

Inhaltsverzeichnis

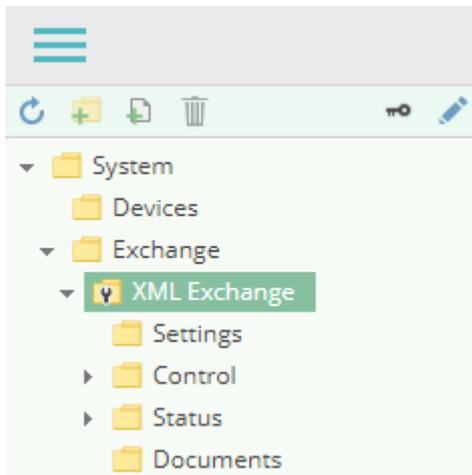
- Konfiguration** 1
 - Document 1
 - Settings 1
 - Hinzufügen eines Documents (Channels) 2
 - Variablen 2
 - Beispiel 3
 - Lesen/Schreiben 3
- Fehlerdiagnose** 5
 - Kanal 5
 - Variablen 6
 - Logdatei 6
- Entities** 6
 - Exchange 6
 - Channel 6
- Ordner & Dateien** 7
 - Ordner 7
 - Dateien 7
- Versionsinformation** 7
 - Dieses Dokument 7
 - Plugin 7
 - Assembly 7

XML Exchange Plugin

Das XML Exchange Plugin ermöglicht das Lesen und Schreiben von Daten (Attribute und Textknoten) in XML-Dateien (mit fester Struktur).

Konfiguration

Die gesamte XML Exchange Plugin-Konfiguration befindet sich unter dem Nodepfad /System/Exchange/XML Exchange.



Document

Ein XML Exchange Document (ähnlich eines **Channels** bei Device-Plugins) repräsentiert eine XML-Datei.

Settings

File Path

Der Pfad zur XML-Datei im Dateisystem.

Exchange Mode

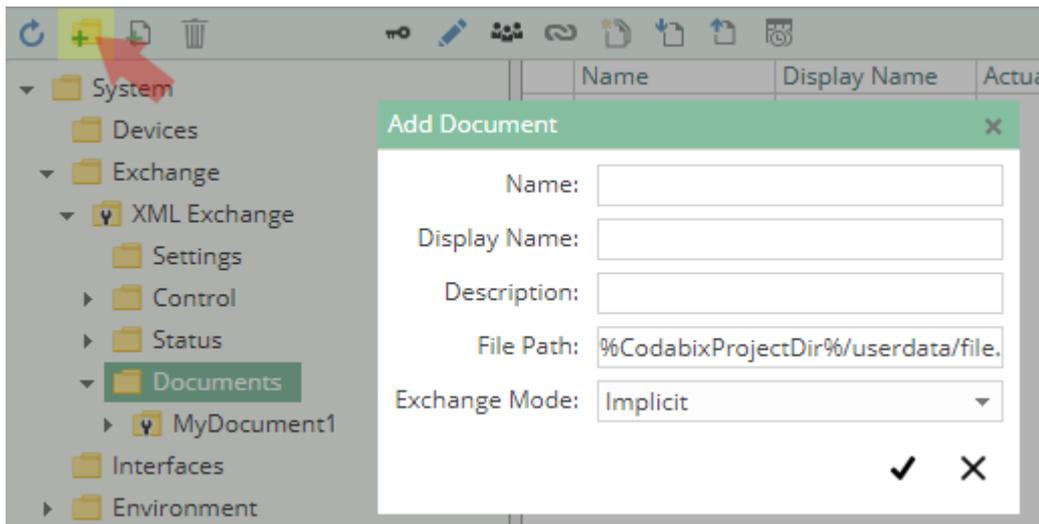
Gibt an, wie auf die XML-Datei zugegriffen werden soll:

- **Implicit:** Die XML-Datei wird eingelesen, sobald eine oder mehrere Variablen gelesen werden; und die Datei wird geschrieben, sobald eine oder mehrere Variablen

geschrieben werden.

- **Explicit:** Beim Lesen oder Schreiben von Variablen findet kein Dateizugriff statt. Die Datei muss explizit gelesen oder geschrieben werden, indem die Load- oder Save-Methode aufgerufen wird, um den Inhalt der Datei in die Variablen zu übertragen und umgekehrt.

Hinzufügen eines Documents (Channels)



Um ein neues **XML Document** (entspricht einem **Channel** bei Device Plugins) zu erstellen, gehen Sie wie folgt vor:

1. Fügen Sie einen Folder Node unter dem Node XML Exchange/Documents hinzu, oder machen Sie einen Rechtsklick auf den XML Exchange/Documents-Node und wählen Sie Add Document aus.
2. Tragen Sie im Add Document-Dialog die Settings für die XML-Datei ein.
3. Nachdem Sie „Save“ geklickt haben, wird die Document-Node erstellt.
4. Sie können den Kanal starten, indem Sie die Document-Node auswählen und den Startbutton klicken.

Variablen

Unter dem DocumentElement-Node können Sie Variablen (Nodes) erstellen, die den Aufbau des XML-Dokuments abbilden (d.h. Baumstruktur der CoDaBix-Variablen entspricht der Struktur der XML-Knoten):

- Ein Folder-Node-Variable bildet ein **XML-Element** ab, wobei die Path-Eigenschaft den Namen des Elements enthält; oder, falls dieser leer ist, der Name der Variablen als Name des XML-Elements verwendet wird.
- Ein Datapoint-Node-Variable bildet ein **XML-Attribut** oder **-Textknoten** ab, das als String gelesen und geschrieben werden kann. Die Path-Eigenschaft kann folgende Ausdrücke enthalten:
 - `text()`: Die Variable repräsentiert einen **XML-Textknoten**.

- Beginnt **nicht** mit / (z.B.: abc): Die Variable repräsentiert ein **XML-Attribut** des übergeordneten XML-Elements mit dem Namen, der angegeben wurde (im Beispiel "abc").
- Beginnt mit / (z.B.: /*/A/B/text() [2]): Die Variable repräsentiert eine **XPath-Abfrage**, die als Ergebnis entweder einen Text zurückliefert (dann kann in die Variable nicht geschrieben werden), oder ein Node Set, von dem der erste Knoten verwendet wird (muss ein Text-, Attribut- oder Kommentar-Knoten sein).
Beachten Sie: Bei Verwendung einer XPath-Abfrage wird die Position der Variable innerhalb des Nodes-Baums nicht berücksichtigt.

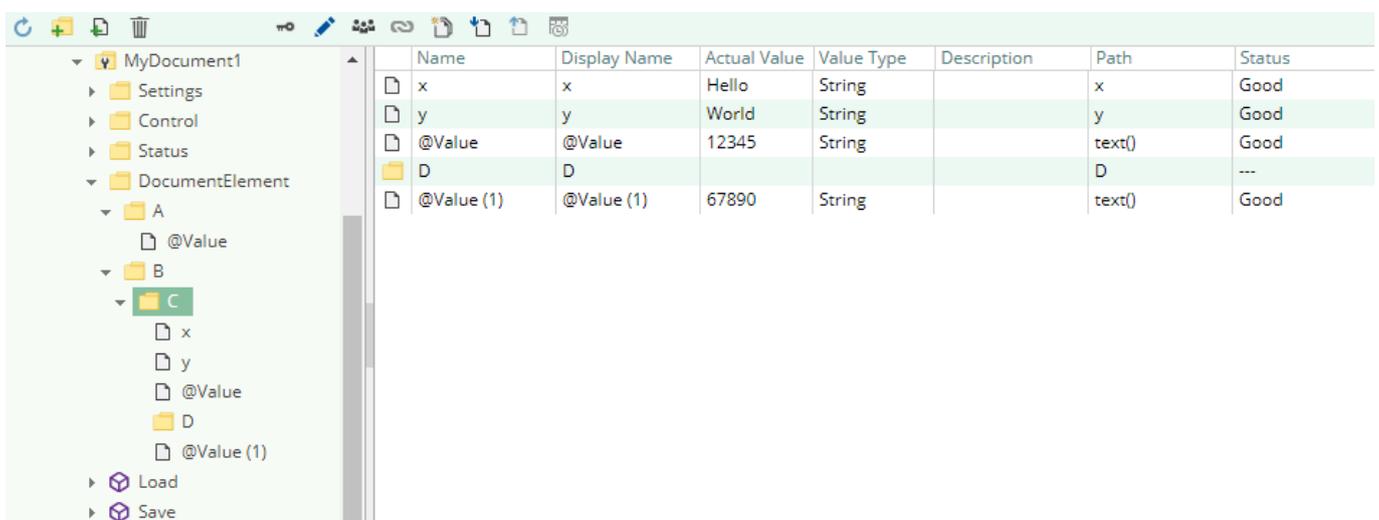
Bei einer vorhandenen XML-Datei können Sie das Document (Channel) browsen, um die CoDaBix-Variablen für die XML-Dokumentenstruktur automatisch anzulegen.

Beispiel

[example.xml](#)

```
<Root>
  <A>Test</A>
  <B>
    <C x="Hello" y="World">
      12345
    <D />
      67890
    </C>
  </B>
</Root>
```

Nachdem Browsen des Documents wird folgende Nodestruktur in CoDaBix angelegt:



Name	Display Name	Actual Value	Value Type	Description	Path	Status
x	x	Hello	String		x	Good
y	y	World	String		y	Good
@Value	@Value	12345	String		text()	Good
D	D				D	---
@Value (1)	@Value (1)	67890	String		text()	Good

Lesen/Schreiben

Sowohl beim Lesen und Schreiben werden CoDaBix-Variablen anhand deren Position innerhalb des

Nodebaums zu XML-Knoten zugeordnet. Falls beim Schreiben von Werten Variablen in CoDaBix existieren, diese aber nicht in der XML-Datei gefunden werden können (gilt beispielsweise auch, wenn die XML-Datei noch nicht existiert), werden die entsprechenden Knoten in der XML-Datei angelegt.

Leseverhalten (abhängig vom eingestellten **Exchange Mode** in den Settings).

- Exchange Mode **„Implicit“**:
 - Synchroner Lesevorgang von einer oder mehreren Variablen:
 - Die XML-Datei wird komplett eingelesen.
 - Die XML-Knoten werden den existierenden CoDaBix-Variablen über die Baumstruktur zugeordnet.
 - Es werden nur Werte in diejenigen CoDaBix-Variablen geschrieben, die Teil des synchronen Lesevorgangs waren.
 - Falls das Lesen der XML-Datei fehlschlägt, wird ein Wert mit dem Status Bad in die Variablen geschrieben.
 - Aufruf der Load-Methode:
 - Es wird ein Fehler ausgelöst, da die Methode nur im Modus **Explicit** verwendet werden kann.
- Exchange Mode **„Explicit“**:
 - Synchroner Lesevorgang von einer oder mehreren Variablen:
 - Es findet kein Dateizugriff statt; als Ergebnis des synchronen Lesevorgangs werden die aktuellen Werte der gelesenen Variablen zurückgegeben.
 - Aufruf der Load-Methode:
 - Die XML-Datei wird komplett eingelesen.
 - Die XML-Knoten werden den CoDaBix-Variablen über die Baumstruktur zugeordnet.
 - Es werden in alle CoDaBix-Variablen die Werte geschrieben, die aus den zugeordneten XML-Knoten gelesen wurden.
 - Falls das Lesen der XML-Datei fehlschlägt, wird nichts in die Variablen geschrieben, sondern die Methode löst einen Fehler aus.

Schreibverhalten:

- Exchange Mode **„Implicit“**:
 - Schreiben von Werten in eine oder mehrere Variablen:
 - Die XML-Datei wird komplett eingelesen.
 - Die XML-Knoten werden den existierenden CoDaBix-Variablen über die Baumstruktur zugeordnet.
Falls für eine existierende CoDaBix-Variable kein XML-Knoten gefunden wird, wird einer in der XML-Datei erzeugt (auch wenn die Variable nicht Teil des Schreibvorgangs ist), jedoch zunächst ohne Wert (Text).
 - Es werden nur Werte in diejenigen XML-Knoten eingefügt, deren zugeordnete CoDaBix-Variablen Teil der Schreibvorgangs waren.
 - Die XML-Datei wird geschrieben.
 - Aufruf der Save-Methode:
 - Es wird ein Fehler ausgelöst, da die Methode nur im Modus **Explicit** verwendet werden kann.
- Exchange Mode **„Explicit“**:
 - Schreiben von Werten in eine oder mehrere Variablen:
 - Es findet kein Dateizugriff statt; als Ergebnis des Schreibvorgangs wird der Status Good zurückgegeben.
 - Aufruf der Save-Methode:
 - Die XML-Datei wird komplett eingelesen.

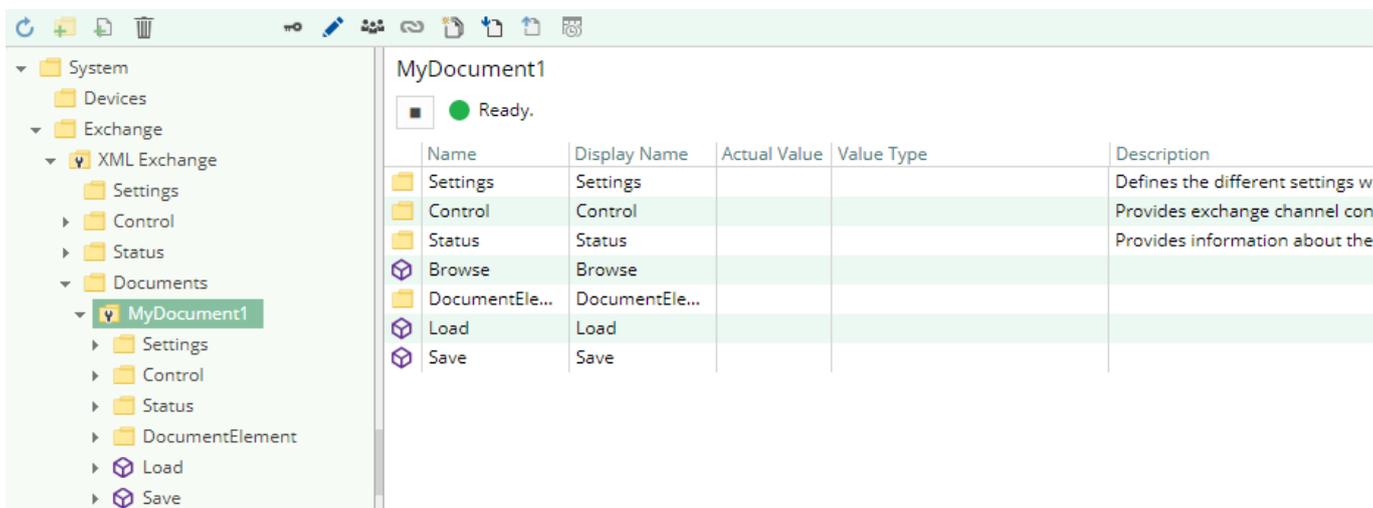
- Die XML-Knoten werden den existierenden CoDaBix-Variablen über die Baumstruktur zugeordnet.
Falls für eine existierende CoDaBix-Variable kein XML-Knoten gefunden wird, wird einer in der XML-Datei erzeugt, jedoch zunächst ohne Wert (Text).
- Es werden in alle XML-Knoten die aktuellen Werte eingefügt, die in den CoDaBix-Variablen enthalten sind.
- Die XML-Datei wird geschrieben.

Fehlerdiagnose

Das XML Exchange Plugin liefert je nach zu untersuchender Schicht verschiedene Statusinformationen. Generell werden die dokumentenbasierten Diagnoseinformationen durch den Zugriffsstatus auf die XML-Datei produziert. Die variablenbasierten Diagnoseinformationen werden während des Lese-/Schreibzugriffs auf die verschiedenen Variablen produziert.

Kanal

Um den Status des XML-Dokuments (Kanal) zu überwachen und zu diagnostizieren, werfen Sie einen Blick auf das folgende Bild:



Das obige Bild zeigt das Bedienfeld des XML-Dokuments, das alle statusrelevanten Informationen anzeigt. Das Bedienfeld aktualisiert automatisch seine Statusinformation, wenn ein neuer Status verfügbar ist.

Statuskreis

Farbe	Bedeutung
●	Der Kanal ist gestoppt. Klicken Sie den ►-Button, um ihn zu starten.
●	Der Kanal startet oder stoppt gerade oder wartet auf den Verbindungsaufbau.
●	Der Kanal ist bereit für Lese-/Schreiboperationen. Sie können ihn durch Klick auf den ■-Button stoppen.
●	Der Kanal läuft, aber die Verbindung ist momentan fehlerhaft. Bitte überprüfen Sie den Statustext für weitere Informationen.

Variablen

Um den Status der verschiedenen XML-Variablen zu überwachen und zu diagnostizieren, werfen Sie einen Blick auf die in CoDaBix angezeigte Status-Eigenschaft der Spalte. Falls der Exchange Mode auf „**Implicit**“ eingestellt ist, benutzen Sie den Button „Read actual Value“, um die Werte von der XML-Datei auszulesen und das Ergebnis in den Variablen zu speichern. Ansonsten können Sie die Load-Methode aufrufen, um zu prüfen, ob die XML-Datei erfolgreich gelesen werden kann.

	Name	Display Name	Actual Value	Value Type	Description	Path	Status
	x	x		String		x	Bad: Name cannot begin with the '<' character, hexadecimal value 0x3C. Line 2, position 1.
	y	y		String		y	Bad: Name cannot begin with the '<' character, hexadecimal value 0x3C. Line 2, position 1.
	D	D				ns2:D	---

Logdatei

Alle kanalbezogenen Statusinformationen werden auch in die kanalspezifische Logdatei im [LoggingFolder] protokolliert. Jede Logdatei wird nach dem Namensschema XML Exchange.<ChannelName>.log benannt.

Der Inhalt einer solchen Logdatei kann wie folgt aussehen:

```
...
2018-04-11 11:32:37.0 +2: [Error] Error (Severity=High): Code=[-1],
Text=[The operation has timed-out.], Details=[]
...
```

Entities

Wie jedes Exchange Plugin erweitert das XML Exchange Plugin das CoDaBix [Exchange-Modell](#).

Exchange

Der Exchange-Typ XmlExchange des Plugins definiert auch das XmlExchangeDocument und erweitert somit die grundlegenden CodabixExchange und CodabixExchangeChannel Entities. Während das XmlExchange nur eine Konkretisierung des CodabixExchange darstellt, erweitert das XmlExchangeDocument den CodabixExchangeChannel mit den XML Variable Entities.

Channel

Der Kanal wird von einem Channel Worker behandelt, der über Dateisystemoperationen Zugriff auf die XML-Datei herstellt.

Der Worker liest standardmäßig keine Werte. Wenn der Exchange Mode auf „**Implicit**“ gestellt ist und ein Anwender oder Plugin in CoDaBix einen synchronen Lesevorgang der Channel Variablen anfordert (z.B. mit der „Read actual value“-Funktion in der CoDaBix Webkonfiguration), liest der Channel Worker diese aus der XML-Datei und schreibt diese in die entsprechenden CoDaBix Nodes. Wenn der Exchange Mode auf **Explicit** gestellt ist, liest der Channel Worker Werte, sobald die Load-

Methode des Channels aufgerufen wird.

Ähnlich schreibt der Channel Worker auch die Werte in die Datei, wenn ein Client oder Plugin Werte in die Channel-Variablen schreibt („Implicit“) oder wenn die Save-Methode des Channels aufgerufen wird („Explicit“).

Damit eine XML-Exchange-Variable im Exchange Mode „Implicit“ regelmäßig gelesen wird, können Sie in der Webkonfiguration bei dem Node „History Options“ auf Yes stellen (was eine interne Subscription erstellt), oder Sie können zum Beispiel einen OPC UA Client verwenden, der mit dem OPC UA Server Plugin verbunden ist und damit eine Subscription für die XML-Variablenodes erstellen. In diesen Fällen liest der Channel Worker in regelmäßigen Intervallen die Variablen aus der XML-Datei und schreibt den neuen Wert nach einer Wertänderung automatisch in die entsprechenden CoDaBix Nodes.

Ordner & Dateien

Ordner

Name	Pfad	Zweck / Verwendung
AssemblyFolder	<CodabixInstallDir>/plugins/XMLExchangePlugin/	Beinhaltet die Plugin-Assemblydatei.
ConfigFolder	<CodabixProjectDir>/plugins/XMLExchangePlugin/	Beinhaltet die Plugin-Konfigurationsdatei.
LoggingFolder	<CodabixProjectDir>/log/	Beinhaltet die Plugin-Logdateien.

Dateien

Typ	Pfad	Zweck / Verwendung
Assembly	[AssemblyFolder]/CoDaBix.XMLExchangePlugin.dll	Die Plugin-Assembly Datei.
Logging	[LoggingFolder]/XML Exchange.<ChannelName>.log	Die Logdatei.

Versionsinformation

Dieses Dokument

Datum	2020-02-06
Version	1.0

Plugin

Name	XML Exchange Plugin
Node	/System/Exchange/XML Exchange
Version	1.0.0

Assembly

Name	CoDaBix.XMLExchangePlugin.dll
-------------	-------------------------------

Datum	2020-02-11
Version	1.0.0.0

From:

<https://www.codabix.com/> - **CoDaBix®**

Permanent link:

<https://www.codabix.com/de/plugins/exchange/xmlexchangeplugin>

Last update: **2021/07/30 13:39**