

# Inhaltsverzeichnis

- Allgemein** ..... 1
- Was tut das Plugin?** ..... 1
- Funktionen** ..... 1
- Unterstützte Server** ..... 1
- Zweck & Anwendung** ..... 1
- Installation** ..... 1
  - Anforderungen ..... 1
- Konfiguration** ..... 2
- Überblick ..... 2
- Benutzung ..... 2
- Einstellungen ..... 3
- Einstellungen ändern ..... 3
- Übersicht ..... 3
- Variablen ..... 4
- OPC Nodend ..... 4
- Fehlerdiagnose** ..... 5
- Channel ..... 5
- Variablen ..... 5
- Logdatei ..... 6
- Status Codes ..... 6
- Entities** ..... 6
- Device ..... 6
- Channel ..... 7
- Ordner & Dateien** ..... 7
- Ordner ..... 7
- Dateien ..... 7
- Versionsinformation** ..... 7
- Dieses Dokument ..... 7
- Plugin ..... 7
- Assembly ..... 7



# OPC UA Client Device Plugin

## Allgemein

Das OPC UA Client Device Plugin bietet eine Verbindung zwischen CoDaBix® und einem OPC UA Server.

## Was tut das Plugin?

Das OPC UA Client Device Plugin baut Verbindungen zu einem oder mehreren OPC Servers auf und hält diese aufrecht. Jeder OPC UA Client kann über einen separaten Kanal verbunden werden.

## Funktionen

- Browsen von OPC UA Servern/OPC Classic Servern, um automatisch verfügbare Variablen in CoDaBix zu erstellen
- Lesen und Schreiben von OPC UA/OPC Classic Data Variables
- Abonnieren von OPC UA Data Variablen/OPC Classic Variablen, wenn die entsprechenden Nodes in CoDaBix abonniert werden

## Unterstützte Server

- OPC UA Server, die das `opc.tcp`- oder `http`-Protokoll verwenden
- OPC Classic (COM) Server (spezifiziert durch die ProgID und ClassID) (nur unter Windows)

## Zweck & Anwendung

Die verbundenen OPC UA Server können durch die Benutzung von CoDaBix® einfach gesteuert werden. Indem die Nodes des OPC UA Servers mit den in CoDaBix® definierten Nodes verbunden werden, kann der OPC Node direct mit vielen anderen Nodes, Geräten, Services etc. interagieren, die in CoDaBix® gewartet werden.

Auch andere CoDaBix®-Teilnehmer können mit dem durch das OPC UA Device Plugin verbundenen OPC UA Server interagieren.

## Installation

Dieses Plugin ist Bestandteil des CoDaBix® Setups. Bitte konsultieren Sie [CoDaBix® Setup und erster Start](#) für weitere Informationen darüber, wie dieses Plugin installiert und deinstalliert werden kann.

## Anforderungen

- Standardanforderungen von CoDaBix®
- Aktivierte ausgehende Verbindung via TCP/IP über den spezifizierten Port

# Konfiguration

## Überblick

Die gesamte OPC UA Client Device Pluginkonfiguration finden Sie unter dem Nodepfad `/System/Devices/OPC-UA Client Device`. Diese Wurzelnode des Device Plugins ermöglicht die vollständige Konfiguration des OPC UA Client Device Plugins.

Name	Display Name	Description	Node Type	Path	Actual Value	Status code
ApplicationName	ApplicationName		String		CoDaBix OPC Client Device	Good
LoginCertificate	LoginCertificate		Blob			---
LoginName	LoginName		String			---
LoginPassword	LoginPassword		Blob			---
ServerAddress	ServerAddress		String		opc.tcp://192.168.0.116:12345	Good

## Benutzung

Der Nodebaum im oberen Bild zeigt den Standardnodebaum des OPC UA Client Device Plugin. Um einen oder mehrere OPC UA Client Device Kanäle zu einzurichten, fügen Sie einen Folder Node unter dem Node `OPC UA Client Device/Channels` hinzu (linkes Bild). Danach erscheint der Standardnodebaum für einen Kanal (rechts Bild).

The screenshot shows a software interface with a tree view on the left and a table on the right. A red arrow points to a '+' icon in the toolbar. A dialog box titled 'Add new Folder Node' is open, with the following fields:

- Name: My Channel 1
- Display Name: My Channel 1
- Node Type: Folder
- Path: (empty)
- Refresh Interval: <Inherit>
- Max Value Age (ms): (empty)
- History Value Interval: (empty)

The tree view on the right shows the updated structure: `System > Devices > OPC-UA Client Device > Channels > My Channel 1`. Under 'My Channel 1', there are sub-nodes for 'Control', 'Settings', 'Application', 'Security', 'Session', 'Transport', 'Status', and 'Variables'.

Nun können die Einstellungen für den spezifizierten Kanal verändert werden. Auch können neue Variablen unter dem Order `OPC UA Client Device/Channels/<Channel>/Variables` erstellt werden. Die Verbindung zum OPC Node wird durch das Benutzen der `Path` Eigenschaft des neuen Variablennodes genehmigt.

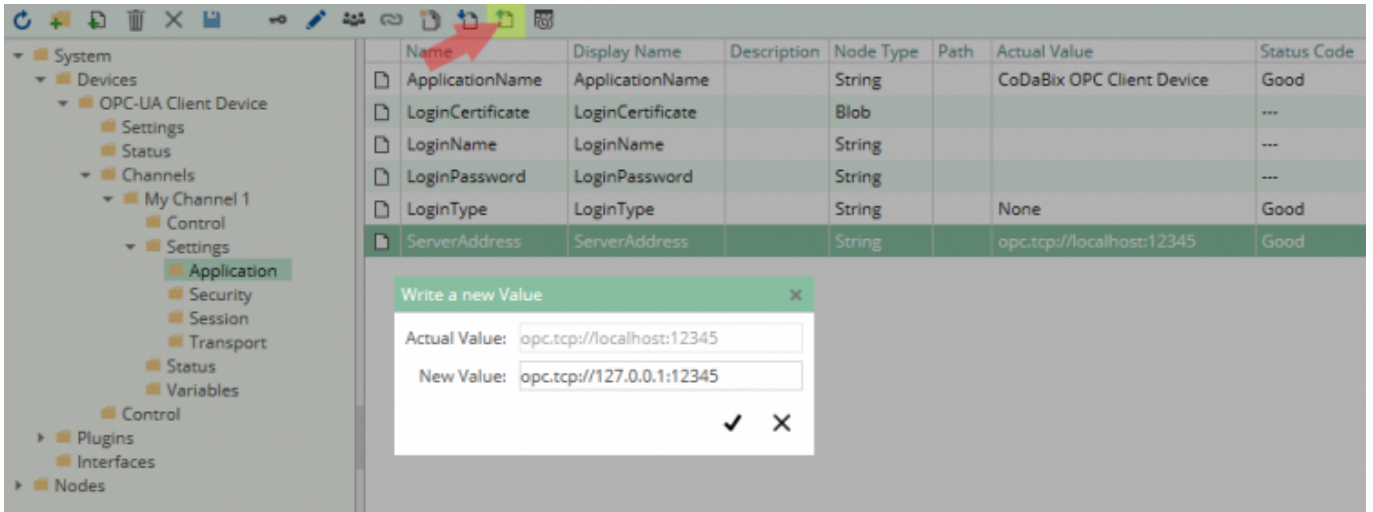
Die `Path` Eigenschaft muss eine gültig formatierte OPC Nodeld sein, zum Beispiel `2:Main-PLC/Office 1 - Lights/Front`.

Jede Änderung im `Settings` Ordner wird angewandt, sobald der Kanal neu gestartet wird.

Jede Änderung im **Variables** Ordner führt automatisch eine Rekonfiguration des OPC Clients im spezifizierten Kanal durch.

## Einstellungen Einstellungen ändern

- Wählen Sie die zu ändernde **Settings** Eigenschaft aus. (z.B. „Application/ServerAddress“)
- Klicken Sie auf „Write a new Value“. (siehe Beispiel im unteren Screenshot)
- Geben Sie den neuen Wert ins Eingabefeld ein und speichern Sie die Änderungen.



## Übersicht

Name	Typ	Standardwert	Beschreibung
<u>Application</u>			
Application Name	String	CoDaBix® OPC UA Client Device	Mit diesem Namen stellt sich der OPC UA Client dem OPC UA Server vor.
Server Address	String	opc.tcp://localhost:12345	Die URL zum OPC UA Server. Mögliche Protokolle sind: <b>opc.tcp://</b> , <b>http://</b> , <b>https://</b> , <b>opc.com://</b> .  Um unter Windows auf OPC Classic (COM)-Server zuzugreifen, können Sie das Format <b>opc.com://&lt;hostname&gt;(:&lt;port&gt;)/&lt;progId&gt;/&lt;classId&gt;</b> verwenden. Die optionale Portnummer wird von dem intern erstellten Wrapperserver verwendet. Falls die Portnummer nicht angegeben ist, wird sie in einem Bereich zwischen 48000 und 48999 aus dem <b>&lt;classId&gt;</b> -Wert generiert. Beispiel für OPC Classic: <b>opc.com://localhost:4711/OPCManager.DA.XML-DA.Server.DA/{E4EBF7FA-CCAC-4125-A611-EAC4981C00EA}</b>  <b>Bitte beachten Sie:</b> Einige OPC Classic (COM)-Server unterstützen nur <b>x86- (32-Bit)-Clients</b> . Daher empfehlen wir, die <b>x86</b> -Version statt der <b>x64</b> -Version von CoDaBix zu installieren, wenn Sie auf solche Server zugreifen möchten.
Login Type	Enum	None	Definiert, welche Authentifizierungsart verwendet wird. 🔍 Gültige Werte: <b>None, Certificate, UserPassword</b>
Login Certificate	Blob		Das Zertifikat des OPC UA Servers, das für die Authentifizierung benutzt wird. Kann als Zertifikatsdatei hochgeladen werden. (Muss den Zertifikatsschlüssel enthalten.)
Login Name	String		Der Benutzer des OPC UA Servers, der für die Authentifizierung benutzt wird.
Login Password	String		Das Passwort des Benutzers des OPC UA Servers.
<u>Session</u>			
Disconnect Timeout	Int32	10000	t.b.a.
Reconnect Timeout	Int32	10000	t.b.a.
Session Timeout	Int32	60000	t.b.a.
Use Break Detection	Boolean	True	Stellt automatisch fest, wenn die Verbindung zum OPC UA Server unterbrochen wird. Dies wird für die automatische Wiederverbindung benutzt, falls die Verbindung zum Server abgebrochen ist.. Gültige Werte: <b>True, False</b>
<u>Security</u>			
Policy Algorithm	String	Auto	🔍 Gültige Werte: <b>Auto, Basic128Rsa15, Basic256, Basic256Sha256, Custom, Https, None</b>
Policy Level	Int32	0	t.b.a.
Policy Mode	String	None	🔍 Gültige Werte: <b>None, Sign, SignAndEncrypt</b>

Name	Typ	Standardwert	Beschreibung
Use Domain Checks	Boolean	False	Zeigt an, ob der OPC UA Client den OPC UA Server auf ein vertrauenswürdigen Zertifikat prüft oder nicht. Dies ist eine Sicherheitsfunktion, um z.B. Man-in-the-middle Attacken zu verhindern. Gültige Werte: <b>True, False</b>  <b>Achtung</b> Ist diese Option auf <b>true</b> eingestellt und der OPC UA Server hat kein vertrauenswürdigen X.509-Zertifikat (z.B. ein selbstsigniertes Zertifikat), wird die Verbindung abgelehnt.
Use Secure Endpoint	Boolean	True	Zeigt an, ob der OPC UA Client die Sicherheit des Endpunkts überprüfen soll oder nicht. Manche OPC UA Server könnten diese Option nicht unterstützen. Gültige Werte: <b>True, False</b>
Certificate	Blob	<Blob>	Das Client-Zertifikat. Mit diesem Zertifikat stellt sich der OPC UA Client dem OPC UA Server vor. Standardmäßig wird ein neues Zertifikat erstellt.
<b>Transport</b>			
Channel Lifetime	Int32	600000	t.b.a.
Max Array Length	Int32	65535	t.b.a.
Max Buffer Size	Int32	65535	t.b.a.
Max Byte String Length	Int32	65535	t.b.a.
Max Message Size	Int32	1048576	t.b.a.
Max String Length	Int32	65535	t.b.a.
Operation Timeout	Int32	60000	t.b.a.
Security Token Lifetime	Int32	3600000	t.b.a.

## Variablen

Jeder Node unter **System/Devices/OPC UA Client Device/Channels/My Channel 1/Variables** kann vom OPC Server des spezifizierten Kanals über die **Path** Eigenschaft mit einem OPC Node verbunden werden.

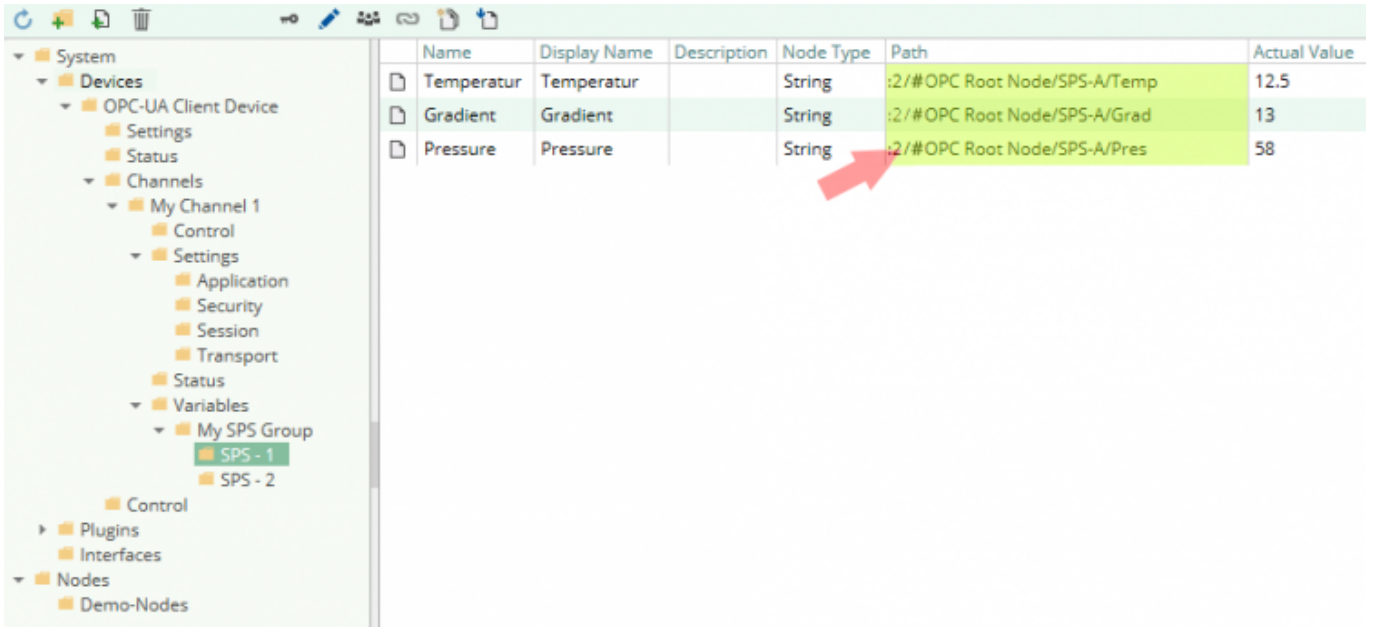
Um einen CoDaBix® Node mit einem OPC Node zu verbinden, muss die **Path** Eigenschaft als eine OPC Nodeld formatiert sein.

## OPC Nodeld

- Beispiel Nodeld: **2:Machine\_1/IsActive**

Die spezifizierte Nodeld ist in zwei Bestandteile unterteilt: Der Index des Namensraums und der Identifier der OPC UA Node. Im Moment werden nur String Identifier unterstützt. Das Schema des Identifiers hängt vom benutzten OPC UA Server ab..

- Namespace Index: **<Number>** e.g. **2**
- Trennzeichen: **:**
- Identifier (String): **<ParentNode1>/<ParentNode2>/.../<ParentNodeX>/<NodeName>**

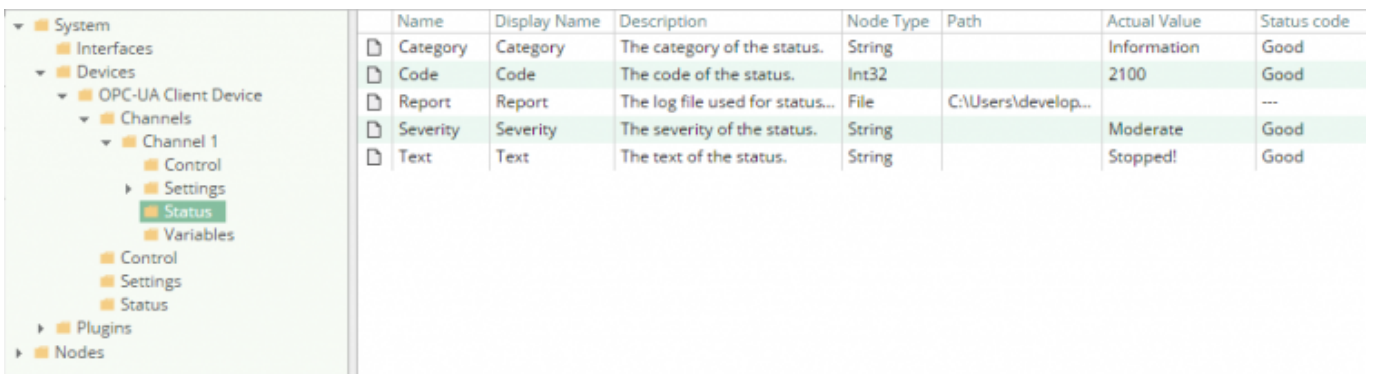


# Fehlerdiagnose

Das OPC UA Client Device Plugin stellt je nach zu untersuchender Schicht verschiedene Statusinformationen bereit. Generell wird die kanalbasierte Diagnoseinformation vom Verbindungsstatus des Kanals zum OPC UA Server produziert. Die variablenbasierte Diagnoseinformation wird während des Lese- / Schreibzugriffs der verschiedenen Variablen oder direkt vom OPC UA Server produziert.

## Channel

Um den Status von verschiedenen Gerätekanälen zu überwachen und zu diagnostizieren, werfen Sie einen Blick auf folgendes Bild::



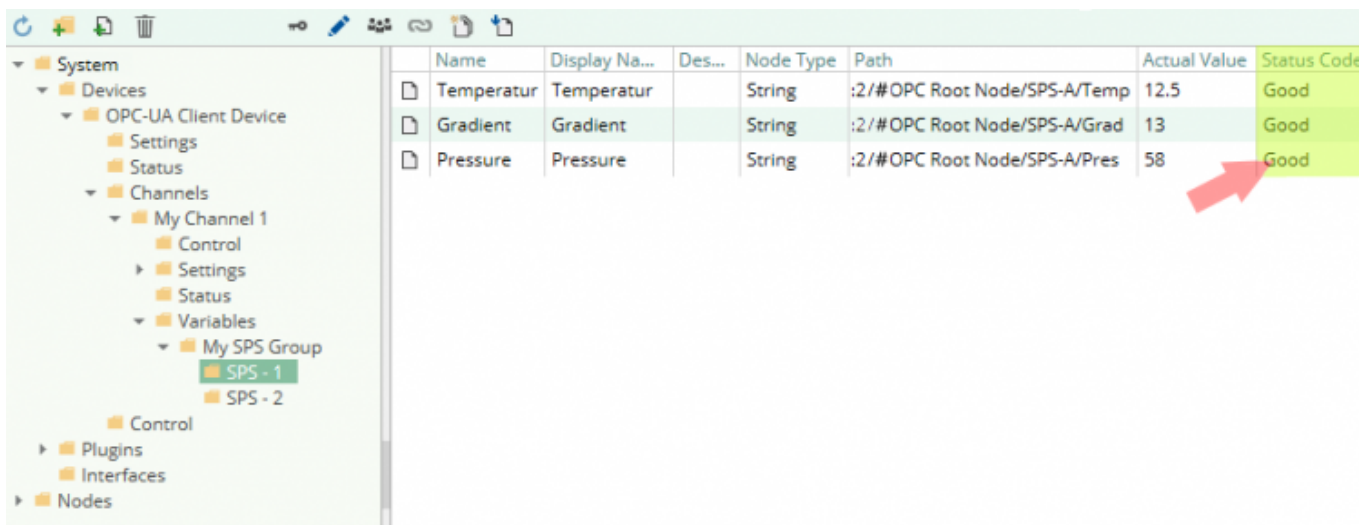
Das obige Bild zeigt den **Status** Node des Kanals, der alle statusrelevanten Informationen abbildet. Die folgenden Datenpunktnodes werden benutzt, um den Kommunikationsstatus zwischen CoDaBix® und dem OPC UA Server zu erhalten.

Node	Beschreibung
Category	Unterteilt den Status in <b>Information</b> , <b>Warning</b> und <b>Error</b> und zeigt somit die generelle Art der Statusinformation an.
Code	Definiert den numerischen Ausdruck / Identifier des Status.
Severity	Stuft die Statusinformation ein in <b>Low</b> , <b>Moderate</b> , <b>High</b> und <b>Critical</b> und zeigt somit die Dringlichkeit eines Einschreitens an.
Text	Beschreibt die durch die <b>Code</b> Eigenschaft identifizierte Statusinformation.

## Variablen

Um den Status der verschiedenen OPC UA Client Variablen zu überwachen und zu diagnostizieren, werfen

Sie einen Blick auf die **Status Code** Eigenschaft der in CoDaBix® angezeigten Variable. Falls die **Status Code**-Spalte den Wert **Bad** anzeigt, ist in den meisten Fällen der adressierte Datenbereich nicht erreichbar.



### Logdatei

Alle Device Channel-relevanten Statusinformationen werden auch in die kanalspezifische Logdatei protokolliert, die sich im [LoggingFolder] befindet. Jede Logdatei wird nach dem folgenden Namensschema benannt: **OPC UA Client Device.<ChannelName>.log**. Der Inhalt einer solchen Logdatei kann wie folgt aussehen:

```

...
[15:31:46 05.09.2016] - Information (Severity=Moderate): Code=[10012], Text=[Creating Client
| opc.tcp://192.168.0.116:12345/]
...

```

In Verwendung des Beispielkanals wäre der Name der Logdatei: **OPC UA Client Device.Channel 1.log**

### Status Codes

Die folgende Tabelle zeigt die verschiedenen möglichen Statusinformationen:

Code	Kategori	Informationsart
-22000 to -22999	Error	Fehler mit Ausnahme
-21000 to -21999	Error	Interner Fehler
-12000 to -12999	Warning	Warnung mit Ausnahme
-11000 to -11999	Warning	Interne Warnung
10000 to 10999	Information	Information
20000 to 20999	Information	Debug

## Entities

Wie jedes Device Plugin erweitert das OPC UA Client Device Plugin das grundlegende CoDaBix® **Device Modell**.

### Device

Der Devicetyp **OpcClientDevice** des Plugins definiert auch den **OpcClientDeviceChannel** und erweitert somit die grundlegenden **CodabixDevice** und **CodabixDeviceChannel** Entities. Während das **OpcClientDevice** nur eine Konkretisierung des **CodabixDevice** repräsentiert, erweiter der **OpcClientDeviceChannel** den **CodabixDeviceChannel** mit den OPC UA Client Variable Entities.



## Channel

Jeder Kanal wird von einem Channel Worker behandelt, der eine physische Verbindung zum OPC UA Server herstellt.

Standardmäßig liest der Worker keine Werte. Wenn ein Client oder Plugin ein synchrones Lesen der Variablen des Kanals in CoDaBix® anfordert (z.B. durch das Benutzen der Funktion „Read actual value“ der CoDaBix® Webkonfiguration), liest der Channel Worker diese aus dem zugrundeliegenden OPC UA Server und schreibt sie dann in die entsprechenden CoDaBix® Nodes.

Ähnlich schreibt der Channel Worker die Werte in den zugrundeliegenden OPC UA Server, wenn ein Client oder Plugin Werte in die Variablen des Kanals schreibt.

Beachten Sie: Nachdem die Werte in die CoDaBix® Nodes geschrieben wurden, werden diese solange die alten Werte beinhalten, bis der OPC UA Server die neuen Werte erhalten und anerkannt hat.

Um eine OPC UA Client Variable regelmäßig gelesen zu bekommen, können Sie den Node in der Webkonfiguration bearbeiten und „History Options“ auf **Yes** setzen (was intern eine Subscription erstellt). In diesem Fall abonniert der Channel Worker die Variablen vom OPC UA Server (der OPC Client bekommt bei einer Wertänderung den neuen Wert automatisch vom OPC Server) und wenn eine der Variablen sich verändert hat, schreibt er den neuen Wert in den entsprechenden CoDaBix® Node.

## Ordner & Dateien

### Ordner

Name	Pfad	Zweck / Verwendung
AssemblyFolder	<CodabixInstallDir>\plugins\OpcUaClientDevicePlugin\	Beinhaltet die Plugin Assembly Datei.
ConfigFolder	<CodabixDataDir>\plugins\OpcUaClientDevicePlugin\	Beinhaltet die Plugin Konfigurationsdatei.
LoggingFolder	<CodabixDataDir>\log\	Beinhaltet die Plugin Log Dateien.

### Dateien

Typ	Pfad	Zweck / Verwendung
Assembly	[AssemblyFolder]\CoDaBix.OpcUaClientDevicePlugin.dll	Die Plugin Assembly Datei.
Logging	[LoggingFolder]\OPC-UA Client Device.<ChannelName>.log	Die Log Datei.

## Versionsinformation

### Dieses Dokument

<b>Datum</b>	2017-02-06
<b>Version</b>	1.2

### Plugin

<b>Name</b>	OPC UA Client Device Plugin
<b>Node</b>	/System/Devices/OPC UA Client Device
<b>Version</b>	1.0.6

### Assembly

<b>Name</b>	CoDaBix.OpcUaClientDevicePlugin.dll
<b>Datum</b>	2017-05-16
<b>Version</b>	1.0.6.0