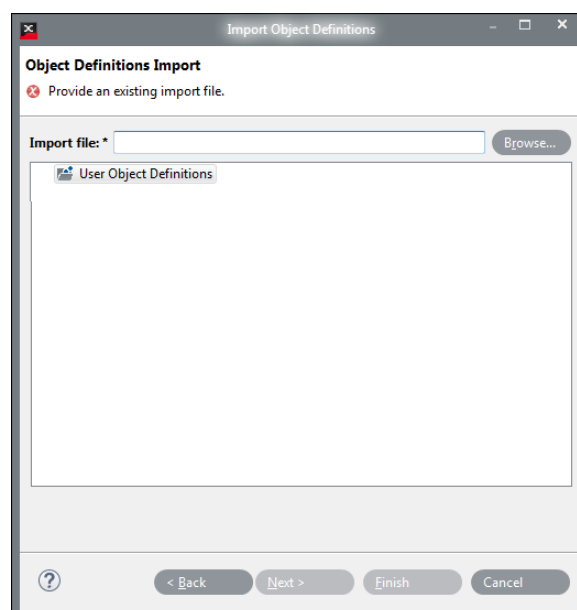


Abbildung 91: Importdatei wählen


Parameter	Bedeutung	Wertebereich / Default-Wert
Import file (Pflichteingabe)	Zeigt nach der Auswahl der Importdatei den Pfad zu der gewählten Datei mit den Objektdefinitionen einschließlich des Namens der zip-Datei.	
	Über Browse ... kann die Importdatei mit den Objektdefinitionen *.zip (z. B., User.zip) gewählt werden.	*.zip, z. B. User.zip
User Object Definitions	Erscheint nur der Ordner ‚User Object Definitions‘, kann eine Importdatei mit beliebigen Objektkategorien und Objektdefinitionen importiert werden. Erscheinen unter dem Ordner ‚User Object Definitions‘ weitere Ordner von Objektkategorien und Sie wählen einen dieser Ordner, müssen Sie, auf Warnungen achten, damit Sie Objektdefinitionen nicht versehentlich überschreiben.	Ordner ‚User Object Definitions‘, bzw. Ordner von untergeordneten Objektkategorien
Errors (optional)	Wird versucht einen Ordner einer Objektkategorie durch eine Importdatei mit Objektdefinitionen zu ersetzen, wobei einige der zu importierenden Objektdefinitionen bereits in einer anderen Objektkategorie des Projektes vorkommen, erscheint die Fehlermeldung: Can not import object with ID 0x4XXXXXXX because such an object exists outside of the destination category. [Das Objekt mit der ID 0x4XXXXXXX kann nicht importiert werden, da ein solches Objekt bereits außerhalb der Zielkategorie vorhanden ist.] Wählen Sie in diesem Fall einen anderen Ordner, z. B. ‚User Object Definitions‘ oder importieren Sie nur Objektdefinitionen, die im Projekt noch nicht außerhalb der Zielkategorie vorkommen.	

Tabelle 28: Erläuterungen zum Fenster Object Definitions import

- Klicken Sie **Browse...** an.
- Das Fenster **Öffnen** erscheint.

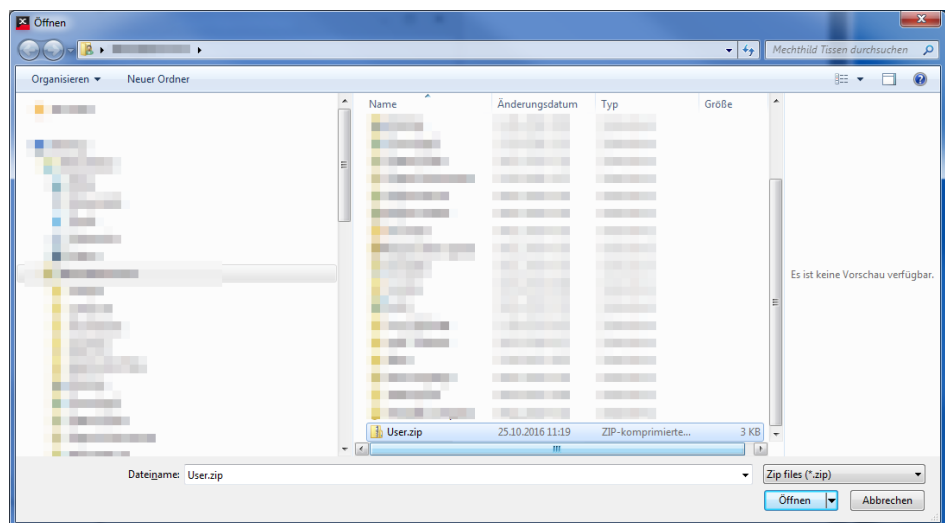


Abbildung 92: Dateiauswahl-Dialog für Importdatei (Beispiel User.zip)

- Wählen Sie die Importdatei mit den Objektdefinitionen (z. B. User.zip) und klicken Sie **Öffnen** an.
- Der Pfad zu der gewählten Datei mit den Objektdefinitionen (z. B. User.zip) wird im Fenster **Import Object Definitions** angezeigt.

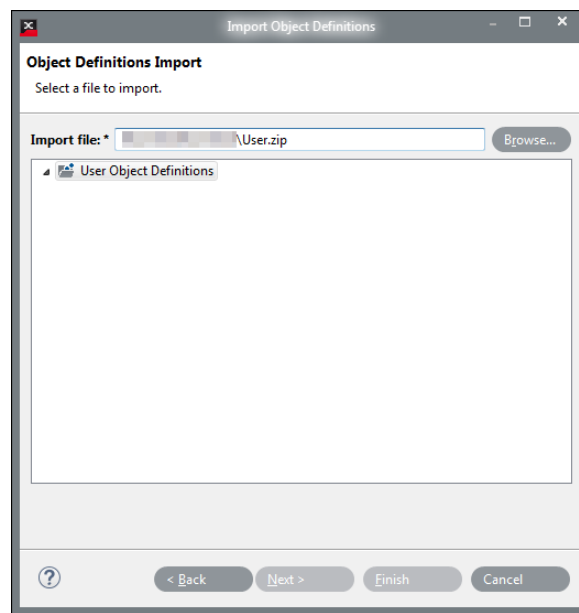


Abbildung 93: Importdatei gewählt (Beispiel User.zip)

- Im Fenster **Import Object Definitions** > **Finish** anklicken.

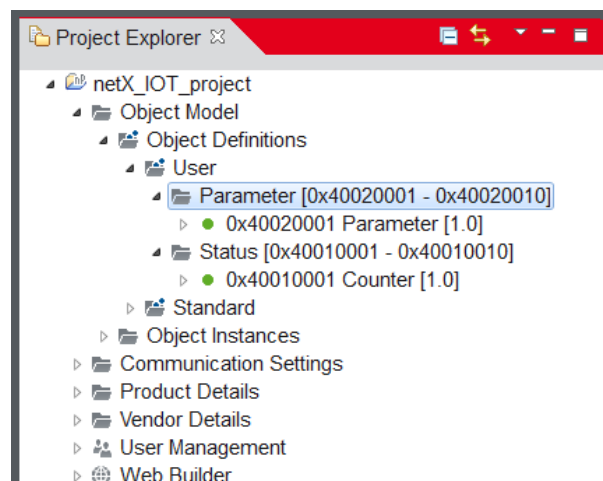


Abbildung 94: Neu importierte Objektdefinitionen (Beispiel)

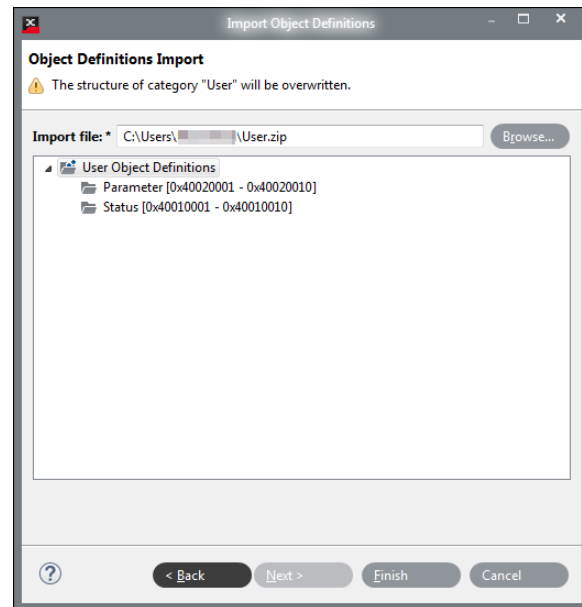
Warnung: Objektkategorie wird überschrieben

Abbildung 95: Importdatei gewählt (Beispiel User.zip) - Warnung, Kategorie „User“ wird überschrieben

6.2 Export

Über die Exportfunktion in netX Studio Engineering Tool können Sie

- die Objektdefinitionen eines bestehenden Projektes exportieren (um sie anschließend in ein bzw. mehrere neue Projekte importieren zu können),
- die Projektdokumentation eines Projektes exportieren.

Die Beschreibungen zur Vorgehensweise im Einzelnen finden Sie in den nachfolgenden Abschnitten.

6.2.1 Objektdefinitionen exportieren

Im Dialog zum Export der Objektdefinitionen (Export Object Definitions) legen Sie den Ausgabeordner für die Datei mit den Objektdefinitionen fest. Gehen Sie wie folgt vor:

Export-Dialog öffnen und Exportinhalt festlegen

- Machen Sie im Fenster **Project Explorer** einen Rechtsklick auf den Projektordner.

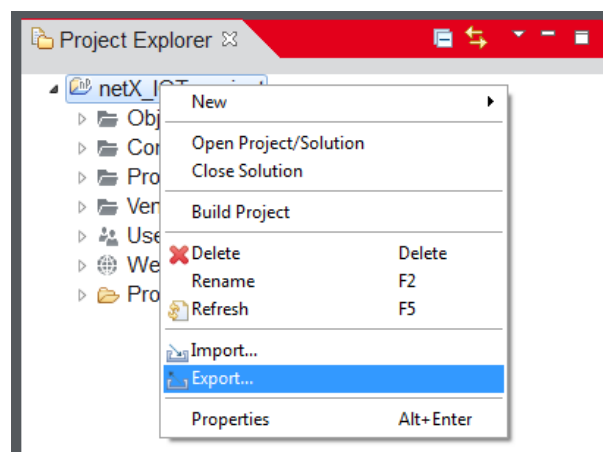


Abbildung 96: Export-Funktion aufrufen (Beispiel)

- Klicken Sie **Export ...** an.
- Das Fenster **Export (Select)** erscheint.

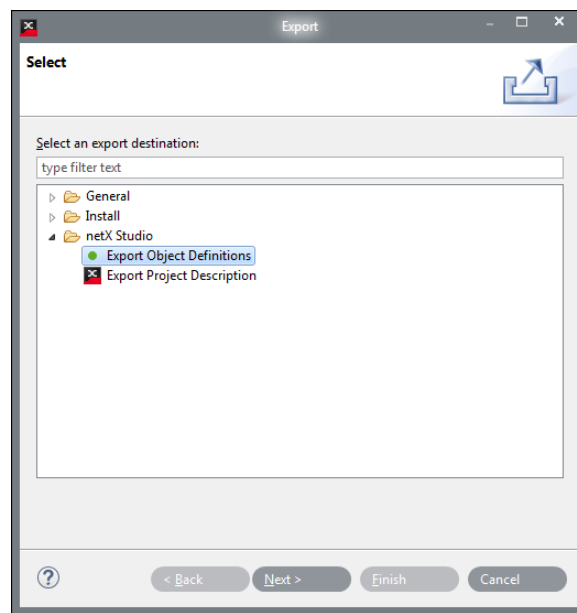


Abbildung 97: Exportinhalt festlegen - Export Object Definitions

Parameter	Bedeutung	Wertebereich / Default-Wert
Select an export destination	Anzeigefenster mit einer Ordnerstruktur zur Auswahl der Exportinhalte. Relevant ist der Ordner netX Studio. Die übrigen Ordner sind nicht relevant. Im Ordner netX Studio können Sie den Exportinhalt festlegen: <ul style="list-style-type: none"> • Export Object Definitions (Objektdefinitionen), • Export Project Description (Projektdokumentation) 	General, Install, netX Studio Default: Ordner netX Studio
Filter [type filter text]	Feld zur Eingabe eines Filtertextes zur Auswahl einer Exportfunktion.	-

Tabelle 29: Erläuterungen zum Fenster Export (Select)

- Wählen Sie den Ordner **netX Studio**.
- Wählen Sie **Export Object Definitions** und klicken Sie **Next >** an.
- Das Fenster **Export Object Definitions** erscheint.

Ausgabeordner für die Datei mit den Objektdefinitionen festlegen

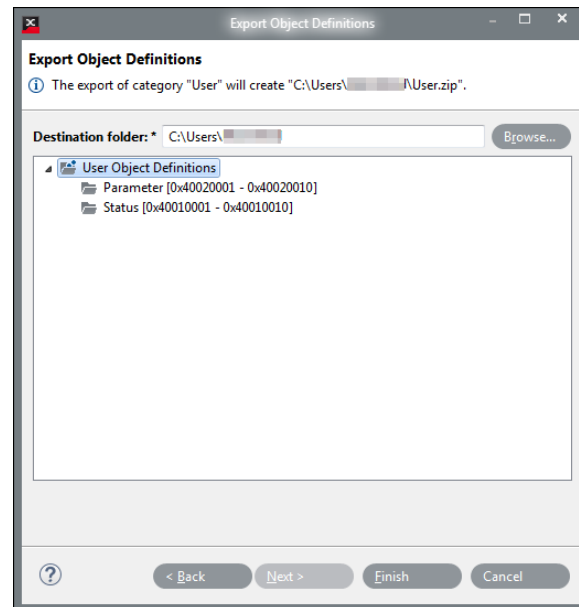


Abbildung 98: Ausgabeordner festgelegt

Parameter	Bedeutung	Wertebereich / Default-Wert
Export Object Definitions	Anweisungstexte: Provide an existing destination folder. [Stellen Sie einen bestehenden Ausgabeordner bereit.] The export of category „XXX“ will create [path and export file name]. [Beim Export der Kategorie „XXX“ wird [Pfad und Exportdateiname] erstellt.]	
Destination folder* (Plichteingabe)	Zeigt nach der Auswahl den Pfad zum aktuellen Ausgabeordner für die Exportdatei mit den Objektdefinitionen.	
Browse...	Über Browse ... kann ein Ausgabeordner für die Exportdatei mit den Objektdefinitionen geändert oder neu angelegt werden (standardmäßig auf Ihrem PC).	C:\User\[Benutzername]

Tabelle 30: Erläuterungen zum Fenster Export Object Definitions

- Über **Browse...** können Sie den Ausgabeordner suchen oder einen neuen Ausgabeordner erstellen.

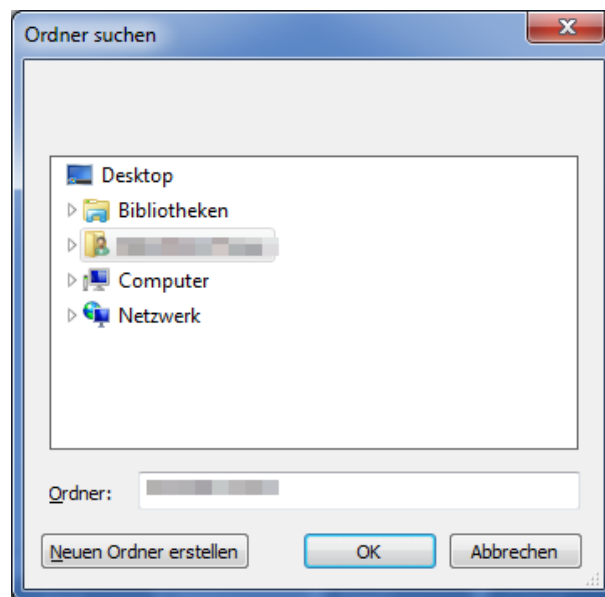


Abbildung 99: Ordner suchen

- Den Pfad wählen und **OK** anklicken.
- Im Fenster **Export Object Definitions > Finish** anklicken.
- ⇒ Die Datei mit den Objektdefinitionen (z. B. User.zip) wird an dem gewählten Ablageort gespeichert. Wird der gleiche Ablageort mehrfach verwendet, erscheint *keine* Abfrage, ob Sie die Datei mit den Objektdefinitionen überschreiben wollen.

6.2.2 Projektbeschreibungen exportieren

Im Dialog zum Export der Projektbeschreibung (Export Project Description) legen Sie die Inhalte für die Projektdokumentation und den Ausgabeordner fest. Gehen Sie wie folgt vor:

Export-Dialog öffnen und Exportinhalt festlegen

- Machen Sie im Fenster **Project Explorer** einen Rechtsklick auf den Projektordner.

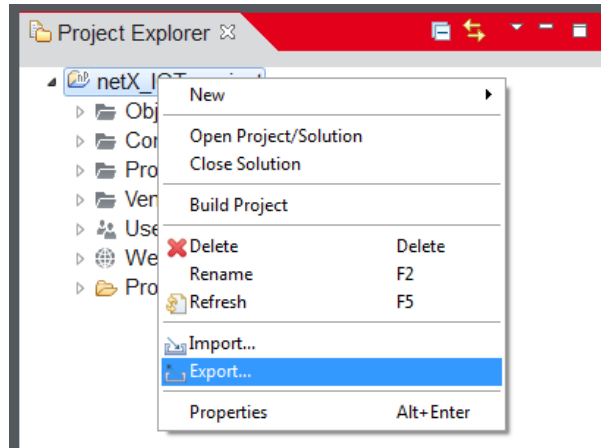


Abbildung 100: Export-Funktion aufrufen (Beispiel)

- Klicken Sie **Export ...** an.
- Das Fenster **Export (Select)** erscheint.

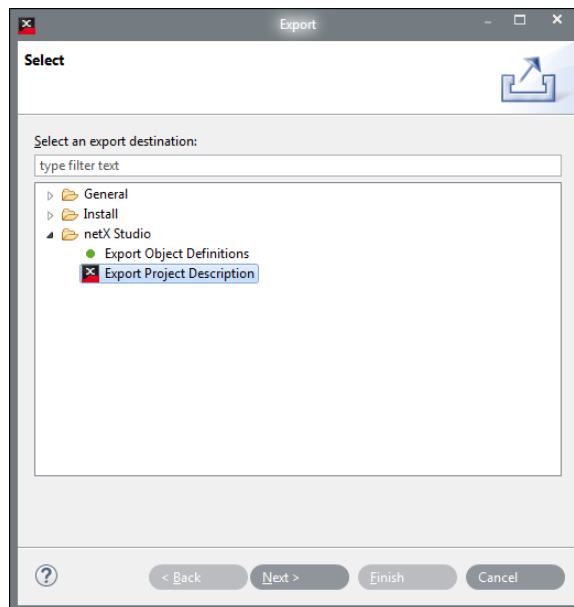


Abbildung 101: Exportinhalt festlegen - Export Project Description

Parameter	Bedeutung	Wertebereich / Default-Wert
Select an export destination	Anzeigefenster mit einer Ordnerstruktur zur Auswahl der Exportinhalte. Relevant ist der Ordner netX Studio. Die übrigen Ordner sind nicht relevant. Im Ordner netX Studio können Sie den Exportinhalt festlegen: <ul style="list-style-type: none"> • Export Object Definitions (Objektdefinitionen), • Export Project Description (Projektdokumentation) 	General, Install, netX Studio Default: Ordner netX Studio
Filter [type filter text]	Feld zur Eingabe eines Filtertextes zur Auswahl eines Unterordners.	-

Tabelle 31: Erläuterungen zum Fenster Export (Select)

- Wählen Sie den Ordner **netX Studio**.
- Wählen Sie **Export Project Description** und klicken Sie **Next >** an.
- Das Fenster **Export Project Description (Select projects to export and destination folder)** erscheint.

Inhalte für die Projektdokumentation festlegen

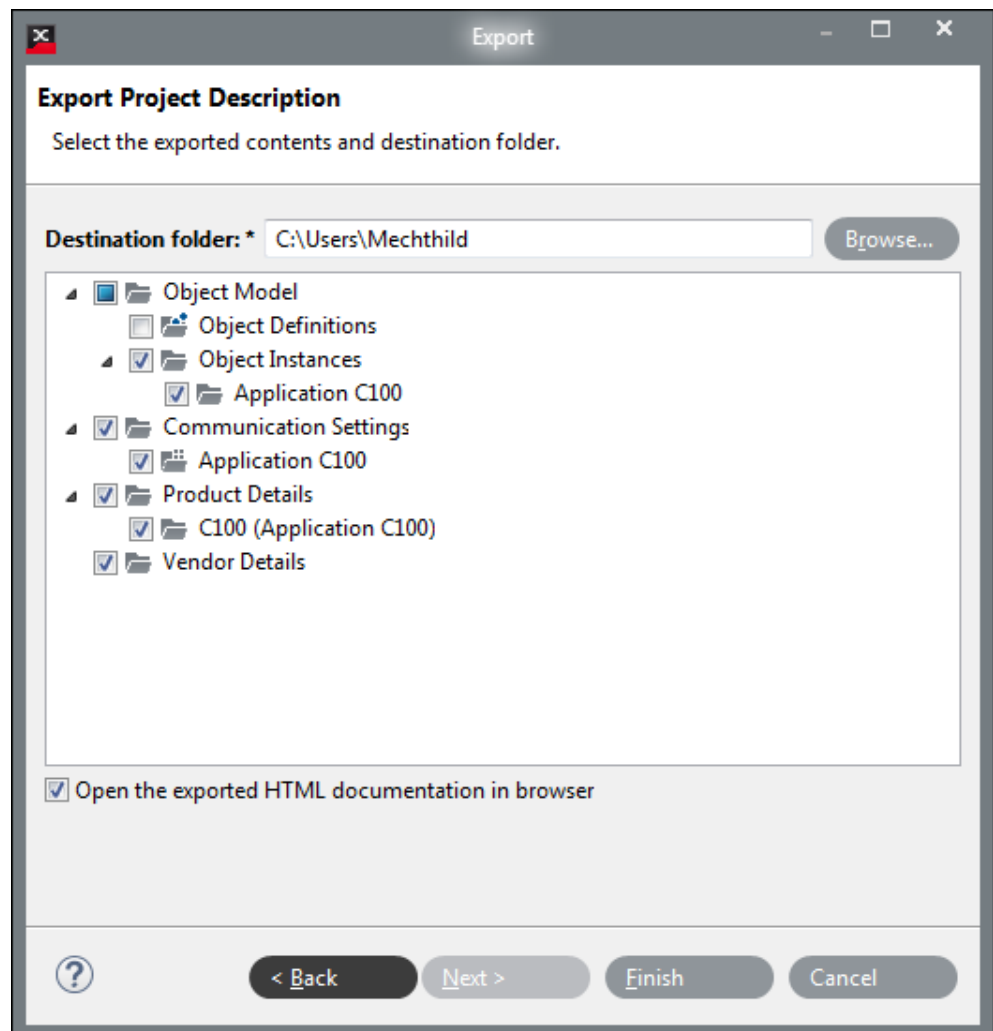


Abbildung 102: Inhalte für die Projektdokumentation und Ausgabeordner festlegen


Parameter	Bedeutung	Wertebereich / Default-Wert
File system	Anweisungstext: Select projects to export and destination folder. [<i>Projekte für den Export und Zielordner auswählen.</i>]	
Destination Folder* (Pflichtfeld)	Zeigt den Pfad zum aktuell Ausgabeordner für die Datei mit der Projektdokumentation. Der Dateiname [<i>Projektname</i>].html (im Beispiel Serie B.html) und kann hier nicht angepasst werden.	
	Über Browse ... kann ein Ausgabeordner für die Datei mit den Projektdokumentation gewählt oder neu angelegt werden (standardmäßig auf Ihrem PC).	C:\User\[<i>Benutzername</i>]
Auswahldialog für die Projekteinhalte	Der Auswahldialog umfasst alle Projekteinhalte, <ul style="list-style-type: none"> • zum Objektmodell (Object Model), • zu den Kommunikationseinstellungen (Communication Settings), • zu den Produktdetails (Product Details), • zu den Herstellerangaben (Vendor Details). Standardmäßig davon ausgenommen sind die Objektdefinitionen, dies können aber manuell gewählt werden.	
Open the exported HTML documentation in browser	Die exportierte Projektdokumentation wird nach jedem Export direkt in einem Browser angezeigt.	angehakt, nicht angehakt, Default: angehakt

Tabelle 32: Erläuterungen zum Fenster Export Project Description

➤ Klicken Sie **Finish**.

⇒ Die Projektdokumentation erscheint in Ihrem Browser.

Die Projektdokumentation umfasst alle zuvor im Auswahldialog gewählten Projekteinhalte. Durch Scrollen im Browser können Sie alle Inhalte anzeigen.

Project : netX_IOT_project**Object Model****Object Instances****Application C100****0x40010001 Status**

Name	Variable name	Data type	Value	Attributes	Description
Counter1	counter1	uint32	0	None	
Counter2	counter2	uint32	0	None	

0x40020001 Parameter

Name	Variable name	Data type	Value	Attributes	Description
Parameter1	parameter1	uint32	0	None	
Parameter2	parameter2	uint32	0	None	

Communication Settings**Application C100****Product Details****C100 (Application C100)****Product Type**

Property	Value
Product Name	C100
Product ID	C100 (Application C100)
Product Type	COMX
Order ID	2000
Profile ID	3000
Profile Type	Standard

Product Version

Property	Value
Product Revision	V1.0.0.0
Hardware Name	C100
Hardware Version	V1.0.0.0
Software Name	C100
Software Version	V1.0.0.0
Bootloader Name	C100_BSL
Bootloader Version	V1.0.0.0

Macros

Macros	Expands to...
--------	---------------

Vendor Details**Vendor Information**

Property	Value
Vendor Name	Vendor Name
Vendor Address	
Vendor Phone	
Vendor URL	

Vendor ID Table

Protocol	Vendor ID
EtherNet/IP	1
PROFIBUS DP	1
PROFINET IO	1
EtherCAT	1
CANopen	1
DeviceNet	1
Open Modbus/TCP	1
Modbus RTU	1
POWERLINK	1
CC-Link	1
SERCOS III	1

Abbildung 103: Ausschnitt Produktdokumentation im html-Format

7 Anhang

7.1 Referenzdaten für netPROXY

7.1.1 Objekt-IDs

Folgende Übersichtstabelle listet die festgelegten Wertebereiche für Objekte auf, die beim Einsatz der objektorientierten Datenstrukturierung mit netPROXY mit dem Geräteentwicklungswerkzeug netX Studio Engineering Tool eine Rolle spielen und unterschieden werden müssen bzw. welche Object-IDs von netX Studio Engineering Tool in den Bereichen 0 bis 3 belegt werden.

Vom Anwender in netX Studio Engineering Tool erstellte Objekte liegen immer im Object-ID-Bereich 4.

Wertebereich Object-ID	Bereichsname	Anmerkung
0x0xxx xxxx (ohne 0x0000 0000)	netPROXY	z. B. Verzeichnis, Event-Log etc. in netX Studio Engineering Tool
0x1xxx xxxx	System	z. B. allgemeine Dienste. Hierzu gehören unter anderem Objekte, die in netX Studio Engineering Tool unter Product Details angelegt wurden.
0x2xxx xxxx	Kommunikationsprotokoll	Objekt-ID-Bereich, der für Objekte des Kommunikationsprotokolls reserviert ist.
0x3xxx xxxx	Hilscher	Von Hilscher erstellte Objekte, z. B. Host-Interface, Peripherie etc.
0x4xxx xxxx	Anwender	Vom Anwender in netX Studio Engineering Tool erstellte Objekte. Diese sind in der Objects.h-Datei enthalten.
0x5xxx xxxx ... 0xFxxx xxxx	Reserviert	

Tabelle 33: Festgelegte Wertebereiche für Objekt-IDs



Weitere Informationen zur Belegung der in Tabelle *Festgelegte Wertebereiche für Objekt-IDs* [► Seite 111] aufgeführten Object-ID-Bereiche finden Sie im Handbuch **Technical reference, netPROXY, Objects and packages** [2] (siehe Abschnitt Referenzen).

7.1.2 Element-Datentypen, Größe, Anzahl, Grenzen

Die nachfolgende Tabelle *Objektelement-Datentypen, Größe, Anzahl, Grenzen* [► Seite 112] führt die für

- die in netX Studio Engineering Tool konfigurierbaren Elemente gültigen Element-Datentypen auf,
- die zugehörigen netPROXY-Datentypen,
- die Größe für jeden Datentyp in Bytes,
- die maximale Anzahl Datentypen pro Element für jeden Datentyp sowie
- die minimale bzw. die maximale Zahl darstellbarer Werte für jeden Datentyp.

Der Default-Wert für alle numerischen Datentypen beträgt 0.

Die Default-Länge für alle numerischen Datentypen beträgt 1 (0 ist nicht erlaubt).

Ein „string“ hat als Default-Wert einen Leerstring. Die Default-Länge „string“ beträgt 255 Bytes.

netX Studio-Element-Datentypen	netPROXY-Datentypen	Größe in Bytes	Max. Anzahl Datentypen = Max. Default-Wert	Minimale Zahl darstellbarer Werte	Maximale Zahl darstellbarer Werte
boolean	NPX_TYPE_BOOLEAN	1	1	0	255
uint8	NPX_TYPE_UNSIGNED	1	1	0	255
uint16	NPX_TYPE_UNSIGNED	2	1	0	65535
uint32	NPX_TYPE_UNSIGNED	4	1	0	4294967295
uint64	NPX_TYPE_UNSIGNED	8	1	0	1,84467E+19
int8	NPX_TYPE_INTEGER	1	1	-128	127
int16	NPX_TYPE_INTEGER	2	1	-32768	32767
int32	NPX_TYPE_INTEGER	4	1	-2,147,483,648	2,147,483,647
int64	NPX_TYPE_INTEGER	8	1	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
real32	NPX_TYPE_REAL	4	1	-3.4E+38 (7 digits)	+3.4E+38 (7 digits)
real64	NPX_TYPE_REAL	8	1	-1.8E+308 (15 digits)	1.8E+308 (15 digits)
string	NPX_TYPE_STRING	variiert	1540	-	-

Tabelle 34: *Objektelement-Datentypen, Größe, Anzahl, Grenzen*

7.2 Referenzen

[1] Technische Referenz, netPROXY, Objekte und Packages, DOC160204TRXXDE, netPROXY - Objekte und Packages TR XX DE.pdf

[2] Protocol API, netPROXY, Function Interface, DOC160205APIXXEN, netPROXY Function interface API XX EN.pdf

[3] Bedienerhandbuch, netX Studio Engineering Tool (V1.400), IOT-Konfiguration, DOC180404OIXXDE, netX Studio Engineering Tool OI IOT XX DE.pdf (dieses Handbuch)

[4] Application note, netX Studio Engineering Tool, Web Builder, DOC160207ANXXDE, netX Studio Web Builder AN 01 DE.pdf

Abbildungsverzeichnis

Abbildung 1:	netX Studio Engineering Tool IOT - Kommunikationskanäle	13
Abbildung 2:	Installationsinformation (Beispiel Windows® 10).....	17
Abbildung 3:	Endbenutzer-Lizenzvereinbarung akzeptieren (Beispiel Windows® 10).....	18
Abbildung 4:	Zielordner für die Installation (Beispiel Windows® 10).....	18
Abbildung 5:	Installation durchführen (Beispiel Windows® 10).....	18
Abbildung 6:	Installationsfortschritt (Beispiel Windows® 10).....	19
Abbildung 7:	Installationsabschluss (Beispiel Windows® 10)	19
Abbildung 8:	Deinstallation wird vorbereitet (Beispiel Windows® 10)	20
Abbildung 9:	Fortschrittsanzeige zur Deinstallation (Beispiel Windows® 10)	20
Abbildung 10:	Startbildschirm netX Studio (netPROXY Basic) mit Meldungsfenster	21
Abbildung 11:	Startbildschirm netX Studio (netPROXY Engineering)	22
Abbildung 12:	Neues Projekt anlegen	22
Abbildung 13:	Projektname und Speicherpfad eingeben	23
Abbildung 14:	Grafische Benutzeroberfläche von netX Studio Engineering Tool mit geöffnetem Projekt	24
Abbildung 15:	Symbolleiste (aktiviert)	25
Abbildung 16:	Menü New Project	25
Abbildung 17:	Project Explorer - New Category	36
Abbildung 18:	Create New Category	36
Abbildung 19:	Create New Category – Advanced	37
Abbildung 20:	Kontextmenü ID range.....	38
Abbildung 21:	Edit Category Object ID Range	39
Abbildung 22:	Beispiel für sich überlappende ID-Bereiche	39
Abbildung 23:	Project Explorer - New Object	40
Abbildung 24:	Neues Objekt anlegen	40
Abbildung 25:	Objektkennung anpassen, Objektname eingeben, Objektkategorie auswählen ..	41
Abbildung 26:	Beispielobjekt '0x40010000 Sensor – simple [1.0]'	41
Abbildung 27:	Copy (Objekt kopieren).....	42
Abbildung 28:	Paste (Objekt einfügen).....	42
Abbildung 29:	Editor mit Properties und Elements - Beispiel	44
Abbildung 30:	Label auswählen für Zugriff über OPC UA, MQTT oder www (Beispielauswahl www).....	45
Abbildung 31:	Object Definitions-Editor – ausgewählte Labels (Beispielauswahl OPC UA, MQTT und www)	46
Abbildung 32:	Eigenes Label erstellen (Beispielname „test“)	47
Abbildung 33:	Beispiel zugeordnete Labels	47
Abbildung 34:	Object Definitions-Editor - Element anlegen (Beispielobjekt ‚0x40010001 Counter‘).....	48
Abbildung 35:	Datentyp auswählen	49
Abbildung 36:	Beispielobjekt ‚0x40010001 Counter [1.0]‘, Elemente ‚Counter1‘ und ‚Counter2‘	51

Abbildung 37: Beispielobjekt '0x40020001 Parameter [1.0]', Elemente 'Parameter1' und 'Parameter2'	51
Abbildung 38: Standardobjekte verwenden	52
Abbildung 39: Auswahl für Standardobjekte - Beispiel Generic Device	52
Abbildung 40: Objekteditor 0x10001000 - Device Description	53
Abbildung 41: Standardobjekte kopieren (Beispiel).....	54
Abbildung 42: Kategorie mit Standardobjekten löschen (Beispiel).....	54
Abbildung 43: Label auswählen für Zugriff über OPC UA, MQTT oder WebServer (Beispielauswahl OPC UA).....	55
Abbildung 44: Object Definitions-Editor – ausgewählte Labels (Beispielauswahl OPC UA)	56
Abbildung 45: Applikationsvariante erstellen	59
Abbildung 46: Name für erste Applikationsvariante eingeben	59
Abbildung 47: Beispiel Application C100	59
Abbildung 48: Applikationsvariante (Beispiel Application C100) - Objekte per Drag und Drop zusammenstellen.....	60
Abbildung 49: Applikationsvariante (Beispiel Application C100)	60
Abbildung 50: Editor der Applikationsvariante (Beispiel).....	61
Abbildung 51: Editor der Applikationsvariante mit Zeilen für die Instanzen und Elemente (Beispiel).....	61
Abbildung 52: Editor der Applikationsvariante - Instanzname editieren (Beispiel)	63
Abbildung 53: Editor der Applikationsvariante mit produktbezogenen Instanznamen (Beispiel).	63
Abbildung 54: Editor der Applikationsvariante – Unit, Min./Max. Value anpassen (Beispiel)	64
Abbildung 55: Project Explorer - New Communication Setting	65
Abbildung 56: Add Communication Settings - IoT protocol	66
Abbildung 57: Add Communication Settings - Application (Beispiel)	66
Abbildung 58: Add Communication Settings – No more applications	66
Abbildung 59: Project Explorer - Remove (Beispiel)	67
Abbildung 60: Kommunikationseinstellungen einer Applikationsvariante, Beispiel Add WebServer.....	69
Abbildung 61: Communication Settings > OPC UA-Editor – OPC UA Server Configuration (Beispiel).....	70
Abbildung 62: Communication Settings > OPC UA-Editor – OPC UA Actions Configuration (Beispiel).....	72
Abbildung 63: Communication Settings > MQTT-Editor - MQTT Component Information (Beispiel).....	73
Abbildung 64: Communication Settings > Beispiel WebServer-Editor (Default-Auswahl www)..	77
Abbildung 65: Product Details – New Product.....	79
Abbildung 66: Name eingeben und Kommunikationseinstellungen auswählen	79
Abbildung 67: Name eingeben und Kommunikationseinstellungen auswählen	79
Abbildung 68: Product Details – Beispiel.....	80
Abbildung 69: Product Details - Umbenennen (Beispiel)	81
Abbildung 70: Rename Product (Produktumbenennen).....	81
Abbildung 71: Product Details - Entfernen (Beispiel)	82

Abbildung 72: Product Details – Product Information (Beispiel)	83
Abbildung 73: Product Details – Product Information (Beispiel).....	84
Abbildung 74: Vendor Details > Vendor Information.	85
Abbildung 75: Vendor Details > Vendor ID (Beispiel mit Herstellerkennungen der Firma Hilscher)	86
Abbildung 76: User Management - User Accounts-Editor	87
Abbildung 77: Benutzerkonto Admin editiert	88
Abbildung 78: User Accounts – Angaben zum neuen Benutzerkonto hinzufügen	89
Abbildung 79: Neues Benutzerkonto für User anlegen (Beispiel)	89
Abbildung 80: Übersicht zur Verteilung der Zugriffsrechte für den WebServer bzw. den FTP-Server	90
Abbildung 81: Geräteerstellungsprozess (Build-Prozess) starten	91
Abbildung 82: Laufender Build-Prozess	91
Abbildung 83: Verzeichnis Product Build.....	92
Abbildung 84: Beispiel, Header-Datei Objects.h	93
Abbildung 85: Project > Clean	94
Abbildung 86: Kontextmenü Product Build > Show in Windows Explorer, Copy, Remove	95
Abbildung 87: Ordner Product Build, Unterverzeichnisse Application Interface und Products....	95
Abbildung 88: Kontextmenübefehle Copy, Remove.....	95
Abbildung 89: Importfunktion aufrufen (Beispiel).....	98
Abbildung 90: Importinhalt festlegen (Beispiel)	99
Abbildung 91: Importdatei wählen	99
Abbildung 92: Dateiauswahl-Dialog für Importdatei (Beispiel User.zip)	100
Abbildung 93: Importdatei gewählt (Beispiel User.zip)	101
Abbildung 94: Neu importierte Objektdefinitionen (Beispiel)	101
Abbildung 95: Importdatei gewählt (Beispiel User.zip) - Warnung, Kategorie „User“ wird überschrieben.....	102
Abbildung 96: Export-Funktion aufrufen (Beispiel).....	103
Abbildung 97: Exportinhalt festlegen - Export Object Definitions	104
Abbildung 98: Ausgabeordner festgelegt	105
Abbildung 99: Ordner suchen.....	106
Abbildung 100:Export-Funktion aufrufen (Beispiel).....	107
Abbildung 101:Exportinhalt festlegen - Export Project Description	107
Abbildung 102:Inhalte für die Projektdokumentation und Ausgabeordner festlegen.....	108
Abbildung 103:Ausschnitt Produktdokumentation im html-Format.....	110

Tabellenverzeichnis

Tabelle 1:	Änderungsübersicht.....	4
Tabelle 2:	Kennzeichnung von Objekteigenschaften	12
Tabelle 3:	Objektfunktionen.....	12
Tabelle 4:	Komponenten netX Firmware mit IOT	14
Tabelle 5:	Symbole in der netX Studio Engineering Tool-Symbolleiste	25
Tabelle 6:	Allgemeine Bedienelemente	26
Tabelle 7:	Objektdaten nach Konfiguration des Zugriffs (Beispiele).....	28
Tabelle 8:	Schrittfolge bei der Projektkonfiguration bzw. zur IOT-Konfiguration	30
Tabelle 9:	Objektdaten für das Beispiel C100	34
Tabelle 10:	New Object – Parameter	40
Tabelle 11:	Labels für OPC UA, MQTT, www bzw. eigene Labels	46
Tabelle 12:	Parameter der Elemente (Object Definitions)	48
Tabelle 13:	Bedienelemente des Object Definitions-Editors	50
Tabelle 14:	Funktionsumfang, Beispiel C100	57
Tabelle 15:	Parameter der Elemente (Object Instances)	62
Tabelle 16:	Parameter der Elemente – Details (Details)	64
Tabelle 17:	Parameter OPC UA Server Configuration	70
Tabelle 18:	Parameter OPC UA Actions Configuration	72
Tabelle 19:	Parameter MQTT Component Information	73
Tabelle 20:	Parameter MQTT - Connection Settings	74
Tabelle 21:	Parameter MQTT Configuration	76
Tabelle 22:	Herstellerinformationen (Beispiel).....	85
Tabelle 23:	Bedienelemente des User Accounts-Editors	87
Tabelle 24:	Angaben zum Benutzerkonto	89
Tabelle 25:	Produkdateien im Build-Prozess von netX Studio Engineering Tool	92
Tabelle 26:	Produkdateien des Build-Prozesses.....	96
Tabelle 27:	Erläuterungen zum Fenster Import (Select).....	99
Tabelle 28:	Erläuterungen zum Fenster Object Definitions import	100
Tabelle 29:	Erläuterungen zum Fenster Export (Select)	104
Tabelle 30:	Erläuterungen zum Fenster Export Object Definitions.....	105
Tabelle 31:	Erläuterungen zum Fenster Export (Select)	108
Tabelle 32:	Erläuterungen zum Fenster Export Project Description.....	109
Tabelle 33:	Festgelegte Wertebereiche für Objekt-IDs	111
Tabelle 34:	Objektelement-Datentypen, Größe, Anzahl, Grenzen	112

Glossar

MQTT	Message Queue Telemetry Transport: Publisher-Subscriber-basiertes "schlankes" Nachrichten-Protokoll für Machine-to-Machine-Kommunikation (M2M) und den Einsatz oberhalb des TCP/IP-Protokolls, welches die Übertragung von Telemetrie-Daten in Form von Nachrichten zwischen Geräten ermöglicht, auch bei langen Verzögerungen oder eingeschränkter Netzwerk-Qualität. Als Protokoll des Internets der Dinge (IoT) wird es seit 2013 von der Organization for the Advancement of Structured Information Standards (OASIS) standardisiert. Zusätzliche Merkmale: einfache Handhabung, effiziente Ressourcen-Nutzung, Sicherheitsmechanismen, Unterstützung mehrerer Cloud-Systeme, Session-Awareness, Daten-agnostisch (d. h. Einsatz flexibler Strukturen zur Übertragung verschiedener Informationsarten)
netPROXY	Rahmenanwendung für die effiziente Entwicklung von vernetzten Geräten basierend auf Hilscher netX, welche ein Objektmodell als generische Abstraktion von Diensten oder Daten bereitstellt. Die Bezeichnung wird auch als Oberbegriff für die netPROXY-Technologie von Hilscher verwendet.
netX	networX on chip, Hilscher-Netzwerk-Kommunikationscontroller. Hoch integrierter Netzwerk-Controller mit einer auf Kommunikation und maximalen Datentransfer optimierten Systemarchitektur
OPC UA	OPC Unified Architecture: industrielles M2M-Kommunikationsprotokoll von der OPC Foundation zu Kompatibilitätszwecken entwickelt, welches insbesondere die Fähigkeit besitzt, Maschinendaten (Regelgrößen, Messwerte, Parameter usw.) nicht nur zu transportieren, sondern auch maschinenlesbar semantisch zu beschreiben.
Scnd-Stage-Boot-Loader	vom netX ROM-Loader gestartet, erstellt er ein Standard-Hilscher-Dual-Port-Memory und stellt Basis-Funktionalitäten bereit. Er ermöglicht den Download von Firmware-Dateien über das Dual-Port-Memory.
WebServer	Bestandteil einer Firmware eines nach dem netPROXY-Modell entwickelten Gerätes, welches es dem Anwender im Feld u. a. ermöglicht, mittels Standard-Webbrowser und Ethernet-Schnittstelle Statusparameter des Automatisierungsgerätes abzurufen, ein Firmware-Update oder ein Reset des Geräts über HTTP durchzuführen und gegebenenfalls Daten zu lesen und zu schreiben, bzw. die IP-Adresse des Gerätes zu konfigurieren.

Kontakte

HAUPTSITZ

Deutschland

Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Telefon: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support

Telefon: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

NIEDERLASSUNGEN

China

Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Telefon: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support

Telefon: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

Frankreich

Hilscher France S.a.r.l.
69500 Bron
Telefon: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support

Telefon: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

Indien

Hilscher India Pvt. Ltd.
Pune, Delhi, Mumbai
Telefon: +91 8888 750 777
E-Mail: info@hilscher.in

Italien

Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Telefon: +39 02 25007068
E-Mail: info@hilscher.it

Support

Telefon: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan

Hilscher Japan KK
Tokyo, 160-0022
Telefon: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support

Telefon: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea

Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Telefon: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

Schweiz

Hilscher Swiss GmbH
4500 Solothurn
Telefon: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support

Telefon: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA

Hilscher North America, Inc.
Lisle, IL 60532
Telefon: +1 630-505-5301
E-Mail: info@hilscher.us

Support

Telefon: +1 630-505-5301
E-Mail: us.support@hilscher.com